User's Manual

R-IN32M4-CL3

User's Manual: Hardware edition

R9A06G064MGBG
R9A06G064SGBG

Rev.2.00 May, 2024

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

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

Trademarks (continued)

Examples of trademark or registered trademark used in the document of R-IN32M4;
Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere.
All rights reserved.
Ethernet is a registered trademark of Fuji Xerox Co., Ltd.
IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers Inc.
TRON is an acronym for “The Real-time Operation system Nucleus”.
ITRON is an acronym for “Industrial TRON”.
μITRON is an acronym for “Micro Industrial TRON”.
TRON, ITRON, and μITRON do not refer to any specific product or products.
CC-Link and CC-Link IE Field are registered trademarks of the CC-Link Partner Association (CLPA).

Note that in each section of the Manual, trademark notation of ® and TM may be omitted.
All other trademarks and registered trademarks are the property of their respective owners.

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Purpose and Target Readers

This manual is intended for users who wish to understand the functions of industrial Ethernet communications ASSP (Application Specific Standard Product) “R-IN32M4-CL3” (R9A06G064MGBG, R9A06G064SGBG) and design application systems using it. It is assumed that the reader of this manual has general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

Related Documents

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such. Please be understanding of this beforehand. In addition, because we make document at development, planning of each core, the related document may be the document for individual customers. Last four digits of document number indicate version information of each document. Please download the latest document from our web site and refer to it.

The document related to R-IN32M4-CL3

Document Name	Document Number
R-IN32M4-CL3 User’s Manual: Hardware edition (This manual)	R18UZ0073EJ0100
R-IN32M4-CL3 User’s Manual: Gigabit Ethernet PHY edition	R18UZ0075EJ0100
R-IN32M4-CL3 User’s Manual: Board Design edition	R18UZ0074EJ0100
R-IN32M4-CL3 User’s Manual: CC-Link IE TSN edition	R18UZ0070EJ0100
R-IN32M4-CL3 User’s Manual: CC-Link IE Field edition	R18UZ0071EJ0100
R-IN32M4-CL3 Programming Manual: Driver	R18UZ0076EJ0100
R-IN32M4-CL3 Programming Manual: OS	R18UZ0072EJ0100

2. Notation of Numbers and Symbols

Weight in data notation: Left is high-order column, right is low-order column

Active low notation:

xxxZ (capital letter Z after pin name or signal name)

or xxx_N (capital letter _N after pin name or signal name)

or xxnx (pin name or signal name contains small letter n)

Note:

Explanation of (Note) in the text

Caution:

Item deserving extra attention

Remark:

Supplementary explanation to the text

Numeric notation:

Binary … xxxx , xxxxB or n’bxxxx (n bits)

Decimal … xxxx

Hexadecimal … xxxxH or n’hxxxx (n bits)

Prefixes representing powers of 2 (address space, memory capacity):

K (kilo)… $2^{10} = 1024$

M (mega)… $2^{20} = 1024^2$

G (giga)… $2^{30} = 1024^3$

Data Type:

Word … 32 bits

Halfword … 16 bits

Byte … 8 bits

Contents

1. Functional Overview

1.1 Overview

Compared to conventional technologies, Ethernet communications in the industrial field are strongly required to have high-performance functions such as higher speed real-time response. These requirements are not necessarily realized by traditional methods such as making the Ethernet processing itself hardware or using a dedicated CPU for high-speed network processing.

The R-IN32M4-CL3 Ethernet communications LSI from Renesas Electronics has the following functions to realize "higher speed real-time response" and "high-accuracy communications control (low jitter communications)" which are important for Ethernet communications in the factory automation field.

■ CPU
- Integrated Arm® Cortex®-M4 core
- Integrated hardware real-time OS accelerator (HW-RTOS)

■ Peripheral
- Timers, WDT, I²C, UART, CSI, CAN, and general-purpose I/O ports (GPIO)
- Serial flash memory controller

■ External memory/external MCU (microcontroller) interface
- External memory connection mode: 16- or 32-bit bus connection to external devices such as SRAM
- External MCU connection mode: 16- or 32-bit bus connection to host MCU

■ Ethernet
- Integrated Gigabit Ethernet MAC
- Integrated switching hub with cut-through transfer, IEEE 1588 timer, and Device Level Ring (DLR) functions
- Integrated 2-port 10/100/1000BASE-PHY
- Dedicated DMA controller and buffer for the network processor

■ Supported industrial Ethernet protocol
- CC-Link IE Field intelligent device station and remote device station
- CC-Link IE TSN remote station (authentication Classes: A and B)
- PROFINET RT
- EtherNet/IP
- Modbus TCP

■ Target application
- Remote I/O
- Inverter and servo drive
- Industrial Ethernet communication unit

■ Others
- Pin assignment considering a replacement from R-IN32M4-CL2
- Integrated 2.5 V regulator for PHY

1.2 Functional Overview

Table 1.1 Functional Overview of R-IN32M4-CL3 (1/3)

Item \ Product	R9A06G064MGBG (23 mm Square Package)	R9A06G064SGBG (17 mm Square Package)
CPU core	Arm Cortex-M4 32-bit RISC CPU + Real-Time OS Accelerator (Hardware Real-Time OS)	
Operating frequency	100 MHz	
Instruction set	Thumb®-2 instruction Armv7-M architecture	
Floating-point UNIT	Armv7M FPv4-SP (32-bit single precision)	
Instruction RAM	768 Kbytes (RAM with ECC)	
Data RAM	512 Kbytes (RAM with ECC)	
Buffer RAM	64 Kbytes (RAM with ECC)	
Network RAM	128 Kbytes (RAM with ECC)	
Internal system bus	32-bit system bus at 100 MHz (AHB-Lite) 64-bit system bus at 125 MHz (AXI) 128-bit communication bus at 100 MHz	
DMA function (system bus side)	4 channels + 1 channel (for real-time port), Supports software and various interrupt-triggered DMA.	
Boot modes	Serial flash ROM boot, External memory boot, External MCU boot	
Support for external memory access	• Bus-size selection (16 or 32 bits) • Paged ROM/ROM/SRAM interface • Synchronous burst memory interface • Programmable wait function	
Chip select signals for static memory	4-line	4- or 3-line*1
External memory space	256 Mbytes	256 or 192 Mbytes
External MCU interface	• Bus-size selection (16 or 32 bits) • General-purpose interface for static memory • Address space: 2 Mbytes (Instruction RAM, Data RAM, Register area) • Internal address space mapping switching function	
Serial flash ROM memory controller	• Supports serial interface compatible with SPI of companies. • Supports direct boot from serial memory device. • Supports Fast Read, Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, and Fast Read Quad I/O modes. • Direct layout in memory space	
Interrupt	• 30 external interrupt ports	

Note 1. When using an asynchronous SRAM controller in the 17 mm square package, the external memory areas of CSZ0–CSZ2 can be accessed.

Table 1.1 Functional Overview of R-IN32M4-CL3 (2/3)

Item \ Product		R9A06G064MGBG (23 mm Square Package)	R9A06G064SGBG (17 mm Square Package)
Internal peripheral modules			
	I/O port	CMOS I/O: 106 maximum	CMOS I/O: 101 maximum
	Timers (4 sub-systems)	• Internal timer of hardware RTOS • Internal timer of the CPU • 32-bit timer (4 channels) • 16-bit timer (16 channels)	
	Watchdog timer	• 1 channel • Software-triggered start mode • Watchdog error response options: – Generation of a non-maskable interrupt (NMI) – Generation of a reset • Interrupt when the counter reaches 75% of its overflow value	
	Asynchronous serial interface	• 2 channels • Full duplex transfer • FIFOs: 10 bits × 16 receive and 8 bits × 16 transmit • Support output of receive errors and status • Character length: 7 or 8 bits • Parity bit options: Odd, even, 0, none • Transmit stop bits: 1 bit or 2 bits	
	I^2C serial interface	• 2 channels • Operating modes: Normal or high-speed • Transfer modes: Single-transfer mode or continuous-transfer mode • Transfer data length: 8 bits	
	CAN controller	• 2 channels • Conforming to ISO11898 • Support for transmission and reception of standard and expanded frames • Transfer rate: Up to 1 Mbps	Not available
	Clocked serial interface	• 2 channels • Synchronized serial data transmission by three-wire system • Master mode or slave mode selectable • Built-in baud-rate generator • Transfer data length: 7 to 16 bits	
10/100/1000Mbps Ethernet MAC		• 1 channel • Built-in 2-port switch	
Ethernet PHY		• 2 ports IEEE 802.3 • 10BASE-T, 100BASE-TX, 1000BASE-T	
CC-Link IE		Two types of CC-Link IE (CC-Link IE Field and CC-Link IE TSN) are supported. They can be used exclusively	
	CC-Link IE Field	CC-Link IE Field (intelligent device station / remote device station)	
	CC-Link IE TSN	CC-Link IE TSN	

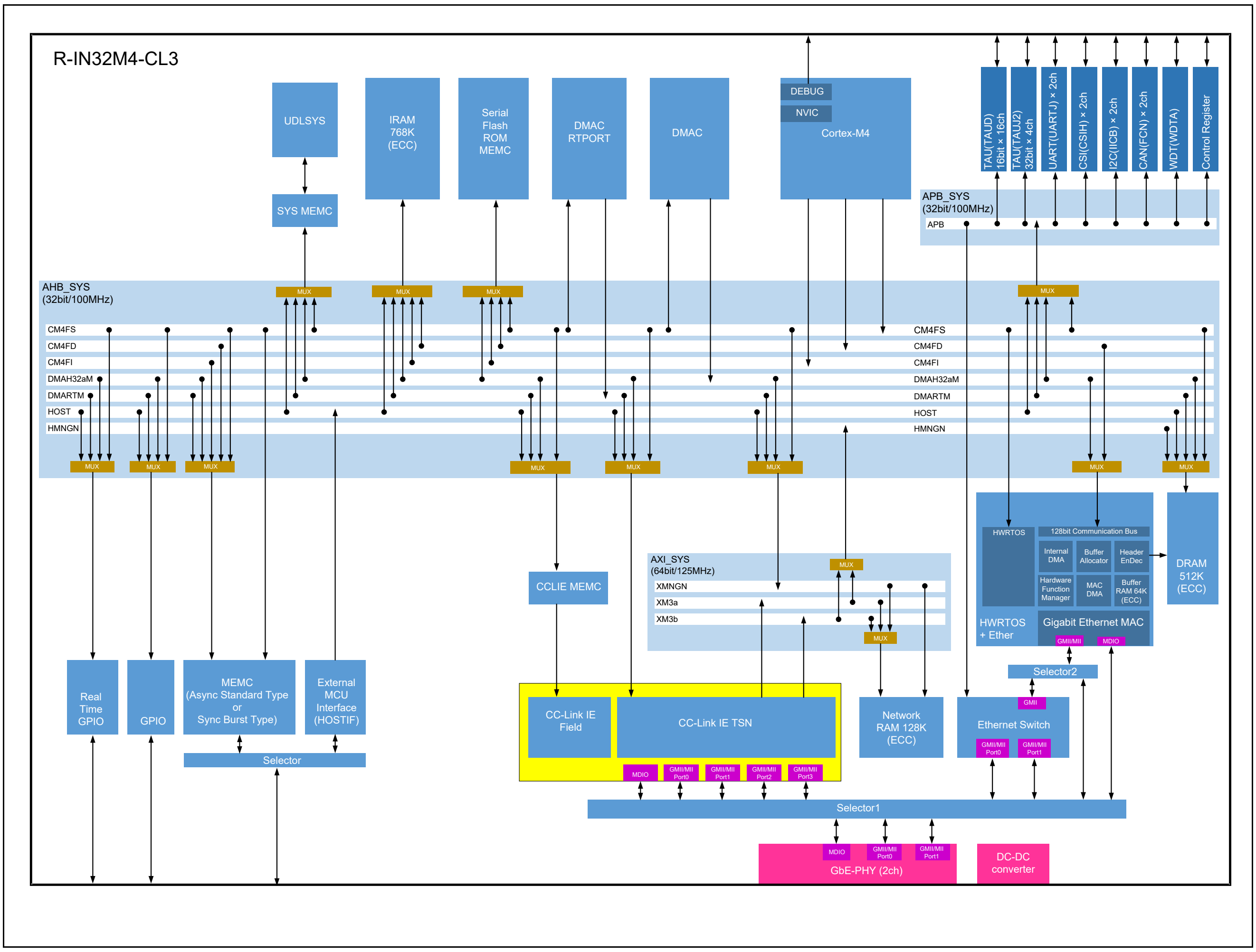
Table 1.1 Functional Overview of R-IN32M4-CL3 (3/3)

Item \ Product	R9A06G064MGBG (23 mm Square Package)	R9A06G064SGBG (17 mm Square Package)
On-chip debugging	• Selecting serial wire or JTAG • Full trace (built-in ETM)	
Internal PLL	Generates various clocks from 25 MHz input clock	
Built-in regulator	2.5V power supply dedicated to PHY can be generated from 3.3V power supply	
Power supply voltage	VDD33 = 3.3 ± 0.165 V*2 VDD11 = 1.15 ± 0.06 V*2 VDD25 = 2.5 ± 0.125 V*1, *2	
Operating temperature	-40°C ≤ Tj ≤ +125°C, -40°C ≤ Ta ≤ +85°C	
Packages	484-ball PBGA 23 mm × 23 mm, 1.0-mm Pitch	356-ball FBGA 17 mm × 17 mm, 0.8-mm Pitch

Note 1. 2.5 V power supply (VDD25) can be generated with the built-in regulator.

2. Ripple incorporated value. As a target value, set the DC component to within ±3% and the ripple component to within ±2%.

1.3 Internal Block Diagram



1.4 Pin Assignments (Top View)

1.4.1 23 mm Square Package Pin Assignments (Top View)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	
22	GND	GND	RP21	RP23	RP25	RP27	RP02	RP00	GND/ OPEN	GND/ OPEN	GND/ OPEN	P20	P22	P25	GND	P67	P65	P63	P60	P30	GND	GND	22
21	GND	RP20	RP22	RP24	RP26	RP04	RP03	RP01	GND/ OPEN	GND/ OPEN	GND/ OPEN	P21	P23	P26	P27	P66	P64	P62	P61	P31	P32	GND	21
20	RP30	RP32	RP10	RP11	RP12	RP13	RP07	RP05	GND	VDD33	GND/ OPEN	GND	P24	EXTP0	EXTP1	EXTP2	EXTP3	HWRZ SEL	HOT RESETZ	PONRZ	P33	RESETZ	20
19	RP31	RP33	RP37	RP14	RP15	RP16	RP17	RP06	GND	VDD33	GND/ OPEN	GND	GND	TEST7	CLK2M SEL	VDD33	MEMC SEL	ADMUX MODE	BUS32 EN	RST OUTZ	P35	P34	19
18	BUS CLK	RP34	RP36	D15	GND	VDD33	GND	VDD33	GND	VDD33	VDD33	GND	GND	VDD11	GND	GND	VDD33	TEST6	MEMIF SEL	BOOT0	GND	CCI_ CLK2_0 97M	18
17	D6	RP35	D13	D14	TEST3	VDD33	GND	VDD11	VDD11	VDD11	VDD11	VDD11	VDD11	VDD11	VDD11	GND	VDD33	PLL_ VDD	HIF SYNC	BOOT1	P36	GND	17
16	D4	D5	D11	D12	GND	VDD33	GND	VDD11	GND	GND	GND	GND	GND	GND	VDD11	GND	VDD33	PLL_ GND	EXTP9	EXTP8	P37	GND	16
15	D2	D3	D9	D10	GND	GND	GND	VDD11	GND	GND	GND	GND	GND	GND	VDD11	GND	GND	GND	EXTP7	EXTP6	P70	XT2	15
14	D0	D1	D7	D8	GND	VDD33	GND	VDD11	GND	GND	GND	GND	GND	GND	VDD11	GND	GND	GND	EXTP5	EXTP4	P71	XT1	14
13	RDZ	WRSTB Z	CSZ0	A20	GND	GND	GND	VDD11	VDD11	VDD11	VDD11	VDD11	VDD11	VDD11	VDD11	GND	VDD33	GND	OSCTH	NMIZ	P73	P72	13
12	P10	P12	WRZ0	A19	GND	VDD33	VDD33	GND	VDD33	GND	VDD33	VDD33	GND	VDD33	GND	VDD33	VDD33	GND	GND	TRACE CLK	P75	P74	12
11	P11	P13	WRZ1	A18	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	TRACE DATA1	TRACE DATA0	P77	P76	11
10	P14	P15	A17	GND	GND	GND	GND	GND	GND	GND	GND/ OPEN	GND	GND	GND	GND	GND	GND	GND	GND	TRACE DATA2	P01	P00	10
9	P16	P17	A15	A16	GND	GND	GND	GND	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	GND	GND	GND	GND	TRSTZ	TRACE DATA3	P03	P02	9
8	P47	P44	A13	A14	GND	GND	GND	GND	VDD33	GND	GND	GND	GND	VDD33	GND	GND	GND	GND	TDO	JTAG SEL	P05	P04	8
7	P45	P46	A11	A12	GND	GND	GND	GND	VDD11	GND	GND	GND	GND	VDD11	GND	GND	GND	GND	TCK	TMODE 2	P07	P06	7
6	P43	P41	A9	A10	REG_ EN	GND	AVDD REG_33	GND	VDD11	GND	GND	GND	GND	VDD11	GND	GND	GND	GND	TDI	TMODE 1	P51	P50	6
5	P42	A7	A8	PHY ADD1	AGND	GND	VDD REG_33	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	TMS	TMODE 0	P53	P52	5
4	P40	A5	A6	PHY ADD2	AGND	REG_ OUT	GND	REG_ FB	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	TEST5	TEST4	P55	P54	4
3	A2	A3	A4	PHY ADD3	GND	GND	GND	GND	GND	VDD11A	VDD11A	GND	VDD25A	VDD25A	VDD25A	GND	GND	GND	GND	GND	P57	P56	3
2	GND	PHY0_ LED0	PHY ADD4	GND	GND	GND	P0_D3N	P0_D2N	P0_D1N	P0_D0N	GND	REF_ FILT	GND	P1_D3N	P1_D2N	P1_D1N	P1_D0N	GND	GND	GND	PHY1_ LED0	GND	2
1	GND	GND	GND	GND	GND	GND	P0_D3P	P0_D2P	P0_D1P	P0_D0P	GND	REF_ REXT	GND	P1_D3P	P1_D2P	P1_D1P	P1_D0P	GND	GND	GND	GND	GND	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	

Figure 1.1 23 mm Square Package Pin Assignments (Top View)

1.4.2 17 mm Square Package Pin Assignments (Top View)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	
20	GND	GND	GND	RP21	RP23	RP25	RP27	RP05	RP02	P24	P25	P27	P66	P64	P60	P30	PONRZ	GND	GND	GND	20
19	GND	RP37	RP20	RP22	RP24	RP26	RP04	RP03	RP01	P21	P26	P67	P65	P63	P61	P31	EXTP7	EXTP12	P32	GND	19
18	RP36	RP30	RP32	RP14	RP11	RP16	RP17	RP06	RP00	P22	EXTP0	EXTP1	EXTP2	P62	EXTP3	HOT RESETZ	EXTP13	BUS32 EN	P33	RESETZ	18
17	RP34	RP31	RP33	RP10	RP15	RP12	RP13	RP07	P20	P23	TEST7	CLK2M SEL	EXTP14	MEMC SEL	HWRZ SEL	ADMUX MODE	TEST6	MEMIF SEL	RST OUTZ	P34	17
16	BUS CLK	RP35	D14	D15													PLL_ VDD	BOOT0	P35	GND	16
15	D11	D12	D13	D10		GND	VDD33	VDD33	GND	GND	GND	VDD33	VDD33	GND	GND		PLL_ GND	BOOT1	P36	CCI_ CLK2_ 097M	15
14	D4	D5	D6	D9		GND	VDD11	GND	GND	VDD11	VDD11	GND	GND	VDD11	GND		EXTP9	HIF SYNC	P37	GND	14
13	D2	D3	D7	D8		GND	VDD11	GND	GND	GND	GND	GND	GND	VDD11	VDD33		EXTP8	EXTP6	P70	XT2	13
12	CSZ0	D0	D1	A20		VDD33	GND	GND	GND	GND	GND	GND	GND	GND	VDD33		EXTP5	EXTP4	P71	XT1	12
11	WRSTB Z	RDZ	WRZ0	A19		VDD33	VDD11	GND	GND	GND	GND	GND	GND	VDD11	GND		NMIZ	OSCTH	P73	P72	11
10	P15	P14	WRZ1	A18		GND	VDD11	GND	GND	GND	GND	GND	GND	VDD11	GND		GND	TRACE CLK	P75	P74	10
9	P16	P17	A17	A15		GND	VDD11	GND	GND	GND	GND	GND	GND	GND	GND		TRACE DATA1	TRACE DATA0	P77	P76	9
8	P47	P44	A13	A16		GND	VDD11	GND	GND	GND	GND	GND	GND	VDD11	VDD33		TRACE DATA3	TRACE DATA2	P01	P00	8
7	P45	P46	A14	A11		VDD33	GND	GND	VDD11	VDD11	GND	GND	GND	GND	VDD33		TRSTZ	EXTP10	P03	P02	7
6	P43	P41	A7	A12		REG_ EN	GND	GND	GND	GND	GND	GND	VDD11A	VDD11A	GND		TMODE 2	TDO	P05	P04	6
5	P42	P40	A9	A6													TCK	TMODE 1	JTAG SEL	EXTP11	5
4	A10	A8	A2	A3	GND	GND	GND	REG_ FB	GND	VDD25A	VDD25A	VDD25A	GND	GND	GND	TEST4	TEST5	TDI	P52	P50	4
3	A5	A4	GND	AGND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	TMODE 0	P57	P51	3
2	GND	PHY0_ LED0	REG_ OUT	AGND	GND	P0_D3N	P0_D2N	P0_D1N	P0_D0N	GND	REF_ FILT	GND	P1_D3N	P1_D2N	P1_D1N	P1_D0N	GND	PHY1_ LED0	TMS	GND	2
1	GND	GND	VDD REG_33	AVDD REG_33	GND	P0_D3P	P0_D2P	P0_D1P	P0_D0P	GND	REF_ REXT	GND	P1_D3P	P1_D2P	P1_D1P	P1_D0P	GND	GND	GND	GND	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	

Figure 1.2 17 mm Square Package Pin Assignments (Top View)

1.5 External Pin List

1.5.1 23 mm Square Package

Table 1.2 23 mm Square Package External Pin List (1/4)

Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name
A1	GND	B15	D3	D7	A12	E21	RP26
A2	GND	B16	D5	D8	A14	E22	RP25
A3	A2	B17	RP35	D9	A16	F1	GND
A4	P40	B18	RP34	D10	GND	F2	GND
A5	P42	B19	RP33	D11	A18	F3	GND
A6	P43	B20	RP32	D12	A19	F4	REG_OUT
A7	P45	B21	RP20	D13	A20	F5	GND
A8	P47	B22	GND	D14	D8	F6	GND
A9	P16	C1	GND	D15	D10	F7	GND
A10	P14	C2	PHYADD4	D16	D12	F8	GND
A11	P11	C3	A4	D17	D14	F9	GND
A12	P10	C4	A6	D18	D15	F10	GND
A13	RDZ	C5	A8	D19	RP14	F11	GND
A14	D0	C6	A9	D20	RP11	F12	VDD33
A15	D2	C7	A11	D21	RP24	F13	GND
A16	D4	C8	A13	D22	RP23	F14	VDD33
A17	D6	C9	A15	E1	GND	F15	GND
A18	BUSCLK	C10	A17	E2	GND	F16	VDD33
A19	RP31	C11	WRZ1	E3	GND	F17	VDD33
A20	RP30	C12	WRZ0	E4	AGND	F18	VDD33
A21	GND	C13	CSZ0	E5	AGND	F19	RP16
A22	GND	C14	D7	E6	REG_EN	F20	RP13
B1	GND	C15	D9	E7	GND	F21	RP04
B2	PHY0_LED0	C16	D11	E8	GND	F22	RP27
B3	A3	C17	D13	E9	GND	G1	P0_D3P
B4	A5	C18	RP36	E10	GND	G2	P0_D3N
B5	A7	C19	RP37	E11	GND	G3	GND
B6	P41	C20	RP10	E12	GND	G4	GND
B7	P46	C21	RP22	E13	GND	G5	VDDREG_33
B8	P44	C22	RP21	E14	GND	G6	AVDDREG_33
B9	P17	D1	GND	E15	GND	G7	GND
B10	P15	D2	GND	E16	GND	G8	GND
B11	P13	D3	PHYADD3	E17	TEST3	G9	GND
B12	P12	D4	PHYADD2	E18	GND	G10	GND
B13	WRSTBZ	D5	PHYADD1	E19	RP15	G11	GND
B14	D1	D6	A10	E20	RP12	G12	VDD33

Table 1.2 23 mm Square Package External Pin List (2/4)

Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name
G13	GND	J9	VDD33	L5	GND	N1	GND
G14	GND	J10	GND	L6	GND	N2	GND
G15	GND	J11	GND	L7	GND	N3	VDD25A
G16	GND	J12	VDD33	L8	GND	N4	GND
G17	GND	J13	VDD11	L9	VDD33	N5	GND
G18	GND	J14	GND	L10	GND/OPEN	N6	GND
G19	RP17	J15	GND	L11	GND	N7	GND
G20	RP07	J16	GND	L12	VDD33	N8	GND
G21	RP03	J17	VDD11	L13	VDD11	N9	VDD33
G22	RP02	J18	GND	L14	GND	N10	GND
H1	P0_D2P	J19	GND	L15	GND	N11	GND
H2	P0_D2N	J20	GND	L16	GND	N12	GND
H3	GND	J21	GND/OPEN	L17	VDD11	N13	VDD11
H4	REG_FB	J22	GND/OPEN	L18	VDD33	N14	GND
H5	GND	K1	P0_D0P	L19	GND/OPEN	N15	GND
H6	GND	K2	P0_D0N	L20	GND/OPEN	N16	GND
H7	GND	K3	VDD11A	L21	GND/OPEN	N17	VDD11
H8	GND	K4	GND	L22	GND/OPEN	N18	GND
H9	GND	K5	GND	M1	REF_REXT	N19	GND
H10	GND	K6	GND	M2	REF_FILT	N20	P24
H11	GND	K7	GND	M3	GND	N21	P23
H12	GND	K8	GND	M4	GND	N22	P22
H13	VDD11	K9	VDD33	M5	GND	P1	P1_D3P
H14	VDD11	K10	GND	M6	GND	P2	P1_D3N
H15	VDD11	K11	GND	M7	GND	P3	VDD25A
H16	VDD11	K12	GND	M8	GND	P4	GND
H17	VDD11	K13	VDD11	M9	VDD33	P5	GND
H18	VDD33	K14	GND	M10	GND	P6	VDD11
H19	RP06	K15	GND	M11	GND	P7	VDD11
H20	RP05	K16	GND	M12	VDD33	P8	VDD33
H21	RP01	K17	VDD11	M13	VDD11	P9	VDD33
H22	RP00	K18	VDD33	M14	GND	P10	GND
J1	P0_D1P	K19	VDD33	M15	GND	P11	GND
J2	P0_D1N	K20	VDD33	M16	GND	P12	VDD33
J3	GND	K21	GND/OPEN	M17	VDD11	P13	VDD11
J4	GND	K22	GND/OPEN	M18	GND	P14	GND
J5	GND	L1	GND	M19	GND	P15	GND
J6	VDD11	L2	GND	M20	GND	P16	GND
J7	VDD11	L3	VDD11A	M21	P21	P17	VDD11
J8	VDD33	L4	GND	M22	P20	P18	VDD11

Table 1.2 23 mm Square Package External Pin List (3/4)

Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name
P19	TEST7	T15	GND	V11	GND	Y7	TMODE2
P20	EXTP0	T16	GND	V12	GND	Y8	JTAGSEL
P21	P26	T17	GND	V13	GND	Y9	TRACEDATA3
P22	P25	T18	GND	V14	GND	Y10	TRACEDATA2
R1	P1_D2P	T19	VDD33	V15	GND	Y11	TRACEDATA0
R2	P1_D2N	T20	EXTP2	V16	PLL_GND	Y12	TRACECLK
R3	VDD25A	T21	P66	V17	PLL_VDD	Y13	NMIZ
R4	GND	T22	P67	V18	TEST6	Y14	EXTP4
R5	GND	U1	P1_D0P	V19	ADMUXMODE	Y15	EXTP6
R6	GND	U2	P1_D0N	V20	HWRZSEL	Y16	EXTP8
R7	GND	U3	GND	V21	P62	Y17	BOOT1
R8	GND	U4	GND	V22	P63	Y18	BOOT0
R9	GND	U5	GND	W1	GND	Y19	RSTOUTZ
R10	GND	U6	GND	W2	GND	Y20	PONRZ
R11	GND	U7	GND	W3	GND	Y21	P31
R12	GND	U8	GND	W4	TEST5	Y22	P30
R13	VDD11	U9	GND	W5	TMS	AA1	GND
R14	VDD11	U10	GND	W6	TDI	AA2	PHY1_LED0
R15	VDD11	U11	GND	W7	TCK	AA3	P57
R16	VDD11	U12	VDD33	W8	TDO	AA4	P55
R17	VDD11	U13	VDD33	W9	TRSTZ	AA5	P53
R18	GND	U14	GND	W10	GND	AA6	P51
R19	CLK2MSEL	U15	GND	W11	TRACEDATA1	AA7	P07
R20	EXTP1	U16	VDD33	W12	GND	AA8	P05
R21	P27	U17	VDD33	W13	OSCTH	AA9	P03
R22	GND	U18	VDD33	W14	EXTP5	AA10	P01
T1	P1_D1P	U19	MEMCSEL	W15	EXTP7	AA11	P77
T2	P1_D1N	U20	EXTP3	W16	EXTP9	AA12	P75
T3	GND	U21	P64	W17	HIFSYNC	AA13	P73
T4	GND	U22	P65	W18	MEMIFSEL	AA14	P71
T5	GND	V1	GND	W19	BUS32EN	AA15	P70
T6	GND	V2	GND	W20	HOTRESETZ	AA16	P37
T7	GND	V3	GND	W21	P61	AA17	P36
T8	GND	V4	GND	W22	P60	AA18	GND
T9	GND	V5	GND	Y1	GND	AA19	P35
T10	GND	V6	GND	Y2	GND	AA20	P33
T11	GND	V7	GND	Y3	GND	AA21	P32
T12	VDD33	V8	GND	Y4	TEST4	AA22	GND
T13	GND	V9	GND	Y5	TMODE0	AB1	GND
T14	GND	V10	GND	Y6	TMODE1	AB2	GND

Table 1.2 23 mm Square Package External Pin List (4/4)

Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name
AB3	P56	AB8	P04	AB13	P72	AB18	CCI_CLK2 _097M
AB4	P54	AB9	P02	AB14	XT1	AB19	P34
AB5	P52	AB10	P00	AB15	XT2	AB20	RESETZ
AB6	P50	AB11	P76	AB16	GND	AB21	GND
AB7	P06	AB12	P74	AB17	GND	AB22	GND

1.5.2 17 mm Square Package

Table 1.3 17 mm Square Package External Pin List (1/3)

Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name
A1	GND	B19	RP37	D17	RP10	G10	VDD11
A2	GND	B20	GND	D18	RP14	G11	VDD11
A3	A5	C1	VDDREG_33	D19	RP22	G12	GND
A4	A10	C2	REG_OUT	D20	RP21	G13	VDD11
A5	P42	C3	GND	E1	GND	G14	VDD11
A6	P43	C4	A2	E2	GND	G15	VDD33
A7	P45	C5	A9	E3	GND	G17	RP13
A8	P47	C6	A7	E4	GND	G18	RP17
A9	P16	C7	A14	E17	RP15	G19	RP04
A10	P15	C8	A13	E18	RP11	G20	RP27
A11	WRSTBZ	C9	A17	E19	RP24	H1	P0_D1P
A12	CSZ0	C10	WRZ1	E20	RP23	H2	P0_D1N
A13	D2	C11	WRZ0	F1	P0_D3P	H3	GND
A14	D4	C12	D1	F2	P0_D3N	H4	REG_FB
A15	D11	C13	D7	F3	GND	H6	GND
A16	BUSCLK	C14	D6	F4	GND	H7	GND
A17	RP34	C15	D13	F6	REG_EN	H8	GND
A18	RP36	C16	D14	F7	VDD33	H9	GND
A19	GND	C17	RP33	F8	GND	H10	GND
A20	GND	C18	RP32	F9	GND	H11	GND
B1	GND	C19	RP20	F10	GND	H12	GND
B2	PHY0_LED0	C20	GND	F11	VDD33	H13	GND
B3	A4	D1	AVDDREG_33	F12	VDD33	H14	GND
B4	A8	D2	AGND	F13	GND	H15	VDD33
B5	P40	D3	AGND	F14	GND	H17	RP07
B6	P41	D4	A3	F15	GND	H18	RP06
B7	P46	D5	A6	F17	RP12	H19	RP03
B8	P44	D6	A12	F18	RP16	H20	RP05
B9	P17	D7	A11	F19	RP26	J1	P0_D0P
B10	P14	D8	A16	F20	RP25	J2	P0_D0N
B11	RDZ	D9	A15	G1	P0_D2P	J3	GND
B12	D0	D10	A18	G2	P0_D2N	J4	GND
B13	D3	D11	A19	G3	GND	J6	GND
B14	D5	D12	A20	G4	GND	J7	VDD11
B15	D12	D13	D8	G6	GND	J8	GND
B16	RP35	D14	D9	G7	GND	J9	GND
B17	RP31	D15	D10	G8	VDD11	J10	GND
B18	RP30	D16	D15	G9	VDD11	J11	GND

Table 1.3 17 mm Square Package External Pin List (2/3)

Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name
J12	GND	L17	TEST7	P1	P1_D2P	T17	ADMUXMODE
J13	GND	L18	EXTP0	P2	P1_D2N	T18	HOTRESETZ
J14	GND	L19	P26	P3	GND	T19	P31
J15	GND	L20	P25	P4	GND	T20	P30
J17	P20	M1	GND	P6	VDD11A	U1	GND
J18	RP00	M2	GND	P7	GND	U2	GND
J19	RP01	M3	GND	P8	VDD11	U3	GND
J20	RP02	M4	VDD25A	P9	GND	U4	TEST5
K1	GND	M6	GND	P10	VDD11	U5	TCK
K2	GND	M7	GND	P11	VDD11	U6	TMODE2
K3	GND	M8	GND	P12	GND	U7	TRSTZ
K4	VDD25A	M9	GND	P13	VDD11	U8	TRACEDATA3
K6	GND	M10	GND	P14	VDD11	U9	TRACEDATA1
K7	VDD11	M11	GND	P15	GND	U10	GND
K8	GND	M12	GND	P17	MEMCSEL	U11	NMIZ
K9	GND	M13	GND	P18	P62	U12	EXTP5
K10	GND	M14	GND	P19	P63	U13	EXTP8
K11	GND	M15	VDD33	P20	P64	U14	EXTP9
K12	GND	M17	CLK2MSEL	R1	P1_D1P	U15	PLL_GND
K13	GND	M18	EXTP1	R2	P1_D1N	U16	PLL_VDD
K14	VDD11	M19	P67	R3	GND	U17	TEST6
K15	GND	M20	P27	R4	GND	U18	EXTP13
K17	P23	N1	P1_D3P	R6	GND	U19	EXTP7
K18	P22	N2	P1_D3N	R7	VDD33	U20	PONRZ
K19	P21	N3	GND	R8	VDD33	V1	GND
K20	P24	N4	GND	R9	GND	V2	PHY1_LED0
L1	REF_REXT	N6	VDD11A	R10	GND	V3	TMODE0
L2	REF_FILT	N7	GND	R11	GND	V4	TDI
L3	GND	N8	GND	R12	VDD33	V5	TMODE1
L4	VDD25A	N9	GND	R13	VDD33	V6	TDO
L6	GND	N10	GND	R14	GND	V7	EXTP10
L7	GND	N11	GND	R15	GND	V8	TRACEDATA2
L8	GND	N12	GND	R17	HWRZSEL	V9	TRACEDATA0
L9	GND	N13	GND	R18	EXTP3	V10	TRACECLK
L10	GND	N14	GND	R19	P61	V11	OSCTH
L11	GND	N15	VDD33	R20	P60	V12	EXTP4
L12	GND	N17	EXTP14	T1	P1_D0P	V13	EXTP6
L13	GND	N18	EXTP2	T2	P1_D0N	V14	HIFSYNC
L14	VDD11	N19	P65	T3	GND	V15	BOOT1
L15	GND	N20	P66	T4	TEST4	V16	BOOT0

Table 1.3 17 mm Square Package External Pin List (3/3)

Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name	Pin ID	Pin Name
V17	MEMIFSEL	W8	P01	W19	P32	Y10	P74
V18	BUS32EN	W9	P77	W20	GND	Y11	P72
V19	EXTP12	W10	P75	Y1	GND	Y12	XT1
V20	GND	W11	P73	Y2	GND	Y13	XT2
W1	GND	W12	P71	Y3	P51	Y14	GND
W2	TMS	W13	P70	Y4	P50	Y15	CCI_CLK2_097M
W3	P57	W14	P37	Y5	EXTP11	Y16	GND
W4	P52	W15	P36	Y6	P04	Y17	P34
W5	JTAGSEL	W16	P35	Y7	P02	Y18	RESETZ
W6	P05	W17	RSTOUTZ	Y8	P00	Y19	GND
W7	P03	W18	P33	Y9	P76	Y20	GND

1.6 Base Addresses

The addresses of each register described in the following sections are the relative addresses from the base addresses. In access to the registers via the external MCU interface, the base address is D_0000H. In access by the internal CPU or DMA controller, the base address is 4001_0000H.

• In access by the internal CPU or DMA controller: BASE = 4001_0000H
• In access via the external MCU interface: BASE = D_0000H

2. Pin Function

The following tables list the meanings of the items, symbols, and abbreviations used in each pin table in this chapter. Some pins and functions are not available depending on the type of package. Refer to the PKG column.

Table 2.1 Meanings of the Items in the Pin Lists

Item	Meaning
Function Name	Name of a function of the pin under “Pin Name” below.
Pin Name	Name of the pin shown in Section 1.4 “Pin Assignments (Top View)”.
PKG	Type of package 23□: 23 mm Square Package 17□: 17 mm Square Package
I/O	I/O direction of the given pin
Description	Summary of the given pin function
Active	Active level of the given pin
Level during Reset	The pin state while RSTOUTZ is Low. For details on the reset specifications, refer to Section 4.3 “Reset Function”.

Table 2.2 Meanings of the Symbols and Abbreviations in the Pin Lists

Target	Symbol and Abbreviation	Meaning
Pin Name	— (hyphen)	The pin is a dedicated pin that is not multiplexed with a port-pin function.
PKG	○	The pin exists.
	×	The pin does not exist.
I/O	— (hyphen)	The pin does not have an I/O direction, such as a power supply or ground pin.
Active	— (hyphen)	There is no active level (clock pins, data pins, and address pins).
	High	The active level is high.
	Low	The active level is low.
Level during Reset	— (hyphen)	This is an input-dedicated pin that has no initial level or state following a reset.
	High	The pin state during a reset is high.
	Low	The pin state during a reset is low.
	Hi-Z (High)	The pin state during a reset is Hi-Z (high) with the internal pull-up resistor pulling it to the high level.
	Hi-Z (Low)	The pin state during a reset is Hi-Z (Low) with the internal pull-down resistor pulling it to the Low level.

2.1 Pin List by function

The pins described in Section 2.1.2 “Ethernet Pins” to Section 2.1.15 “Operating Mode Setting Pins" are multiplexed with port pins described in Section 2.1.1 “Port Pins and Real-Time Port Pins”. For details, refer to Multiplexed function 1 to Multiplexed function 4 in Section 2.1.1 “Port Pins and Real-Time Port Pins”.

2.1.1 Port Pins and Real-Time Port Pins

The LSI has 13 ports for the 3.3 V interface, all of which are 8-bit ports except for EXTP, which has 15 bits. Grouping them into sets of four ports allows 32-bit access: for example, through ports 0 to 3 (P00–P37), ports 4 to 7 (P40–P77), and real-time ports 0 to 3 (RP00–RP37).

(1/5)

Pin Name	PKG		Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function 4	Level during Reset
	23□	17□					
P00	○	○	INTPZ0	—	CCI_RUNLEDZ	—	Hi-Z (High)
P01	○	○	INTPZ1	—	—	—	
P02	○	○	INTPZ2	—	CCI_DLINKLEDZ	—	
P03	○	○	INTPZ3	—	CCI_ERRLEDZ	—	
P04	○	○	INTPZ4	—	CCI_LERR1LEDZ	—	
P05	○	○	INTPZ5	—	CCI_LERR2LEDZ	—	
P06	○	×	—	—	CCI_SDLEDZ	—	
P07	○	×	—	—	CCI_RDLEDZ	—	
P10	○	×	SMIO2	—	—	—	
P11	○	×	SMIO3	—	—	—	
P12	○	×	CSZ3	—	CCI_WDTIZ	—	
P13	○	×	CSZ2	—	—	—	
P14	○	○	SMSCK	—	—	—	
P15	○	○	SMIO0	—	—	—	
P16	○	○	SMIO1	—	—	—	
P17	○	○	SMCSZ	—	—	—	
P20	○	○	RXD0	—	—	—	
P21	○	○	TXD0	—	—	—	
P22	○	○	INTPZ8	—	—	—	
P23	○	○	INTPZ9	—	—	—	
P24	○	○	INTPZ10	ETHSWSYNCOUT	—	—	
P25	○	○	WDTOUTZ	—	—	—	
P26	○	○	TINJ1 / TIND5*1	TOUTJ1 / TOUTD5*1	—	—	
P27	○	○	TINJ0 / TIND4*1	TOUTJ0 / TOUTD4*1	—	—	

Note 1. Enabling the TAUJ2 or TAUD pin function is selectable by using the TMISEL register.
For details, refer to Section 28.17 “Timer Interface Select Register (TMISEL)”.

(2/5)

Pin Name	PKG 23□	PKG 17□	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function 4	Level during Reset
P30	○	○	RXD1	—	—	—	Hi-Z (High)
P31	○	○	TXD1	—	—	—	
P32	○	○	DMAREQZ1	—	—	—	
P33	○	○	DMAACKZ1	—	—	—	
P34	○	○	DMATCZ1	—	—	—	
P35	○	○	CSISCK1	INTPZ22	—	—	Hi-Z (Low)
P36	○	○	CSISI1	INTPZ23	—	—	Hi-Z (High)
P37	○	○	CSISO1	INTPZ24	—	—	Hi-Z (Low)
P40	○	○	A1 / MA0	HA1	—	—	Hi-Z (High)
P41	○	○	WAITZ	HWAITZ	INTPZ29	—	
P42	○	○	CSICS00	HERROUTZ	—	—	
P43	○	○	CSICS01	HBUSCLK	—	—	
P44	○	○	CSZ1	HPGCSZ	—	—	
P45	○	○	CSISCK0	WAITZ1	—	—	
P46	○	○	CSISI0	WAITZ2	—	—	
P47	○	○	CSISO0	WAITZ3	—	—	
P50	○	○	INTPZ6	—	—	—	
P51	○	○	INTPZ7	—	—	—	Hi-Z (Low)
P52	○	○	TINJ3 / TIND7*1	TOUTJ3 / TOUTD7*1	CCI_NMIZ	—	Hi-Z (High)
P53	○	×	CRXD0	CCI_INTZ	—	—	
P54	○	×	CTXD0	—	—	—	
P55	○	×	CRXD1	—	—	—	
P56	○	×	CTXD1	—	—	—	
P57	○	○	TINJ2 / TIND6*1	TOUTJ2 / TOUTD6*1	—	—	

Note 1. Enabling the TAUJ2 or TAUD pin function is selectable by using the TMISEL register.
For details, refer to Section 28.17 "Timer Interface Select Register (TMISEL)".

(3/5)

Pin Name	PKG		Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function 4	Level during Reset
	23□	17□					
P60	○	○	SCL0	—	—	—	Hi-Z (High)
P61	○	○	SDA0	—	—	—	
P62	○	○	RTDMAREQZ	—	—	—	
P63	○	○	RTDMAACKZ	—	—	—	
P64	○	○	RTDMATCZ	—	—	—	
P65	○	○	DMAREQZ0	—	—	—	
P66	○	○	DMAACKZ0	—	—	—	
P67	○	○	DMATCZ0	—	—	—	
P70	○	○	CSICS10	—	—	—	
P71	○	○	CSICS11	—	—	—	
P72	○	○	SLEEPING	—	—	—	
P73	○	○	INTPZ11	—	—	—	
P74	○	○	INTPZ12	—	—	—	
P75	○	○	INTPZ13	—	—	—	
P76	○	○	INTPZ14	—	—	—	
P77	○	○	INTPZ15	—	—	—	

(4/5)

Pin Name	PKG		Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function 4	Level during Reset
	23□	17□					
EXTP0	○	○	—	TOUTD0	—	TIND0	Hi-Z (High)
EXTP1	○	○	—	TOUTD1	—	TIND1	
EXTP2	○	○	—	TOUTD2	—	TIND2	
EXTP3	○	○	WDTOUTZ	TOUTD3	—	TIND3	
EXTP4	○	○	—	—	—	—	
EXTP5	○	○	—	—	—	—	
EXTP6	○	○	—	—	—	—	Hi-Z (Low)
EXTP7	○	○	—	—	—	—	Hi-Z (High)
EXTP8	○	○	—	—	—	—	
EXTP9	○	○	—	—	—	—	
EXTP10	×	○	SMIO2	CCI_INTZ	—	—	
EXTP11	×	○	SMIO3	CCI_WDTIZ	—	—	
EXTP12	×	○	CSZ3	CCI_SDLEDZ	—	—	
EXTP13	×	○	CSZ2	CCI_RDLEDZ	—	—	
EXTP14	×	○	IETYPE_LED	—	—	—	

Ports RP0x to RP3x (x: 0–7) operate as real-time ports. These ports can handle input and output in 32-bit units in synchronization with the DMA transfer trigger from the dedicated DMA controller for the real-time ports.

(5/5)

Pin Name	PKG 23□	PKG 17□	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function 4	Level during Reset
RP00	○	○	INTPZ16	SCL1	—	—	Hi-Z (High)
RP01	○	○	INTPZ17	SDA1	—	—	
RP02	○	○	INTPZ18	—	—	—	
RP03	○	○	INTPZ19	—	—	—	
RP04	○	○	INTPZ20	—	—	—	
RP05	○	○	INTPZ21	—	—	—	
RP06	○	○	WRZ2 / BENZ2	HWRZ2 / HBENZ2	—	—	
RP07	○	○	WRZ3 / BENZ3	HWRZ3 / HBENZ3	—	—	
RP10	○	○	D24 / MD24 / HD24	LED0_PHY0	—	—	
RP11	○	○	D25 / MD25 / HD25	LED1_PHY0	—	—	
RP12	○	○	D26 / MD26 / HD26	LED2_PHY0	—	—	
RP13	○	○	D27 / MD27 / HD27	LED3_PHY0	—	—	
RP14	○	○	D28 / MD28 / HD28	LED0_PHY1	—	—	
RP15	○	○	D29 / MD29 / HD29	LED1_PHY1	—	—	
RP16	○	○	D30 / MD30 / HD30	LED2_PHY1	—	—	
RP17	○	○	D31 / MD31 / HD31	LED3_PHY1	—	—	
RP20	○	○	BCYSTZ / ADVZ	HBCYSTZ	—	—	
RP21	○	×	A21 / MA20	—	—	—	Hi-Z (Low)
	×	○					Hi-Z (High)
RP22	○	×	A22 / MA21	—	—	—	Hi-Z (Low)
	×	○					Hi-Z (High)
RP23	○	×	A23 / MA22	—	—	—	Hi-Z (Low)
	×	○					Hi-Z (High)
RP24	○	×	A24 / MA23	INTPZ25	—	—	Hi-Z (Low)
	×	○					Hi-Z (High)
RP25	○	×	A25 / MA24	INTPZ26	—	—	Hi-Z (Low)
	×	○					Hi-Z (High)
RP26	○	×	A26 / MA25	INTPZ27	—	—	Hi-Z (Low)
	×	○					Hi-Z (High)
RP27	○	×	A27 / MA26	INTPZ28	—	—	Hi-Z (Low)
	×	○					Hi-Z (High)
RP30	○	○	D16 / MD16 / HD16	TOUTD8	TIND8	—	
RP31	○	○	D17 / MD17 / HD17	TOUTD9	TIND9	—	
RP32	○	○	D18 / MD18 / HD18	TOUTD10	TIND10	—	
RP33	○	○	D19 / MD19 / HD19	TOUTD11	TIND11	—	
RP34	○	○	D20 / MD20 / HD20	TOUTD12	TIND12	—	
RP35	○	○	D21 / MD21 / HD21	TOUTD13	TIND13	—	
RP36	○	○	D22 / MD22 / HD22	TOUTD14	TIND14	—	
RP37	○	○	D23 / MD23 / HD23	TOUTD15	TIND15	—	

2.1.2 Ethernet Pins

(1/2)

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
P0_D0N	—	○	○	I/O	PHY 0 Tx/Rx channel A negative signal	—	—
P0_D0P	—	○	○		PHY 0 Tx/Rx channel A positive signal		
P0_D1N	—	○	○		PHY 0 Tx/Rx channel B negative signal		
P0_D1P	—	○	○		PHY 0 Tx/Rx channel B positive signal		
P0_D2N	—	○	○		PHY 0 Tx/Rx channel C negative signal		
P0_D2P	—	○	○		PHY 0 Tx/Rx channel C positive signal		
P0_D3N	—	○	○		PHY 0 Tx/Rx channel D negative signal		
P0_D3P	—	○	○		PHY 0 Tx/Rx channel D positive signal		
P1_D0N	—	○	○		PHY 1 Tx/Rx channel A negative signal		
P1_D0P	—	○	○		PHY 1 Tx/Rx channel A positive signal		
P1_D1N	—	○	○		PHY 1 Tx/Rx channel B negative signal		
P1_D1P	—	○	○		PHY 1 Tx/Rx channel B positive signal		
P1_D2N	—	○	○		PHY 1 Tx/Rx channel C negative signal		
P1_D2P	—	○	○		PHY 1 Tx/Rx channel C positive signal		
P1_D3N	—	○	○		PHY 1 Tx/Rx channel D negative signal		
P1_D3P	—	○	○		PHY 1 Tx/Rx channel D positive signal		
PHYADD1	—	○	×	I	Device SMI Address bit 1 (with Pull-Down resistance)		
PHYADD2	—	○	×		Device SMI Address bit 2 (with Pull-Down resistance)		
PHYADD3	—	○	×		Device SMI Address bit 3 (with Pull-Down resistance)		
PHYADD4	—	○	×		Device SMI Address bit 4 (with Pull-Down resistance)		
REF_FILT	—	○	○	I/O	Copper media reference filter pin.		
REF_REXT	—	○	○		Copper media reference external pin.		
VDD11A	—	○	○	—	1.15 V analog power requiring additional PCB power supply filtering		
VDD25A	—	○	○		2.5 V general analog power supply		
PHY0_LED0	—	○	○	O	GbE-PHY LED0_PHY0 output signal	Low	High
PHY1_LED0	—	○	○		GbE-PHY LED0_PHY1 output signal		
LED0_PHY0	RP10	○	○		GbE-PHY LED signal output (Same signal as external pin PHY0_LED0)		Hi-Z (High)
LED1_PHY0	RP11	○	○		GbE-PHY LED signal output		
LED2_PHY0	RP12	○	○		GbE-PHY LED signal output		
LED3_PHY0	RP13	○	○		GbE-PHY LED signal output		
LED0_PHY1	RP14	○	○		GbE-PHY LED signal output (Same signal as external pin PHY1_LED0)		
LED1_PHY1	RP15	○	○		GbE-PHY LED signal output		
LED2_PHY1	RP16	○	○		GbE-PHY LED signal output		
LED3_PHY1	RP17	○	○		GbE-PHY LED signal output		
ETHSWSYNCOUT	P24	○	○		Ethernet switch event output	High	

(2/2)

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
VDDREG_33	—	○	○	—	3.3 V power for 2.5 V regulator	—	—
AVDDREG_33	—	○	○		3.3 V analog power for 2.5 V regulator		
REG_EN	—	○	○	I	2.5 V Regulator enable		
REG_FB	—	○	○		Feedback from the supply regulation point		
AGND	—	○	○	—	Analog ground for regulator		
REG_OUT*[1]	—	○	○	O	2.5 V Regulator output		

Note 1. The power supply of 2.5 V via the REG_OUT pin is dedicated to the VDD25A.
The pin is not available for the power supply of 2.5 V to other devices.

2.1.3 External SRAM and External MCU Interface Pins

Usage of the external SRAM interface pins and external MCU interface pins is exclusive.
This setting is selected by the level of the MEMIFSEL pin. (Setting value: Low level for the external SRAM interface pins and High level for the external MCU interface pins)

2.1.3.1 External SRAM Interface Pins

(a) External SRAM Interface Pins (when Asynchronous SRAM Controller is Selected (MEMCSEL = 0))

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
BUSCLK	—	○	○	O	Bus clock output	—	Clock output
CSZ0	—	○	○		Chip select signal output	Low	Hi-Z (High)
CSZ1	P44	○	○				
CSZ2	P13	○	×				
	EXTP13	×	○				
CSZ3	P12	○	×				
	EXTP12	×	○				
A1	P40	○	○		Address output	—	
A2–A20	—	○	○				Hi-Z (Low)
A21–A27	RP21–RP27	○	○				Hi-Z (High)
D0–D15	—	○	○	I/O	Data bus		Hi-Z (Low)
D16–D31	RP30–RP37, RP10–RP17	○	○				Hi-Z (High)
RDZ	—	○	○	O	Read strobe output	Low	
WRSTBZ	—	○	○		Write strobe output		
WRZ0 / BENZ0*1	WRZ0	○	○		Valid byte lane strobe output		
WRZ1 / BENZ1*1	WRZ1	○	○				
WRZ2 / BENZ2*1	RP06	○	○				
WRZ3 / BENZ3*1	RP07	○	○				
WAITZ	P41	○	○	I	Wait signal input		
BCYSTZ	RP20	○	○	O	Bus cycle start status output		

Remark. The external memory interface pins other than BUSCLK are inputs as long as the internal reset signal (HRESETZ) is active.

Note 1. The WREN register is used to switch the pin functions between WRZ3–WRZ0 and BENZ3–BENZ0. For details on this register, refer to Section 14.3.5 “Write Enable Switching Register (WREN)”.

(b) External SRAM Interface Pins (When Synchronous Burst Access Memory Controller is Selected (MEMCSEL = 1))

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
BUSCLK	—	○	○	O	Bus clock output	—	Low
CSZ0	—	○	○		Chip select signal output	Low	Hi-Z (High)
CSZ1	P44	○	○				
CSZ2	P13	○	×				
	EXTP13	×	○				
CSZ3	P12	○	×				
	EXTP12	×	○				
MA0	P40	○	○		Address output	—	
MA1–MA19	A2–A20	○	○				Hi-Z (Low)
MA20–MA26	RP21–RP27	○	○				Hi-Z (High)
MD0–MD15 / MA0–MA15*1	D0–D15	○	○	I/O	Data bus		Hi-Z (Low)
MD16–MD31 / MA16–MA31*1	RP30–RP37, RP10–RP17	○	○				Hi-Z (High)
RDZ	—	○	○	O	Read strobe output	Low	
WRSTBZ	—	○	○		Write strobe output		
WRZ0 / BENZ0*2	WRZ0	○	○		Valid byte lane strobe output		
WRZ1 / BENZ1*2	WRZ1	○	○				
WRZ2 / BENZ2*2	RP06	○	○				
WRZ3 / BENZ3*2	RP07	○	○				
WAITZ	P41	○	○	I	Wait signal input		
WAITZ1–WAITZ3	P45–P47	○	○				
ADVZ	RP20	○	○	O	Address valid output		

Remark. The external memory interface pins other than BUSCLK are inputs as long as the internal reset signal (HRESETZ) is active.

Note 1. When the ADMUXMODE pin is at the High level, these pin functions are multiplexed with address pin functions.
ADMUXMODE = 0: MD0–MD31 (separated address and data lines)
ADMUXMODE = 1: MD0–MD31 / MA0 to MA31 (multiplexed address and data lines)

2. The SET_OPMODE register is used to the switch pin functions between WRZ3–WRZ0 and BENZ3–BENZ0.
For details on this register, refer to Section 15.2.7 "Synchronous Burst Access Memory Controller Mode Setting Register (SET_OPMODE)".

2.1.3.2 External MCU Interface Pins

(a) External MCU Interface Pins (When Asynchronous SRAM Memory Controller is Selected (MEMCSEL = 0))

<table>
<tr><th rowspan="2">Function Name</th><th rowspan="2">Pin Name</th><th colspan="2">PKG</th><th rowspan="2">I/O</th><th rowspan="2">Description</th><th rowspan="2">Active</th><th rowspan="2">Level during Reset</th></tr>
<tr><th>23□</th><th>17□</th></tr>
<tr><td>HBUSCLK*1</td><td>P43</td><td>○</td><td>○</td><td rowspan="3">I</td><td>Bus clock input</td><td>—</td><td rowspan="5">Hi-Z (High)</td></tr>
<tr><td>HCSZ</td><td>CSZ0</td><td>○</td><td>○</td><td>Chip select signal input</td><td rowspan="3">Low</td></tr>
<tr><td>HPGCSZ</td><td>P44</td><td>○</td><td>○</td><td>Page ROM mode chip select signal input</td></tr>
<tr><td>HWAITZ</td><td>P41</td><td>○</td><td>○</td><td>O</td><td>Wait signal output</td></tr>
<tr><td>HA1</td><td>P40</td><td>○</td><td>○</td><td rowspan="2">I</td><td rowspan="2">Address signal input</td><td rowspan="4">—</td></tr>
<tr><td>HA2–HA20</td><td>A2–A20</td><td>○</td><td>○</td><td rowspan="2">Hi-Z (Low)</td></tr>
<tr><td>HD0–HD15</td><td>D0–D15</td><td>○</td><td>○</td><td rowspan="2">I/O</td><td rowspan="2">Data bus</td></tr>
<tr><td>HD16–HD31</td><td>RP30–RP37, RP10–RP17</td><td>○</td><td>○</td><td rowspan="7">Hi-Z (High)</td></tr>
<tr><td>HRDZ</td><td>RDZ</td><td>○</td><td>○</td><td rowspan="6">I</td><td>Read strobe input</td><td rowspan="8">Low</td></tr>
<tr><td>HWRSTBZ</td><td>WRSTBZ</td><td>○</td><td>○</td><td>Write strobe input</td></tr>
<tr><td>HWRZ0 / HBENZ0*2</td><td>WRZ0</td><td>○</td><td>○</td><td rowspan="4">Valid byte lane strobe input</td></tr>
<tr><td>HWRZ1 / HBENZ1*2</td><td>WRZ1</td><td>○</td><td>○</td></tr>
<tr><td>HWRZ2 / HBENZ2*2</td><td>RP06</td><td>○</td><td>○</td></tr>
<tr><td>HWRZ3 / HBENZ3*2</td><td>RP07</td><td>○</td><td>○</td></tr>
<tr><td>HERROUTZ</td><td>P42</td><td>○</td><td>○</td><td>O</td><td>Error interrupt output</td><td>High</td></tr>
<tr><td>HBCYSTZ</td><td>RP20</td><td>○</td><td>○</td><td>I</td><td>Bus cycle input</td><td>Hi-Z (High)</td></tr>
</table>

Note 1. The HBUSCLK pin is used only in case of synchronous SRAM supported MCU connection mode (HIFSYNC pin is High). The HBUSCLK pin is not used in case of asynchronous SRAM supported MCU connection mode (HIFSYNC pin is Low). Furthermore, the other signal connection is common in each mode.

For details on the connection example, refer to the "R-IN32M4-CL3 User's Manual: Board Design edition".

2. The level being input on the HWRZSEL pin controls switching between HWRZ3–HWRZ0 and HBENZ3–HBENZ0 signals.

Remark. The external MCU interface pins continue to operate as those pins even during a reset.

(b) External MCU Interface Pins (When Synchronous Burst Access Memory Controller is Selected (MEMCSEL = 1))

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
HBUSCLK	P43	○	○	I	Bus clock input	—	Hi-Z (High)
HCSZ	CSZ0	○	○		Chip select signal input	Low	
HPGCSZ	P44	○	○		Page ROM mode chip select signal input		
HWAITZ	P41	○	○	O	Wait signal output		
HA1*1	P40	○	○	I	Address signal input	—	
HA2–HA20*1	A2–A20	○	○				Hi-Z (Low)
HD0–HD15*1	D0–D15	○	○	I/O	Data bus		
HD16–HD31*1	RP30–RP37, RP10–RP17	○	○				Hi-Z (High)
HRDZ	RDZ	○	○	I	Read strobe input	Low	
HWRSTBZ	WRSTBZ	○	○		Write strobe input		
HWRZ0 / HBENZ0*2	WRZ0	○	○		Valid byte lane strobe input		
HWRZ1 / HBENZ1*2	WRZ1	○	○				
HWRZ2 / HBENZ2*2	RP06	○	○				
HWRZ3 / HBENZ3*2	RP07	○	○				
HERROUTZ	P42	○	○	O	Error interrupt output		High
HBCYSTZ	RP20	○	○	I	Bus cycle input		Hi-Z (High)

Note 1. The address/data pin connection depends on address/data multiplex mode (ADMUXMODE pin is High) or address/data separate mode (ADMUXMODE pin is Low).
For details on the connection example, refer to the "R-IN32M4-CL3 User's Manual: Board Design edition".
2. When the MEMCSEL pin is High, setting the HWRZSEL pin to High is prohibited.

Remark. The external MCU interface pins continue to operate as those pins even during a reset.

2.1.4 Serial Flash ROM Interface Pins

The serial flash ROM interface pins are the pins of the serial flash ROM controller.
These pins support the following instruction formats: Fast Read, Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, and Fast Read Quad I/O.

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
SMSCK	P14	○	○	O	Serial clock output signal for serial flash ROM	—	Hi-Z (High)
SMIO0	P15	○	○	I/O	Serial data I/O signals for serial flash ROM (Connected to the IO0 pin of serial flash ROM)		
SMIO1	P16	○	○		Serial data I/O signals for serial flash ROM (Connected to the IO1 pin of serial flash ROM)		
SMIO2	P10	○	×		Serial data I/O signals for serial flash ROM (Connected to the /WP(IO2) pin of serial flash ROM)		
	EXTP10	×	○				
SMIO3	P11	○	×		Serial data I/O signals for serial flash ROM (Connected to the /HOLD(IO3) pin of serial flash ROM)		
	EXTP11	×	○				
SMCSZ	P17	○	○	O	Chip select output signal for serial flash ROM	Low	

2.1.5 DMA Interface Pins

The DMA interface pins are the external interface pins of the DMA controllers.
Two types of DMA controllers built into the R-IN32M4-CL3, general-purpose DMA controllers (channels 0 and 1) and real-time port DMA controller, can be used as external DMA interfaces.

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
RTDMAREQZ	P62	○	○	I	RTDMAC DMA transfer request input	Low	Hi-Z (High)
RTDMAACKZ	P63	○	○	O	RTDMAC DMA acknowledge output		
RTDMATCZ	P64	○	○		RTDMAC terminal count output		
DMAREQZ0	P65	○	○	I	DMA transfer request input 0		
DMAACKZ0	P66	○	○	O	DMA acknowledge output 0		
DMATCZ0	P67	○	○		Terminal count output 0		
DMAREQZ1	P32	○	○	I	DMA transfer request input 1		
DMAACKZ1	P33	○	○	O	DMA acknowledge output 1		
DMATCZ1	P34	○	○		Terminal count output 1		

Note. These pins are fixed to channels of the DMA controllers. The pins cannot be assigned to the desired channel. For details, refer to Section 18 “DMA Controllers”.

2.1.6 External Interrupt Input Pins

The LSI has one non-maskable interrupt and 30 maskable interrupt input pins.

Function Name	Pin Name	PKG		I/O	Description	Active	Level during Reset
		23□	17□				
NMIZ	—	○	○	I	Non-maskable external interrupt input	Low	Hi-Z (High)
INTPZ0–INTPZ5	P00–P05	○	○	I	External interrupt input		
INTPZ6	P50	○	○				
INTPZ7	P51	○	○				Hi-Z (Low)
INTPZ8–INTPZ10	P22–P24	○	○				Hi-Z (High)
INTPZ11–INTPZ15	P73–P77	○	○				
INTPZ16–INTPZ21	RP00–RP05	○	○				
INTPZ22	P35	○	○				Hi-Z (Low)
INTPZ23	P36	○	○				Hi-Z (High)
INTPZ24	P37	○	○				Hi-Z (Low)
INTPZ25–INTPZ28	RP24–RP27	○	○				Hi-Z (High)
INTPZ29	P41	○	○				

2.1.7 Timer I/O Pins

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
TINJ0 / TOUTJ0*[1]	P27	○	○	I/O	Timer TAUJ2 I/O pin	—	Hi-Z (High)
TINJ1 / TOUTJ1*[1]	P26	○	○				
TINJ2 / TOUTJ2*[1]	P57	○	○				
TINJ3 / TOUTJ3*[1]	P52	○	○				
TIND0 / TOUTD0	EXTP0	○	○		Timer TAUD I/O pin		
TIND1 / TOUTD1	EXTP1	○	○				
TIND2 / TOUTD2	EXTP2	○	○				
TIND3 / TOUTD3	EXTP3	○	○				
TIND4 / TOUTD4*[1]	P27	○	○				
TIND5 / TOUTD5*[1]	P26	○	○				
TIND6 / TOUTD6*[1]	P57	○	○				
TIND7 / TOUTD7*[1]	P52	○	○				
TIND8 / TOUTD8	RP30	○	○				
TIND9 / TOUTD9	RP31	○	○				
TIND10 / TOUTD10	RP32	○	○				
TIND11 / TOUTD11	RP33	○	○				
TIND12 / TOUTD12	RP34	○	○				
TIND13 / TOUTD13	RP35	○	○				
TIND14 / TOUTD14	RP36	○	○				
TIND15 / TOUTD15	RP37	○	○				

Note 1. TINJ0–TINJ3 and TIND4–TIND7, and TOUTJ0–TOUTJ3 and TOUTD4–TOUTD7 are assigned as multiplexed functions of the same port pins. Use the TMISEL register to select the pin functions to be used.
For details on this register, refer to Section 28.17 "Timer Interface Select Register (TMISEL)".
When the external pin functions such as the interval timer function of the internal clock are not used, both TAUJ2 and TAUD channels can be used at the same time.

2.1.8 Watchdog Timer Output Pin

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
WDTOUTZ	P25 / EXTP3	○	○	O	Watchdog timer output pin	Low	Hi-Z (High)

2.1.9 Serial Interface Pins

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
TXD0	P21	○	○	O	UART0 serial data output	—	Hi-Z (High)
RXD0	P20	○	○	I	UART0 serial data input		
TXD1	P31	○	○	O	UART1 serial data output		
RXD1	P30	○	○	I	UART1 serial data input		
CSISCK0	P45	○	○	I/O	CSI0 serial clock input/output		
CSISI0	P46	○	○	I	CSI0 serial data input		
CSISO0	P47	○	○	O	CSI0 serial data output		
CSICS00	P42	○	○		CSI0 chip select signal output 0	Low	
CSICS01	P43	○	○		CSI0 chip select signal output 1		
CSISCK1	P35	○	○	I/O	CSI1 serial clock input/output	—	Hi-Z (Low)
CSISI1	P36	○	○	I	CSI1 serial data input		Hi-Z (High)
CSISO1	P37	○	○	O	CSI1 serial data output		Hi-Z (Low)
CSICS10	P70	○	○		CSI1 chip select signal output 0	Low	Hi-Z (High)
CSICS11	P71	○	○		CSI1 chip select signal output 1		
SCL0[*1]	P60	○	○	I/O	I^{2}C0 serial clock	—	
SDA0[*1]	P61	○	○		I^{2}C0 serial data		
SCL1[*1]	RP00	○	○		I^{2}C1 serial clock		
SDA1[*1]	RP01	○	○		I^{2}C1 serial data		
CRXD0	P53	○	×	I	CAN0 receive data input (5 V tolerant)		
CTXD0	P54	○	×	O	CAN0 transmit data output (5 V tolerant)		
CRXD1	P55	○	×	I	CAN1 receive data input (5 V tolerant)		
CTXD1	P56	○	×	O	CAN1 transmit data output (5 V tolerant)		

Note 1. The SCLn and SDAn pins (n = 0, 1) are open-drain outputs.
For details, refer to Section 24.4.1 "Pin Configuration".

2.1.10 CC-Link IE Pins

These pins are used for CC-Link IE Field and CC-Link IE TSN.

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
CCI_RUNLEDZ	P00	○	○	O	RUN status output	Low	Hi-Z (High)
CCI_DLINKLEDZ	P02	○	○		Cyclic communication status output		
CCI_ERRLEDZ	P03	○	○		Field network error status output		
CCI_LERR1LEDZ	P04	○	○		Link error status output 1		
CCI_LERR2LEDZ	P05	○	○		Link error status output 2		
CCI_SDLEDZ	P06	○	×		Transmission status output		
	EXTP12	×	○				
CCI_RDLEDZ	P07	○	×		Port reception status output		
	EXTP13	×	○				
CCI_NMIZ	P52	○	○		Output NMI interrupt to MCU		
CCI_WDTIZ	P12	○	×	I	Input from external WDT		
	EXTP11	×	○				
CCI_INTZ	P53	○	×	O	Output interrupt to MCU		
	EXTP10	×	○				
CCI_CLK2_097M	—	○	○	I	2.097152 MHz clock (crystal oscillator)	—	—
CLK2MSEL*1	—	○	○		CC-Link IE Field clock selection signal input 0: 2.097152 MHz (CCI_CLK2_097M) 1: 2 MHz (PLL divided clock)		
IETYPE_LED	EXTP14	×	○	O	The signal pin for LED indicates whether the CC-Link IE function is operating in TSN or Field. TSN operating: LED on Field operating: LED blinking	Low	Hi-Z (High)

Note 1. For details, refer to Section 26.2 "CC-Link IE Clock Selection Function".

2.1.11 System Pins

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
XT1	—	○	○	O	Clock input pin When using an oscillator (OSCTH = 1): XT1 is connected to GND and XT2 is connected to the oscillator. When using a Resonator (OSCTH = 0): XT1 and XT2 are connected to the resonator.	—	—
XT2	—	○	○	I			
RESETZ	—	○	○		Reset input	Low	
PONRZ	—	○	○		Power-on reset input		
HOTRESETZ	—	○	○		Hot reset input		
OSCTH	—	○	○		External clock input mode setting 0: Resonator connection mode 1: External clock input mode	High	
JTAGSEL	—	○	○		JTAG pin operating mode setting 0: Cortex-M4 JTAG mode 1: B-SCAN JTAG mode	—	
RSTOUTZ	—	○	○	O	External reset output	Low	Low
PLL_VDD	—	○	○	—	PLL power supply (1.15 V)	—	—
PLL_GND	—	○	○		PLL GND		
VDD33	—	○	○		I/O power supply (3.3 V)		
VDD11	—	○	○		Internal power supply (1.15 V)		
GND	—	○	○		Power supply ground voltage (GND)		
GND/OPEN	—	○	×		Power supply ground voltage (GND) or open		

2.1.12 Trace Pins

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
TRACECLK	—	○	○	O	Trace port clock output	—	Clock output
TRACEDATA3*1	—	○	○		Trace port data output		Hi-Z (High)
TRACEDATA2*1	—	○	○				
TRACEDATA1*1	—	○	○				
TRACEDATA0*1	—	○	○				

Note 1. These pins are used as an input port in the initial state. The pins are switched to an output port after the reset state is released (the RSTOUTZ pin is switched from Low to High) and 150 to 170 ns are passed.

2.1.13 CPU Power Control Pin

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
SLEEPING	P72	○	○	O	CPU core sleep mode output	High	Hi-Z (High)

2.1.14 Test Pins

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
TMODE0–TMODE2	—	○	○	I	Renesas dedicated test pins	—	—
TEST3		○	×				
TEST4, TEST5		○	○	I/O			
TEST6, TEST7		○	○	I			
TMS		○	○	I/O	Mode select signal		
TDI		○	○	I	Serial data input		
TDO		○	○	O	Serial data output		
TRSTZ		○	○	I	Reset signal	Low	
TCK		○	○		Clock signal (JTAG clock)	—	

2.1.15 Operating Mode Setting Pins

Function Name	Pin Name	PKG 23□	PKG 17□	I/O	Description	Active	Level during Reset
BOOT1–BOOT0	—	○	○	I	Boot mode selection 00: External memory boot 01: External serial flash ROM boot 10: External MCU boot 11: Instruction RAM boot (only available for debugging)	—	—
MEMIFSEL	—	○	○		External memory interface selection 0: Slave memory interface 1: External MCU interface		
MEMCSEL	—	○	○		Internal memory controller selection 0: Asynchronous SRAM controller 1: Synchronous burst access memory controller		
BUS32EN	—	○	○		External memory interface bus width selection 0: 16-bit bus 1: 32-bit bus		
HIFSYNC	—	○	○		External MCU interface operating mode selection 0: Asynchronous SRAM interface 1: Synchronous SRAM interface		
HWRZSEL	—	○	○		External MCU interface HWRZ/HBENZ selection 0: Used as HBENZ 1: Used as HWRZ		
ADMUXMODE	—	○	○		Multiplexing of address and data lines 0: Separated address and data lines 1: Multiplexed address and data lines		

The following table lists the combinations of available operating mode setting pins for this product.

Boot mode	External memory boot				External MCU boot				External serial flash ROM boot							
External memory interface	Slave memory interface				External MCU interface				Slave memory interface				External MCU interface			
Memory controller type	Asynchronous		Synchronous		Asynchronous		Synchronous		Asynchronous		Synchronous		Asynchronous		Synchronous	
External bus width	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit
BOOT1–0	00	00	00	00	10	10	10	10	01	01	01	01	01	01	01	01
MEMIFSEL	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
MEMCSEL	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
BUS32EN	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
HIFSYNC	0	0	0	0	*1	*1	1	1	0	0	0	0	*1	*1	1	1
HWRZSEL	0	0	0	0	*2	*2	0	0	0	0	0	0	*2	*2	0	0
ADMUXMODE	0	0	*3	*3	0	0	*3	*3	0	0	*3	*3	0	0	*3	*3

Note. Any combination of operating mode setting pins other than above is prohibited.

Note 1. The mode of the external MCU interface is selectable by the level on the HIFSYNC pin.
HIFSYNC = 0: Asynchronous SRAM supported MCU connection mode
HIFSYNC = 1: Synchronous SRAM supported MCU connection mode
For details, refer to Section 16 “External MCU Interface”.

2. The external MCU interface HWRZ or HBENZ is selectable by the level on the HWRZSEL pin. For details, refer to Section 2.1.3.2(a) “External MCU Interface Pins (When Asynchronous SRAM Memory Controller is Selected (MEMCSEL = 0))”.

3. Multiplexing of address and data lines is selectable by the level on the ADMUXMODE pin. For details, refer to Section 2.1.3.1(b) “External SRAM Interface Pins (When Synchronous Burst Access Memory Controller is Selected (MEMCSEL = 1))”.

Remarks 1. The combination of operating mode setting pins used to select booting for instruction RAM (BOOT1–0 = 11) is the same as that for booting from external memory (BOOT1–0 = 00).

2. Asynchronous: Asynchronous SRAM controller (MEMCSEL = 0)
Synchronous: Synchronous burst access memory controller (MEMCSEL = 1)

2.2 Pin States

The initial state of the port functions after release from the reset state differs depending on the state of the operating mode setting pins. For the state of the operating mode setting pins in each boot mode and the supported combinations, refer to Section 2.1.15 “Operating Mode Setting Pins”.

Remarks 1. Entries in cells shaded in light green indicate multiplexed pin functions that are enabled in the initial state.

2. The initial state of booting for instruction RAM is the same as that for booting from external memory.

2.2.1 Pin States when Booting is from External Memory

(1/4)

Pin Name	PKG		External memory boot (BOOT1–0 = 00)			
			Slave memory interface (MEMIFSEL = 0)			
			Asynchronous SRAM controller (MEMCSEL = 0)		Synchronous burst access memory controller (MEMCSEL = 1)	
	23□	17□	16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)	16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)
P00	○	○	P00	P00	P00	P00
P01	○	○	P01	P01	P01	P01
P02	○	○	P02	P02	P02	P02
P03	○	○	P03	P03	P03	P03
P04	○	○	P04	P04	P04	P04
P05	○	○	P05	P05	P05	P05
P06	○	×	P06	P06	P06	P06
P07	○	×	P07	P07	P07	P07
P10	○	×	P10	P10	P10	P10
P11	○	×	P11	P11	P11	P11
P12	○	×	P12	P12	P12	P12
P13	○	×	P13	P13	P13	P13
P14	○	○	P14	P14	P14	P14
P15	○	○	P15	P15	P15	P15
P16	○	○	P16	P16	P16	P16
P17	○	○	P17	P17	P17	P17
P20	○	○	P20	P20	P20	P20
P21	○	○	P21	P21	P21	P21
P22	○	○	P22	P22	P22	P22
P23	○	○	P23	P23	P23	P23
P24	○	○	P24	P24	P24	P24
P25	○	○	P25	P25	P25	P25
P26	○	○	P26	P26	P26	P26
P27	○	○	P27	P27	P27	P27
P30	○	○	P30	P30	P30	P30
P31	○	○	P31	P31	P31	P31
P32	○	○	P32	P32	P32	P32
P33	○	○	P33	P33	P33	P33
P34	○	○	P34	P34	P34	P34

(2/4)

Pin	PKG		External memory boot (BOOT1–0 = 00)			
			Slave memory interface (MEMIFSEL = 0)			
			Asynchronous SRAM controller (MEMCSEL = 0)		Synchronous burst access memory controller (MEMCSEL = 1)	
Name	23□	17□	16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)	16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)
P35	○	○	P35	P35	P35	P35
P36	○	○	P36	P36	P36	P36
P37	○	○	P37	P37	P37	P37
P40	○	○	A1	P40	MA0	MA0
P41	○	○	P41	P41	P41	P41
P42	○	○	P42	P42	P42	P42
P43	○	○	P43	P43	P43	P43
P44	○	○	P44	P44	P44	P44
P45	○	○	P45	P45	P45	P45
P46	○	○	P46	P46	P46	P46
P47	○	○	P47	P47	P47	P47
P50	○	○	P50	P50	P50	P50
P51	○	○	P51	P51	P51	P51
P52	○	○	P52	P52	P52	P52
P53	○	×	P53	P53	P53	P53
P54	○	×	P54	P54	P54	P54
P55	○	×	P55	P55	P55	P55
P56	○	×	P56	P56	P56	P56
P57	○	○	P57	P57	P57	P57
P60	○	○	P60	P60	P60	P60
P61	○	○	P61	P61	P61	P61
P62	○	○	P62	P62	P62	P62
P63	○	○	P63	P63	P63	P63
P64	○	○	P64	P64	P64	P64
P65	○	○	P65	P65	P65	P65
P66	○	○	P66	P66	P66	P66
P67	○	○	P67	P67	P67	P67
P70	○	○	P70	P70	P70	P70
P71	○	○	P71	P71	P71	P71
P72	○	○	P72	P72	P72	P72
P73	○	○	P73	P73	P73	P73
P74	○	○	P74	P74	P74	P74
P75	○	○	P75	P75	P75	P75
P76	○	○	P76	P76	P76	P76
P77	○	○	P77	P77	P77	P77

(3/4)

Pin Name	PKG 23□	PKG 17□	External memory boot (BOOT1–0 = 00) / Slave memory interface (MEMIFSEL = 0) / Asynchronous SRAM controller (MEMCSEL = 0) 16-bit (BUS32EN = 0)	Asynchronous SRAM controller (MEMCSEL = 0) 32-bit (BUS32EN = 1)	Synchronous burst access memory controller (MEMCSEL = 1) 16-bit (BUS32EN = 0)	Synchronous burst access memory controller (MEMCSEL = 1) 32-bit (BUS32EN = 1)
EXTP0	○	×	EXTP0	EXTP0	EXTP0	EXTP0
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP1	○	×	EXTP1	EXTP1	EXTP1	EXTP1
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP2	○	×	EXTP2	EXTP2	EXTP2	EXTP2
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP3	○	○	EXTP3	EXTP3	EXTP3	EXTP3
EXTP4	○	×	EXTP4	EXTP4	EXTP4	EXTP4
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP5	○	○	EXTP5	EXTP5	EXTP5	EXTP5
EXTP6	○	○	EXTP6	EXTP6	EXTP6	EXTP6
EXTP7	○	○	EXTP7	EXTP7	EXTP7	EXTP7
EXTP8	○	×	EXTP8	EXTP8	EXTP8	EXTP8
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP9	○	×	EXTP9	EXTP9	EXTP9	EXTP9
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP10	×	○	EXTP10	EXTP10	EXTP10	EXTP10
EXTP11	×	○	EXTP11	EXTP11	EXTP11	EXTP11
EXTP12	×	○	EXTP12	EXTP12	EXTP12	EXTP12
EXTP13	×	○	EXTP13	EXTP13	EXTP13	EXTP13
EXTP14	×	○	EXTP14	EXTP14	EXTP14	EXTP14

Note. The initial state of the "Reserved" pins is not specified. These pins can be used as normal ports by setting the port function after release from the reset state.

(4/4)

Pin	PKG		External memory boot (BOOT1–0 = 00)			
			Slave memory interface (MEMIFSEL = 0)			
			Asynchronous SRAM controller (MEMCSEL = 0)		Synchronous burst access memory controller (MEMCSEL = 1)	
Name	23□	17□	16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)	16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)
RP00	○	○	RP00	RP00	RP00	RP00
RP01	○	○	RP01	RP01	RP01	RP01
RP02	○	○	RP02	RP02	RP02	RP02
RP03	○	○	RP03	RP03	RP03	RP03
RP04	○	○	RP04	RP04	RP04	RP04
RP05	○	×	RP05	RP05	RP05	RP05
	×	○	Reserved	Reserved	Reserved	Reserved
RP06	○	○	RP06	WRZ2	RP06	WRZ2
RP07	○	○	RP07	WRZ3	RP07	WRZ3
RP10	○	○	RP10	D24	RP10	MD24
RP11	○	○	RP11	D25	RP11	MD25
RP12	○	○	RP12	D26	RP12	MD26
RP13	○	○	RP13	D27	RP13	MD27
RP14	○	○	RP14	D28	RP14	MD28
RP15	○	○	RP15	D29	RP15	MD29
RP16	○	○	RP16	D30	RP16	MD30
RP17	○	○	RP17	D31	RP17	MD31
RP20	○	○	RP20	RP20	ADVZ	ADVZ
RP21	○	○*1	RP21	RP21	RP21	RP21
RP22	○	○*1	RP22	RP22	RP22	RP22
RP23	○	○*1	RP23	RP23	RP23	RP23
RP24	○	○*1	RP24	RP24	RP24	RP24
RP25	○	○*1	RP25	RP25	RP25	RP25
RP26	○	○*1	RP26	RP26	RP26	RP26
RP27	○	○*1	RP27	RP27	RP27	RP27
RP30	○	○	RP30	D16	RP30	MD16
RP31	○	○	RP31	D17	RP31	MD17
RP32	○	○	RP32	D18	RP32	MD18
RP33	○	○	RP33	D19	RP33	MD19
RP34	○	○	RP34	D20	RP34	MD20
RP35	○	○	RP35	D21	RP35	MD21
RP36	○	○	RP36	D22	RP36	MD22
RP37	○	○	RP37	D23	RP37	MD23

Note 1. In the initial state of the 17 mm square package, the A21–A27 inputs (external pins: RP21–RP27) of the external SRAM interface are pulled up. For details, refer to Section 7.1 "Selecting the Boot Mode".

Note. The initial state of the "Reserved" pins is not specified. These pins can be used as normal ports by setting the port function after release from the reset state.

2.2.2 Pin States when Booting is from External Serial Flash ROM

Remarks 1. Asynchronous type: Asynchronous SRAM controller (MEMCSEL = 0)
Synchronous type: Synchronous burst access memory controller (MEMCSEL = 1)
2. 16-bit: 16-bit bus width of the external memory interface (BUS32EN = 0)
32-bit: 32-bit bus width of the external memory interface (BUS32EN = 1)

(1/4)

Pin Name	PKG		External serial flash ROM boot (BOOT1–0 = 01)							
			Slave memory interface (MEMIFSEL = 0)				External MCU interface (MEMIFSEL = 1)			
			Asynchronous type		Synchronous type		Asynchronous type		Synchronous type	
	23□	17□	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit
P00	○	○	P00	P00	P00	P00	P00	P00	P00	P00
P01	○	○	P01	P01	P01	P01	P01	P01	P01	P01
P02	○	○	P02	P02	P02	P02	P02	P02	P02	P02
P03	○	○	P03	P03	P03	P03	P03	P03	P03	P03
P04	○	○	P04	P04	P04	P04	P04	P04	P04	P04
P05	○	○	P05	P05	P05	P05	P05	P05	P05	P05
P06	○	×	P06	P06	P06	P06	P06	P06	P06	P06
P07	○	×	P07	P07	P07	P07	P07	P07	P07	P07
P10	○	×	P10	P10	P10	P10	P10	P10	P10	P10
P11	○	×	P11	P11	P11	P11	P11	P11	P11	P11
P12	○	×	P12	P12	P12	P12	P12	P12	P12	P12
P13	○	×	P13	P13	P13	P13	P13	P13	P13	P13
P14	○	○	SMSCK	SMSCK	SMSCK	SMSCK	SMSCK	SMSCK	SMSCK	SMSCK
P15	○	○	SMIO0	SMIO0	SMIO0	SMIO0	SMIO0	SMIO0	SMIO0	SMIO0
P16	○	○	SMIO1	SMIO1	SMIO1	SMIO1	SMIO1	SMIO1	SMIO1	SMIO1
P17	○	○	SMCSZ	SMCSZ	SMCSZ	SMCSZ	SMCSZ	SMCSZ	SMCSZ	SMCSZ
P20	○	○	P20	P20	P20	P20	P20	P20	P20	P20
P21	○	○	P21	P21	P21	P21	P21	P21	P21	P21
P22	○	○	P22	P22	P22	P22	P22	P22	P22	P22
P23	○	○	P23	P23	P23	P23	P23	P23	P23	P23
P24	○	○	P24	P24	P24	P24	P24	P24	P24	P24
P25	○	○	P25	P25	P25	P25	P25	P25	P25	P25
P26	○	○	P26	P26	P26	P26	P26	P26	P26	P26
P27	○	○	P27	P27	P27	P27	P27	P27	P27	P27
P30	○	○	P30	P30	P30	P30	P30	P30	P30	P30
P31	○	○	P31	P31	P31	P31	P31	P31	P31	P31
P32	○	○	P32	P32	P32	P32	P32	P32	P32	P32
P33	○	○	P33	P33	P33	P33	P33	P33	P33	P33
P34	○	○	P34	P34	P34	P34	P34	P34	P34	P34
P35	○	○	P35	P35	P35	P35	P35	P35	P35	P35
P36	○	○	P36	P36	P36	P36	P36	P36	P36	P36
P37	○	○	P37	P37	P37	P37	P37	P37	P37	P37

(2/4)

Pin Name	PKG		External serial flash ROM boot (BOOT1–0 = 01)							
			Slave memory interface (MEMIFSEL = 0)				External MCU interface (MEMIFSEL = 1)			
			Asynchronous type		Synchronous type		Asynchronous type		Synchronous type	
	23□	17□	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit
P40	○	○	A1	P40	MA0	MA0	HA1	HA1	HA1	HA1
P41	○	○	P41	P41	P41	P41	HWAITZ	HWAITZ	HWAITZ	HWAITZ
P42	○	○	P42	P42	P42	P42	HERROUTZ	HERROUTZ	HERROUTZ	HERROUTZ
P43	○	○	P43	P43	P43	P43	HBUSCLK	HBUSCLK	HBUSCLK	HBUSCLK
P44	○	○	P44	P44	P44	P44	HPGCSZ	HPGCSZ	HPGCSZ	HPGCSZ
P45	○	○	P45	P45	P45	P45	P45	P45	P45	P45
P46	○	○	P46	P46	P46	P46	P46	P46	P46	P46
P47	○	○	P47	P47	P47	P47	P47	P47	P47	P47
P50	○	○	P50	P50	P50	P50	P50	P50	P50	P50
P51	○	○	P51	P51	P51	P51	P51	P51	P51	P51
P52	○	○	P52	P52	P52	P52	P52	P52	P52	P52
P53	○	×	P53	P53	P53	P53	P53	P53	P53	P53
P54	○	×	P54	P54	P54	P54	P54	P54	P54	P54
P55	○	×	P55	P55	P55	P55	P55	P55	P55	P55
P56	○	×	P56	P56	P56	P56	P56	P56	P56	P56
P57	○	○	P57	P57	P57	P57	P57	P57	P57	P57
P60	○	○	P60	P60	P60	P60	P60	P60	P60	P60
P61	○	○	P61	P61	P61	P61	P61	P61	P61	P61
P62	○	○	P62	P62	P62	P62	P62	P62	P62	P62
P63	○	○	P63	P63	P63	P63	P63	P63	P63	P63
P64	○	○	P64	P64	P64	P64	P64	P64	P64	P64
P65	○	○	P65	P65	P65	P65	P65	P65	P65	P65
P66	○	○	P66	P66	P66	P66	P66	P66	P66	P66
P67	○	○	P67	P67	P67	P67	P67	P67	P67	P67
P70	○	○	P70	P70	P70	P70	P70	P70	P70	P70
P71	○	○	P71	P71	P71	P71	P71	P71	P71	P71
P72	○	○	P72	P72	P72	P72	P72	P72	P72	P72
P73	○	○	P73	P73	P73	P73	P73	P73	P73	P73
P74	○	○	P74	P74	P74	P74	P74	P74	P74	P74
P75	○	○	P75	P75	P75	P75	P75	P75	P75	P75
P76	○	○	P76	P76	P76	P76	P76	P76	P76	P76
P77	○	○	P77	P77	P77	P77	P77	P77	P77	P77

(3/4)

Pin	PKG		External serial flash ROM boot (BOOT1–0 = 01)							
			Slave memory interface (MEMIFSEL = 0)				External MCU interface (MEMIFSEL = 1)			
			Asynchronous type		Synchronous type		Asynchronous type		Synchronous type	
Name	23□	17□	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit
EXTP0	○	×	EXTP0	EXTP0	EXTP0	EXTP0	EXTP0	EXTP0	EXTP0	EXTP0
	×	○	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
EXTP1	○	×	EXTP1	EXTP1	EXTP1	EXTP1	EXTP1	EXTP1	EXTP1	EXTP1
	×	○	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
EXTP2	○	×	EXTP2	EXTP2	EXTP2	EXTP2	EXTP2	EXTP2	EXTP2	EXTP2
	×	○	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
EXTP3	○	○	EXTP3	EXTP3	EXTP3	EXTP3	EXTP3	EXTP3	EXTP3	EXTP3
EXTP4	○	×	EXTP4	EXTP4	EXTP4	EXTP4	EXTP4	EXTP4	EXTP4	EXTP4
	×	○	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
EXTP5	○	○	EXTP5	EXTP5	EXTP5	EXTP5	EXTP5	EXTP5	EXTP5	EXTP5
EXTP6	○	○	EXTP6	EXTP6	EXTP6	EXTP6	EXTP6	EXTP6	EXTP6	EXTP6
EXTP7	○	○	EXTP7	EXTP7	EXTP7	EXTP7	EXTP7	EXTP7	EXTP7	EXTP7
EXTP8	○	×	EXTP8	EXTP8	EXTP8	EXTP8	EXTP8	EXTP8	EXTP8	EXTP8
	×	○	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
EXTP9	○	×	EXTP9	EXTP9	EXTP9	EXTP9	EXTP9	EXTP9	EXTP9	EXTP9
	×	○	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
EXTP10	×	○	EXTP10	EXTP10	EXTP10	EXTP10	EXTP10	EXTP10	EXTP10	EXTP10
EXTP11	×	○	EXTP11	EXTP11	EXTP11	EXTP11	EXTP11	EXTP11	EXTP11	EXTP11
EXTP12	×	○	EXTP12	EXTP12	EXTP12	EXTP12	EXTP12	EXTP12	EXTP12	EXTP12
EXTP13	×	○	EXTP13	EXTP13	EXTP13	EXTP13	EXTP13	EXTP13	EXTP13	EXTP13
EXTP14	×	○	EXTP14	EXTP14	EXTP14	EXTP14	EXTP14	EXTP14	EXTP14	EXTP14

Note. The initial state of the “Reserved” pins is not specified. These pins can be used as normal ports by setting the port function after release from the reset state.

(4/4)

Pin	PKG		External serial flash ROM boot (BOOT1–0 = 01)							
			Slave memory interface (MEMIFSEL = 0)				External MCU interface (MEMIFSEL = 1)			
			Asynchronous type		Synchronous type		Asynchronous type		Synchronous type	
Name	23□	17□	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit
RP00	○	○	RP00	RP00	RP00	RP00	RP00	RP00	RP00	RP00
RP01	○	○	RP01	RP01	RP01	RP01	RP01	RP01	RP01	RP01
RP02	○	○	RP02	RP02	RP02	RP02	RP02	RP02	RP02	RP02
RP03	○	○	RP03	RP03	RP03	RP03	RP03	RP03	RP03	RP03
RP04	○	○	RP04	RP04	RP04	RP04	RP04	RP04	RP04	RP04
RP05	○	×	RP05	RP05	RP05	RP05	RP05	RP05	RP05	RP05
	×	○	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
RP06	○	○	RP06	WRZ2	RP06	WRZ2	RP06	HWRZ2	RP06	HWRZ2
RP07	○	○	RP07	WRZ3	RP07	WRZ3	RP07	HWRZ3	RP07	HWRZ3
RP10	○	○	RP10	D24	RP10	MD24	RP10	HD24	RP10	HD24
RP11	○	○	RP11	D25	RP11	MD25	RP11	HD25	RP11	HD25
RP12	○	○	RP12	D26	RP12	MD26	RP12	HD26	RP12	HD26
RP13	○	○	RP13	D27	RP13	MD27	RP13	HD27	RP13	HD27
RP14	○	○	RP14	D28	RP14	MD28	RP14	HD28	RP14	HD28
RP15	○	○	RP15	D29	RP15	MD29	RP15	HD29	RP15	HD29
RP16	○	○	RP16	D30	RP16	MD30	RP16	HD30	RP16	HD30
RP17	○	○	RP17	D31	RP17	MD31	RP17	HD31	RP17	HD31
RP20	○	○	RP20	RP20	ADVZ	ADVZ	HBCYSTZ	HBCYSTZ	HBCYSTZ	HBCYSTZ
RP21	○	○	RP21	RP21	RP21	RP21	RP21	RP21	RP21	RP21
RP22	○	○	RP22	RP22	RP22	RP22	RP22	RP22	RP22	RP22
RP23	○	○	RP23	RP23	RP23	RP23	RP23	RP23	RP23	RP23
RP24	○	○	RP24	RP24	RP24	RP24	RP24	RP24	RP24	RP24
RP25	○	○	RP25	RP25	RP25	RP25	RP25	RP25	RP25	RP25
RP26	○	○	RP26	RP26	RP26	RP26	RP26	RP26	RP26	RP26
RP27	○	○	RP27	RP27	RP27	RP27	RP27	RP27	RP27	RP27
RP30	○	○	RP30	D16	RP30	MD16	RP30	HD16	RP30	HD16
RP31	○	○	RP31	D17	RP31	MD17	RP31	HD17	RP31	HD17
RP32	○	○	RP32	D18	RP32	MD18	RP32	HD18	RP32	HD18
RP33	○	○	RP33	D19	RP33	MD19	RP33	HD19	RP33	HD19
RP34	○	○	RP34	D20	RP34	MD20	RP34	HD20	RP34	HD20
RP35	○	○	RP35	D21	RP35	MD21	RP35	HD21	RP35	HD21
RP36	○	○	RP36	D22	RP36	MD22	RP36	HD22	RP36	HD22
RP37	○	○	RP37	D23	RP37	MD23	RP37	HD23	RP37	HD23

Note. The initial state of the “Reserved” pins is not specified. These pins can be used as normal ports by setting the port function after release from the reset state.

2.2.3 Pin States when Booting is for External MCU

(1/4)

Pin Name	PKG		External MCU boot (BOOT1–0 = 10)			
			External MCU interface (MEMIFSEL = 1)			
			Asynchronous SRAM controller (MEMCSEL = 0)		Synchronous burst access memory controller (MEMCSEL = 1)	
	23□	17□	16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)	16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)
P00	○	○	P00	P00	P00	P00
P01	○	○	P01	P01	P01	P01
P02	○	○	P02	P02	P02	P02
P03	○	○	P03	P03	P03	P03
P04	○	○	P04	P04	P04	P04
P05	○	○	P05	P05	P05	P05
P06	○	×	P06	P06	P06	P06
P07	○	×	P07	P07	P07	P07
P10	○	×	P10	P10	P10	P10
P11	○	×	P11	P11	P11	P11
P12	○	×	P12	P12	P12	P12
P13	○	×	P13	P13	P13	P13
P14	○	○	P14	P14	P14	P14
P15	○	○	P15	P15	P15	P15
P16	○	○	P16	P16	P16	P16
P17	○	○	P17	P17	P17	P17
P20	○	○	P20	P20	P20	P20
P21	○	○	P21	P21	P21	P21
P22	○	○	P22	P22	P22	P22
P23	○	○	P23	P23	P23	P23
P24	○	○	P24	P24	P24	P24
P25	○	○	P25	P25	P25	P25
P26	○	○	P26	P26	P26	P26
P27	○	○	P27	P27	P27	P27
P30	○	○	P30	P30	P30	P30
P31	○	○	P31	P31	P31	P31
P32	○	○	P32	P32	P32	P32
P33	○	○	P33	P33	P33	P33
P34	○	○	P34	P34	P34	P34
P35	○	○	P35	P35	P35	P35
P36	○	○	P36	P36	P36	P36
P37	○	○	P37	P37	P37	P37

(2/4)

Pin Name	PKG 23□	PKG 17□	External MCU boot (BOOT1–0 = 10) External MCU interface (MEMIFSEL = 1) Asynchronous SRAM controller (MEMCSEL = 0) 16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)	Synchronous burst access memory controller (MEMCSEL = 1) 16-bit (BUS32EN = 0)	32-bit (BUS32EN = 1)
P40	○	○	HA1	HA1	HA1	HA1
P41	○	○	HWAITZ	HWAITZ	HWAITZ	HWAITZ
P42	○	○	HERROUTZ	HERROUTZ	HERROUTZ	HERROUTZ
P43	○	○	HBUSCLK	HBUSCLK	HBUSCLK	HBUSCLK
P44	○	○	HPGCSZ	HPGCSZ	HPGCSZ	HPGCSZ
P45	○	○	P45	P45	P45	P45
P46	○	○	P46	P46	P46	P46
P47	○	○	P47	P47	P47	P47
P50	○	○	P50	P50	P50	P50
P51	○	○	P51	P51	P51	P51
P52	○	○	P52	P52	P52	P52
P53	○	×	P53	P53	P53	P53
P54	○	×	P54	P54	P54	P54
P55	○	×	P55	P55	P55	P55
P56	○	×	P56	P56	P56	P56
P57	○	○	P57	P57	P57	P57
P60	○	○	P60	P60	P60	P60
P61	○	○	P61	P61	P61	P61
P62	○	○	P62	P62	P62	P62
P63	○	○	P63	P63	P63	P63
P64	○	○	P64	P64	P64	P64
P65	○	○	P65	P65	P65	P65
P66	○	○	P66	P66	P66	P66
P67	○	○	P67	P67	P67	P67
P70	○	○	P70	P70	P70	P70
P71	○	○	P71	P71	P71	P71
P72	○	○	P72	P72	P72	P72
P73	○	○	P73	P73	P73	P73
P74	○	○	P74	P74	P74	P74
P75	○	○	P75	P75	P75	P75
P76	○	○	P76	P76	P76	P76
P77	○	○	P77	P77	P77	P77

(3/4)

Pin Name	PKG 23□	PKG 17□	External MCU boot (BOOT1–0 = 10) External MCU interface (MEMIFSEL = 1) Asynchronous SRAM controller (MEMCSEL = 0) 16-bit (BUS32EN = 0)	Asynchronous SRAM controller (MEMCSEL = 0) 32-bit (BUS32EN = 1)	Synchronous burst access memory controller (MEMCSEL = 1) 16-bit (BUS32EN = 0)	Synchronous burst access memory controller (MEMCSEL = 1) 32-bit (BUS32EN = 1)
EXTP0	○	×	EXTP0	EXTP0	EXTP0	EXTP0
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP1	○	×	EXTP1	EXTP1	EXTP1	EXTP1
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP2	○	×	EXTP2	EXTP2	EXTP2	EXTP2
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP3	○	○	EXTP3	EXTP3	EXTP3	EXTP3
EXTP4	○	×	EXTP4	EXTP4	EXTP4	EXTP4
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP5	○	○	EXTP5	EXTP5	EXTP5	EXTP5
EXTP6	○	○	EXTP6	EXTP6	EXTP6	EXTP6
EXTP7	○	○	EXTP7	EXTP7	EXTP7	EXTP7
EXTP8	○	×	EXTP8	EXTP8	EXTP8	EXTP8
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP9	○	×	EXTP9	EXTP9	EXTP9	EXTP9
	×	○	Reserved	Reserved	Reserved	Reserved
EXTP10	×	○	EXTP10	EXTP10	EXTP10	EXTP10
EXTP11	×	○	EXTP11	EXTP11	EXTP11	EXTP11
EXTP12	×	○	EXTP12	EXTP12	EXTP12	EXTP12
EXTP13	×	○	EXTP13	EXTP13	EXTP13	EXTP13
EXTP14	×	○	EXTP14	EXTP14	EXTP14	EXTP14

Note. The initial state of the “Reserved” pins is not specified. These pins can be used as normal ports by setting the port function after release from the reset state.

(4/4)

Pin Name	PKG 23□	PKG 17□	External MCU boot (BOOT1–0 = 10) External MCU interface (MEMIFSEL = 1) Asynchronous SRAM controller (MEMCSEL = 0) 16-bit (BUS32EN = 0)	Asynchronous SRAM controller (MEMCSEL = 0) 32-bit (BUS32EN = 1)	Synchronous burst access memory controller (MEMCSEL = 1) 16-bit (BUS32EN = 0)	Synchronous burst access memory controller (MEMCSEL = 1) 32-bit (BUS32EN = 1)
RP00	○	○	RP00	RP00	RP00	RP00
RP01	○	○	RP01	RP01	RP01	RP01
RP02	○	○	RP02	RP02	RP02	RP02
RP03	○	○	RP03	RP03	RP03	RP03
RP04	○	○	RP04	RP04	RP04	RP04
RP05	○	×	RP05	RP05	RP05	RP05
	×	○	Reserved	Reserved	Reserved	Reserved
RP06	○	○	RP06	HWRZ2	RP06	HWRZ2
RP07	○	○	RP07	HWRZ3	RP07	HWRZ3
RP10	○	○	RP10	HD24	RP10	HD24
RP11	○	○	RP11	HD25	RP11	HD25
RP12	○	○	RP12	HD26	RP12	HD26
RP13	○	○	RP13	HD27	RP13	HD27
RP14	○	○	RP14	HD28	RP14	HD28
RP15	○	○	RP15	HD29	RP15	HD29
RP16	○	○	RP16	HD30	RP16	HD30
RP17	○	○	RP17	HD31	RP17	HD31
RP20	○	○	HBCYSTZ	HBCYSTZ	HBCYSTZ	HBCYSTZ
RP21	○	○	RP21	RP21	RP21	RP21
RP22	○	○	RP22	RP22	RP22	RP22
RP23	○	○	RP23	RP23	RP23	RP23
RP24	○	○	RP24	RP24	RP24	RP24
RP25	○	○	RP25	RP25	RP25	RP25
RP26	○	○	RP26	RP26	RP26	RP26
RP27	○	○	RP27	RP27	RP27	RP27
RP30	○	○	RP30	HD16	RP30	HD16
RP31	○	○	RP31	HD17	RP31	HD17
RP32	○	○	RP32	HD18	RP32	HD18
RP33	○	○	RP33	HD19	RP33	HD19
RP34	○	○	RP34	HD20	RP34	HD20
RP35	○	○	RP35	HD21	RP35	HD21
RP36	○	○	RP36	HD22	RP36	HD22
RP37	○	○	RP37	HD23	RP37	HD23

Note. The initial state of the “Reserved” pins is not specified. These pins can be used as normal ports by setting the port function after release from the reset state.

2.3 Operating Mode Monitoring

The levels on the operating mode setting pins can be checked by using the operating mode monitoring register. The following table lists the operating mode setting pins for which the settings can be checked.

For details on the operating mode monitoring register, refer to Section 28.2 "Operating Mode Monitor Register (MDMNT)".

Table 2.3 Operating Mode Setting Pins for which the Settings can be Checked

Pin Name	Function
BUS32EN	Selects the bus width when the external memory interface is started.
MEMIFSEL	Selects the type of external memory interface.
HIFSYNC	Sets the operating mode of the external MCU interface.
HWRZSEL	Selects HWRZ or HBENZ of the external MCU interface.
JTAGSEL	Sets the operating mode of JTAG pins.
OSCTH	Inputs High level in external clock input mode.
BOOT0, BOOT1	Selects boot mode.
MEMCSEL	Selects the internal memory controller.
ADMUXMODE	Sets multiplexing of address and data lines.
CLK2MSEL	Selects the clock to be input to CC-Link IE.

2.4 Buffer Switching

The driving ability and use of a pull-up or pull-down resistor are programmable for the real-time port pins and general-purpose port pins (with some exceptions).
This function provides stable operation in systems with large loads by providing the ability to raise the driving ability.
The buffer switching register (DRCTLx) is used to switch the buffer function.

For details on the buffer switching register, refer to Section 27.6 “Buffer Switching Register (DRCTL)”.

2.5 Buffer types of Port Pins and Real-Time Port Pins and Handling of Unused Pins

2.5.1 Port Pins and Real-Time Port Pins

Pin Name	PKG 23□	PKG 17□	I/O	Interface	Recommended connection when not in use
P00–P05	○	○	I/O	Programmable I/O Buffer (3.3 V) Driving ability selection (8 mA, 12 mA) Resistor selection (Pull-up or Pull-down or less)	Open
P06–P07	○	×			
P34–P35, P37, P50, P70–P77, RP10–RP17, RP33, RP34, EXTP0, EXTP6, EXTP7	○	○			
P20–P23, P25–P27, P30–P33, P44, P60–P67, RP00–RP04, RP20–RP27	○	○	I/O	Programmable I/O Buffer (3.3 V) Schmitt in driving ability selection (8 mA, 12 mA) Resistor selection (Pull-up or Pull-down or less)	
P10–P13	○	×	I/O	Programmable I/O Buffer (3.3 V, 8 mA) Resistor selection (Pull-up or Pull-down or less)	
P14–P17	○	○			
P24, P36, P40–P43, P45–P47, P51, P52, P57, RP05–RP07, RP30–RP32, RP35–RP37, EXTP1–EXTP5, EXTP8, EXTP9	○	○			
P53–P56	○	×	I/O	5 V tolerant I/O Buffer Schmitt in 4 mA 130 kΩ Pull-up	
EXTP10–EXTP14	×	○	I/O	Programmable I/O Buffer (3.3 V) Driving ability selection 8 mA, 12 mA) Resistor selection (Pull-up or Pull-down or less)	

2.5.2 Ethernet pins

Pin Name	PKG 23□	PKG 17□	I/O	Interface	Recommended connection when not in use
P0_D0N	〇	〇	I/O	Management Data Interface (Analog)	Open
P0_D0P	〇	〇			
P0_D1N	〇	〇			
P0_D1P	〇	〇			
P0_D2N	〇	〇			
P0_D2P	〇	〇			
P0_D3N	〇	〇			
P0_D3P	〇	〇			
P1_D0N	〇	〇			
P1_D0P	〇	〇			
P1_D1N	〇	〇			
P1_D1P	〇	〇			
P1_D2N	〇	〇			
P1_D2P	〇	〇			
P1_D3N	〇	〇			
P1_D3P	〇	〇			
PHYADD1	〇	×	I	Input Buffer (3.3 V) Schmitt in 160 kΩ Pull-down	
PHYADD2	〇	×			
PHYADD3	〇	×			
PHYADD4	〇	×			
REF_FILT	〇	〇	I/O	Copper media reference filter pin.	Pin handling is required even when not in use.*1
REF_REXT	〇	〇		Copper media reference external pin.	
VDD11A	〇	〇	—	1.15 V analog power requiring additional PCB power supply filtering	Connect to VDD (1.15 V)
VDD25A	〇	〇		2.5 V general analog power supply	Connect to VDD (2.5 V)
PHY0_LED0	〇	〇	O	GbE-PHY LED0_PHY0 output signal Output Buffer (3.3 V) 12 mA	Open
PHY1_LED0	〇	〇		GbE-PHY LED0_PHY1 output signal Output Buffer (3.3 V) 12 mA	
VDDREG_33	〇	〇	—	3.3 V power for 2.5 V regulator	Connect to GND
AVDDREG_33	〇	〇		3.3 V analog power for 2.5 V regulator	
REG_EN	〇	〇	I	Input Buffer (3.3 V) Schmitt in	
REG_FB	〇	〇		Feedback from the supply regulation point	Connect to GND or Open
AGND	〇	〇	—	Analog ground for regulator	Connect to GND
REG_OUT	〇	〇	O	2.5 V Regulator output	Connect to GND or Open

Note 1. For the pin handling, refer to the "R-IN32M4-CL3 User's Manual: Board Design edition".

2.5.3 External SRAM Interface Pins and External MCU Interface Pins

Pin Name	PKG		I/O	Interface	Recommended connection when not in use
	23□	17□			
BUSCLK	○	○	O	Output Buffer (3.3 V) 10 mA	Open
CSZ0	○	○	I/O	I/O Buffer (3.3 V) 8 mA 130 kΩ Pull-up	
A2–A20	○	○		I/O Buffer (3.3 V) 8 mA 160 kΩ Pull-down	
D0–D15	○	○			
RDZ	○	○		I/O Buffer (3.3 V) 8 mA 130 kΩ Pull-up	
WRSTBZ	○	○			
WRZ0, WRZ1	○	○			

2.5.4 External Interrupt Input Pins

Pin Name	PKG		I/O	Interface	Recommended connection when not in use
	23□	17□			
NMIZ	○	○	I	Input Buffer (3.3 V) Schmitt in 130 kΩ Pull-up	Connect to VDD (3.3 V)

2.5.5 CC-Link IE Pins

Pin Name	PKG		I/O	Interface	Recommended connection when not in use
	23□	17□			
CCI_CLK2_097M*1	○	○	I	Input Buffer (3.3 V)	When using CC-Link IE Field (CLK2MSEL = 0): A 2.097152MHz oscillator must be connected. When using only CC-Link IE TSN (CLK2MSEL = 1): Connect the pin to GND.
CLK2MSEL	○	○		Input Buffer (3.3 V) Schmitt in	Set this pin in accordance with the operating mode.

Note 1. For details, refer to Section 26.2 "CC-Link IE Clock Selection Function".

2.5.6 System Pins

Pin Name	PKG 23□	PKG 17□	I/O	Interface	Recommended connection when not in use
XT1	○	○	O	Oscillator	*1
XT2	○	○	I		
RSTOUTZ	○	○	O	Output Buffer (3.3 V) 8 mA	Open
RESETZ	○	○	I	Input Buffer (3.3 V) Schmitt in	Connect to VDD (3.3 V)
PONRZ	○	○			Connect a reset signal since this pin is always used.
HOTRESETZ	○	○			Connect to VDD (3.3 V)
OSCTH	○	○		Input Buffer (3.3 V) Schmitt in 160 kΩ Pull-down	Set these pins in accordance with the operating mode.
JTAGSEL	○	○			
PLL_VDD	○	○	—	PLL power supply (1.15 V)	Connect to VDD (1.15 V)
PLL_GND	○	○		PLL GND	Connect to GND
VDD33	○	○		I/O power supply (3.3 V)	Connect to VDD (3.3 V)
VDD11	○	○		Internal power supply (1.15 V)	Connect to VDD (1.15 V)
GND	○	○		Power supply ground voltage (GND)	Connect to GND
GND/OPEN*2	○	×		Power supply ground voltage (GND) or Open	Connect to GND or Open

Note 1. The pin connection differs depending on the setting of OSCTH pin.
For details, refer to the “R-IN32M4-CL3 User's Manual: Board Design edition”.
2. In consideration of the replacement from R-IN32M4-CL2 to R-IN32M4-CL3, OPEN is also permitted.
Connect to GND, if not the replacement from R-IN32M4-CL2.

2.5.7 Trace Pins

Pin Name	PKG 23□	PKG 17□	I/O	Interface	Recommended connection when not in use
TRACECLK	○	○	O	Output Buffer (3.3 V) 8 mA	Open
TRACEDATA3–TRACEDATA0	○	○	I/O	Programmable I/O Buffer (3.3 V, 8 mA), 130 kΩ Pull-up	

2.5.8 Test Pins

Pin Name	PKG 23□	PKG 17□	I/O	Interface	Recommended connection when not in use (Required)
TMODE0–TMODE2	○	○	I	Input Buffer (3.3 V) Schmitt in, 160 kΩ Pull-down	Connect to GND
TMS	○	○	I/O	I/O Buffer (3.3 V) 8 mA 130 kΩ Pull-up	Open
TDI	○	○	I	Input Buffer (3.3 V) 130 kΩ Pull-up	
TDO	○	○	O	3-state Output Buffer (3.3 V) 8 mA	
TRSTZ	○	○	I	Input Buffer (3.3 V) Schmitt in, 130 kΩ Pull-up	
TCK	○	○		Input Buffer (3.3 V) 130 kΩ Pull-up	
TEST3	○	×		Renesas Test signal	Connect to VDD (3.3 V) or GND
TEST4, TEST5	○	○	I/O		Open
TEST6, TEST7	○	○	I		Connect to GND

2.5.9 Operating Mode Setting Pins

Pin Name	PKG 23□	PKG 17□	I/O	Interface	Recommended connection when not in use
BOOT0, BOOT1	○	○	I	Input Buffer (3.3 V) Schmitt in	Set these pins in accordance with the operating mode.
MEMIFSEL	○	○			
BUS32EN	○	○			
HIFSYNC	○	○			
HWRZSEL	○	○			
MEMCSEL	○	○			
ADMUXMODE	○	○			

3. Memory Maps

The memory of R-IN32M4-CL3 consists of three spaces: internal AHB space (4 Gbytes, 0000 0000H to FFFF FFFFH), internal AXI space (1 Mbyte, 0 0000H to F FFFFH), and external MCU interface space (2 Mbytes).
An accessible area depends on the operation mode and the state of each control register.

Note. Access to the "Reserved" areas is prohibited.
The operation is not guaranteed if an attempt is made to access these areas.

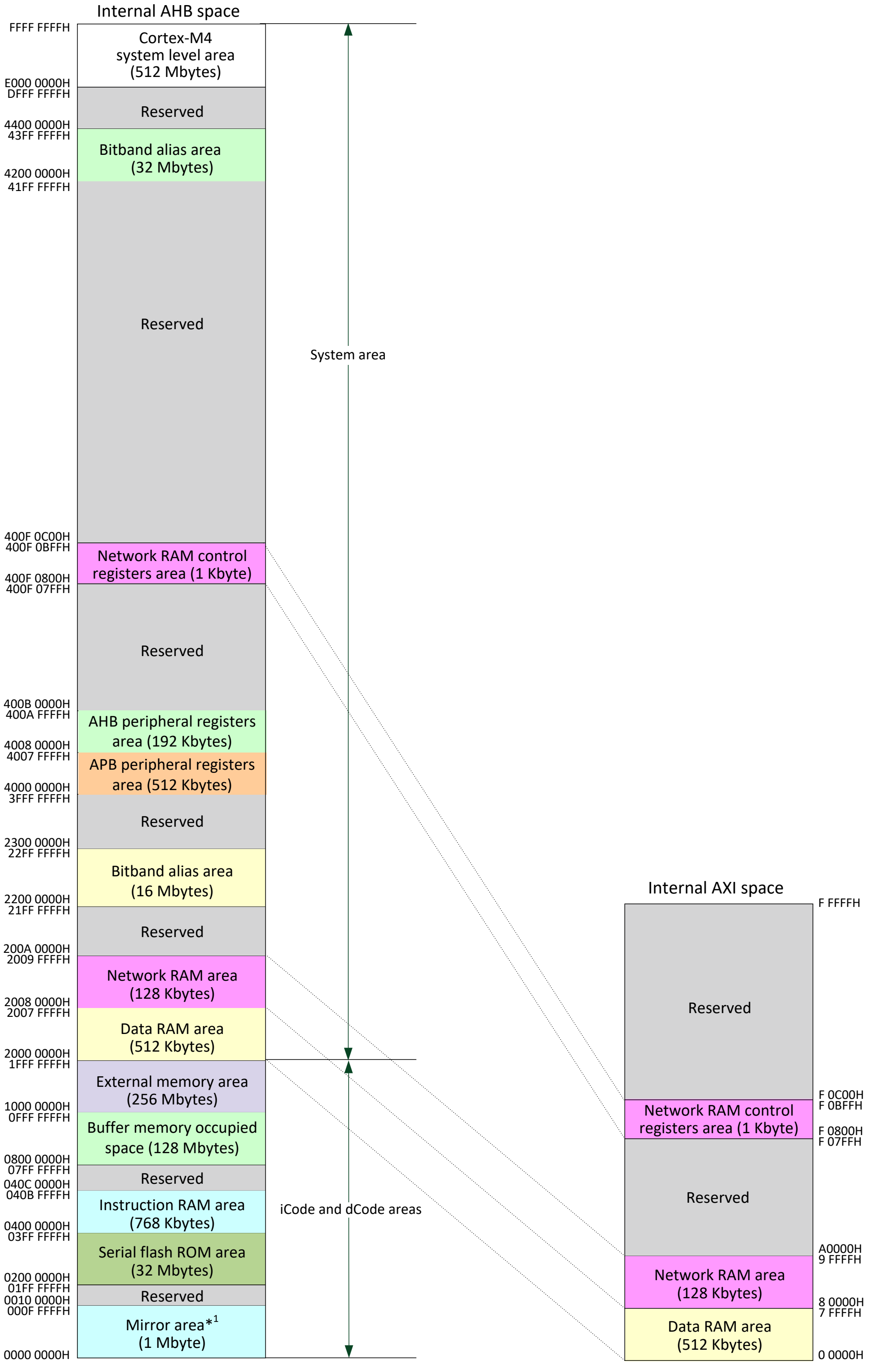


Figure 3.1 Memory Map (All)

Note 1. The mirror area is an area where one of the external memory area/serial flash memory area/instruction RAM area can be mirrored by setting the external pins (BOOT0, BOOT1). For details, refer to Section 7 "Booting Procedure".

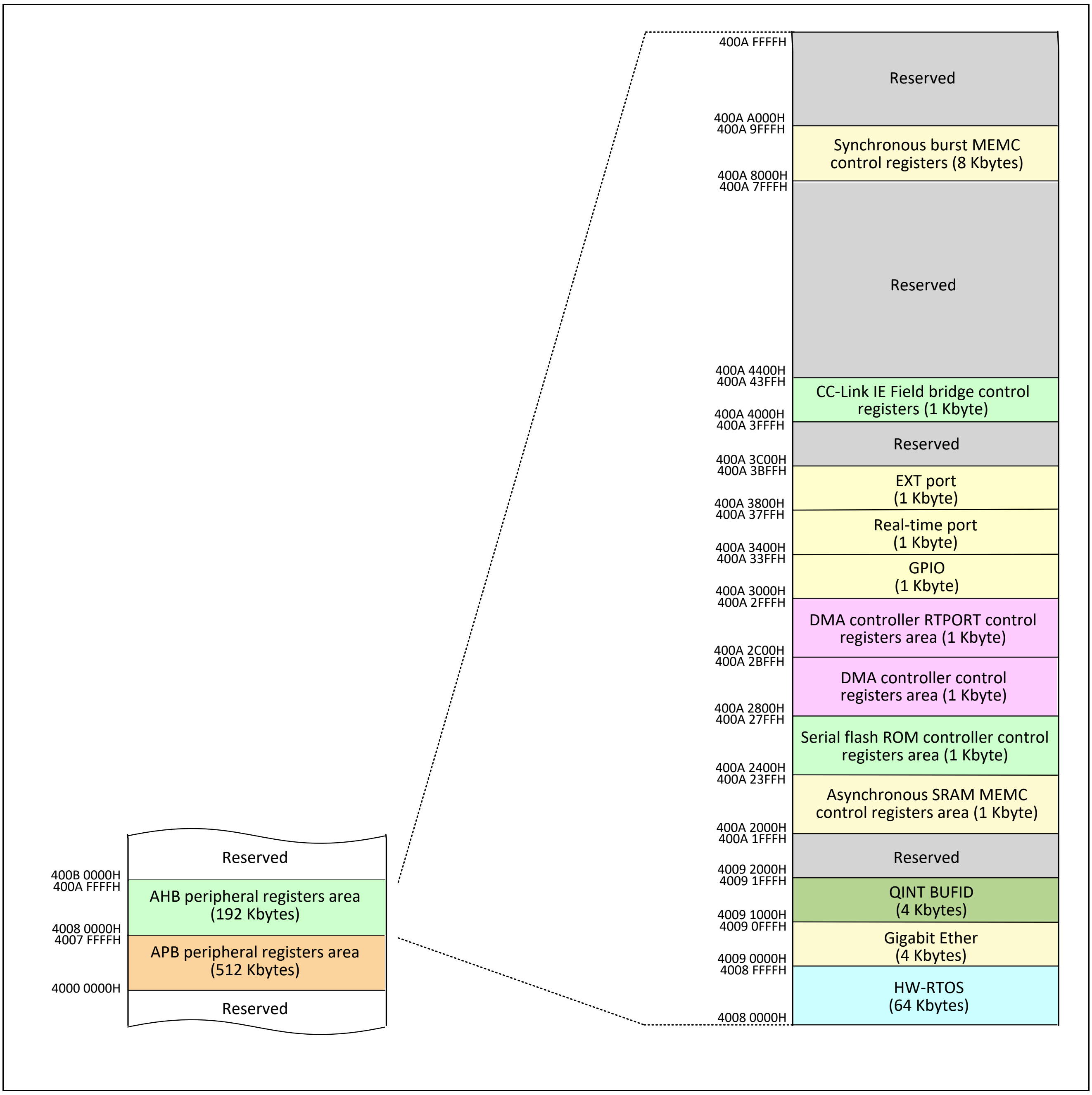


Figure 3.2 Memory Map (AHB Peripheral Registers Area)

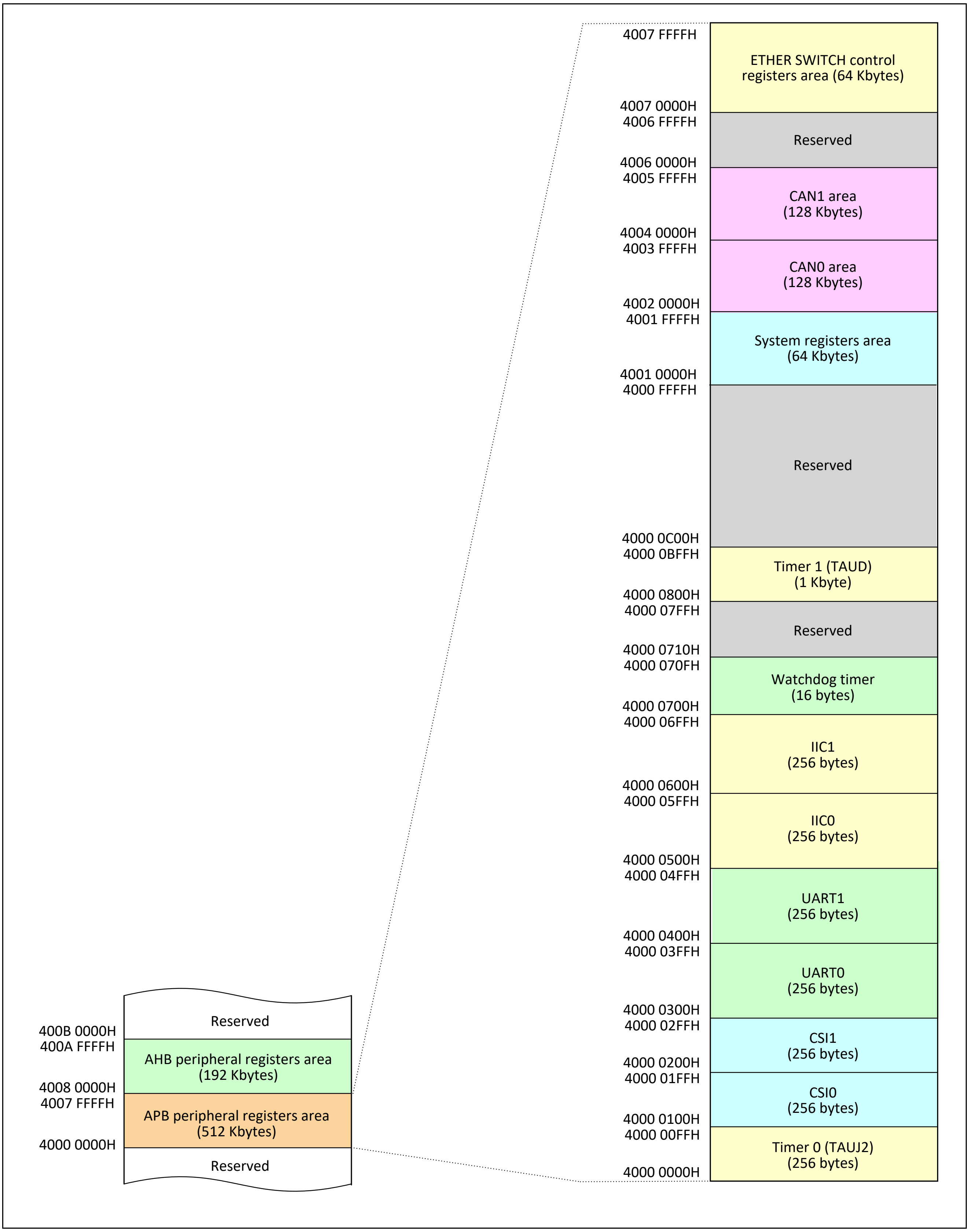


Figure 3.3 Memory Map (APB Peripheral Registers Area)

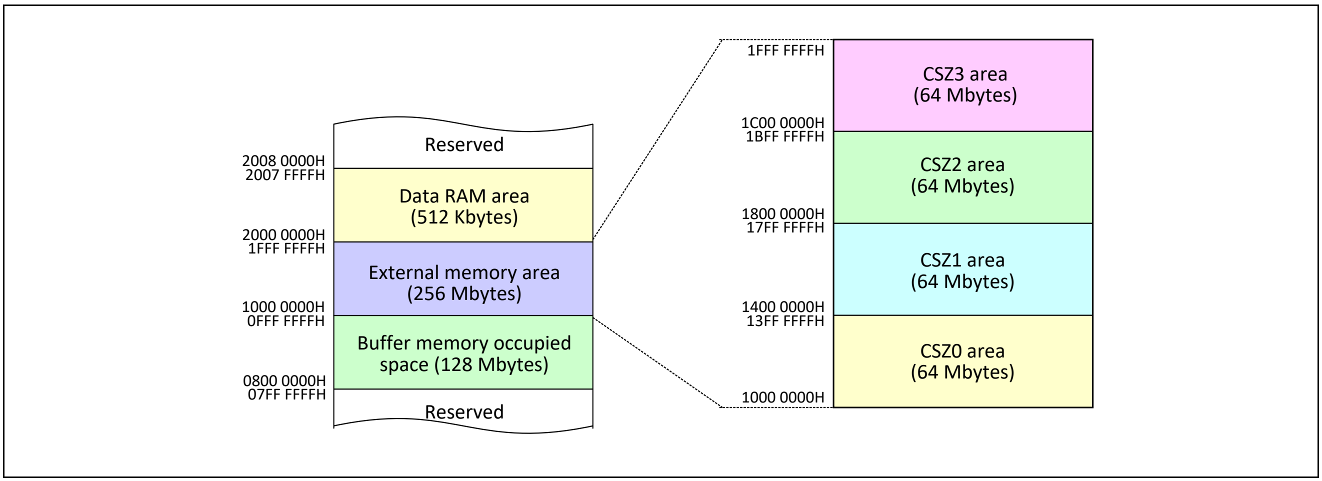


Figure 3.4 Memory Map (External Memory Area of the 23 mm Square Package)

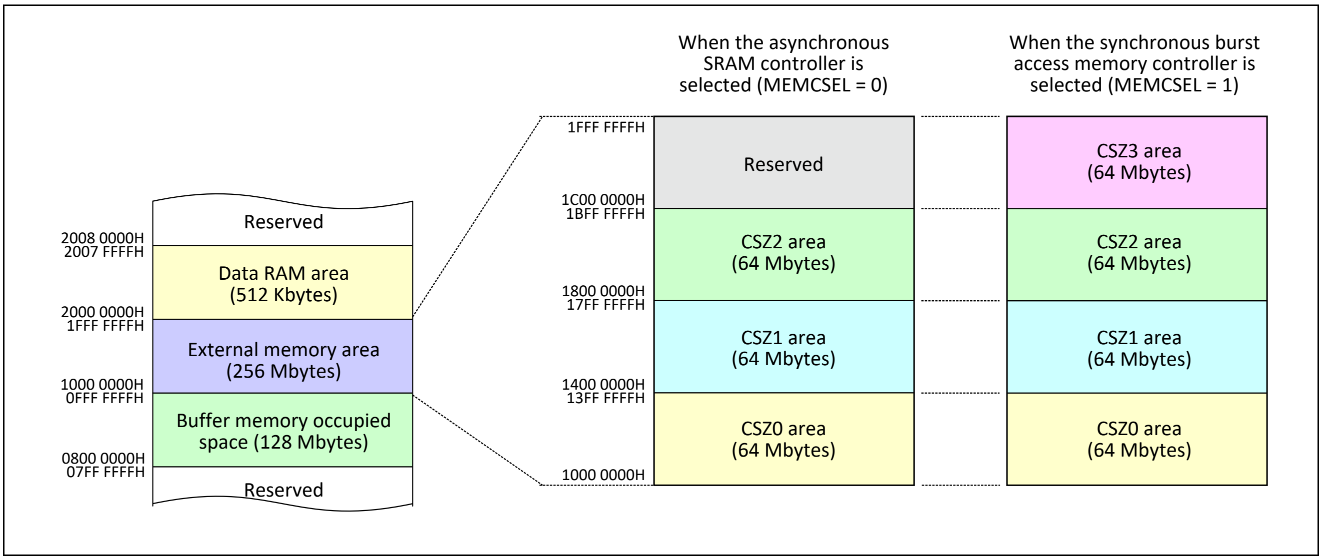


Figure 3.5 Memory Map (External Memory Area of the 17 mm Square Package)

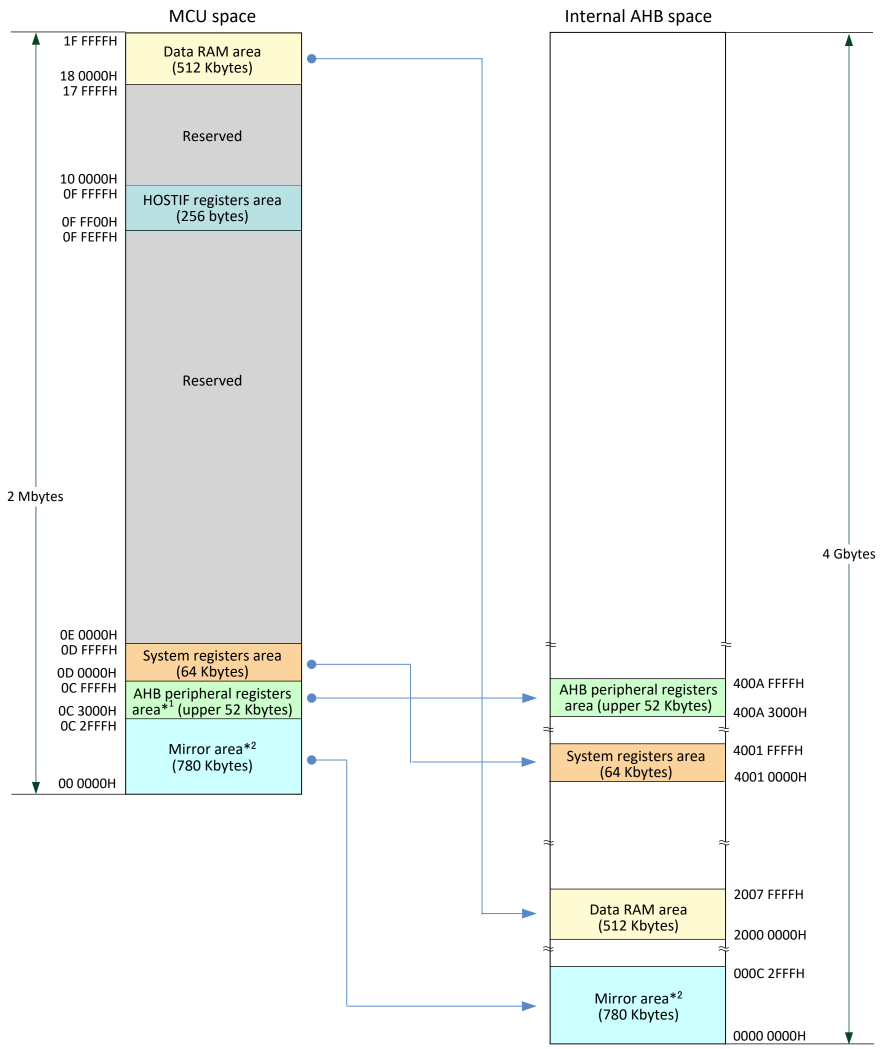


Figure 3.6 External MCU Interface Space

Note 1. The upper 52 Kbytes of the AHB peripheral registers area cover the range from the GPIO area to the synchronous burst memory controller control registers. For details, refer to Figure 3.2 “Memory Map (AHB Peripheral Registers Area)”.

3. The mirror area is an area where one of the external memory area/serial flash memory area/instruction RAM area can be mirrored by setting the external pins (BOOT0, BOOT1). When access is from the external MCU interface, the following areas can be accessed.

• When BOOT0, BOOT1 = 01 (External serial flash ROM boot): Reserved area (access prohibited)

• When BOOT0, BOOT1 = 10 (External MCU boot): Instruction RAM area

For the lists of slaves accessible by the bus masters, refer to Section 6 “Bus Architecture”.

For details on boot mode, refer to Section 7 “Booting Procedure”.

4. Clock/Reset Function

4.1 Clock Configuration

4.1.1 Description of Internal Clocks

R-IN32M4-CL3 uses various clocks.
The following lists the major clock signals covered in this manual.

Clock Signal	Application
OSCCLK	A clock before passing through the internal PLL. This is a 25-MHz clock with no frequency multiplication.
FCLK	A clock signal of the internal AHB system bus. This clock is the base clock signal for use in access to the CPU and Ethernet MAC. The duty of this clock signal is 50%.
HCLK	A clock signal of the internal AHB system bus. This clock is the base clock signal for use in access to the HW-RTOS, DMA controller, and memory controller. The duty of this clock signal is 50%. The watchdog timer A uses this clock as is, or with its frequency divided.
PCLK	A clock signal for internal peripheral macros. This clock is the base clock used for accessing peripheral circuits such as the timers, serial interfaces, and I^2C. The frequency of PCLK is the same as that of HCLK, therefore the frequency of PCLK is also 100 MHz. The duty of this clock signal is 50%.
ACLK	A clock signal of the internal AXI system bus. This is a 125-MHz clock. The duty of this clock signal is 50%.
BUSCLK*1	A clock signal of the external bus interface used by the memory controller. The frequency of this clock signal is the same as that of HCLK.
HBUSCLK*2	A clock signal for the external MCU interface. This clock signal is used in clock-synchronous access by an external MCU.

Note 1. BUSCLK operates as follows when the synchronous burst access memory controller is selected.

1. Frequency division setting:

Division settings of 1/2 to 1/6 are selectable by using the BCLKSEL register.

2. Clock operation by access to memory

• No memory access: Clock is stopped. (fixed to the Low level)

• At the time of access to asynchronous memory: Clock is stopped. (fixed to the Low level)

• At the time of access to synchronous memory: Clock is output only at the time of access.

2. When using asynchronous mode, input a Low level to the HBUSCLK pin.

4.1.2 Clock Configuration Diagram

Figure 4.1 shows the clock configuration diagram.
This is a schematic diagram. It does not show all the clock configurations.

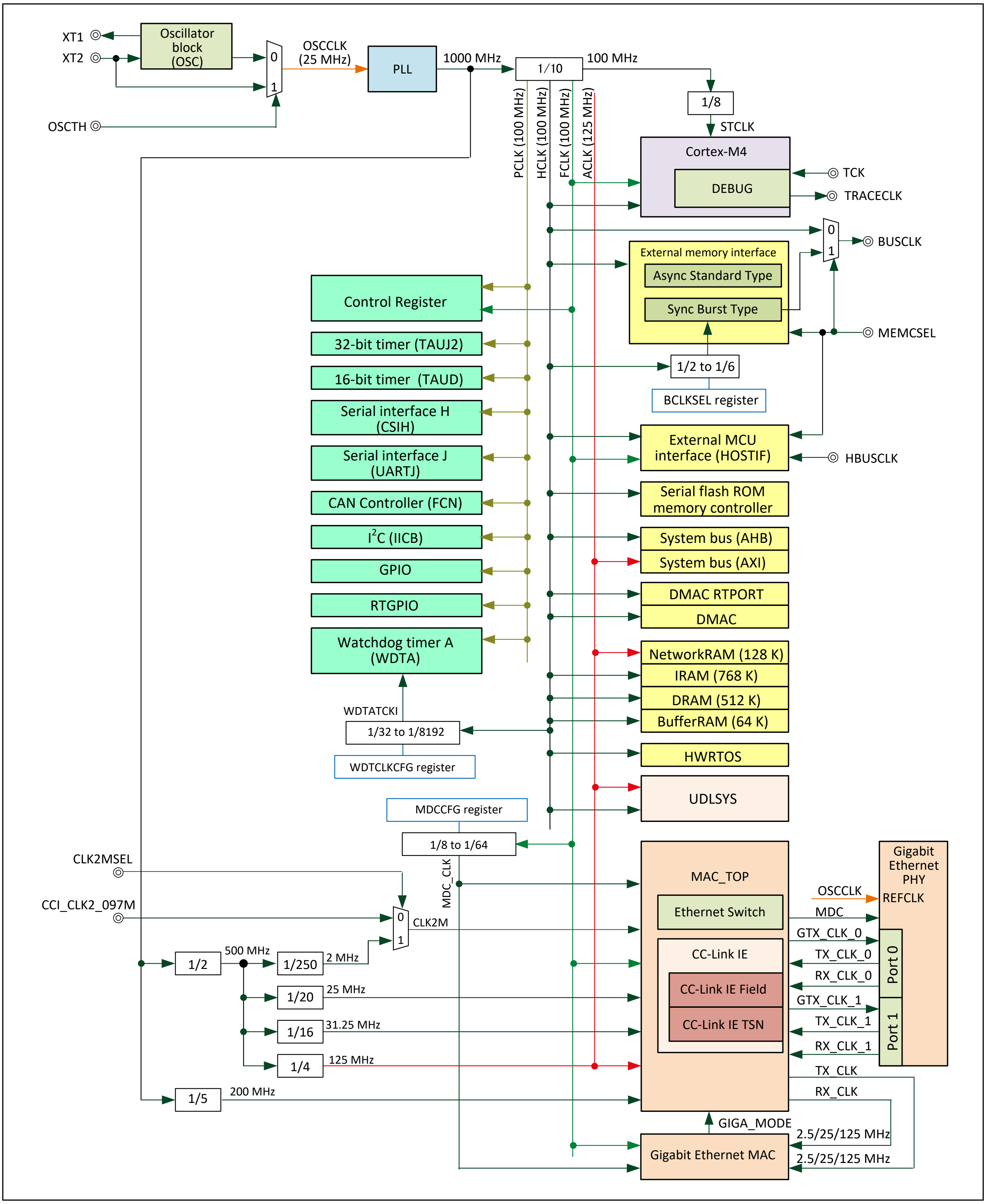


Figure 4.1 Clock Configuration Diagram (Schematic)

4.2 Clock Supply Stop Function

4.2.1 Overview

R-IN32M4-CL3 is capable of stopping clock supply to unused functions. Once the clock supply is stopped by using the CLKGTD register, it cannot be resumed. To supply the clock signal again, reset the entire system.

4.2.2 Clock Control Register (CLKGTD0, CLKGTD1, CLKGTD2)

This register is used to stop clock supply to unused modules to save power.

• This register can be read or written in 32- or 16-bit units.

Caution 1. This register is writable only when a protection release sequence is performed using the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

2. Once the clock supply is stopped by using the CLKGTD register, it cannot be resumed. To supply the clock signal again, reset the entire system.

3. Access to stopped modules is prohibited. Operation is not guaranteed if an attempt is made to access these modules.

CLKGTD0 — Address: BASE + 01A0H; Initial value: 0000 FFFFH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLKGTD0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKGTD0	GCI2C1	GCI2C0	GCUART1	GCUART0	GCTIMER1	GCTIMER0	GCRTOS	GCHOSTIF	1	1	1	1	GCMEMC	GCSROM	GCDMACRT	GCDMAC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	1	1	1	1	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15	GCI2C1	I^2C bus (I2C1) function (1: Operating, 0: Stopped)
14	GCI2C0	I^2C bus (I2C0) function (1: Operating, 0: Stopped)
13	GCUART1	Asynchronous serial interface 1 (UART1) function (1: Operating, 0: Stopped)
12	GCUART0	Asynchronous serial interface 0 (UART0) function (1: Operating, 0: Stopped)
11	GCTIMER1	16-bit timer array unit (TAUD) function (1: Operating, 0: Stopped)
10	GCTIMER0	32-bit timer array unit (TAUJ2) function (1: Operating, 0: Stopped)
9	GCRTOS	HW-RTOS (including Gigabit Ethernet MAC) function (1: Operating, 0: Stopped)
8	GCHOSTIF	External MCU interface function (1: Operating, 0: Stopped)
7–4	—	Reserved (When writing to these bits, write 1. When read, 1 is returned.)
3	GCMEMC*	Memory controller (asynchronous/synchronous burst access) function (1: Operating, 0: Stopped)
2	GCSROM	Serial flash ROM function (1: Operating, 0: Stopped)
1	GCDMACRT	DMAC for real-time ports function (1: Operating, 0: Stopped)
0	GCDMAC	General-purpose DMAC function (1: Operating, 0: Stopped)

Note. Whether to stop or supply the clock signal to the asynchronous SRAM controller and synchronous burst access memory controller cannot be specified separately.

Address: BASE + 01A4H

Initial value: 23 mm Square Package: 0000 6EDFH; 17 mm Square Package: 0000 EEDFH

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKGTD1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	GCIE	1	0	GCIE2	1	1	0	GCBCLK	GCESW	0	GCWDT	GCFCN1	GCFCN0	GCCSI1	GCCSI0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	R/W	1	0	R/W	1	1	0	R/W	R/W	0	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description	
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)	
15	—	23 mm Square Package	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
		17 mm Square Package	Reserved (When writing to these bits, write 1. When read, 1 is returned.)
14	GCIE	CC-Link IE function (1: Operating, 0: Stopped)	
13	—	Reserved (When writing to these bits, write 1. When read, 1 is returned.)	
12	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)	
11	GCIE2	CC-Link IE function (CLK2M clock only) (1: Operating, 0: Stopped)	
10–9	—	Reserved (When writing to these bits, write 1. When read, 1 is returned.)	
8	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)	
7	GCBCLK*	BUSCLK output function (1: Operating, 0: Stopped)	
6	GCESW	Ethernet Switch function (1: Operating, 0: Stopped)	
5	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)	
4	GCWDT	Watch Dog Timer function (1: Operating, 0: Stopped)	
3	GCFCN1	CAN1 function (1: Operating, 0: Stopped)	
2	GCFCN0	CAN0 function (1: Operating, 0: Stopped)	
1	GCCSI1	CSI1 function (1: Operating, 0: Stopped)	
0	GCCSI0	CSI0 function (1: Operating, 0: Stopped)	

Note. When the synchronous burst access memory controller is used, the GCBCLK bit needs to be fixed to 1. When the GCBCLK bit is set to 0, the synchronous burst access memory controller stops its operation.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKGTD2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	GCNWRAM	1	1	1	1
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	R/W	1	1	1	1

Address: BASE + 01A8H
Initial value: 0000 00FFH

Bit	Bit Name	Description
31–8	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
7–5	—	Reserved (When writing to these bits, write 1. When read, 1 is returned.)
4	GCNWRAM	Network RAM function (1: Operating, 0: Stopped)
3–0	—	Reserved (When writing to these bits, write 1. When read, 1 is returned.)

4.3 Reset Function

R-IN32M4-CL3 has various reset functions.
The following are the major reset functions described in this manual.

4.3.1 Overview

- Reset control by hardware
  - Reset by signal input from the RESETZ pin
  - Power-on reset function by signal input from the PONRZ pin (including initialization processing of the R-IN32M4-CL3 internal RAM)
  - Reset by signal input from the HOTRESETZ pin
- ResetReset control by software
- Reset control by the window watchdog timer A (WDTA)
- Reset output (RSTOUTZ)
- Noise elimination for external reset input signals (Applicable pins: RESETZ, PONRZ, HOTRESETZ, and TRSTZ)

Table 4.1 Reset Sources and Targets to be Reset

Reset source		Target to be reset						RSTOUTZ output
		Instruction RAM, Data RAM, Buffer RAM, Network RAM, Control register*2	PLL	CC-Link IE power-on reset control register*3	CPU debugging unit	Gigabit Ethernet PHY	Other peripheral circuits (including CPU)	
Hardware	PONRZ	○	○	○	—	○	○	○
	RESETZ	—	○	○	—	○	○	○
	HOTRESETZ	—	—	—	—	—	○	○
	TRSTZ	—	—	—	○	—	—	—
Software	Window watchdog timer A (WDTA0TRES)	—	—	—	—	—	○	○
	System reset register (SYSRESET)	—	—	—	—	—	○	○
	AIRCR register*1	—	—	—	—	—	○	○
	PHYRSTregister (PHYRST)	—	—	—	—	○	—	—

○: Applicable, —: Not applicable

Notes 1. This is a system control register (0xE000_ED18) in the Cortex-M4.
It can be reset by setting "1" to AIRCR[2].SYSRESETREQ.
2. The control registers that can be reset only by signal input from the PONRZ pin are PHYADD, MACSEL, and MDIOSEL.
3. The control register that can be reset only by signal input from the PONRZ pin and the RESETZ pin is PHYRST.

4.3.2 Reset Configuration Diagram

Figure 4.2 shows the reset configuration diagram.
This is a schematic diagram. It does not show all the reset configurations.

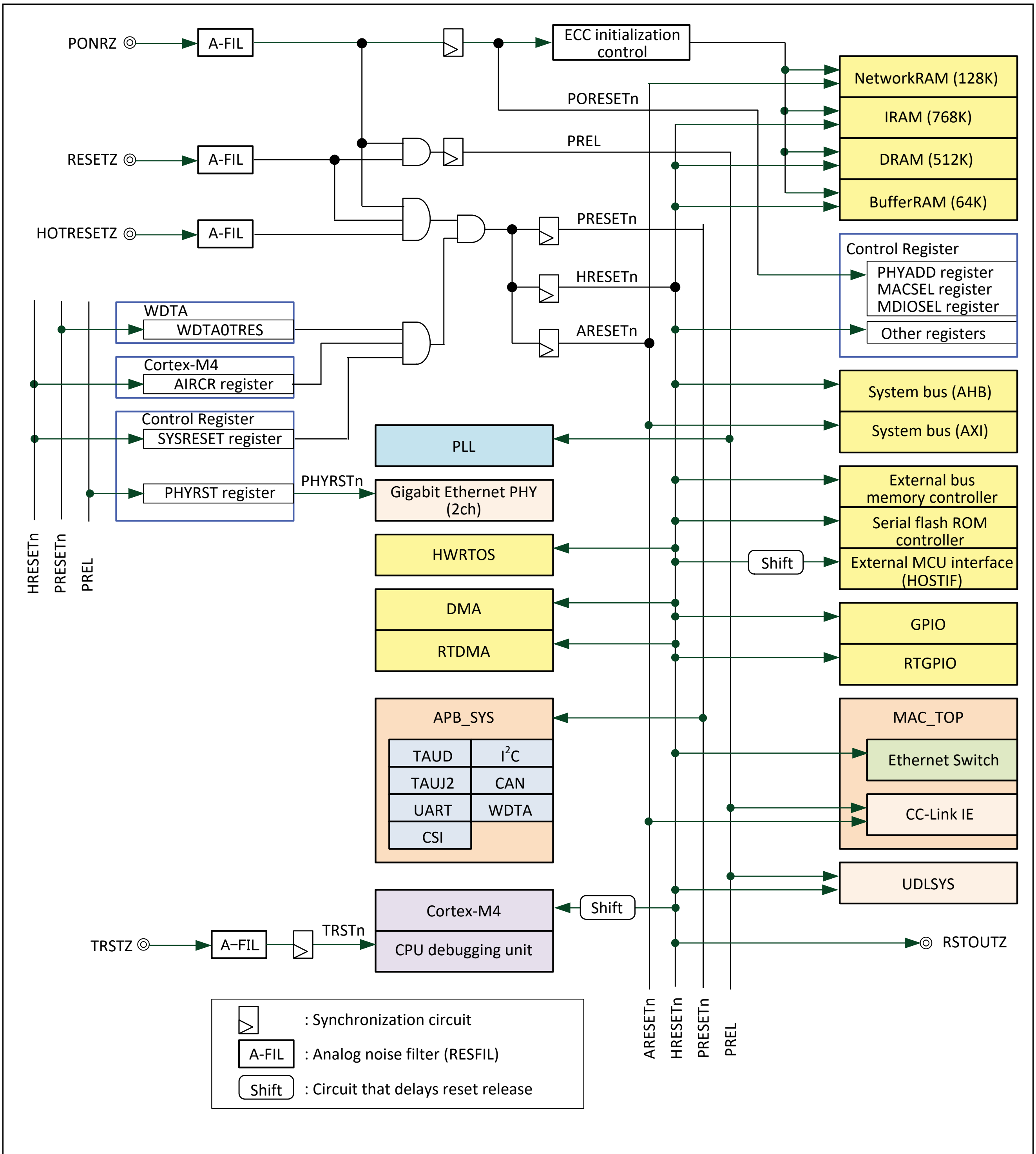


Figure 4.2 Reset Configuration Diagram (Schematic)

4.3.3 Types of Reset

(1) Reset Control by Hardware

When the reset signal PONRZ, RESETZ, or HOTRESETZ is input, the CPU core and internal peripheral modules are initialized. Note that the input of HOTRESETZ does not reset the internal PLL.
The width at Low level of each reset signal must be at least 1 µs.
The oscillation stabilization time of the external oscillator clock (25 MHz) must be secured using the width at Low level for the signal on each reset pin.
TRSTZ is only connected to the Cortex-M4 debugging unit. When resetting the CPU core and internal peripheral modules from the in-circuit emulator (ICE), connect the target reset signal (nSRST) input via the ICE connector to the RESETZ pin by using logic such as wired OR. For an example of the connection, refer to Section 9 "Debugging".

(2) Reset Control by Software

R-IN32M4-CL3 can be reset by using the system reset register (SYSRESET). This reset is equivalent to a reset executed by the input of a signal to the HOTRESETZ pin. The internal RAM is not initialized.
Hardware reset for Gigabit Ethernet PHY can be controlled by using the reset signal RESETZ or PONRZ, or the CC-Link IE dedicated register or the PHYRST register.

(3) Reset Control by the Window Watchdog Timer A (WDTA)

When a reset request is generated by the window watchdog timer A (WDTA), the CPU core and internal peripheral modules are initialized. This reset is equivalent to a reset executed by the input of a signal to the HOTRESETZ pin. The internal RAM is not initialized.

(4) Reset output (RSTOUTZ output)

When a reset is generated in R-IN32M4-CL3, a low-level signal is output from the RSTOUTZ pin. This reset can be used as a general-purpose reset for external devices.

(5) Analog Noise Filter (RESFIL)

The RESFIL is a circuit for eliminating noise from a power-on reset input (PONRZ), reset input (RESETZ), hot reset input (HOTRESETZ), and JTAG reset (TRSTZ). This circuit eliminates noise shorter than 100 ns. The noise eliminator is skipped in test mode.

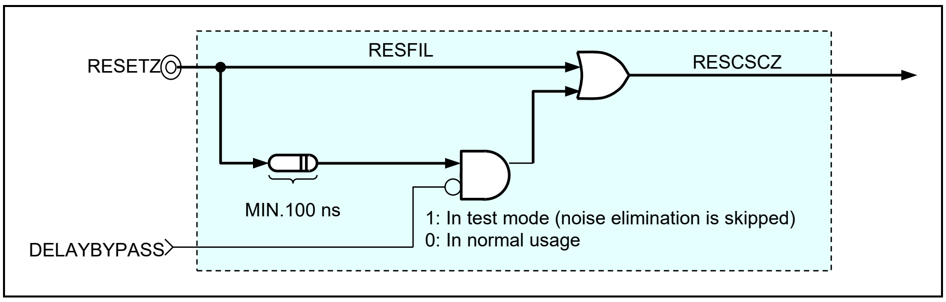


Figure 4.3 Configuration of the Noise Eliminator for Reset Input

4.3.4 Reset Control Registers

(1) System Reset Register (SYSRESET)

This register resets R-IN32M4-CL3 (equivalent to the HOTRESETZ input pin). The registers targeted only by the PONRZ or RESETZ pin are not reset. When system reset is performed for this register (by writing "0" to SYSRESET), the reset is automatically released (SYSRESET will be "1") after R-IN32M4-CL3 is reset.

• This register can be read or written in 32- or 16-bit units.

Caution: This register is writable only when a protection release sequence is performed using the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
SYSRESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SYS RST	BASE + 01C0H	0001H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W		

Bit	Bit Name	Description
0	SYSRST	This register performs system reset of R-IN32M4-CL3. 0: System reset start 1: System reset is released

(2) PHY Reset Register (PHYRST)

This register performs reset control of the PHY.

• This register can be read or written in 32- or 16-bit units.

Caution: This register is writable only when a protection release sequence is performed using the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

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

Address
BASE + 1220H
Initial value
0000 0000H

Bit	Bit Name	Description
0	PHYRST	This register performs reset control of the PHY. 0: PHY is in the reset state 1: PHY reset is released

This register can be initialized only by signal input from the PONRZ pin and the RESETZ pin, which are external pins.

4.3.5 Operations for Reset

The charts below show the timing of the reset at power-on and when a system reset is issued for R-IN32M4-CL3.

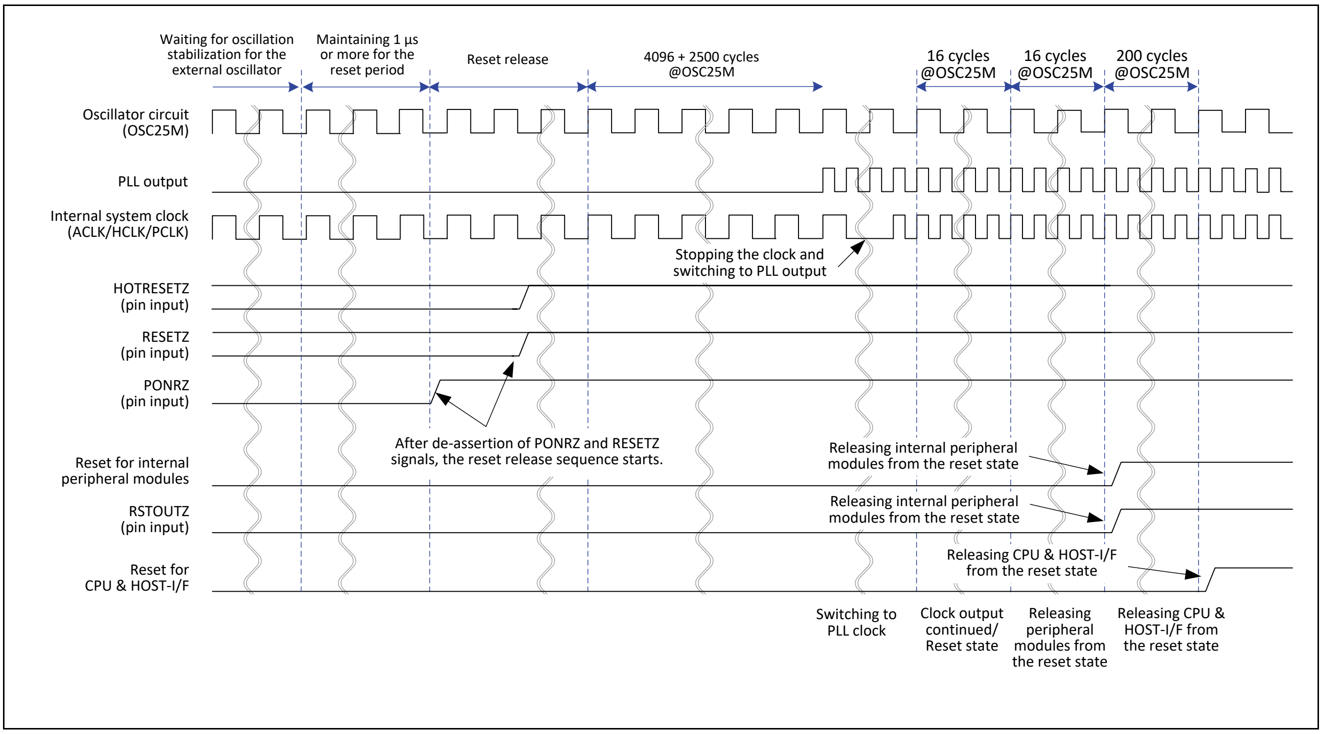


Figure 4.4 Timing of Reset at Power-On

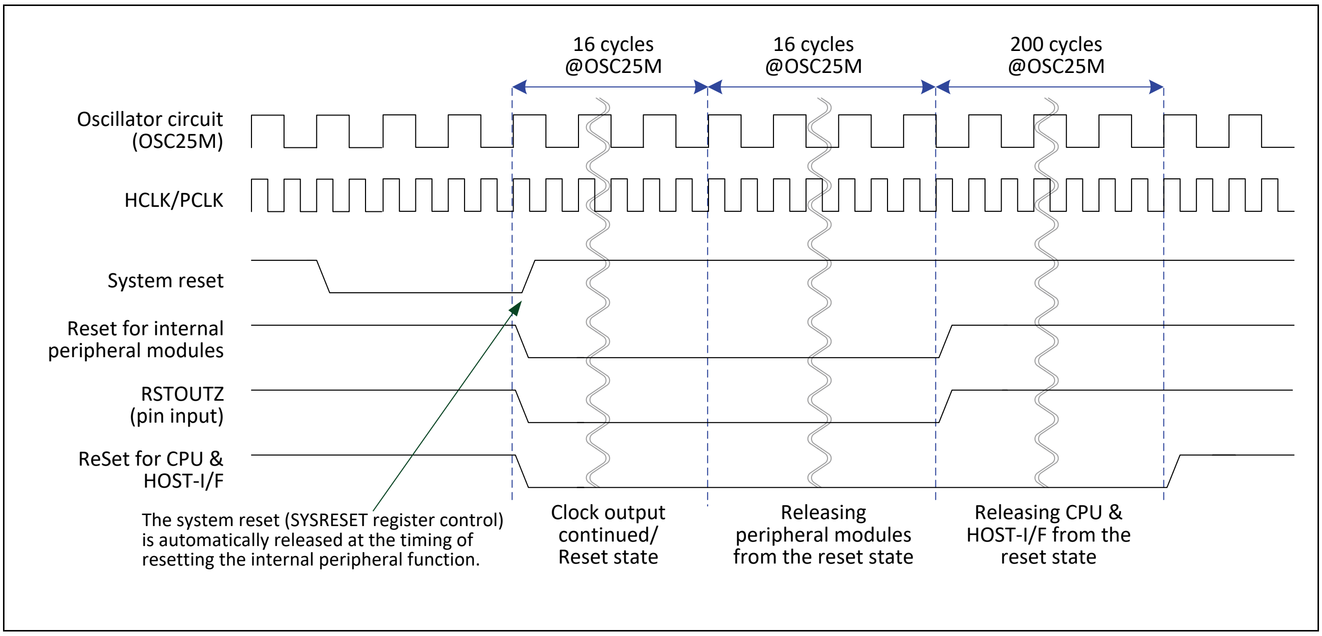


Figure 4.5 Timing of Reset at System Reset

5. CPU and Internal RAMs

R-IN32M4-CL3 incorporates a high-performance 32-bit processor (Arm Cortex-M4 core).
This section describes an overview of the CPU and internal RAMs of R-IN32M4-CL3.

5.1 CPU-core Information

The revision of the Cortex-M4 core currently used in R-IN32M4-CL3 is Cortex-M4 r0p1.
For details on the CPU core section, architecture, or others, refer to the following Arm Limited website.

http://infocenter.arm.com/help/topic/com.arm.doc.set.cortexm/index.html

5.2 CPU-core Configuration Information

The Cortex-M4 of R-IN32M4-CL3 has the following configurations.

Category	Configuration Item	Setting	Remark
Interrupt	NUM_IRQ	240	The number of IRQ interrupts to be input: 1 to 240 (NMI interrupts are counted separately.)
Interrupt Priority	LVL_WIDTH	4	Priority bit numbers 3 to 8 (8 to 256 priority levels)
MPU	MPU_PRESENT	Yes	Presence of the memory protection unit
Debug Level	DEBUG_LVL	3	Debug level 1 to 3
Trace Level	TRACE_LVL	2	Trace level 0 to 2
SW/SWJ-DP Selection	JTAG_PRESENT	SWJ-DP	SWJ-DP is selected when the JTAG access circuit is built-in.
Bit-band Area	BB_PRESENT	Yes	Presence of bit-banding

Debug Level	1	2	3 (setting in R-IN32M4-CL3)
Function Outline	Minimum debug configuration	Full debug configuration (without data matching)	Full debug configuration (with data matching)
Debugging Halt	Yes	Yes	Yes
Breakpoints	2 (Instruction)	6 (Instruction) 2 (Literal)	6 (Instruction) 2 (Literal)
DWT Comparator Number	1 (Data matching is not available.)	4 (Data matching is not available.)	4
Flash Patch Function	No	Yes	Yes

Trace Level	0	1	2 (setting in R-IN32M4-CL3)
Function Outline	No trace	Standard trace	Full trace
ITM and TPIU Functions	No	Yes	Yes
DWT Trigger and Counter	No	Yes	Yes
ETM function	No	No	Yes

Caution: R-IN32M4-CL3 does not support SLEEPDEEP mode. Do not set the SLEEPDEEP bit of the SCR register to 1.

5.3 Internal Instruction RAM

The internal instruction RAM is a 768-Kbyte RAM. It is accessible by AHB.

5.3.1 Outline of Features

- Includes a 128-bit (32 bits x 4) read buffer
- Latency: latency is 2 in read access in general but 1 in the case of hitting the read buffer.
  latency is 1 in write access.
- AHB bus width: 32 bits
- RAM data bus width: 128 bits (without ECC circuit)
- Transfer size: 16- or 32-bit transfer selectable
- Support for burst transfer
- Little endian fixed
- Support for ECC (1-bit error correction, 2-bit error detection)

Table 5.1 Interrupt from Internal Instruction RAM and Request for Peripheral Modules

Internal Instruction RAM Interrupt Signal	Function	Connected to
IRAMECCSEC	Instruction RAM ECC 1-bit error correct interrupt	Interrupt controller
IRAMECCDED	Instruction RAM ECC 2-bit error detect interrupt	Interrupt controller

5.3.2 Read Buffer

- 128-bit (32 bits × 4) read buffer
- Response to the AHB involves with latency 0 in the case of hitting the read buffer.
- Clear the data in the read buffer when a 2-bit ECC error occurs.
- A 2-bit ECC error at the time of the read response generates an ECC error interrupt.

5.3.3 Write Interface

- When 16-bit write access arises, write to the RAM in 32-bit units through two consecutive rounds of access.
- When 8-bit write access arises, return an error response.

Caution: Write access by an external MCU in 16-bit units may occur.
The specification assumes that such access to the RAM will always proceed two consecutive times (for the writing of data in 32-bit units).

5.4 Internal Data RAM

The internal data RAM is a 512-Kbyte RAM. It is accessible by the AHB and Header Endec (Communication-BUS).

5.4.1 Outline of Features

- AHB latency: latency is 1 in read and write access (latency is 2 in read access following write access).
- Communication-BUSlatency: latency is 1 in read and write access
- Arbitration of access when contention arises: Round robin
- AHB bus width: 32 bits
- Communication-BUS width: 128 bits
- RAM bus width: 128 bits (without ECC circuit)
- AHB transfer size: 8-, 16-, or 32-bit transfer selectable
- Communication-BUS transfer size: 8-, 16-, 32-, 128-bit transfer selectable
- Support for burst transfer
- Little endian fixed
- Support for ECC (1-bit error correction, 2-bit error detection)

Table 5.2 Interrupt from Internal Data RAM and Request for Peripheral Modules

Internal Data RAM Interrupt Signal	Function	Connected to
DRAMECCSEC	Data RAM ECC 1-bit error correct interrupt	Interrupt controller
DRAMECCDED	Data RAM ECC 2-bit error detect interrupt	Interrupt controller

5.5 Buffer RAM

The buffer RAM is a 64-Kbyte RAM. It is accessible by the Communication-BUS.

5.5.1 Outline of Features

- Communication-bus latency: latency is 1 in read and write access
- Communication-BUS width: 128 bits
- RAM bus width: 128 bits (without ECC circuit)
- Communication-BUS transfer size: 8-, 16-, 32-, 128-bit transfer selectable
- Support for ECC (1-bit error correction, 2-bit error detection)

Table 5.3 Interrupt from Buffer RAM and Request for Peripheral Modules

Buffer RAM Interrupt Signal	Function	Connected to
BRAMECCSEC	Buffer RAM ECC 1-bit error correct interrupt	Interrupt controller
BRAMECCDED	Buffer RAM ECC 2-bit error detect interrupt	Interrupt controller

5.6 Network RAM

The network RAM is a 128-Kbyte RAM. The RAM is accessible by AXI.

5.6.1 Outline of Features

- Includes a write buffer
- AXI latency:
  - —When the ECC function is enabled
    - Read access: Single transfer: Latency is 3.
      Burst transfer: First data latency is 3, second data or later latency is 2.
    - Write access: 8- or 16-bit writing: Latency is 3.
      32- or 64-bit writing: Latency is 1.
  - —When the ECC function is disabled
    - Read access: Single transfer: Latency is 2.
      Burst transfer: First data latency is 2, second data or later latency is 1.
    - Write access: 8- or 16-bit writing: Latency is 1.
      32- or 64-bit writing: Latency is 1.
- AXI bus width: 64 bits
- RAM bus width: 64 bits (without ECC circuit)
- AXI transfer size: 8-, 16-, 32-, or 64-bit transfer selectable
- Support for ECC (1-bit error correction, 2-bit error detection)

Table 5.4 Interrupt from Network RAM and Request for Peripheral Modules

Network RAM Interrupt Signal	Function	Connected to
NRAMECCSEC	Network RAM 1-bit ECC error detection interrupt	Interrupt controller
NRAMECCDED	Network RAM 2-bit ECC error detection interrupt	Interrupt controller

5.6.2 Control Register

(1) Register List

Register Name	Abbreviation	Address
Network RAM ECC Function Control Register	NRAMCTL	BASE + 2010H
Network RAM Lower 32-bit ECC control register	NRAM_CTL_0	400F 0800H
Network RAM Upper 32-bit ECC control register	NRAM_CTL_1	400F 0840H

(2) Network RAM ECC Function Control Register (NRAMCTL)

This register is used to enable or disable the ECC function of the network RAM.

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

Caution 1. This register is writable only when a protection release sequence is performed using the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

2. Change the setting of this register from the initial value, immediately after the boot or system reset release and before the access to the network RAM.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRAMCTL	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	ECCERMK	0	0	0	ECCEN
R/W	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	R/W	0	0	0	R/W

Address BASE + 2010H

Initial Value 0000 0001H

Bit	Bit Name	Description
31–5	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
4	ECCERMK	Configure whether to return an error response over the AXI bus when a 2-bit ECC error occurs during the network RAM read. 0: Returns an error response over the AXI bus 1: Not returns an error response over the AXI bus
3–1	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
0	ECCEN	Network RAM ECC Function Enable 0: ECC function disabled 1: ECC function enabled

(3) Network RAM ECC control register (NRAM_CTL_n)

These registers are used to set the operation when an ECC error occurs.
There are registers for the upper 32 bits (NRAM_CTL_1) and the lower 32 bits (NRAM_CTL_0) since the ECC is performed in units of 32 bits.

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

NRAM_CTL_0 / NRAM_CTL_1 — Address: 400F 0800H / 400F 0840H — Initial Value: 0000 0010H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NRAM_CTL_0 / NRAM_CTL_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRAM_CTL_0 / NRAM_CTL_1	ECMA		0	0	0	0	0	0	0	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	0	0	1
R/W	R/W		0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	0	0	R

Bit	Bit Name	Description
31–16	—	Reserved
15–14	ECMA[1:0]	Enable Writing to ECERVF 2'b01: Writing enabled Others: Writing disabled If these bits are 2'b01, writing to the ECERVF bit is enabled. Other than 2'b01, writing to the ECERVF bit is disabled. When these bits are read, 2'b00 are always returned.
13–7	—	Reserved
6	ECERVF*1	ECC Error Detection Enable This bit is used to enable or disable the ECC Error Detection. 0: Disables ECC error detection. 1: Enables ECC error detection. When writing to this bit, 15–14 bits must be set to 2'b01. If 15–14 bits are other than 2'b01, writing to this bit is not performed.
5	EC1ECP*1	Single-bit Error Correction Enable This bit is used to enable or disable error correction when an ECC single-bit error is detected. 0: Single-bit error correction enabled 1: Single-bit error correction disabled
4	EC2EDIC*1 *2	2-bit Error Detection Interrupt Control This bit is used to control whether or not to output an interrupt when an ECC 2-bit error is detected. 0: Do not generate an interrupt when a 2-bit error is detected 1: NRAMECCDED interrupt signal is output when a 2-bit error is detected
3	EC1EDIC*1 *2	Single-bit Error Detection Interrupt Control This bit is used to control whether or not to output an interrupt when a single-bit error is detected. 0: Do not generate an interrupt when a single-bit error is detected 1: NRAMECCSEC interrupt signal is output when a single-bit error is detected
2–0	—	Reserved

Notes 1. Changes the ECERVF, EC1ECP, EC2EDIC, and EC1EDIC bits which related to the ECC function control, when the network RAM is not being accessed.

2. For the ECnEDIC control, the ECERVF must be enabled.
For details, refer to Section 5.6.3 "ECC Function Initial Setting".

Remark: m = 0–7, n = 0–1

5.6.3 ECC Function Initial Setting

The procedure for setting the ECC function of the network RAM is described below.

(1) Enabling interrupt function when ECC error is detected

After immediately a power-on reset or system reset, the ECC function (single-bit error correction) is enabled, but the interrupt generation is disabled when an ECC error is detected.
Use the following procedure to enable the interrupt generation when an ECC error is detected.

(1) Network RAM ECC control register setting (NRAM_CTL_0, NRAM_CTL_1)
Set ECERVF to 1 to enable error detection.
Set EC2EDIC to 1 to enable the interrupt generation when an ECC 2-bit error is detected.
Set EC1EDIC to 1 to enable the interrupt generation when an ECC single-bit error is detected.

(2) Disabling ECC

The ECC function is enabled immediately after a power-on reset or system reset.
Use the following procedure to disable the ECC function.

(2) Release the protection using the system protection command register (SYSPCMD).
(3) Sets the ECCEN bit of the ECC function control register to 0, and disables the ECC function.
(4) Enable the protection using the system protection command register (SYSPCMD).

Note 1. Change the ECC function control register, immediately after the boot or system reset release and before the access to the network RAM.

6. Bus Architecture

R-IN32M4-CL3 has two types of internal buses: the AHB and AXI buses. The AHB and AXI internal buses connected by bus conversion bridges, and mutually accessible.

6.1 AHB Internal Bus

A multi-layered configuration is used for the internal AHB buses of an R-IN32M4-CL3, and a bus layer is provided for every seven bus masters. For this reason, except when two or more masters request access to the same slave, queuing for buses does not occur, making for efficient bus usage. In cases of contention for access by two or more masters to the same slave, arbitration proceeds according to the default priority and priority decision system.

Table 6.1 AHB internal bus of an R-IN32M4-CL3

Slave \ Master	High ← (Default Priority) → Low							Priority Decision System
	DMAC for Real-time Ports	Host CPU	Cortex-M4 CPU D Code Bus	Cortex-M4 CPU System Bus	General Purpose DMAC	AXI Master*6	Cortex-M4 CPU I Code Bus	
Data RAM	◎	○	—	○	○	○	—	Round Robin (alternate)*3
Instruction RAM	○	○	○	—	○	—	○	Fixed Priority
Buffer RAM	—	—	○	—	○	—	—	Round Robin (fair)
External Memory	◎	—	○	○	○	—	○	Round Robin (alternate)*3
Serial Flash ROM	—	—	○	○	○	—	○	Round Robin (fair)
Ethernet MAC (CC–Link IE Field)	◎	—	—	○	○	—	—	Round Robin (alternate)*3
APB Internal Peripheral Module*1	◎	○	—	○	○	—	—	Round Robin (alternate)*3
Real-time Ports	◎	○	—	○	○	—	—	Round Robin (alternate)*3
General Ports	◎	○	—	○	○	—	—	Round Robin (alternate)*3
HW-RTOS*2	—	—	—	○	—	—	—	—
DMAC for Real-time Ports*4	—	—	—	○	—	—	—	—
General Purpose DMAC*4	—	—	—	○	—	—	—	—
Synchronous Burst Access Memory Controller	—	—	—	○	—	—	—	—
CC-Link IE TSN	◎	—	—	○	○	—	—	Round Robin (alternate)*3
AXI Slave*5	◎	—	—	○	○	—	—	Round Robin (alternate)*3

Remark **◎: Fixed top priority when round robin (alternate) is specified**
○: Accessible
—: Not accessible

Note 1. This refers to the internal timer, serial interface, system registers, etc. However, the only area accessible by the host CPU is that of the system registers.

2. Hardware read-time OS.

3. RR (alternate): Round robin with fixed priority.

A particular master and slave can be specified as having the fixed-top priority; otherwise, the round-robin system is used for arbitration.

4. The registers areas of the respective DMA controllers

5. The network RAM and network RAM control register connected on the AXI internal bus are the subjects of access.

6. The CC-Link IE TSN (transmitter) and CC-Link IE TSN (receiver) are masters on the AXI internal bus.

6.2 AXI Internal Bus

Three bus masters are connected to the AXI internal bus of an R-IN32M4-CL3. In cases of contention for access by two or more masters to the same slave, arbitration proceeds according to the default priority and priority decision system.

Table 6.2 AXI internal bus of an R-IN32M4-CL3

Master / Slave	High ← (Default Priority) → Low			Priority Decision System
	CC-Link IE TSN (transmitter)	CC-Link IE TSN (receiver)	AHB Master*1	
Network RAM	○	○	○	Round Robin (fair)
Network RAM Control Register	—	—	○	Round Robin (fair)
AHB Slave (Data RAM)	○	○	—	Round Robin (fair)

Remark ○: Accessible

—: Not accessible

Note 1. The real-time port DMAC, Cortex-M4 CPU system bus, and general-purpose DMAC are bus masters on the AHB internal bus.

6.3 Bus Bridges

In R-IN32M4-CL3, the AXI and AHB buses are interconnected via bus bridges. The bus bridges have a bufferable function during the write access.
The error response from the slave changes according to the bufferable attribute of the write access issued by the master.

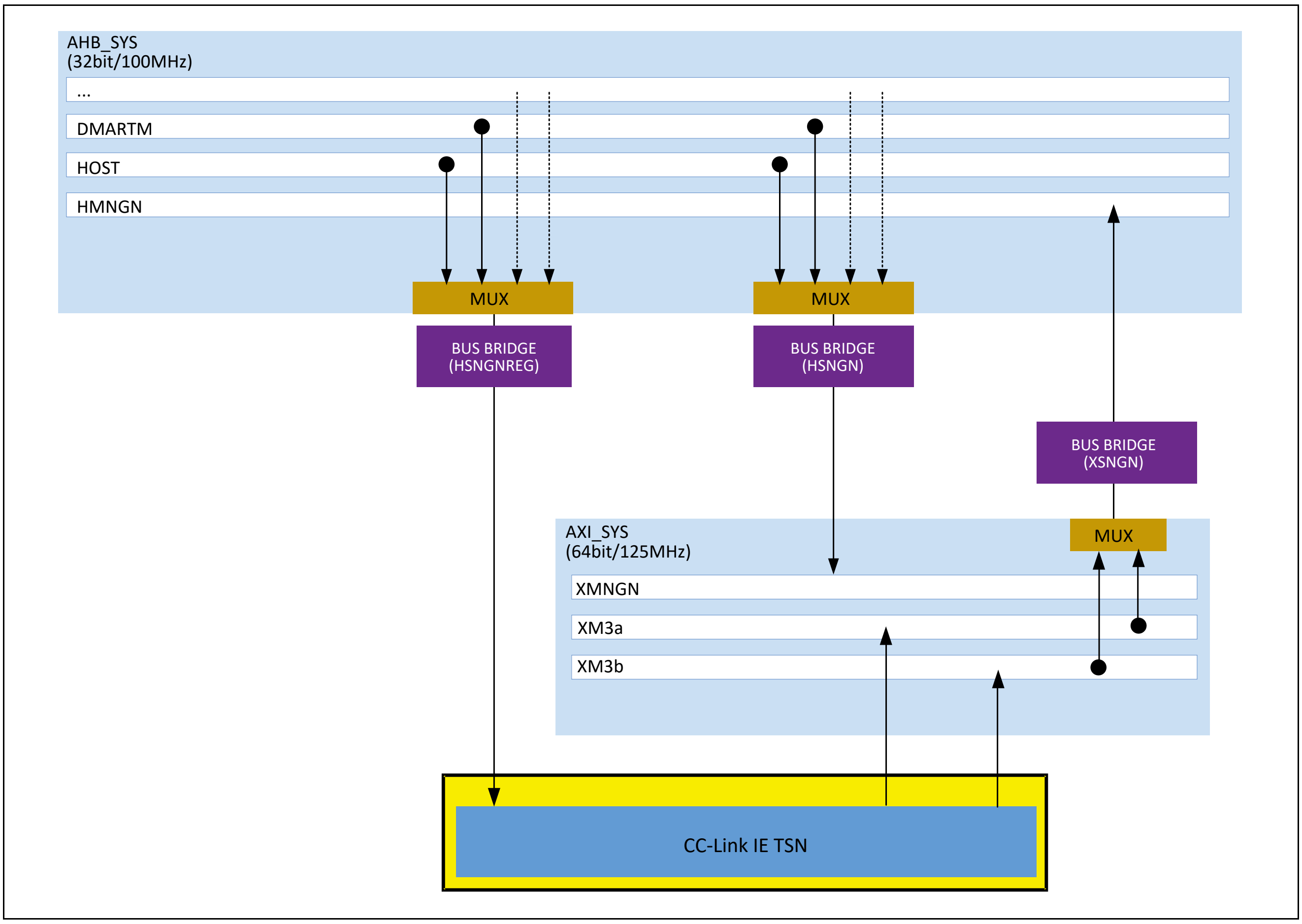


Figure 6.1 AHB–AXI Bus Configuration Diagram

Table 6.3 Error Response of Bus Bridge

Bus Bridge Mode	Response from Slave	Response to Master
Bufferable operation	Normal response	Normal response
	Error response	Normal response + Error interrupt (INTHSNGNREGBE, INTHSNGNBE, INTXSNGNBE)
Non-bufferable operation	Normal response	Normal response
	Error response	Error response

7. Booting Procedure

R-IN32M4-CL3 allows four locations from which to boot the CPU, which can be selected by using external pins BOOT0 and BOOT1: the external memory, the external serial flash ROM, the external MCU, and the instruction RAM. The reset vector and interrupt vector can also be switched by register settings. This section describes booting from the external memory, the external serial flash ROM, and the external MCU, copying the program to the instruction RAM, and then specifying exception vectors to the instruction RAM.

7.1 Selecting the Boot Mode

One of four boot modes, external memory boot, serial flash ROM boot, external MCU boot, and instruction RAM boot can be selected.

Table 7.1 Selecting the Boot Mode

BOOT1	BOOT0	Boot Mode	Boot Area
0	0	External memory boot	Memory connected to CSZ0 area of external memory interface
0	1	External serial flash ROM boot	External serial flash ROM
1	0	External MCU boot	Instruction RAM
1	1	Instruction RAM boot (for debugging only)	Instruction RAM

(1) External memory boot

The CPU is booted from the external memory connected to the CSZ0 area of the external memory interface.
However, in the case of the 17 mm square package, the initial state of the address pins A27–A21 (external pins: RP27–RP21) are input ports with pull-up resistors, so High level propagates to the address bus of the connected external memory device. Therefore, note that the boot code must be placed in the upper space of the memory device address when booting from the external memory with a capacity of 1 MBytes or more in the 17 mm square package.

(2) External serial flash ROM boot

The CPU is booted from the serial flash ROM.

(3) External MCU boot

The program is downloaded to the instruction RAM via the external MCU interface and then the CPU in R-IN32M4-CL3 is booted from the instruction RAM. After the program is downloaded to the instruction RAM, fetching the program from the instruction RAM starts after the CPU reset is released by using the CPURESET register.

(4) Instruction RAM boot (for debugging only)

This mode is for directly downloading a program from a debugger to the instruction RAM to run during software development.

7.2 Initializing the Internal RAM

R-IN32M4-CL3 incorporates large-capacity instruction RAM, data RAM, buffer RAM, and network RAM. When the power-on reset input signal (PONRZ) is de-asserted, all the bits in these RAM areas are initialized to 0 by hardware within the internal circuit reset period. This significantly reduces the time required by the program to initialize the internal RAM.

7.3 Memory Map in Each Boot Mode

In R-IN32M4-CL3, the memory map from addresses 0000_0000H to 000F_FFFFH differs depending on the selected boot mode.

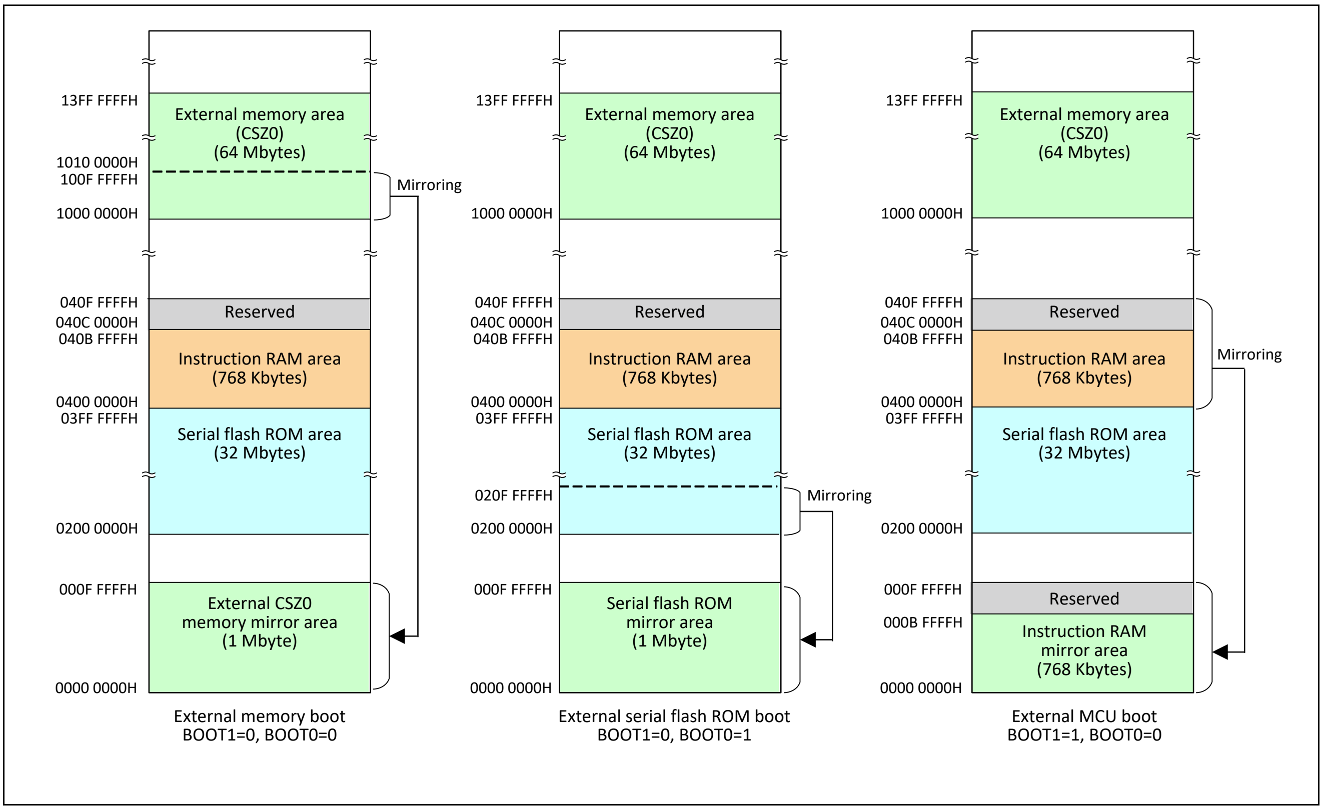


Figure 7.1 Memory Map in Each Boot Mode

7.4 Booting Sequence

The following describes the procedures up to specifying exception vectors to the instruction RAM.

7.4.1 When Booting from the External Memory

(1) Fetching the program from the external CSZ0 memory mirror area after release from the reset state

When external memory boot mode is selected by using the BOOT0 and BOOT1 pins, the lower 1 MB space in the external CSZ0 memory area is allocated to the area starting from address 0000 0000H, as a mirror area. After release from the reset state, the CPU is booted by the program allocated to address 0000 0000H. After the CPU is booted, make the settings for the memory controller registers, which affect the external bus access performance.

(2) Transferring the program code to the internal instruction RAM

Transfer the program code to the internal instruction RAM by using program processing or a DMA transfer.

(3) Masking interrupts

Mask all interrupt operations before switch vector addresses. Software exceptions and exception traps must also not be executed.

(4) Switching the vector address

Specify the instruction RAM area (0400 0000H) in the VTOR register, and then unmask the interrupts.

(5) Branching to main routine (regular operation)

Fetching the program from the instruction RAM starts.

7.4.2 When Booting from the External Serial Flash ROM

(1) Fetching the program from the serial flash ROM mirror area after release from the reset state

When external serial flash ROM boot mode is selected by using the BOOT0 and BOOT1 pins, the lower 1 MB space in the external serial flash ROM area is allocated to the area starting from address 0000 0000H, as a mirror area. After release from the reset state, the CPU is booted by the program allocated to address 0000 0000H. After the CPU is booted, make the settings for the memory controller registers, which affect the external bus access performance.

(2) Transferring the program code to the internal instruction RAM

Transfer the program code to the internal instruction RAM by using program processing or a DMA transfer.

(3) Masking interrupts

Mask all interrupt operations before switching the vector address. Software exceptions and exception traps must also not be executed.

(4) Switching the vector address

Specify the instruction RAM area (0400 0000H) in the VTOR register, and then unmask the interrupts.

(5) Branching to main routine (regular operation)

Fetching the program from the instruction RAM starts.

7.4.3 When Downloading the Program from the External MCU and Booting the CPU

When external MCU boot mode is selected by using the BOOT0 and BOOT1 pins, the instruction RAM area is allocated to the area starting from address 0000 0000H, as a mirror area. Even after releasing R-IN32M4-CL3 from the reset state, the CPU maintains the reset state. Release the CPU from the reset state by using the CPURESET register after the program has been downloaded to the internal instruction RAM.

(1) Transferring the program code to the internal instruction RAM after release from the reset state

Transfer the program code from the external host microcontroller connected to R-IN32M4-CL3 to the internal instruction RAM.

(2) Releasing the CPU from the reset state

After the program code is downloaded to the internal instruction RAM, write 0001H to the CPURESET register to release the CPU from the reset state.

8. Exception Handling

R-IN32M4-CL3 uses the interrupt controller of Cortex-M4.

For Cortex-M4 exception handling operations, visit the following Arm Limited website.

http://infocenter.arm.com/help/topic/com.arm.doc.set.cortexm/index.html

8.1 Exception List

Exception numbers 1–15 are the system exceptions for the Cortex-M4 CPU. Interrupts from the R-IN32M4-CL3 internal hardware and external pins are assigned to exception number 16 and higher numbers.

Exception Number	Exception Type	Priority	Description
1	Reset	-3 (highest)	• Input on the reset pins (RESETZ, PONRZ, and HOTRESETZ) • Reset by the window watchdog timer A • Setting the SYSRESETREQ bit in NVIC of the Cortex-M4 CPU to 1 • Reset by the SYSRESET register
2	NMI	-2	• Input on the NMI pin • Generation of NMI by the window watchdog timer A
3	Hard fault	-1	Used to promote the exception faults for all classes that cannot be handled by other exceptions
4	Memory management fault	Programmable	Exception from the MPU
5	Bus fault	Programmable	Bus error in access through the bus to the area outside the scope of management by the MPU
6	Usage fault	Programmable	Error in instruction execution, including the execution of an undefined instruction
7–10	Reserved	—	—
11	SVCall	Programmable	System service call by an SVC instruction
12	Debug monitor	Programmable	Debug monitor
13	Reserved	—	—
14	PendSV	Programmable	Request for system service that can be kept pending
15	SysTick	Programmable	Notification from the system timer
16–	R-IN32M4-CL3 specific interrupt	Programmable	Interrupt from the R-IN32M4-CL3 internal hardware or external pins

8.2 Interrupt List

The following table lists exceptions (interrupts) with exception numbers 16 and higher assigned to the NVIC of the Cortex-M4 CPU.

In R-IN32M4-CL3, interrupts from the internal hardware and external pins are connected not only to the NVIC of the Cortex-M4 but also to the internal hardware real-time OS (HW-RTOS), trigger for starting the internal DMA controllers (common to both the general DMAC and real-time port DMAC), and timers.

R-IN32M4-CL3 supports the following interrupts.

Table 8.1 Interrupt List

(1/6)

Exception Number	Name	Cause	Connected to				
			NVIC	HW-RTOS	DMAC	Real-time Port	Timer TAUJ2 /TAUD
16	INTTAUJ2I0	TAUJ2 channel 0 interrupt	○	○	○	○	○
17	INTTAUJ2I1	TAUJ2 channel 1 interrupt	○	○	○	○	○
18	INTTAUJ2I2	TAUJ2 channel 2 interrupt	○	○	○	○	○
19	INTTAUJ2I3	TAUJ2 channel 3 interrupt	○	○	○	○	○
20	INTUAJ0TIT	UARTJ0 send interrupt	○	○	○	○	○
21	INTUAJ0TIR	UARTJ0 receive interrupt	○	○	○	○	○
22	INTUAJ1TIT	UARTJ1 send interrupt	○	○	○	○	○
23	INTUAJ1TIR	UARTJ1 receive interrupt	○	○	○	○	○
24	INTCSIH0IC	CSIH0 communication status interrupt	○	○	○	○	○
25	INTCSIH0IR	CSIH0 receive status interrupt	○	○	○	○	○
26	INTCSIH0IJC	CSIH0 job completion interrupt	○	○	○	○	○
27	INTCSIH1IC	CSIH1 communication status interrupt	○	○	○	○	○
28	INTCSIH1IR	CSIH1 receive status interrupt	○	○	○	○	○
29	INTCSIH1IJC	CSIH1 job completion interrupt	○	○	○	○	○
30	INTIICB0TIA	IICB0 data send/receive interrupt	○	○	○	○	○
31	INTIICB1TIA	IICB1 data send/receive interrupt	○	○	○	○	○
32	INTFCN0REC	FCN0 receive completion interrupt	○	○	○	○	○
33	INTFCN0TRX	FCN0 send completion interrupt	○	○	○	○	○
34	INTFCN0WUP	FCN0 sleep and wakeup/send suspension interrupt	○	○	○	○	○
35	INTFCN1REC	FCN1 receive completion interrupt	○	○	○	○	○
36	INTFCN1TRX	FCN1 send completion interrupt	○	○	○	○	○
37	INTFCN1WUP	FCN1 sleep and wakeup/send suspension interrupt	○	○	○	○	○
38	INTDMA00	General-purpose DMAC channel 0 transfer completion interrupt	○	○	○	○	○
39	INTDMA01	General-purpose DMAC channel 1 transfer completion interrupt	○	○	○	○	○

Remarks. **○…Connectable** **—… Not used**

(2/6)

Exception Number	Name	Cause	Connected to				
			NVIC	HW-RTOS	DMAC	Real-time Port	Timer TAUJ2 /TAUD
40	INTDMA02	General-purpose DMAC channel 2 transfer completion interrupt	○	○	○	○	○
41	INTDMA03	General-purpose DMAC channel 3 transfer completion interrupt	○	○	○	○	○
42	INTRTDMA	Real-time port DMAC transfer completion interrupt	○	○	○	○	○
43	INTTAUDI0	TAUD channel 0 interrupt	○	○	○	○	○
44	INTTAUDI1	TAUD channel 1 interrupt	○	○	○	○	○
45	INTTAUDI2	TAUD channel 2 interrupt	○	○	○	○	○
46	INTTAUDI3	TAUD channel 3 interrupt	○	○	○	○	○
47	INTTAUDI4	TAUD channel 4 interrupt	○	○	○	○	○
48	INTBUFDMA	Inter-Buffer DMA transfer completion interrupt	○	○	○	○	○
49	INTETHPHY0	Gigabit Ethernet PHY Port0 interrupt	○	○	○	○	○
50	INTETHPHY1	Gigabit Ethernet PHY Port1 interrupt	○	○	○	○	○
51	INTETHMIICMP	Ethernet MII management access completion interrupt	○	○	○	○	○
52	INTETHPAUSECMP	Ethernet pause packet send completion interrupt	○	○	○	○	○
53	INTETHTXCMP	Ethernet send completion interrupt	○	○	○	○	○
54	INTETHSW	Ethernet SWITCH interrupt	○	○	○	○	○
55	INTETHSWDLR	Ethernet SWITCH DLR interrupt	○	○	○	○	○
56	INTETHSWSYNC	Ethernet SWITCH SYNC interrupt	○	○	○	○	○
57	INTETHRXFIFO	RX FIFO overflow interrupt	○	○	—	—	—
58	INTETHTXFIFO	TX FIFO underflow interrupt	○	○	—	—	—
59	INTETHRXDMA	Ethernet MAC DMA receive completion interrupt	○	○	○	○	○
60	INTETHTXDMA	Ethernet MAC DMA send completion interrupt	○	○	○	○	○
61	INTMACDMARXFRM	Receive frame successful interrupt	○	○	○	○	○
62	—	Reserved	—	—	—	—	—
63	INTPZ0	INTPZ0 input	○	○	○	○	○
64	INTPZ1	INTPZ1 input	○	○	○	○	○
65	INTPZ2	INTPZ2 input	○	○	○	○	○
66	INTPZ3	INTPZ3 input	○	○	○	○	○
67	INTPZ4	INTPZ4 input	○	○	○	○	○
68	INTPZ5	INTPZ5 input	○	○	○	○	○
69	INTPZ6	INTPZ6 input	○	○	○	○	○
70	INTPZ7	INTPZ7 input	○	○	○	○	○

Remarks. **○…Connectable** **—… Not used**

(3/6)

Exception Number	Name	Cause	Connected to NVIC	HW-RTOS	DMAC	Real-time Port	Timer TAUJ2 /TAUD
71	INTPZ8	INTPZ8 input	○	○	○	○	○
72	INTPZ9	INTPZ9 input	○	○	○	○	○
73	INTPZ10	INTPZ10 input	○	○	○	○	○
74	INTPZ11	INTPZ11 input/TAUD channel 5 interrupt*	○	○	○	○	○
75	INTPZ12	INTPZ12 input/TAUD channel 6 interrupt*	○	○	○	○	○
76	INTPZ13	INTPZ13 input/TAUD channel 7 interrupt*	○	○	○	○	○
77	INTPZ14	INTPZ14 input/TAUD channel 8 interrupt*	○	○	○	○	○
78	INTPZ15	INTPZ15 input/TAUD channel 9 interrupt*	○	○	○	○	○
79	INTPZ16	INTPZ16 input/TAUD channel 10 interrupt*	○	○	○	○	○
80	INTPZ17	INTPZ17 input/TAUD channel 11 interrupt*	○	○	○	○	○
81	INTPZ18	INTPZ18 input/TAUD channel 12 interrupt*	○	○	○	○	○
82	INTPZ19	INTPZ19 input/TAUD channel 13 interrupt*	○	○	○	○	○
83	INTPZ20	INTPZ20 input/TAUD channel 14 interrupt*	○	○	○	○	○
84	INTPZ21	INTPZ21 input/TAUD channel 15 interrupt*	○	○	○	○	○
85	INTPZ22	INTPZ22 input	○	○	○	○	○
86	INTPZ23	INTPZ23 input	○	○	○	○	○
87	INTPZ24	INTPZ24 input	○	○	○	○	○
88	INTPZ25	INTPZ25 input	○	○	○	○	○
89	INTPZ26	INTPZ26 input	○	○	○	○	○
90	INTPZ27	INTPZ27 input	○	○	○	○	○
91	INTPZ28	INTPZ28 input	○	○	○	○	○
92	INTHWRTOS	HW-RTOS interrupt	○	—	—	—	—
93	INTBRAMERR	Buffer RAM area access error interrupt	○	○	—	—	—
94	INTIICB0TIS	IICB0 status interrupt	○	○	—	—	—
95	INTIICB1TIS	IICB1 status interrupt	○	○	—	—	—

Remarks. **○...Connectable** **—... Not used**

Note. **INTPZ/TAUD interrupts are selected using the INTSEL register.**
For details, refer to Section 28.18 "INTPZ/Timer Interrupt Select Register (INTSEL)".

(4/6)

Exception Number	Name	Cause	Connected to				
			NVIC	HW-RTOS	DMAC	Real-time Port	Timer TAUJ2 /TAUD
96	INTWDTA	WDT alarm interrupt (including the 75% of timeout interrupt)	○	○	—	—	—
97	INTSFLASH	Serial flash ROM controller error interrupt	○	○	—	—	—
98	INTUAJ0TIS	UARTJ0 status interrupt	○	○	—	—	—
99	INTUAJ1TIS	UARTJ1 status interrupt	○	○	—	—	—
100	INTCSIH0IRE	CSIH0 communication error interrupt	○	○	—	—	—
101	INTCSIH1IRE	CSIH1 communication error interrupt	○	○	—	—	—
102	INTFCN0ERR	FCN0 error detection interrupt	○	○	—	—	—
103	INTFCN1ERR	FCN1 error detection interrupt	○	○	—	—	—
104	INTDERR0	General-purpose DMAC error response interrupt	○	○	—	—	—
105	INTDERR1	Real-time port DMAC error response interrupt	○	○	—	—	—
106	INTETHTXFIFOERR	TX-FIFO error interrupt	○	○	—	—	—
107	INTETHRXERR	Ethernet receive frame error interrupt	○	○	—	—	—
108	INTETHRXDERR	MAC DMA receive error interrupt	○	○	—	—	—
109	INTETHTXDERR	MAC DMA send error interrupt	○	○	—	—	—
110	INTBUFDMAERR	Internal buffer DMA error interrupt	○	○	—	—	—
111	INTLED0PHY0	Gigabit Ethernet PHY LED0_PHY0 input interrupt	○	○	○	○	○
112	INTLED0PHY1	Gigabit Ethernet PHY LED0_PHY1 input interrupt	○	○	○	○	○
113	—	Reserved	—	—	—	—	—
114	—	Reserved	—	—	—	—	—
115	IRAMECCSEC	Internal instruction RAM 1-bit ECC error correction interrupt	○	—	—	—	—
116	DRAMECCSEC	Data RAM 1-bit ECC error correction interrupt	○	—	—	—	—
117	BRAMECCSEC	Buffer RAM 1-bit ECC error correction interrupt	○	—	—	—	—
118	IRAMECCDED	Internal instruction RAM 2-bit ECC error detection interrupt	○	—	—	—	—
119	DRAMECCDED	Data RAM 2-bit ECC error detection interrupt	○	—	—	—	—
120	BRAMECCDED	Buffer RAM 2-bit ECC error detection interrupt	○	—	—	—	—
121	—	Reserved	—	—	—	—	—
122	—	Reserved	—	—	—	—	—

Remarks. **○…Connectable** **—… Not used**

(5/6)

Exception Number	Name	Cause	Connected to				
			NVIC	HW-RTOS	DMAC	Real-time Port	Timer TAUJ2 /TAUD
123	INTCCINMIZ	CC-Link IE Field NMIZ interrupt	○	○	○	○	○
124	INTCCIWDTZ	CC-Link IE Field WDTZ interrupt	○	○	○	○	○
125	INTCCIINTZ	CC-Link IE Field INTZ interrupt	○	○	○	○	○
126	INTCCICLKLOSSZ	CC-Link IE Field CLKLOSSZ interrupt	○	○	○	○	○
127	—	Reserved	—	—	—	—	—
128	—	Reserved	—	—	—	—	—
129	—	Reserved	—	—	—	—	—
130	—	Reserved	—	—	—	—	—
131	—	Reserved	—	—	—	—	—
132	—	Reserved	—	—	—	—	—
133	—	Reserved	—	—	—	—	—
134	—	Reserved	—	—	—	—	—
135	—	Reserved	—	—	—	—	—
136	—	Reserved	—	—	—	—	—
137	INTGBEPHYFLF	Gigabit Ethernet PHY FASTLINK_FAIL interrupt	○	○	—	—	—
138	INTLED1PHY0	Gigabit Ethernet PHY LED1_PHY0 input interrupt	○	○	○	○	○
139	INTLED1PHY1	Gigabit Ethernet PHY LED1_PHY1 input interrupt	○	○	○	○	○
140	INTLED2PHY0	Gigabit Ethernet PHY LED2_PHY0 input interrupt	○	○	—	—	—
141	INTLED2PHY1	Gigabit Ethernet PHY LED2_PHY1 input interrupt	○	○	—	—	—
142	INTFPU	FPU interrupt	○	○	—	—	—
143	INTPZ29	INTPZ29 input	○	○	○	○	○
144	—	Reserved	—	—	—	—	—
145	NRAMECCSEC	Network RAM 1-bit ECC error detection interrupt	○	—	—	—	—
146	NRAMECCDED	Network RAM 2-bit ECC error detection interrupt	○	—	—	—	—
147	—	Reserved	—	—	—	—	—

Remarks. **○...Connectable** **—... Not used**

(6/6)

Exception Number	Name	Cause	Connected to				
			NVIC	HW-RTOS	DMAC	Real-time Port	Timer TAUJ2 /TAUD
148	INTHSNGNREGBE	Bus error response interrupt (HSNGNREG)	○	—	○	○	○
149	INTHSNGNBE	Bus error response interrupt (HSNGN)	○	—	○	○	○
150	INTXSNGNBE	Bus error response interrupt (XSNGN)	○	—	○	○	○
151	—	Reserved	—	—	—	—	—
152	INTMCINTWDTERR	External CPU WDT error interrupt	○	—	○	○	○
153	—	Reserved	—	—	—	—	—
154	—	Reserved	—	—	—	—	—
155	—	Reserved	—	—	—	—	—
156	—	Reserved	—	—	—	—	—
157	—	Reserved	—	—	—	—	—
158	—	Reserved	—	—	—	—	—
159	—	Reserved	—	—	—	—	—
160	—	Reserved	—	—	—	—	—
161	—	Reserved	—	—	—	—	—
162	—	Reserved	—	—	—	—	—
163	—	Reserved	—	—	—	—	—
164	—	Reserved	—	—	—	—	—
165	—	Reserved	—	—	—	—	—
166	—	Reserved	—	—	—	—	—
167	—	Reserved	—	—	—	—	—
168	INTROK	Built-in regulator interrupt	○	—	○	○	○
169	—	Reserved	—	—	—	—	—
…	…	…	…	…	…	…	…
255	—	Reserved	—	—	—	—	—

Remarks. ○…Connectable —… Not used

9. Debugging

The Cortex-M4 of an R-IN32M4-CL3 has a range of on-chip debugging features. These features include downloading, running, and breaking programs, as well as a trace feature to output program execution logs.

An R-IN32M4-CL3 provides JTAG and SWD interfaces that can be used as general debugging interfaces, as well as trace port and SWV interfaces for tracing. For details of the Cortex-M4 of an R-IN32M4-CL3, see Section 5 “CPU and Internal RAMs”.

9.1 JTAG Interface

The JTAG interface handles transfer to and from the host computer via the ICE by using five signals (TCK, TMS, TDO, TDI, and TRSTZ).

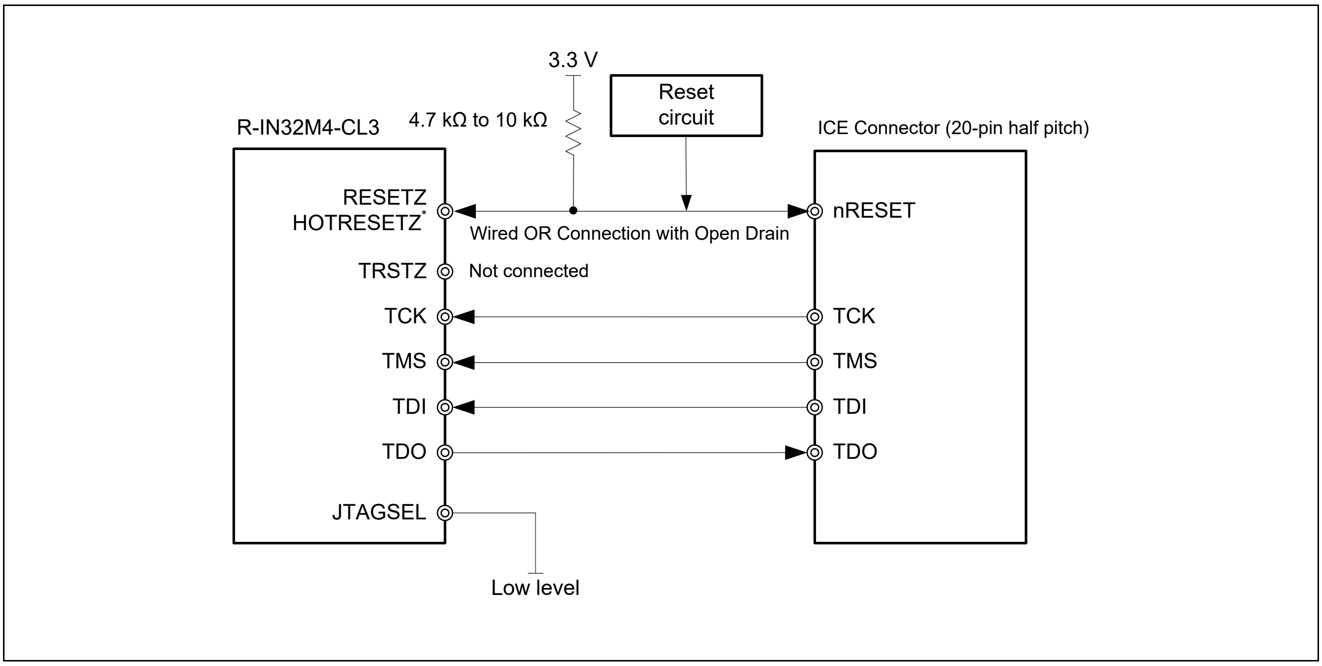


Figure 9.1 JTAG Interface Connection Example (20-Pin Half Pitch without Trace)

Caution: The input of the nRESET signal to HOTRESETZ is not required if it is connected to RESETZ. RESETZ resets the entire LSI chip, but only HOTRESETZ does not reset the internal PLL. Connect the JTAG interface in a way that suits the application. Also, do not connect the nRESET signal with the PONRZ signal.

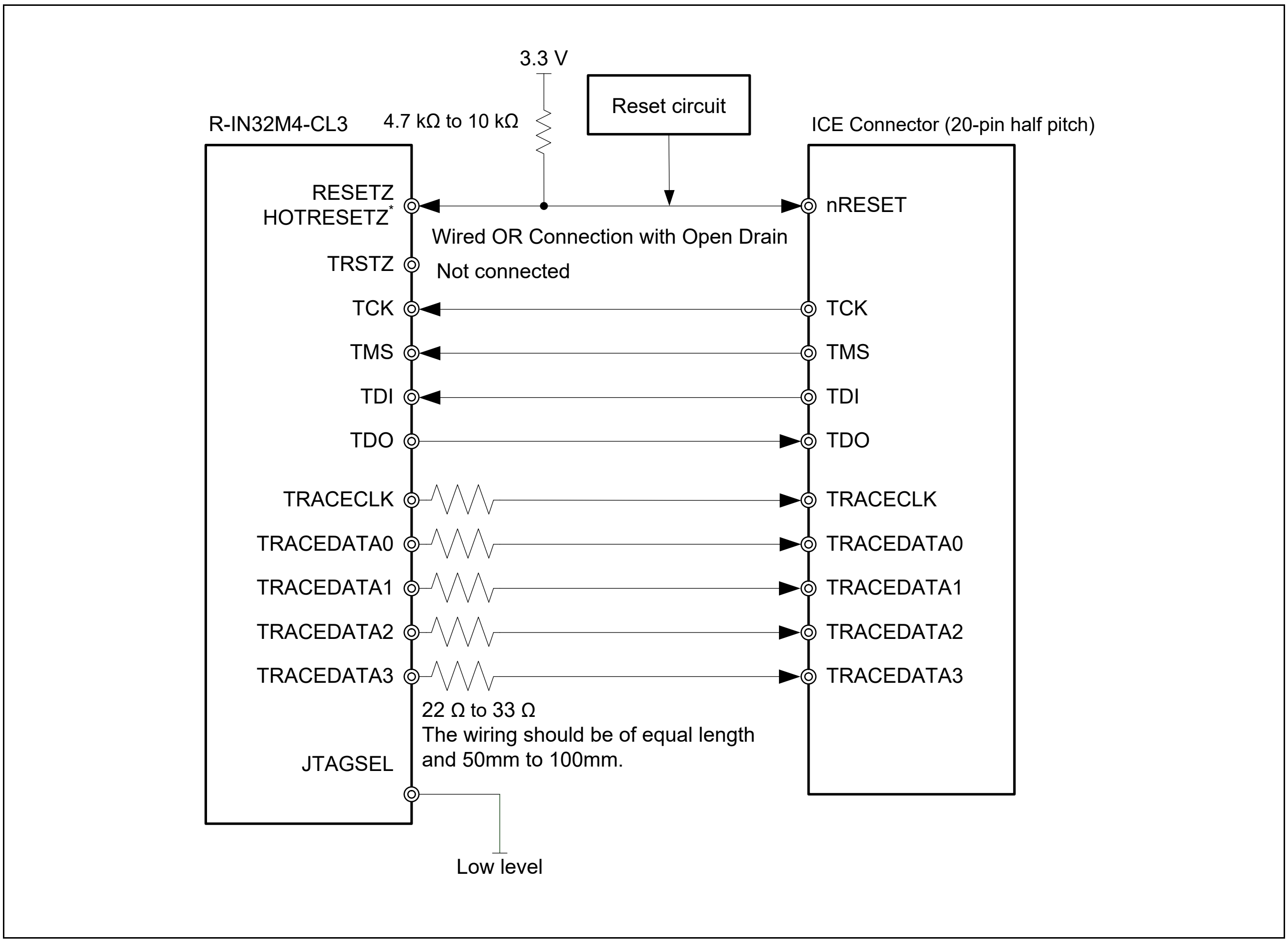


Figure 9.2 JTAG Interface Connection Example (20-Pin Half Pitch with Trace)

Caution: The input of the nRESET signal to HOTRESETZ is not required if it is connected to RESETZ.
RESETZ resets the entire LSI chip, but only HOTRESETZ does not reset the internal PLL.
Connect the JTAG interface in a way that suits the application.
Also, do not connect the nRESET signal with the PONRZ signal.

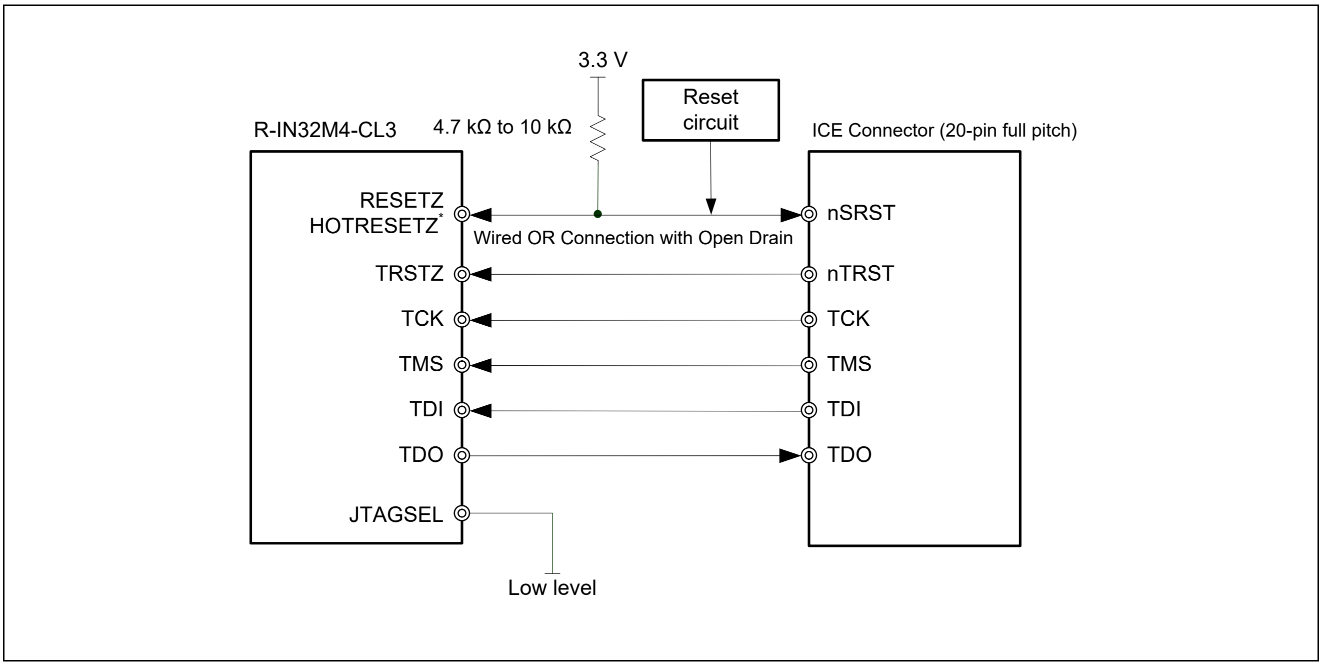


Figure 9.3 JTAG Interface Connection Example (20-Pin Full Pitch)

Caution: The input of the nRESET signal to HOTRESETZ is not required if it is connected to RESETZ. RESETZ resets the entire LSI chip, but only HOTRESETZ does not reset the internal PLL. Connect the JTAG interface in a way that suits the application.
Also, do not connect the nRESET signal with the PONRZ signal.

9.2 SWD Interface

The SWD (Serial Wire Debug) interface handles transfer to and from the host computer via the ICE by using two signals (SWCLK (TCK) and SWDIO (TMS)).

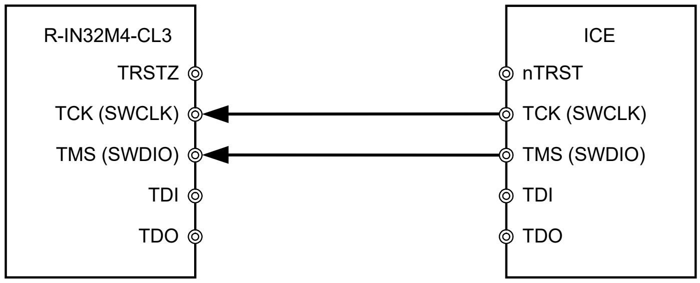


Figure 9.4 SWD Interface Connection Example

9.3 Trace Port Interface

The trace port interface outputs trace information by using five signals (TRACECLK and TRACEDATA[3:0]). The trace port interface outputs information obtained by the ETM trace feature concerning the branch instructions executed in the program. This information is supplemented by the debugger, making it possible to check the branch source and destinations. For details about trace information, see the user's manual of the ICE you are using.

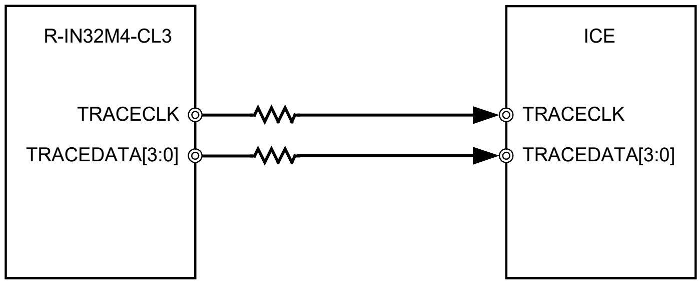


Figure 9.5 Trace Port Interface Connection Example

9.4 SWV Interface

The SWV (Serial Wire Viewer) interface outputs trace information by using one signal (TDO (SWV) or TRACEDATA0 (SWV)). Note that TDO (SWV) cannot be used when using the JTAG interface. SWV tracing involves sampling the specified data at a specific sampling interval. For details about trace information, see the user's manual of the ICE you are using.

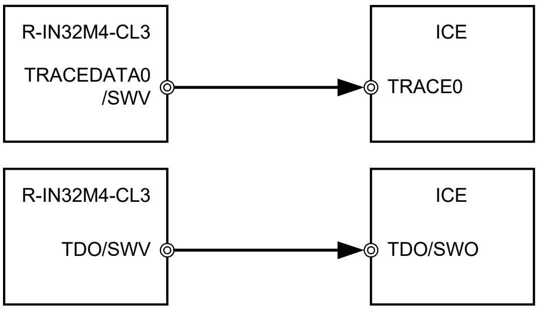


Figure 9.6 SWV Interface Connection Example

10. Hardware Real-Time OS (HW-RTOS)

The hardware Real-Time Operating System (HW-RTOS) supports 30 types of system calls, such as events, semaphores, and mailboxes.

10.1 Outline of Features

- µITRON-like system calls
  - 30 system calls for elements such as events, semaphores, and mailboxes
- Task Scheduler (Ver. 4.2)
  - Hardware ISR: 32 routines selectable from 128 interrupt sources
  - Number of context elements: 64
  - Number of semaphore identifiers: 128
  - Number of event identifiers: 64
  - Number of mailbox identifiers: 64
  - Number of mailbox elements: 192
  - Number of context priority levels: 16
- Hardware function manager

Remark: The hardware real-time OS can be controlled by using the µITRON system calls provided by the sample driver. For how to use the driver, see the R-IN32M4-CL3 Programming Manual: OS.

10.2 Semaphores

The semaphores handled by HW-RTOS in an R-IN32M4-CL3 are 5-bit counting semaphores. For semaphores, whether to use wait queues for each context priority or use wait queues regardless of the context priority can be programmed for individual semaphore identifiers. 125 semaphore identifiers can be handled.

10.3 Events

The event flags handled by HW-RTOS in an R-IN32M4-CL3 are 16-bit event flags. For events, whether to use wait queues for each context priority or use wait queues regardless of the context priority can be programmed for individual event identifiers. 64 event identifiers can be handled.

10.4 Mailboxes

The mailboxes handled by HW-RTOS in an R-IN32M4-CL3 are used for transmitting and receiving 32-bit messages. There are eight message priority levels. For mailboxes, whether to use wait queues for each context priority or use wait queues regardless of the context priority can be specified for individual mailbox identifiers. 64 mailbox identifiers can be handled.

10.5 Operation of HW-RTOS

Handshaking between the CPU and HW-RTOS is performed by using OS interrupts and commands. The relationship between the CPU and the task scheduler is as follows:

- The CPU executes software based on the context scheduled by the task scheduler. Therefore, the CPU does not execute any software other than the target context.
- When an interrupt occurs or a system call is issued, a conventional typical OS performs the following:
  - saves the contents of CPU registers such as general-purpose registers, the program counter, and flag registers to the context management area, and
  - loads the register data for the context to run to the CPU registers, dispatches the task, and then executes software based on the program counter value.

When an interrupt occurs or a system call is issued, HW-RTOS in an R-IN32M4-CL3 performs the following:

- saves the contents of CPU registers such as general-purpose registers, the program counter, and flag registers to the stack area allocated in the data RAM during the OS interrupt exception routine,
- performs processing of system calls and other processing,
- selects the context to be dispatched, and
- loads the values of the CPU register corresponding to the context from the HW-RTOS context control memory, writes them to the CPU registers, and then returns from the OS interrupt exception routine.

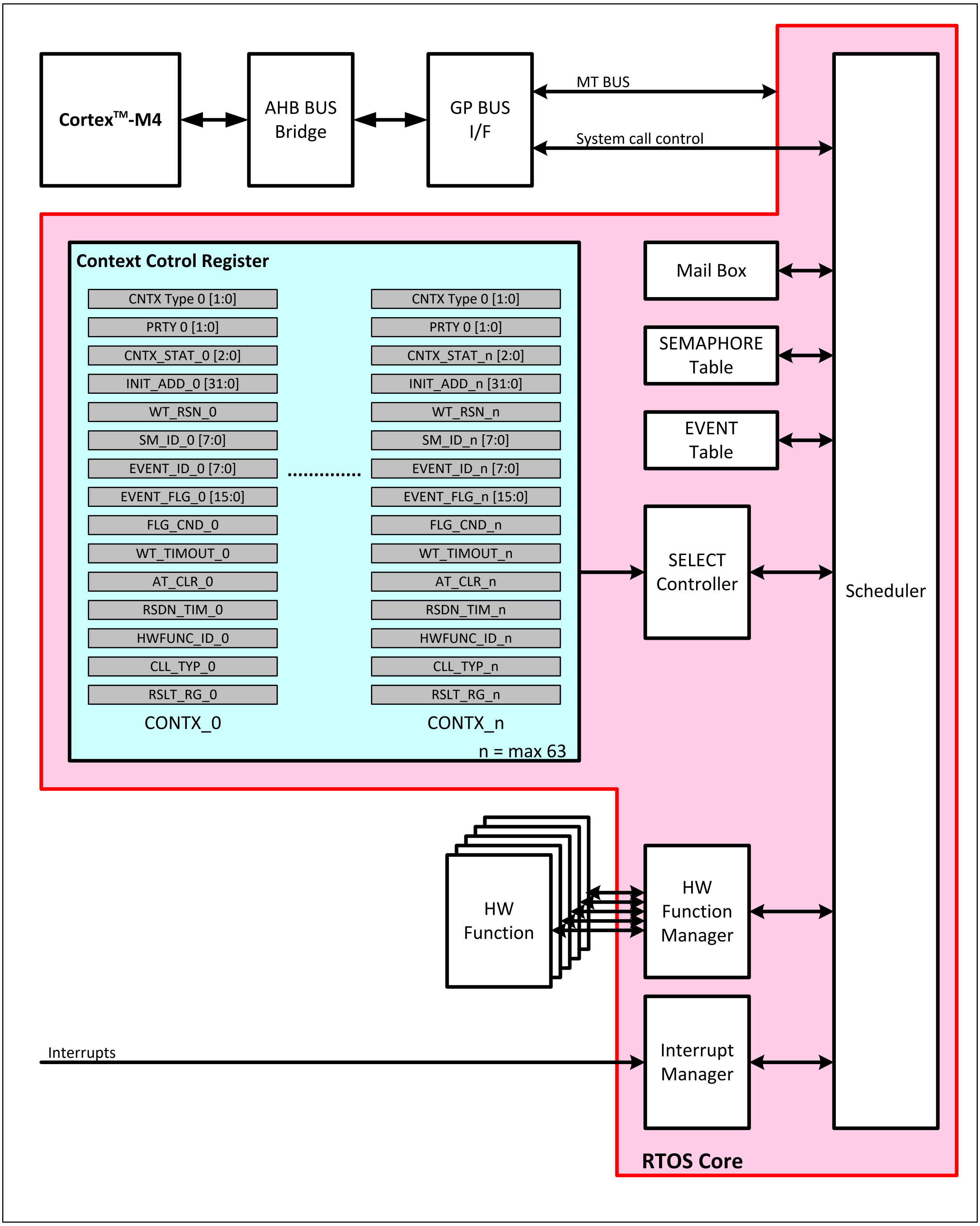


Figure 10.1 Overall Structure of Hardware Real-Time OS (HW-RTOS)

11. Gigabit Ethernet PHY

R-IN32M4-CL3 incorporates a Gigabit Ethernet PHY.
This section explains the peripheral circuit of Gigabit Ethernet PHY and the regulator for the Gigabit Ethernet PHY.
For details on the Gigabit Ethernet PHY, refer to the “R-IN32M4-CL3 User's Manual: Gigabit Ethernet PHY edition”.

11.1 Features

- IEEE 802.3 (10BASE-T, 100BASE-TX, 1000BASE-T) compliant
- Number of ports: 2
- Power saving (ActiPHY™)
- Autonegotiation
- Autocrossover

11.2 PHY Peripheral Circuit Control Register

11.2.1 List of PHY Peripheral Circuit Control Registers

(1) PHY Control Register

Register Name	Abbreviation	Address
PHY Address Setting Register	PHYADD	BASE + 2020H

(2) Built-in Regulator Control Registers

Register Name	Abbreviation	Address
Built-in Regulator Monitoring Register 0	DCDCMON0	BASE + 2030H
Built-in Regulator Monitoring Register 1	DCDCMON1	BASE + 2034H

11.2.2 PHY Address Setting Register (PHYADD)

This register controls the PHYADD1, PHYADD2, PHYADD3, and PHYADD4 pins of the built-in Gigabit Ethernet PHY.

- This register can be read and written in 32- or 16-bit units.

Caution 1. This register is writable only when a protection release sequence is performed using the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

2. When changing the value of this register, do so while the Gigabit Ethernet PHY is reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
PHYADD	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	PHYADD4	PHYADD3	PHYADD2	PHYADD1	BASE + 2020H Initial value Refer to the table below
R/W	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	R/W	R/W	R/W	R/W	

Bit	Bit Name	Description
3–0	PHYADD	Sets to 0 or 1 to the PHYADD1–PHYADD4 pins of the built-in Gigabit Ethernet PHY. The initial value depends on the package. Package: 23 mm Square PBGA — Initial value: 4-bit captured data from the external pins (PHYADD4, PHYADD3, PHYADD2, and PHYADD1) Package: 17 mm Square FBGA — Initial value: 4'b0000 In the 17 mm square package, the initial value is 4'b0000 since the PHYADD1–PHYADD4 pins do not exist as external pins and clamped to Low-level in the package.

Package	Initial value
23 mm Square PBGA	4-bit captured data from the external pins (PHYADD4, PHYADD3, PHYADD2, and PHYADD1)
17 mm Square FBGA	4’b0000

This register is initialized only by the reset signal from the external PONRZ pin.

11.2.3 Built-in Regulator Monitoring Register 0 (DCDCMON0)

This register indicates the state of the built-in regulator.

• This register can be read in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DCDCMON0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EN	BASE + 2030H Initial value Pin setting
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	

Bit	Bit Name	Description
0	EN	This bit is used to read the level of the REG_EN pin. 0: Built-in regulator is disabled. 1: Built-in regulator is enabled

11.2.4 Built-in Regulator Monitoring Register 1 (DCDCMON1)

This register is used to monitor the state of the built-in regulator.

Table 11.1 Built-in Regulator 2.5 V Output Voltage Signal Specifications (ROK)

ROK value	Description	2.5 V output voltage state			Unit
		MIN	TYP	MAX	
0	The 2.5 V output voltage is off or the output voltage has not reached 2.5 V.	—	—	2.5V × 93%	V
1	The 2.5 V output voltage is supplied normally.	2.5V × 84%	—	—	V

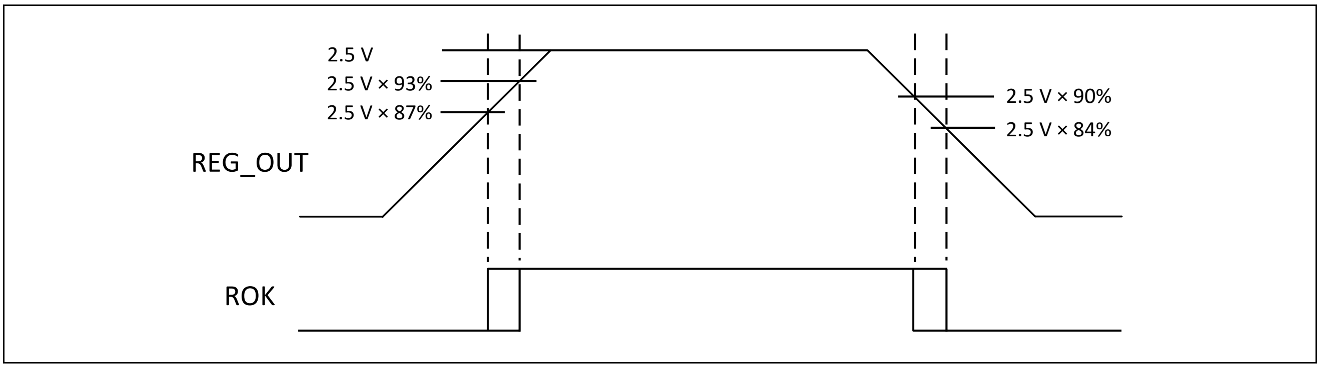


Figure 11.1 Built-in Regulator 2.5 V Output Voltage Signal ROK Operation Waveform

- This register can be read in 32- or 16-bit units.

Address: BASE + 2034H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCDCMON1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ROK
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R

Initial value

Bit	Bit Name	Description
0	ROK	This bit is used to monitor the state of the built-in regulator. 0: The 2.5 V output voltage is off, or the output voltage has not reached 2.5 V. 1: The 2.5 V output voltage is supplied normally. ROK is connected as INTROK of the interrupt signal.

11.3 Initialization Sequence

The Gigabit Ethernet PHY installed on the R-IN32M4-CL3 must be initialized after reset is released. In this initialization, a firmware provided by RENESAS can be used for register setting. After initialization is complete, select the management interface according to the application.

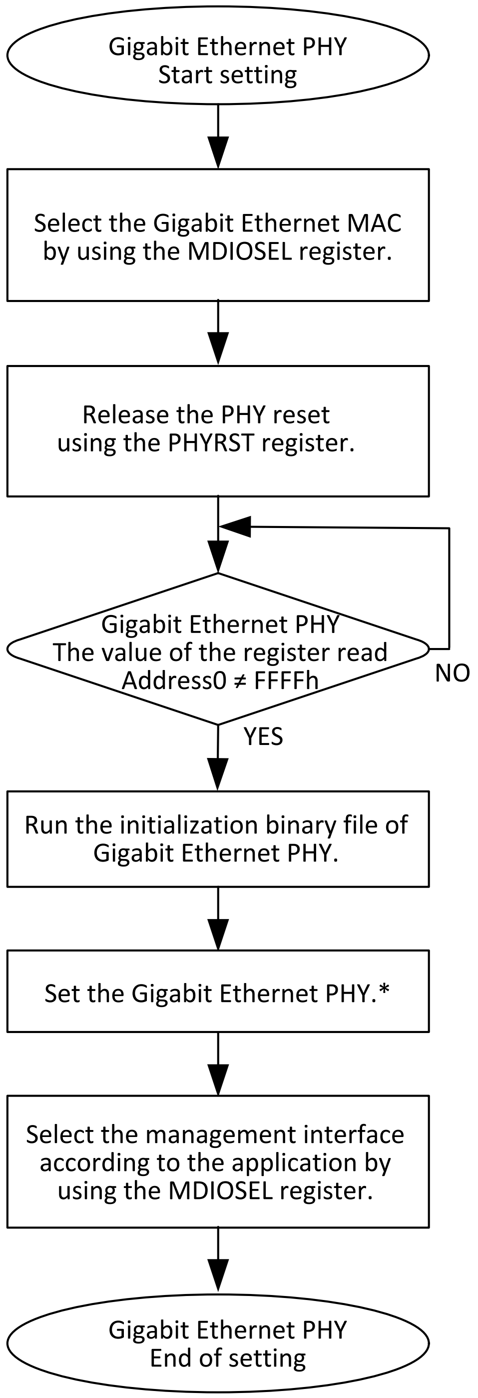


Figure 11.2 Gigabit Ethernet PHY Initialization Procedure

Note. For the PHY settings, refer to the "R-IN32M4-CL3 User's Manual: Gigabit Ethernet PHY edition" and firmware (Gigabit Ethernet PHY initial setting file) provided by Renesas Electronics.

12. Gigabit Ethernet MAC

This section describes the Ethernet MAC function with accelerator installed in the R-IN32M4-CL3.
For detail on the Ethernet Switch function, refer to Section 13, Ethernet Switch.

12.1 Overview

R-IN32M4-CL3 can also be used for a general Gigabit Ethernet interface as well as for an Ethernet interface compliant with the industrial Ethernet protocol (CC-Link IE Field).
Switching of the Ethernet interface and mode settings are controlled by using a register.

12.1.1 Ethernet Interface Architecture

The control registers for selecting the Ethernet interface and the architecture of the control target are shown below.
The control registers in the figure are described later.

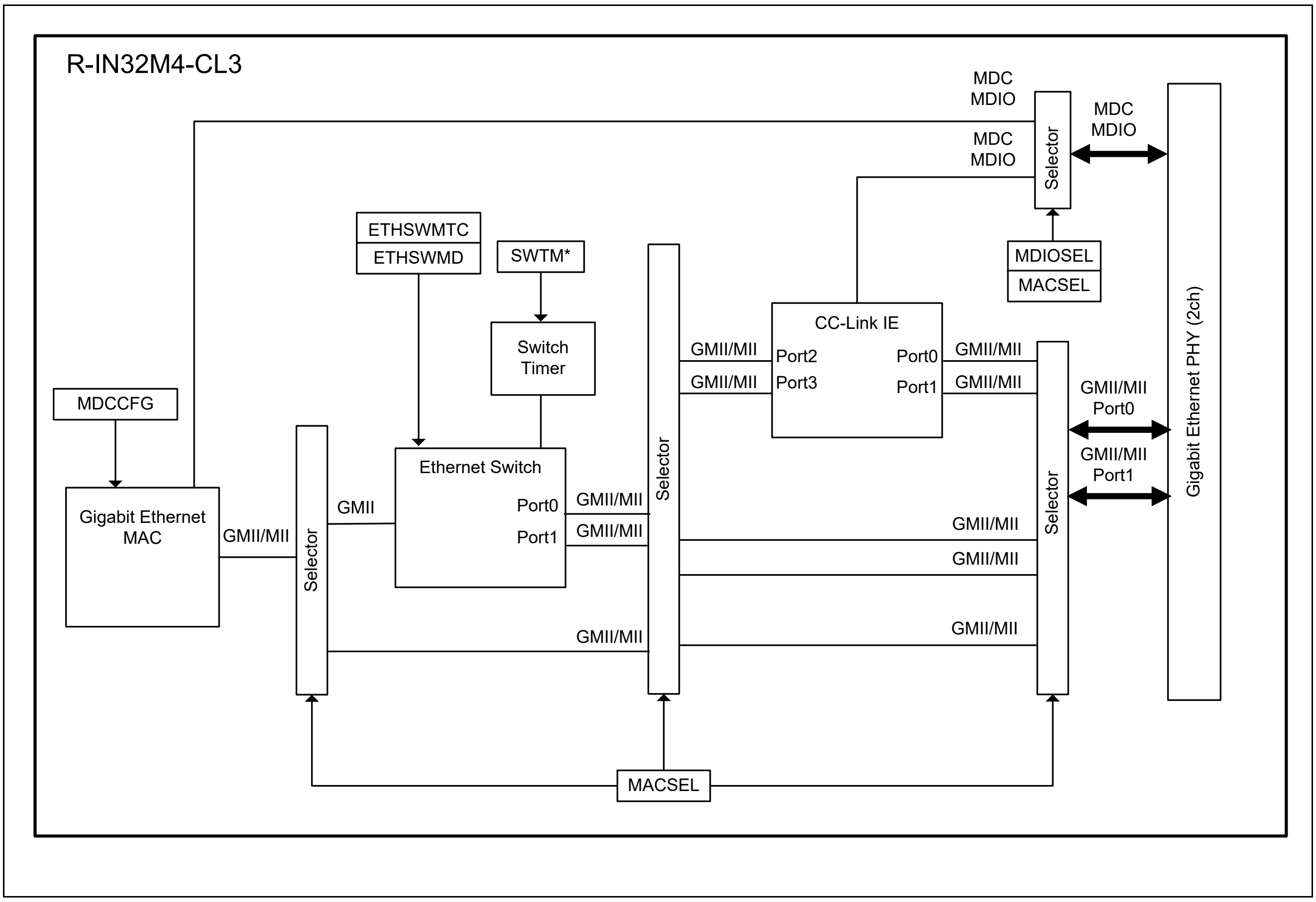


Figure 12.1 Ethernet Interface Peripheral Architecture

12.1.2 PHY Interface Selection

In R-IN32M4-CL3, the Ethernet interface and management interface are selectable by controlling the registers (MACSEL, MDIOSEL). Table 12.1 lists a selection of PHY interfaces.

Table 12.1 PHY Interface Selection

MACSEL	MDIOSEL	Ethernet Port0	Ethernet Port1	Management I/F
0000B	X*1	General-purpose Ethernet port 0 (with an Ethernet switch)	General-purpose Ethernet port 1 (with an Ethernet switch)	Gigabit Ethernet MAC
0001B	0B	CC-Link IE port 0	General-purpose Ethernet port 1 (with an Ethernet switch)	Gigabit Ethernet MAC
	1B			CC-Link IE
0010B	0B	General-purpose Ethernet port 0 (with an Ethernet switch)	CC-Link IE port 1	Gigabit Ethernet MAC
	1B			CC-Link IE
0011B	X*1	— (Not used)	General-purpose Ethernet port 1 (without an Ethernet switch)	Gigabit Ethernet MAC
0100B	X*2	CC-Link IE port 0	CC-Link IE port 1	CC-Link IE
0101B	0B	CC-Link IE port 0	General-purpose Ethernet port 1 (without an Ethernet switch)	Gigabit Ethernet MAC
	1B			CC-Link IE
0110B	0B	General-purpose Ethernet port 0 (without an Ethernet switch)	CC-Link IE port 1	Gigabit Ethernet MAC
	1B			CC-Link IE
0111B	0B	—	—	—
	1B			
1000B	0B	General-purpose Ethernet port 0 (with an Ethernet switch) CC-Link IE port 0 via CC-Link IE port 2	General-purpose Ethernet port 1 (with an Ethernet switch) CC-Link IE port 1 via CC-Link IE port 3	Gigabit Ethernet MAC
	1B			CC-Link IE
1001B	0B	CC-Link IE port 0	General-purpose Ethernet port 1 (without an Ethernet switch) CC-Link IE port 1 via CC-Link IE port 3	Gigabit Ethernet MAC
	1B			CC-Link IE
1010B – 1111B	0B	—	—	—
	1B			

Note 1. A don't care value. The Gigabit Ethernet MAC is selected as the management interface regardless of the setting of MDIOSEL.

2. A don't care value. The CC-Link IE Field is selected as the management interface regardless of the setting of MDIOSEL.

12.2 Features

The functions of the Ethernet interface of the R-IN32M4-CL3 (when the Ethernet switch is not in use) are given below.

- 1 Port (two-port switching)
- Support for EEE802.3
- 10BASE-T, 100BASE-TX, 1000BASE-T
- Full duplex and half duplex communications
- Automatic pause packet transmission
- Auto broadcast suspension in response to reception of a pause packet
- Support for MII/GMII interfaces

12.3 Control Registers

12.3.1 List of Registers

(1) Ethernet interface select registers

Register Name	Abbreviation	Address
MAC select register	MACSEL	BASE + 0600H
GMII/MII management I/F select register	MDIOSEL	BASE + 0E00H

(2) Ethernet interface mode register

Register Name	Abbreviation	Address
MDC clock select register	MDCCFG	BASE + 0604H

(3) Gigabit Ethernet MAC control registers

(1/2)

Register Name	Abbreviation	Address
MIIM register	GMAC_MIIM	4009 00A0H
TX ID register	GMAC_TXID	4009 000CH
TX result register	GMAC_TXRESULT	4009 0010H
Mode register	GMAC_MODE	4009 0020H
RX mode register	GMAC_RXMODE	4009 0024H
TX mode register	GMAC_TXMODE	4009 0028H
Reset register	GMAC_RESET	4009 0030H
Pause packet data register 1	GMAC_PAUSE1	4009 0080H
Pause packet data register 2	GMAC_PAUSE2	4009 0084H
Pause packet data register 3	GMAC_PAUSE3	4009 0088H
Pause packet data register 4	GMAC_PAUSE4	4009 008CH
Pause packet data register 5	GMAC_PAUSE5	4009 0090H
RX flow control register	GMAC_FLWCTL	4009 0098H
Pause packet register	GMAC_PAUSPKT	4009 009CH
MAC address register 0A	GMAC_ADR0A	4009 0100H
MAC address register 0B	GMAC_ADR0B	4009 0104H
MAC address register 1A	GMAC_ADR1A	4009 0108H
MAC address register 1B	GMAC_ADR1B	4009 010CH
MAC address register 2A	GMAC_ADR2A	4009 0110H
MAC address register 2B	GMAC_ADR2B	4009 0114H
MAC address register 3A	GMAC_ADR3A	4009 0118H
MAC address register 3B	GMAC_ADR3B	4009 011CH
MAC address register 4A	GMAC_ADR4A	4009 0120H
MAC address register 4B	GMAC_ADR4B	4009 0124H
MAC address register 5A	GMAC_ADR5A	4009 0128H
MAC address register 5B	GMAC_ADR5B	4009 012CH

(2/2)

Register Name	Abbreviation	Address
MAC address register 6A	GMAC_ADR6A	4009 0130H
MAC address register 6B	GMAC_ADR6B	4009 0134H
MAC address register 7A	GMAC_ADR7A	4009 0138H
MAC address register 7B	GMAC_ADR7B	4009 013CH
MAC address register 8A	GMAC_ADR8A	4009 0140H
MAC address register 8B	GMAC_ADR8B	4009 0144H
MAC address register 9A	GMAC_ADR9A	4009 0148H
MAC address register 9B	GMAC_ADR9B	4009 014CH
MAC address register 10A	GMAC_ADR10A	4009 0150H
MAC address register 10B	GMAC_ADR10B	4009 0154H
MAC address register 11A	GMAC_ADR11A	4009 0158H
MAC address register 11B	GMAC_ADR11B	4009 015CH
MAC address register 12A	GMAC_ADR12A	4009 0160H
MAC address register 12B	GMAC_ADR12B	4009 0164H
MAC address register 13A	GMAC_ADR13A	4009 0168H
MAC address register 13B	GMAC_ADR13B	4009 016CH
MAC address register 14A	GMAC_ADR14A	4009 0170H
MAC address register 14B	GMAC_ADR14B	4009 0174H
MAC address register 15A	GMAC_ADR15A	4009 0178H
MAC address register 15B	GMAC_ADR15B	4009 017CH
RX FIFO status register	GMAC_RXFIFO	4009 0200H
TX FIFO status register	GMAC_TXFIFO	4009 0204H
TCP/IPACC register	GMAC_ACC	4009 0208H
RX MAC enable register	GMAC_RXMAC_ENA	4009 0220H
LPI mode control register	GMAC_LPI_MODE	4009 0224H
LPI client timing control register	GMAC_LPI_TIMING	4009 0228H
Receive buffer information register	BUFID	4009 1100H

(4) Hardware function call registers

Register Name	Abbreviation	Address
Hardware function system call register	SYSC	4008 F000H
Hardware function argument register 4	R4	4008 F004H
Hardware function argument register 5	R5	4008 F008H
Hardware function argument register 6	R6	4008 F00CH
Hardware function argument register 7	R7	4008 F010H
Hardware function operating mode control register	CMD	4008 F014H
Hardware function return value register 0	R0	4008 F020H
Hardware function return value register 1	R1	4008 F024H
Hardware function type register	CNTX_TYPE0	4008 0000H
Hardware function state register	CNTX_STAT0	4008 0008H

12.3.2 Ethernet Interface Select Register

12.3.2.1 MAC Select Register (MACSEL)

This register is used to select the Ethernet interface. After setting this register, reset the PHY and then set the Ethernet MAC.

- This register can be read and written in 32- or 16-bit units.

Cautions 1. This register is writable only when a protection release sequence is performed using the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

2. When changing the value of this register, do so while the Gigabit Ethernet PHY is reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACSEL	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	MAC3	MAC2	MAC1	MAC0
R/W	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	R/W	R/W	R/W	R/W

Address: BASE + 0600H

Initial value: 0000 0008H

Bit	Bit Name	Description
3–0	MAC3–MAC0	Select the function of the MAC interface to be used.*

MAC3–MAC0	Ethernet Port 0	Ethernet Port 1
0000	General-purpose Ethernet port 0 (with Ethernet switch)	General-purpose Ethernet port 1 (with Ethernet switch)
0001	CC-Link IE port 0	General-purpose Ethernet port 1 (with Ethernet switch)
0010	General-purpose Ethernet port 0 (with Ethernet switch)	CC-Link IE port 1
0011	— (Not used)	General-purpose Ethernet port 1 (without Ethernet switch)
0100	CC-Link IE port 0	CC-Link IE port 1
0101	CC-Link IE port 0	General-purpose Ethernet port 1 (without Ethernet switch)
0110	General-purpose Ethernet port 0 (without Ethernet switch)	CC-Link IE port 1
1000	General-purpose Ethernet port 0 (with Ethernet switch) CC-Link IE port 0 via CC-Link IE port 2	General-purpose Ethernet port 1 (with Ethernet switch) CC-Link IE port 1 via CC-Link IE port 3
1001	CC-Link IE port 0	General-purpose Ethernet Port 1 (without Ethernet switch) CC-Link IE port 1 via CC-Link IE port 3
Settings other than the above are prohibited (If set, the setting is 1000).		

This register is initialized only by the reset signal from the external PONRZ pin.

Note. For available combinations, refer to Table 12.1 "PHY Interface Selection".

12.3.2.2 GMII/MII Management I/F Select Register (MDIOSEL)

This register is used to select the management interface when only one channel is selected for CC-Link IE.

- This register can be read and written in 32- or 16-bit units.

Cautions 1. This register is writable only when a protection release sequence is performed using the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

2. When changing the value of this register, do so while the Gigabit Ethernet PHY is reset.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
MDIOSEL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MDIOSEL	BASE + 0E00H Initial value 0000 0001H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>0</td><td>MDIOSEL</td><td>Selects the MAC that manages the PHY layer.
<table><tr><th>MDIOSEL</th><th>Management Pin</th></tr><tr><td>0</td><td>Gigabit Ethernet MAC</td></tr><tr><td>1</td><td>CC-Link IE</td></tr></table>
The desired management pin is specifiable, while one channel of CC-Link IE is selected (MACSEL = 0000 0001H, 0000 0002H, 0000 0005H, 0000 0006H, 0000 0008H, 0000 0009H). Otherwise, select the Gigabit Ethernet MAC or CC-Link IE Field regardless of the setting of this register.</td></tr>
</table>

This register is initialized only by the reset signal from the external PONRZ pin.

12.3.3 Ethernet Interface Mode Register

12.3.3.1 MDC Clock Select Register (MDCCFG)

This register selects the frequency of the management clock (MDC).

- This register can be read and written in 32- or 16-bit units.

Cautions 1. This register is writable only when a protection release sequence is performed using the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

2. The setting of this register is only effective while the Gigabit Ethernet MAC is selected as the management interface. When CC-Link IE Field is selected, MDC is controlled according to the function of the CC-Link IE Field.

[When the setting of this register is effective]

- **MACSEL = 0000 0000H**
- **MACSEL = 0000 0001H and MDIOSEL = 0000 0000H**
- **MACSEL = 0000 0002H and MDIOSEL = 0000 0000H**
- **MACSEL = 0000 0003H**
- **MACSEL = 0000 0005H and MDIOSEL = 0000 0000H**
- **MACSEL = 0000 0006H and MDIOSEL = 0000 0000H**
- **MACSEL = 0000 0008H and MDIOSEL = 0000 0000H**
- **MACSEL = 0000 0009H and MDIOSEL = 0000 0000H**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDCCFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MDCCNF1	MDCCNF0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W

Address: BASE + 0604H
Initial value: 0000 0000H

Bit	Bit Name	Description
1–0	MDCCNF	Select the frequency of the management clock (MDC). 00: 12.5000 MHz 01: 6.2500 MHz 10: 3.1250 MHz 11: 1.5625 MHz

12.3.4 Gigabit Ethernet MAC Control Register

12.3.4.1 MIIM Register (GMAC_MIIM)

This register is used to access registers of the given Ethernet PHY. Follow the procedure below for access to the registers.

For writing:

1. Start write operation: Set 1 to RWDV bit, PHY address to PHYADDR4 to 0 bits, PHY register address to REGADDR4 to 0 bits, and write data to DATA15 to 0 bits.
2. Wait for the completion of the operation: Wait until 1 is read from the RWDV bit.
3. Completion of the operation: Read 1 from RWDV bit and write operation is complete.

For reading:

1. Start read operation: Set 0 to RWDV bit, PHY address to PHYADDR4 to 0 bits, and PHY register address to REGADDR4 to 0 bits.
2. Wait for the completion of the operation: Wait until 1 is read from the RWDV bit.
3. Completion of the operation: Read 1 from the RWDV bit, valid data from the DATA15 to 0 bits, and read operation is complete.

Caution: The setting of this register is only effective when the general-purpose Ethernet port is selected by the MAC select register (MACSEL). In other cases, writing to this register has no effect and the value read is undefined.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_MIIM	0	0	0	0	0	RWDV	PHYADDR4	PHYADDR3	PHYADDR2	PHYADDR1	PHYADDR0	REGADDR4	REGADDR3	REGADDR2	REGADDR1	REGADDR0	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
R/W	0	0	0	0	0	R/W	W	W	W	W	W	W	W	W	W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 4009 00A0H
Initial value 0400 0000H*

Bit	Bit Name	Description
26	RWDV	Read/write operation starts by writing the following value to this bit. Set other associated bits at the same time. 1: Write operation starts. 0: Read operation starts.
		The state of the operation can be confirmed by reading the following value from this bit.* 1: Operation is completed (bits 25 to 0 are valid). 0: Operation is running.
25–21	PHYADDR4–0	These bits specify the destination PHY address. Since these bits are write-only, the value read is undefined.
20–16	REGADDR4–0	These bits specify the destination PHY register address. Since these bits are write-only, the value read is undefined.
15–0	DATA15–0	These bits indicate write data or read data

Note: The RWDV bit becomes 1 after release from the reset state, but the settings of the DATA 15–0 bits are not effective at this time. When the RWDV bit is used to check the state of operation, start operation to read the correct state.

12.3.4.2 TX ID Register (GMAC_TXID)

This register indicates the ID of the transmission frame corresponding to the setting of the GMAC_TXRESULT register.

To check the transmission frame result, be sure to read this register before reading the GMAC_TXRESULT register. If the GMAC_TXRESULT register is read first, the transmission frame result is updated and the updated transmission frame ID is read from this register.

GMAC_TXID — Address: 4009 000CH, Initial value: 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_TXID	TXID31	TXID30	TXID29	TXID28	TXID27	TXID26	TXID25	TXID24	TXID23	TXID22	TXID21	TXID20	TXID19	TXID18	TXID17	TXID16	TXID15	TXID14	TXID13	TXID12	TXID11	TXID10	TXID9	TXID8	TXID7	TXID6	TXID5	TXID4	TXID3	TXID2	TXID1	TXID0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Description
31–0	TXID31–0	These bits indicate the ID of the transmission frame corresponding to the setting of the TX result register.

12.3.4.3 TX Result Register (GMAC_TXRESULT)

This register indicates the transmission frame result. It is only available while GMAC_TXMODE.TRBMODE1-0 bits are 00 or 01.

The transmission frame result is stored in the transmission result buffer when the Ethernet transmission complete interrupt (INTETHTXCMP) occurs. The transmission result buffer can hold 4 frames of information. Reading this register leads to the frame information being removed from the transmission result buffer. The number of frames stored in this buffer can be obtained from the GMAC_TXFIFO.TRBFR bit.

If transmission starts while the transmission result buffer has 4 frames, transmission is invalid and the TX-FIFO error interrupt (INTETHTXFIFOERR) occurs. While this register is enabled, read it appropriately so that no error occurs.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
GMAC_ TXRESULT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TCMP	TABT	TFAIL	SCOLLIS	MCOLLIS	CSERR	OVERFW	UNDERFW	LCOLLIS	RETRYN3	RETRYN2	RETRYN1	RETRYN0	FIFOUFLOW	4009 0010H Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Description
13	TCMP	Transmission was completed.
12	TABT	Transmission was aborted.
11	TFAIL	Transmission failed because of excessive collision.
10	SCOLLIS	One collision occurred.
9	MCOLLIS	Multi collisions occurred.
8	CSERR	The carrier sense has disappeared during a transmission.
7	OVERFW	A frame longer than 1,518 octets was written to the transmission FIFO.
6	UNDERFW	A frame shorter than the minimum frame length was written to the transmission FIFO.
5	LCOLLIS	Late collision was detected.
4–1	RETRYN3–0	Number of retry times
0	FIFOUFLOW	The FIFO buffer underflowed during a transmission.

12.3.4.4 Mode Register (GMAC_MODE)

This register is used to control the operating mode of the Gigabit Ethernet MAC.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_MODE	ETHMODE	DUPMODE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address 4009 0020H
Initial value 0000 0000H

Bit	Bit Name	Description
31	ETHMODE	Ethernet Mode 1: Operation is in Gigabit Ethernet mode. Use this mode when the Gigabit Ethernet MAC is connected the Ethernet switch. 0: Operation is in 10/100 Ethernet mode.
30	DUPMODE	Duplex Mode 1: Operation is in Full duplex mode. Use this mode when the Gigabit Ethernet MAC is connected the Ethernet switch. 0: Operation is in Half duplex mode.

12.3.4.5 RX Mode Register (GMAC_RXMODE)

This register is used to control operation for reception of frames. The RX FIFO treats a word as 64-bits, and the FIFO size is 4 KB.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_RXMODE	AFILLTEREN	MFILLTEREN	SFRXFIFO	RAMASKEN	0	0	0	0	0	0	0	0	0	0	0	0	REMPTH1	REMPTH0	RFULLTH1	RFULLTH0	RRTTH2	RRTTH1	RRTTH0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	0

Address 4009 0024H
Initial value 2000 0000H

Bit	Bit Name	Description
31	AFILLTEREN	Address Filtering Enable 1: Enable address filtering.* 0: Acquire all frames with addresses.
30	MFILLTEREN	Multicast Filtering Enable 1: Discard frames with multicast addresses other than those registered in the MAC address registers (GMAC_ADRnA, GMAC_ADRnB). (n = 0 to 15) 0: Acquire all frames with multicast addresses.
29	SFRXFIFO	Store & Forward For RX FIFO 1: Store & Forward mode The reception DMA controller starts to operate after data up to the end of the frame is written to the RX FIFO buffer. 0: Cut through mode The reception DMA controller starts to operate after the number of words set in the RRTTH2-0 bits is written to the RX FIFO buffer.

Bit	Bit Name	Description
28	RAMASKEN	RX Address Mask Enable 1: Enable the function that can be set by the BITMSK[7:0] bits of the GMAC_ADRnB register (masking of matching in the comparison of Destination MAC Address[7:0]). (n = 0 to 15) 0: Disable the above function
15–14	REMPTH1–0	Receive Almost Empty Threshold When the number of data words in the FIFO buffer is below this value, the REMP bit of the GMAC_RXFIFO register is set to ‘1’. 00: 4 words 01: 8 words 10: 16 words 11: 32 words
13–12	RFULLTH1–0	Receive Almost Full Threshold When the empty space in the FIFO buffer is below this value, the RFULL bit in the GMAC_RXFIFO register becomes ‘1’. 00: 4 words 01: 8 words 10: 16 words 11: 32 words
11–9	RRTTH2–0	RX FIFO Read Trigger Threshold If the number of data words in the FIFO buffer exceeds this value, the RRT bit of the GMAC_RXFIFO register is set to ‘1’. 000: 4 words 001: 8 words 010: 16 words 011: 32 words 100: 64 words 101: 128 words 110: 256 words 111: 512 words

Note: Even though Address filtering is enabled, MAC Control Frames (ex. Pause Packet) are always received regardless contents of MAC Address Register. MAC Control Frame is the frame that the destination address is 01-80-C2-00-00-01.

12.3.4.6 TX Mode Register (GMAC_TXMODE)

This register is used to control operation for transmission of frames. The TX FIFO treats a word as 64-bits, and the FIFO size is 4 KB.

GMAC_TXMODE — Address 4009 0028H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GMAC_TXMODE	RTRANSDEN	LPTXEN	SF	SPTXEN	RTRANSLC	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_TXMODE	0	0	TEMPTH2	TEMPTH1	TEMPTH0	TFULLTH1	TFULLTH0	0	TRBMODE1	TRBMODE0	0	0	0	0	0	0
R/W	0	0	R/W	R/W	R/W	R/W	R/W	0	R/W	R/W	0	0	0	0	0	0

Bit	Bit Name	Description
31	RTRANSDEN	No Retransmission 1: When a collision occurs, data will not be retransmitted. 0: When a collision occurs, data will be retransmitted according to the standard.
30	LPTXEN	Long Packet TX Enable 1: Transmission of frames which exceed the length specified by the IEEE802.3 standard is enabled.*1 0: Transmission of frames which exceed the length specified by the IEEE802.3 standard is disabled.
29	SF	Store & Forward 1: Transmission starts after the end of a frame is written to the TX FIFO buffer. If you are using a TCP/IP accelerator, this must be selected. 0: Setting prohibited.*2
28	SPTXEN	Short Packet TX Enable 1: Transmission of frames shorter than the length specified by the IEEE802.3 standard is enabled. 0: Transmission of frames shorter than the length specified by the IEEE802.3 standard is disabled.
27	RTRANSLC	Retransmission at Late Collision 1: When a late collision occurs, data will be retransmitted. 0: When a late collision occurs, transmission will be aborted.
15–14	—	Reserved. The write value should be 0. Reading these bits returns 0.

Notes 1. LPTXEN must be set to 1 since the frame size may exceed the maximum size of 1518 bytes while management tag insertion of the Ethernet switch is enabled (the SWTAGEN bit in the ETHSWMTC register is 1).

2. Setting the SF bit to 0 is prohibited. Always start operation after setting this bit to 1. For details, see section 12.5.5, Transmitting Data in Cut-Through Mode.

Bit	Bit Name	Description
13–11	TEMPTH2–0	Transmit Almost Empty Threshold If fewer words of data are in the TX FIFO buffer than the value specified by these bits, the TEMP bit in the GMAC_TXFIFO register becomes 1. 000: 4 words 001: 8 words 010: 16 words 011: 32 words 100: 64 words 101: 128 words 110: 256 words 111: 512 words
10–9	TFULLTH1–0	Transmit Almost Full Threshold If the empty space in the TX FIFO buffer is below the value specified by these bits, the TFULL bit in the GMAC_TXFIFO becomes 1. 00: 4 words 01: 8 words 10: 16 words 11: 32 words
7–6	TRBMODE1–0	Transmission Result Buffer Mode Control how to write the transmission result to the GMAC_TXRESULT register. 00: Always writing 01: Writing only proceeds when an error occurs. 10: Writing does not proceed 11: Setting prohibited

12.3.4.7 Reset Register (GMAC_RESET)

This register is a trigger register for resetting the Gigabit Ethernet MAC by software.

The modules can be reset by setting the corresponding bit to 1.

The waiting time for the completion of a reset depends on the operating mode of the MAC as listed below.

Operation at 1 Gbps (125 MHz): 60 ns

Operation at 100 Mbps (25 MHz): 200 ns

Operation at 10 Mbps (2.5 MHz): 2000 ns

Address 4009 0030H, Initial value 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_RESET	ALLRST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TXRST	RXRST	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Bit Name	Description
31	ALLRST	All Ethernet MAC modules are reset. This bit is write-only. 0: No operation 1: Reset the modules.
15	TXRST	The TX MAC, TX FIFO, and TX DMA modules are reset. This bit is write-only. 0: No operation 1: Reset the modules.
14	RXRST	The RX MAC, RX FIFO, and RX DMA modules are reset. This bit is write-only. 0: No operation 1: Reset the modules.

12.3.4.8 Pause Packet Data Register (GMAC_PAUSEn)

This register is used to specify a pause packet for transmission.

GMAC_PAUSE1 — Address 4009 0080H, Initial value 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_PAUSE1	PPDATA31	PPDATA30	PPDATA29	PPDATA28	PPDATA27	PPDATA26	PPDATA25	PPDATA24	PPDATA23	PPDATA22	PPDATA21	PPDATA20	PPDATA19	PPDATA18	PPDATA17	PPDATA16	PPDATA15	PPDATA14	PPDATA13	PPDATA12	PPDATA11	PPDATA10	PPDATA9	PPDATA8	PPDATA7	PPDATA6	PPDATA5	PPDATA4	PPDATA3	PPDATA2	PPDATA1	PPDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	PPDATA31–0	The 4th to 1st bytes of a pause packet for transmission are designated. Data are transmitted in order from the LSB.

GMAC_PAUSE2 — Address 4009 0084H, Initial value 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_PAUSE2	PPDATA31	PPDATA30	PPDATA29	PPDATA28	PPDATA27	PPDATA26	PPDATA25	PPDATA24	PPDATA23	PPDATA22	PPDATA21	PPDATA20	PPDATA19	PPDATA18	PPDATA17	PPDATA16	PPDATA15	PPDATA14	PPDATA13	PPDATA12	PPDATA11	PPDATA10	PPDATA9	PPDATA8	PPDATA7	PPDATA6	PPDATA5	PPDATA4	PPDATA3	PPDATA2	PPDATA1	PPDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	PPDATA31–0	The 8th to 5th bytes of a pause packet for transmission are designated. Data are transmitted in order from the LSB.

GMAC_PAUSE3 — Address 4009 0088H, Initial value 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_PAUSE3	PPDATA31	PPDATA30	PPDATA29	PPDATA28	PPDATA27	PPDATA26	PPDATA25	PPDATA24	PPDATA23	PPDATA22	PPDATA21	PPDATA20	PPDATA19	PPDATA18	PPDATA17	PPDATA16	PPDATA15	PPDATA14	PPDATA13	PPDATA12	PPDATA11	PPDATA10	PPDATA9	PPDATA8	PPDATA7	PPDATA6	PPDATA5	PPDATA4	PPDATA3	PPDATA2	PPDATA1	PPDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	PPDATA31–0	The 12th to 9th bytes of a pause packet for transmission are designated. Data are transmitted in order from the LSB.

GMAC_PAUSE4 — Address 4009 008CH, Initial value 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_ PAUSE4	PPDATA31	PPDATA30	PPDATA29	PPDATA28	PPDATA27	PPDATA26	PPDATA25	PPDATA24	PPDATA23	PPDATA22	PPDATA21	PPDATA20	PPDATA19	PPDATA18	PPDATA17	PPDATA16	PPDATA15	PPDATA14	PPDATA13	PPDATA12	PPDATA11	PPDATA10	PPDATA9	PPDATA8	PPDATA7	PPDATA6	PPDATA5	PPDATA4	PPDATA3	PPDATA2	PPDATA1	PPDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	PPDATA31–0	The 16th to 13th bytes of a pause packet for transmission are designated. Data are transmitted in order from the LSB.

GMAC_PAUSE5 — Address 4009 0090H, Initial value 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_ PAUSE5	PPDATA31	PPDATA30	PPDATA29	PPDATA28	PPDATA27	PPDATA26	PPDATA25	PPDATA24	PPDATA23	PPDATA22	PPDATA21	PPDATA20	PPDATA19	PPDATA18	PPDATA17	PPDATA16	PPDATA15	PPDATA14	PPDATA13	PPDATA12	PPDATA11	PPDATA10	PPDATA9	PPDATA8	PPDATA7	PPDATA6	PPDATA5	PPDATA4	PPDATA3	PPDATA2	PPDATA1	PPDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	PPDATA31–0	The 20th to 17th bytes of a pause packet for transmission are designated. Data are transmitted in order from the LSB.

12.3.4.9 RX Flow Control Register (GMAC_FLWCTL)

This register is used to control operation after reception of a pause packet.

If a pause packet is received while this function is enabled, transmission is suspended for the time specified by the pause packet.

GMAC_FLWCTL — Address 4009 0098H, Initial value 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_ FLWCTL	PPRXEN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Bit Name	Description
31	PPRXEN	1: Enable auto broadcast suspension in response to reception of a pause packet. 0: Disable auto broadcast suspension in response to reception of a pause packet.

12.3.4.10 Pause Packet Register (GMAC_PAUSPKT)

This register is used to control transmission of a pause packet.

When 1 is written to the PPR bit, transmission of a pause packet specified by GMAC_PAUSEn registers starts. The bit is automatically set to 0 following the completion of the transmission.

GMAC_PAUSPKT — Address 4009 009CH, Initial value 0000 0000H

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

Bit	Bit Name	Description
31	PPR	This bit controls transmission of a pause packet. 0: Nothing is to be done. 1: Start pause packet transmission.

The transmission packet format is shown below.

	31–16	15–0
GMAC_PAUSE1	Destination Address	
GMAC_PAUSE2	Source Address	Destination Address
GMAC_PAUSE3	Source Address	
GMAC_PAUSE4	Opcode	Type/Length
GMAC_PAUSE5	(Not used)	Time

12.3.4.11 MAC Address Registers (GMAC_ADRnA, GMAC_ADRnB)

These registers are used to configure the MAC addresses.

A total of 16 MAC addresses can be registered. Multiple addresses can be filtered by using the BITMSK7 to 0 bits of the GMAC_ADRnB register (n = 0, 1, ..., 15).

GMAC_ADRnA — Address: 4009 0100H +8H*n; Initial value: 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_ADRnA	MADDR4B7	MADDR4B6	MADDR4B5	MADDR4B4	MADDR4B3	MADDR4B2	MADDR4B1	MADDR4B0	MADDR3B7	MADDR3B6	MADDR3B5	MADDR3B4	MADDR3B3	MADDR3B2	MADDR3B1	MADDR3B0	MADDR2B7	MADDR2B6	MADDR2B5	MADDR2B4	MADDR2B3	MADDR2B2	MADDR2B1	MADDR2B0	MADDR1B7	MADDR1B6	MADDR1B5	MADDR1B4	MADDR1B3	MADDR1B2	MADDR1B1	MADDR1B0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–24	MADDR4B7–0	Indicate the 4th byte from the beginning of the MAC address to be acquired.
23–16	MADDR3B7–0	Indicate the 3rd byte from the beginning of the MAC address to be acquired.
15–8	MADDR2B7–0	Indicate the 2nd byte from the beginning of the MAC address to be acquired.
7–0	MADDR1B7–0	Indicate the 1st byte of the MAC address to be acquired.

GMAC_ADRnB — Address: 4009 0104H +8H*n; Initial value: 00FF 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_ADRnB	0	0	0	0	0	0	0	0	BITMASK7	BITMASK6	BITMASK5	BITMASK4	BITMASK3	BITMASK2	BITMASK1	BITMASK0	MADDR6B7	MADDR6B6	MADDR6B5	MADDR6B4	MADDR&B3	MADDR6B2	MADDR6B1	MADDR6B0	MADDR5B7	MADDR5B6	MADDR5B5	MADDR5B4	MADDR5B3	MADDR5B2	MADDR5B1	MADDR5B0
R/W	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
23–16	BITMSK7–0	These bits mask given bits for matching in the comparison of the destination MAC address [7:0] bits. Bits [23:16] of this register correspond to the destination MAC address [7:0] bits, respectively, and those for which the BITMSK setting is 0 are excluded from comparison for matching. For example, if mask register bits BITMSK2-0 are all 0s, the destination MAC address [2:0] bits are excluded from comparison for matching. In other words, if the destination MAC address [47:3] bits match, the given frame is acquired.
15–8	MADDR6B7–0	Indicate the 1st byte to 6th byte of the MAC address to be acquired.
7–0	MADDR5B7–0	Indicate the 1st byte to 5th byte of the MAC address to be acquired.

12.3.4.12 RX FIFO Status Register (GMAC_RXFIFO)

This register is a status register which indicates the state of the reception FIFO.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_ RXFIFO	RFULL	REMP	RRT	RSW11	RSW10	RSW9	RSW8	RSW7	RSW6	RSW5	RSW4	RSW3	RSW2	RSW1	RSW0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address 4009 0200H

Initial value 4000 0000H

Bit	Bit Name	Description
31	RFULL	RX FIFO Almost Full 1: Indicate that the data in the RX FIFO buffer is over the Receive Almost Full Threshold. (This threshold is configured by the GMAC_RXMODE register.)
30	REMP	RX FIFO Almost Empty 1: Indicate that the data in the RX FIFO buffer is below the Receive Almost Empty Threshold. (This threshold is configured by the GMAC_RXMODE register.)
29	RRT	RX FIFO Read Trigger 1: Indicate that the data in the RX FIFO buffer is over the RX FIFO Read Threshold. (This threshold is configured by the GMAC_RXMODE register.)
28–17	RSW11–0	Stored Words in RX FIFO Indicate the number of words of the data in the RX FIFO buffer.

12.3.4.13 TX FIFO Status Register (GMAC_TXFIFO)

This register is a status register which indicates the state of the transmission FIFO.

Address 4009 0204H
Initial value 6000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_TXFIFO	TFULL	TEMP	TSTAUS2	TSTAUS1	TSTAUS0	TRBFR2	TRBFR1	TRBFR0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Bit Name	Description
31	TFULL	TX FIFO Almost Full 1: Indicate that the empty space in the TX FIFO buffer is below the threshold set by the TFULLTH1-0 bits of the GMAC_TXMODE register.
30	TEMP	TX FIFO Almost Empty 1: Indicate that the number of data words in the TX FIFO buffer is below the threshold set by the TEMPTH2-0 bits of the GMAC_TXMODE register.
29–27	TSTATUS2–0	TX FIFO Status These bits indicate the state of the TX FIFO buffer. The meaning of the bits is as follows. 100: ACC NEW FR: The TX FIFO buffer is ready to receive a new frame. 101: WRITE ENABLE: The TX FIFO buffer is ready to receive frame data continuously. 110: CMPLT: Indicates the completion of the acquisition of one frame. 111: FULL: The TX FIFO buffer is full. 0xx: STOP: The TX FIFO buffer is stopped (or being initialized).
26–24	TRBFR2–0	The number of frames in the transmission result buffer.

12.3.4.14 TCPIPACC Register (GMAC_ACC)

This register is used to control operation of the TCPIP accelerator.

Address 4009 0208H
Initial value 0000 0003H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_ACC	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	RTCPIPACC	TTCPIPEN	RTCPIPEN
R/W	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	R/W	R/W	R/W

Bit	Bit Name	Description
2	RTCPIPACC	1: RX TCPIPACC Off Disable the checksum support for the RX TCPIP accelerator. Padding in the MAC header section is inserted. 0: The checksum support for the RX TCPIP accelerator remains enabled.
1	TTCPIPEN	1: TX TCPIP Enable Enable the TX TCPIP accelerator. 0: TX TCPIP Disable Disable the TX TCPIP accelerator completely. Padding in the MAC header section is also disabled.
0	RTCPIPEN	1: RX TCPIP Enable Enable the RX TCPIP accelerator 0: RX TCPIP Disable Disable the RX TCPIP accelerator completely. Padding in the MAC header section is not inserted.

12.3.4.15 RX MAC Enable Register (GMAC_RXMAC_ENA)

This register is used to control operation of the reception MAC.

Address 4009 0220H
Initial value 0000 0001H

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

Bit	Bit Name	Description
0	RMACEN	RX MAC ENABLE 1: Enables reception. 0: Disables reception.

12.3.4.16 LPI Mode Control Register (GMAC_LPI_MODE)

This register is used to control LPI (Low Power Idle) mode. When the LPMEN bit is set to 1, an LPI request is automatically sent to the link partner in the case there is no transmission request over the time specified by the LPRDEF bit of the GMAC_LPI_TIMING register. If a transmission request is generated during the LPI state, the MAC finishes this state and waits for the time specified by the LPWTIME bit of the GMAC_LPI_TIMING register, and then transmits a frame. If the Gigabit Ethernet MAC is connected via the Ethernet switch, do not use this register to set LPI mode.

Address 4009 0224H, Initial value 0000 0000H

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

Bit	Bit Name	Description
31	LPMEN	Low Power Idle Mode 1: The Gigabit Ethernet MAC operates in LPI mode. 0: The Gigabit Ethernet MAC does not operate in LPI mode.

12.3.4.17 LPI Client Timing Control Register (GMAC_LPI_TIMING)

This register is used to control the signal timing in LPI mode.

Do not use this register if the Gigabit Ethernet MAC is connected via the Ethernet switch.

Address 4009 0228H, Initial value 0000 080FH

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMAC_LPI_TIMING	LPRDEFR15	LPRDEFR14	LPRDEFR13	LPRDEFR12	LPRDEFR11	LPRDEFR10	LPRDEFR9	LPRDEFR8	LPRDEFR7	LPRDEFR6	LPRDEFR5	LPRDEFR4	LPRDEFR3	LPRDEFR2	LPRDEFR1	LPRDEFR0	LPWTIME15	LPWTIME14	LPWTIME13	LPWTIME12	LPWTIME11	LPWTIME10	LPWTIME9	LPWTIME8	LPWTIME7	LPWTIME6	LPWTIME5	LPWTIME4	LPWTIME3	LPWTIME2	LPWTIME1	LPWTIME0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–16	LPRDEF15–0	Low Power Idle Request Deferral These bits set a delay time until an LPI request is transmitted to a link partner. Setting is in units of 8 nanoseconds in Gigabit mode, and in units of 40 nanoseconds in 100 Mbps mode.
15–0	LPWTIME15–0	Low Power Idle Wake time These bits set a time until a frame can be transmitted to a link partner after an IDLE signal has been transmitted to a link partner when LPI is finished. Setting is in units of 8 nanoseconds in Gigabit mode, and in units of 40 nanoseconds in 100 Mbps mode.

Caution: The settings of the GMAC_LPI_MODE and GMAC_LPI_TIMING registers are only effective while the MACSEL register value is 0000 0003H, 0000 0005H, or 0000 0006H (without an Ethernet switch).

12.3.4.18 Receive Buffer Information Register (BUFID)

This register indicates information of the receive buffer (whether or not data exists, the address of the buffer holding received data, and the number of words of data). If the reception MACDMAC has completed data transfer, the receive buffer information is written to this register and held up to 32 pieces of information. If the receive buffer has data, the Ethernet MACDMA reception complete interrupt (INTETHRXDMA) occurs. This interrupt stays active until the receive buffer becomes empty (i.e. the receive buffer information is read and the NOEMP bit becomes 0).

Address: 4009 1100H
Initial value: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BUFID	NOEMP	0	0	VALID	WORD11	WORD10	WORD9	WORD8	WORD7	WORD6	WORD5	WORD4	WORD3	WORD2	WORD1	WORD0
R/W	R	0	0	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFID	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Description
31	NOEMP	1: The receive buffer has data. 0: The receive buffer has no data.
28	VALID	1: The data in the receive buffer is valid. 0: The data in the receive buffer is not valid.
27–16	WORD11–0	Number of words of received data (including the received MAC information) A word unit is 32 bits.
15–0	ADDR15–0	Address of the receive buffer (26 to 11 bits)

Note: Since this register indicates the information of the next received data every time it is read, the value of this register changes every time it is read.

The ADDR bits cannot indicate a 32-bit address space. Therefore, access to the memory-mapped buffer requires an offset of 0x08000000.

[Method of calculating the receive buffer address]

1. Obtain the value of the ADDR bit.
2. Shift the value by 11 bits to the left.
3. Add the offset of 0x08000000.

The number of words indicated by the WORD bits also includes the received frame information. The start address of the received frame information is calculated by following the procedure below.

[Method of calculating the start address of the received frame information]

1. Obtain the value of the WORD bits.
2. Shift the value by 16 bits to the right.
3. Add the number of words shifted in step 2 to the receive buffer address as an offset.
4. Offset by subtracting the size of the received frame information (2 words).

12.3.5 Hardware Function Call Register

The hardware function call registers are used to acquire buffers and start transmission or reception (hardware function).

The hardware function is executed by writing the given command to the system call register (SYSC) after configuring the argument registers (R4-R7). For how to configure the hardware function call registers, see section 12.4.1, Hardware Functions.

Note: The hardware function related registers are also used for controlling the hardware real-time OS.

12.3.5.1 Hardware Function System Call Register (SYSC)

By writing the given command to this register, the corresponding function is executed.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SYSC15	SYSC14	SYSC13	SYSC12	SYSC11	SYSC10	SYSC9	SYSC8	SYSC7	SYSC6	SYSC5	SYSC4	SYSC3	SYSC2	SYSC1	SYSC0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address
4008 F000H
Initial value
—

Bit	Bit Name	Description
15–0	SYSC15–0	Select the hardware function. The following functions are available.

SYSC15–0	Function
0x5000	Acquires a long buffer.
0x5006	Acquires a short buffer.
0x5001	Releases the whole area of the buffer.
0x5002	Releases the part of the buffer.
0x5101	Enables DMA for the reception MAC.
0x5102	Disables DMA for the reception MAC.
0x510B	Controls interrupts for the reception MACDMAC.
0x510D	Obtains error sources for the reception MACDMAC.
0x5100	Starts transfer by the transmission MACDMAC.
0x510C	Obtains error sources in the transmission MACDMAC.
0x5211	Starts DMA transfer between the buffer RAM and data RAM.
0x5212	Starts replacing data in the buffer RAM or data RAM.
0x5104	Starts DMA transfer between the buffer RAMs.
0x5114	Starts DMA transfer between the buffer RAMs (descriptor method).
Others	Setting prohibited

12.3.5.2 Hardware Function Argument Registers (R4 to R7)

These registers are for writing arguments transferred to a hardware function. Which register is used differs with the hardware function. For details, see section 12.4, Functions.

R4 — Address: 4008 F004H, Initial value: —

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R4B31	R4B30	R4B29	R4B28	R4B27	R4B26	R4B25	R4B24	R4B23	R4B22	R4B21	R4B20	R4B19	R4B18	R4B17	R4B16	R4B15	R4B14	R4B13	R4B12	R4B11	R4B10	R4B9	R4B8	R4B7	R4B6	R4B5	R4B4	R4B3	R4B2	R4B1	R4B0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	R4B31–0	Specify the argument for transfer to the hardware function.

R5 — Address: 4008 F008H, Initial value: —

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R5	R5B31	R5B30	R5B29	R5B28	R5B27	R5B26	R5B25	R5B24	R5B23	R5B22	R5B21	R5B20	R5B19	R5B18	R5B17	R5B16	R5B15	R5B14	R5B13	R5B12	R5B11	R5B10	R5B9	R5B8	R5B7	R5B6	R5B5	R5B4	R5B3	R5B2	R5B1	R5B0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	R5B31–0	Specify the argument for transfer to the hardware function.

R6 — Address: 4008 F00CH, Initial value: —

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R6	R6B31	R6B30	R6B29	R6B28	R6B27	R6B26	R6B25	R6B24	R6B23	R6B22	R6B21	R6B20	R6B19	R6B18	R6B17	R6B16	R6B15	R6B14	R6B13	R6B12	R6B11	R6B10	R6B9	R6B8	R6B7	R6B6	R6B5	R6B4	R6B3	R6B2	R6B1	R6B0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	R6B31–0	Specify the argument for transfer to the hardware function.

R7 — Address: 4008 F010H, Initial value: —

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R7	R7B31	R7B30	R7B29	R7B28	R7B27	R7B26	R7B25	R7B24	R7B23	R7B22	R7B21	R7B20	R7B19	R7B18	R7B17	R7B16	R7B15	R7B14	R7B13	R7B12	R7B11	R7B10	R7B9	R7B8	R7B7	R7B6	R7B5	R7B4	R7B3	R7B2	R7B1	R7B0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Bit Name	Description
31–0	R7B31–0	Specify the argument for transfer to the hardware function.

12.3.5.3 Hardware Function Operating Mode Control Register (CMD)

This register controls the operating mode of the hardware functions.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	CMDB31	CMDB30	CMDB29	CMDB28	CMDB27	CMDB26	CMDB25	CMDB24	CMDB23	CMDB22	CMDB21	CMDB20	CMDB19	CMDB18	CMDB17	CMDB16	CMDB15	CMDB14	CMDB13	CMDB12	CMDB11	CMDB10	CMDB9	CMDB8	CMDB7	CMDB6	CMDB5	CMDB4	CMDB3	CMDB2	CMDB1	CMDB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 4008 F014H
Initial value —

Bit	Bit Name	Description
31–0	CMDB31–0	These bits control the operating mode of the hardware functions. 0x0000 8004: Start the hardware function and hardware real-time OS.

12.3.5.4 Hardware Function Return Value Registers (R0, R1)

These registers hold the value returned from a hardware function. The value returned depends on the hardware function. For details, see section 12.4, Functions.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	R0B31	R0B30	R0B29	R0B28	R0B27	R0B26	R0B25	R0B24	R0B23	R0B22	R0B21	R0B20	R0B19	R0B18	R0B17	R0B16	R0B15	R0B14	R0B13	R0B12	R0B11	R0B10	R0B9	R0B8	R0B7	R0B6	R0B5	R0B4	R0B3	R0B2	R0B1	R0B0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address 4008 F020H
Initial value —

Bit	Bit Name	Description
31–0	R0B31–0	Store the returned value from the hardware function.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1	R1B31	R1B30	R1B29	R1B28	R1B27	R1B26	R1B25	R1B24	R1B23	R1B22	R1B21	R1B20	R1B19	R1B18	R1B17	R1B16	R1B15	R1B14	R1B13	R1B12	R1B11	R1B10	R1B9	R1B8	R1B7	R1B6	R1B5	R1B4	R1B3	R1B2	R1B1	R1B0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address 4008 F024H
Initial value —

Bit	Bit Name	Description
31–0	R1B31–0	Store the returned value from the hardware function.

12.3.5.5 Hardware Function Type Register (CNTX_TYPE0)

This register is for setting the type of hardware function.

CNTX_TYPE0 — Address 4008 0000H, Initial value —

Bit	CNTX_TYPE0	R/W
31	CNTX_TYPE031	R/W
30	CNTX_TYPE030	R/W
29	CNTX_TYPE029	R/W
28	CNTX_TYPE028	R/W
27	CNTX_TYPE027	R/W
26	CNTX_TYPE026	R/W
25	CNTX_TYPE025	R/W
24	CNTX_TYPE024	R/W
23	CNTX_TYPE023	R/W
22	CNTX_TYPE022	R/W
21	CNTX_TYPE021	R/W
20	CNTX_TYPE020	R/W
19	CNTX_TYPE019	R/W
18	CNTX_TYPE018	R/W
17	CNTX_TYPE017	R/W
16	CNTX_TYPE016	R/W
15	CNTX_TYPE015	R/W
14	CNTX_TYPE014	R/W
13	CNTX_TYPE013	R/W
12	CNTX_TYPE012	R/W
11	CNTX_TYPE011	R/W
10	CNTX_TYPE010	R/W
9	CNTX_TYPE09	R/W
8	CNTX_TYPE08	R/W
7	CNTX_TYPE07	R/W
6	CNTX_TYPE06	R/W
5	CNTX_TYPE05	R/W
4	CNTX_TYPE04	R/W
3	CNTX_TYPE03	R/W
2	CNTX_TYPE02	R/W
1	CNTX_TYPE01	R/W
0	CNTX_TYPE00	R/W

Bit	Bit Name	Description
31–0	CNTX_TYPE031–0	These bits set the type of hardware function.

12.3.5.6 Hardware Function State Register (CNTX_STAT0)

This register is for setting the state of the hardware functions.

CNTX_STAT0 — Address 4008 0008H, Initial value —

Bit	CNTX_STAT0	R/W
31	CNTX_STAT031	R/W
30	CNTX_STAT030	R/W
29	CNTX_STAT029	R/W
28	CNTX_STAT028	R/W
27	CNTX_STAT027	R/W
26	CNTX_STAT026	R/W
25	CNTX_STAT025	R/W
24	CNTX_STAT024	R/W
23	CNTX_STAT023	R/W
22	CNTX_STAT022	R/W
21	CNTX_STAT021	R/W
20	CNTX_STAT020	R/W
19	CNTX_STAT019	R/W
18	CNTX_STAT018	R/W
17	CNTX_STAT017	R/W
16	CNTX_STAT016	R/W
15	CNTX_STAT015	R/W
14	CNTX_STAT014	R/W
13	CNTX_STAT013	R/W
12	CNTX_STAT012	R/W
11	CNTX_STAT011	R/W
10	CNTX_STAT010	R/W
9	CNTX_STAT09	R/W
8	CNTX_STAT08	R/W
7	CNTX_STAT07	R/W
6	CNTX_STAT06	R/W
5	CNTX_STAT05	R/W
4	CNTX_STAT04	R/W
3	CNTX_STAT03	R/W
2	CNTX_STAT02	R/W
1	CNTX_STAT01	R/W
0	CNTX_STAT00	R/W

Bit	Bit Name	Description
31–0	CNTX_STAT031–0	These bits set the state of the hardware functions.

12.4 Functions

12.4.1 Hardware Functions

A hardware function (HWF) is defined as a function for reducing the load on the CPU, such as a DMAC or Ethernet communications accelerator.

A hardware function consists of a combination of hardware modules which are divided by function, and an overall function is defined for the set of individual hardware modules.

The following three functions are defined as hardware functions.

- Buffer Allocator
- MAC DMA Controller
- Buffer RAM DMA Controller

The figure below is a schematic block diagram of these hardware functions in context. Solid lines in the figure indicate the flow of data, while broken lines indicate a command interface with the hardware function.

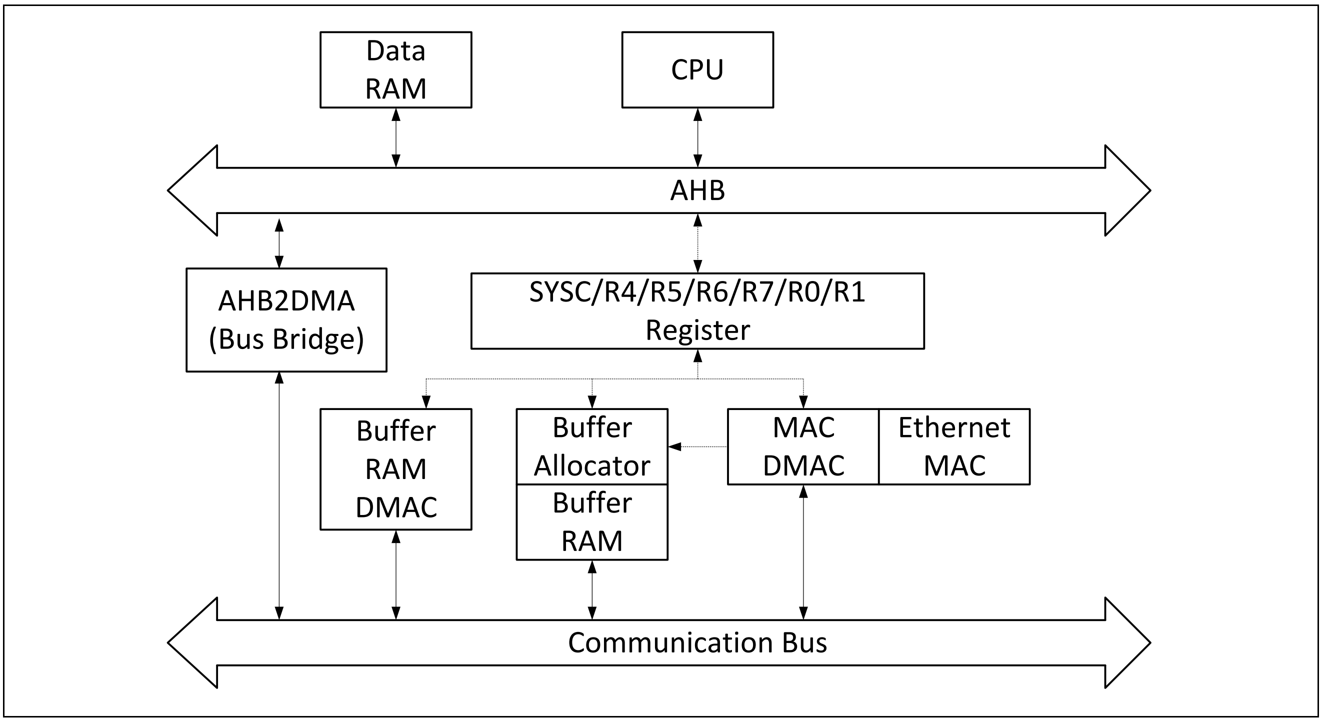


Figure 12.2 Schematic Block Diagram of the Hardware Functions

Caution: Calling the hardware function while the hardware real-time OS is prohibited from dispatching does not successfully execute the call. Be sure to call the hardware function while dispatching is allowed.

12.4.1.1 Initial Settings

Execute the commands listed below to set up the hardware functions.

Procedure for setting up the hardware functions

<1> Set 0x0000 0003 in the CNTX_TYPE0 register.

<2> Set 0x0000 0003 in the CNTX_STAT0 register.

<3> Set 0x0000 8004 in the CMD register.

<4> Wait until 0x8000 0000 is read from the R0 register. Afterwards, dummy-read the R1 register.

<5> Set 0x8000 0000 in the GMAC_RESET register to initialize the Gigabit Ethernet MAC.

Caution: When the hardware real-time OS is used, these settings are not required since it is controlled by setting up the hardware real-time OS functions.

After the completion of setup, make initial settings in the registers below.

- MAC address register (→ 12.3.4.11)
- TX MODE register (→ 12.3.4.6)
- RX MODE register (→ 12.3.4.5)

12.4.1.2 Flow of Processing for Issuing the Hardware Function Call

If you are using a hardware function, follow the flowchart below to issue the hardware function call.

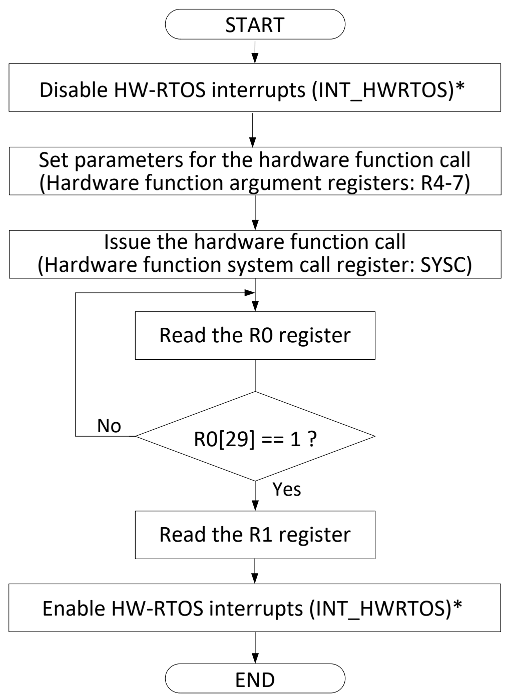


Figure 12.3 Flow of Processing for Issuing the Hardware Function

Note: This processing is required only when the hardware real-time OS is used.

12.4.1.3 Buffer Allocator

(1) Functional Overview

The buffer allocator is a module for controlling the buffer RAM.

The buffer RAM is a communications buffer to improve throughput in Ethernet transfer. Although the buffer RAM has 64 Kbytes, an area of 128 Mbytes is used as the logical space for the dynamic securing and releasing of memory space by the buffer allocator.

To use the buffer RAM, secure the required area (hereafter "buffer") beforehand, and then issue the hardware function calls provided for the buffer allocator. When writing to an area which has not been secured, access by the CPU or MAC DMA controller generates an interrupt, whereas access to such area by the buffer RAM DMA controller generates an interrupt or returns an exception to the return value register R0 depending on the type of hardware function calls.

To reuse a buffer after having secured it, the buffer must be released after it has been used.

The outline of the functions is as follows:

- A long buffer of up to 2048 bytes and short buffer of up to 512 bytes are available.
- When securing a buffer, the size is specified in bytes.
- When releasing a buffer, the size can be specified for the whole area or as the location of a byte (the part of the buffer from that address is released).

The segments which constitute a buffer are of 128 bytes each. The buffer allocator controls each of these 128-byte segments, and connect these segments in response to hardware function calls to provide these as buffers. Addresses are seen as continuous across contiguous segments.

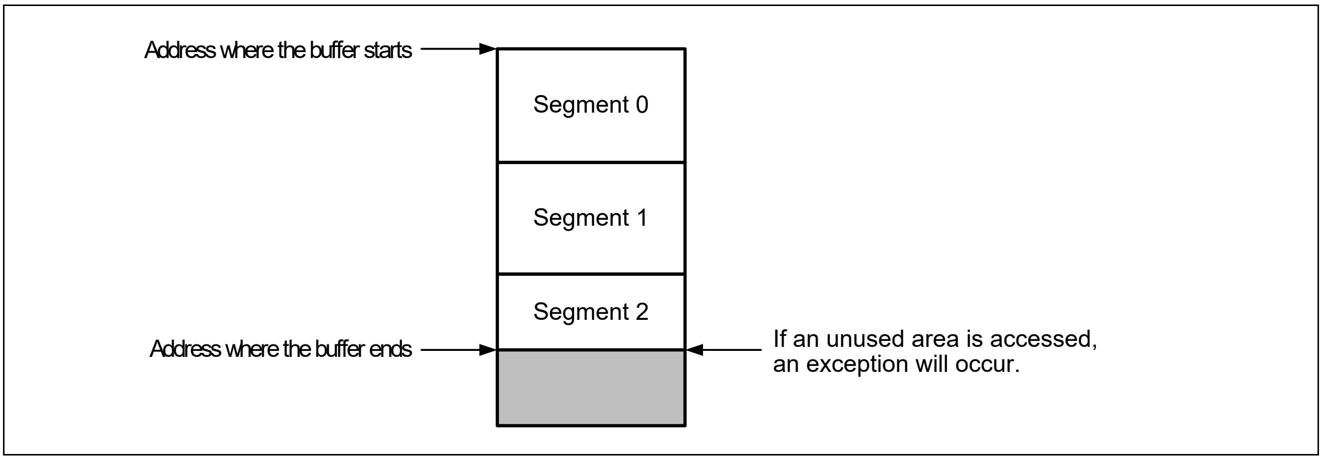


Figure 12.4 Method of Controlling a Buffer

(2) Buffer Control Operation

In this section, short and long buffers are collectively referred to as "buffers". A short buffer has up to four segments and a long buffer has up to 16 segments.

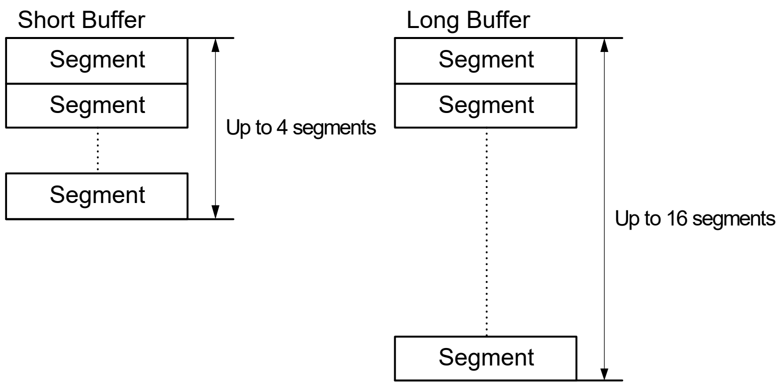


Figure 12.5 Buffer Structure

(a) Acquisition of buffers (HWFNC_ShortBuffer_Get, HWFNC_LongBuffer_Get)

Buffers can be acquired by issuing an HWFNC_ShortBuffer_Get or HWFNC_LongBuffer_Get hardware function call. The size of the buffer is specified in bytes when calling these hardware functions. The number of bytes does not have to reach a segment boundary. The value returned is the address where the buffer starts.

The maximum numbers of short and long buffers that can be acquired are as listed in Table 12.2. Even if fewer short and long buffers are acquired than the maximum, acquisition will fail if the total size of buffers of both sizes exceeds the maximum size imposed by the 64 Kbytes of buffer RAM.

Table 12.2 Number of Buffers that can be Acquired

Buffer Type	Maximum Number of Buffers that can be Acquired	Remarks
Short buffer	64	Up to 256 segments (= 32 KB)
Long buffer	32	Up to 512 segments (= 64 KB)

The address structure of buffers is shown below. When a buffer is acquired, the function returns the address range from 0x0C00 0000 to 0x0FFF FFFF and 0x0800 0000 to 0x0BFF FFFF for a long buffer and short buffer, respectively.

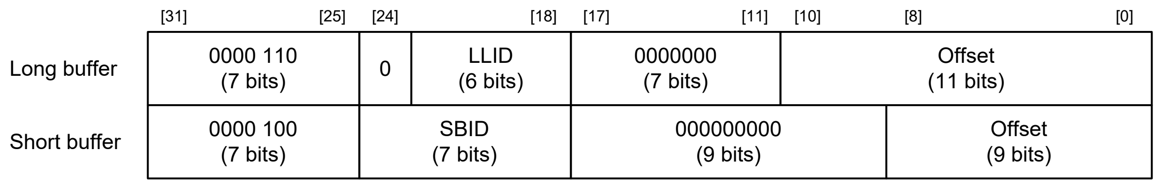


Figure 12.6 Address Structure of Buffers

If a short buffer is acquired, bits [24:18] are given an SBID (short buffer ID), which is used as an identifier for the buffer. The buffer area is allocated with the offset field as 0 to indicate the address where the buffer starts.

If a long buffer is acquired, bits [23:18] are given an LLID (linked long buffer ID), which is used as an identifier for the buffer. The buffer area is allocated with the offset field as 0 to indicate the address where the buffer starts.

(b) Releasing a buffer (HWFNC_Buffer_Release)

The whole area of an acquired buffer can be released by calling the HWFNC_Buffer_Release hardware function. When calling the hardware function, specify the address where the acquired buffer to be released starts.

(c) Releasing part of a buffer (HWFNC_Buffer_Return)

By calling the HWFNC_Buffer_Return hardware function, desired bytes can be released, starting from the location of a byte within the acquired buffer. This is provided for efficiency in using the space; for example, when a frame is received, another resource can use the area obtained by releasing the area following the end of the received frame data. When executing this system call, the addresses where the buffer and the space to be released start must be given as arguments.

(d) Testing memory and initializing buffers

Since it is not allocated at the time of a reset, buffer RAM is neither writable nor readable in that situation. Accordingly, to test the memory, execute the HWFNC LongBuffer_Get system call, etc., to secure the full capacity of the buffer RAM and make that memory available for access. This enables subsequent checking of the memory and initializing its contents.

(e) List of hardware function calls

The table below lists the hardware function calls.

If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.

Table 12.3 HWFNC_LongBuffer_Get

Name	HWFNC_LongBuffer_Get	
Function	Acquires a long buffer for use in the transmission and reception of frames. A buffer can be acquired with any size in bytes between 1 and 2048. Long buffers are mainly used to hold the data sections of frames. The address where the acquired buffer starts is returned in R1 as the value returned.	
Command register		
SYSC[15:0]	0x5000	
Argument registers		
R4[15:0]	Buffer Length	Required buffer length. Unit: bytes. 1 to 2048
R4[23:16]	Reserved	Always 0
R4[31:24]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	2'b0x and R0[29] = 1: Success 2'b10: Invalid system call 2'b11: The buffer is insufficient.
R0[28:2]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	First logical address of the buffer	[31:27] 5'b00001 [26:24] 3'b100 [23:18] LLID [17: 0] 0

Caution: Issuing of this command while the hardware real-time OS is prohibited from dispatching does not successfully execute the hardware function call.
In this case, bits [15:0] of return value register R0 indicates FFE7h.

Table 12.4 HWFNC_ShortBuffer_Get

Name	HWFNC_ShortBuffer_Get	
Function	Acquires a short buffer for use in the transmission and reception of frames. A buffer can be acquired with any size in bytes between 1 and 512. Short buffers are mainly used to hold the header sections of frames, the data sections of ICMP and MAC management frames, etc. The address where the acquired buffer starts is returned in R1 as the value returned.	
Command register		
SYSC[15:0]	0x5006	
Argument registers		
R4[15:0]	Buffer Length	Required buffer length. Unit: bytes. 1 to 512
R4[31:16]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	2’b0x: Success 2’b10: Invalid system call 2’b11: The buffer is insufficient.
R0[28:2]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	First logical address of the buffer	[31:27] 5’b00001 [26:25] 2'b00 [24:18] SBID [17: 0] 0

Table 12.5 HWFNC_Buffer_Release

Name	HWFNC_Buffer_Release	
Function	Releases an acquired long or short buffer.	
Command register		
SYSC[15:0]	0x5001	
Argument registers		
R4[31:0]	First logical address of the buffer	First logical address of the buffer to be released The value is returned in R1 following a call of HWFNC_LongBuffer_Get or HWFNC_ShortBuffer_Get.
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	2’b0x: Success 2’b10: Invalid system call 2’b11: A buffer is not definable at the given address.
R0[28:2]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

Table 12.6 HWFNC_Buffer_Return

Name	HWFNC_Buffer_Return	
Function	Releases some of the latter half of an acquired short or long buffer. Specifying the location where the address range to be released starts leads to the release of the part of the buffer beginning at that address. The address can be set as any byte. This HWF is for the efficient use of buffer resources, for example when a received frame is short.	
Command register		
SYSC[15:0]	0x5002	
Argument registers		
R4[31:0]	First logical address of the buffer	First logical address of the buffer to be released The value is returned in R1 following a call of HWFNC_LongBuffer_Get or HWFNC_ShortBuffer_Get.
R5[31:0]	First logical address of the part for release	First address of the part for release (the part of the buffer at addresses beginning from this address is released)
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[2:0]	Result	3'b00x: Success 3'b010: Invalid system call 3'b011: A buffer is not definable at the address specified by R4. 3'b100: The part of the buffer at the address specified by R5 has already been released.
R0[28:3]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

12.4.1.4 MAC DMA Controller

(1) Functional Overview

The MAC DMA controller is used to transfer data between the buffer RAM and Ethernet MAC. In transmission, the DMAC transfers data to be transmitted from the buffer RAM to the Ethernet MAC; in reception, the DMAC transfers data received by the Ethernet MAC to the buffer RAM. This allows improved throughput for communications.

Figure 12.7 is a block diagram of the MACDMAC in context and the respective interrupt signals.

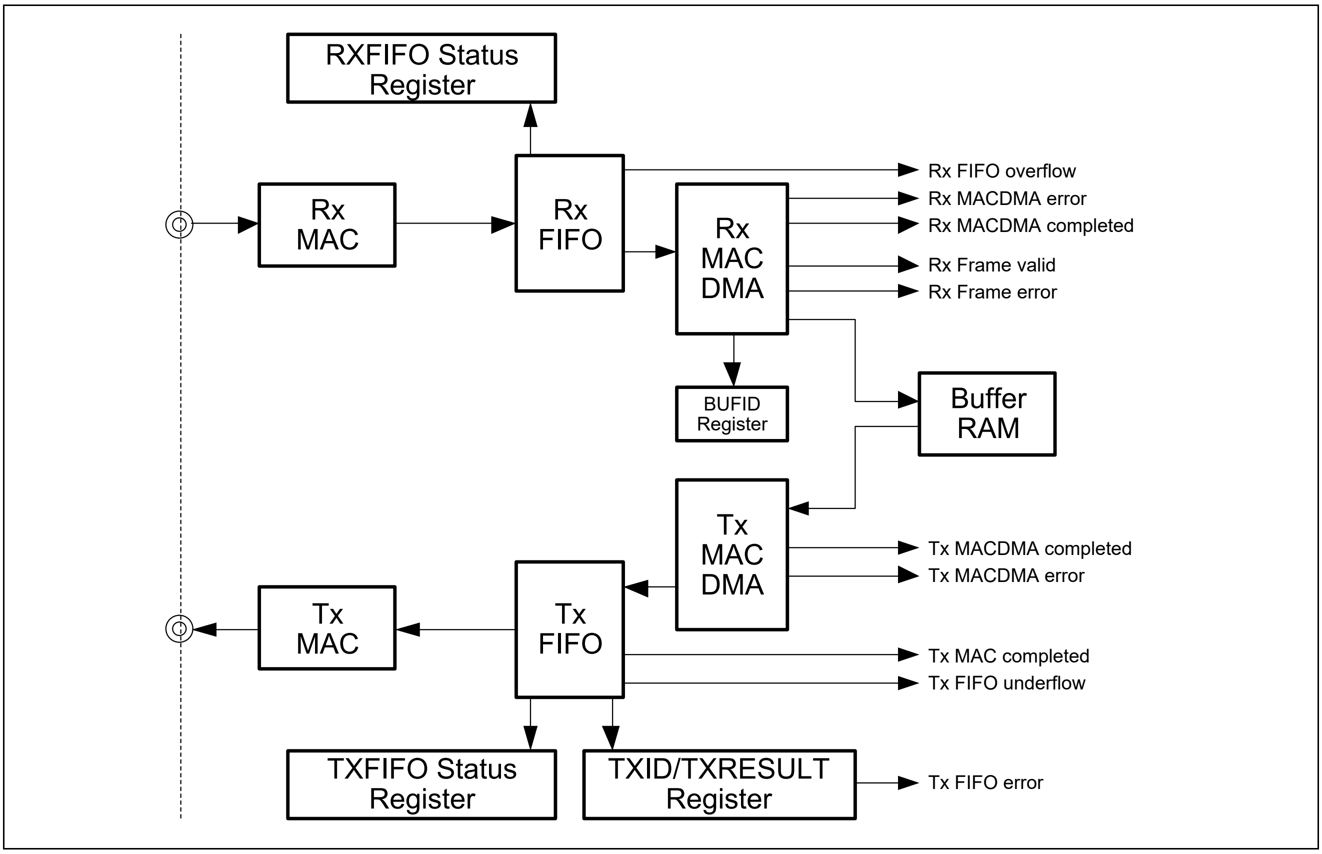


Figure 12.7 Block Diagram of the MACDMAC in Context and Interrupt Signals

(2) DMA for the Reception MAC

Figure 12.8 shows an outline of processing by the reception MACDMAC. A hardware function call (HWFNC_MACDMA_RX_Enable) must be issued to enable operation of the reception MACDMAC. The reception MACDMAC remains active until HWFNC_MACDMAC_RX_Disable is issued.

While active, the reception MACDMAC constantly monitors the state of the MAC RX FIFO. When the FIFO holds a received frame, the reception MACDMAC sends a request for the acquisition of a long (2048-byte) buffer to the buffer allocator. Once the long buffer has been acquired, the reception MACDMAC reads data from the MAC RX FIFO and writes the data sequentially from the start of the acquired long buffer.

After the completion of the full transfer of one frame, the reception MACDMAC writes the number of received words (one word: 32 bits) and the first logical address of the buffer to the BUFID register as information on reception. The information written to the BUFID is described in section 12.3.4.18, Receive Buffer Information Register (BUFID). The BUFID can be read by the CPU and is capable of holding up to 32 pieces of information.

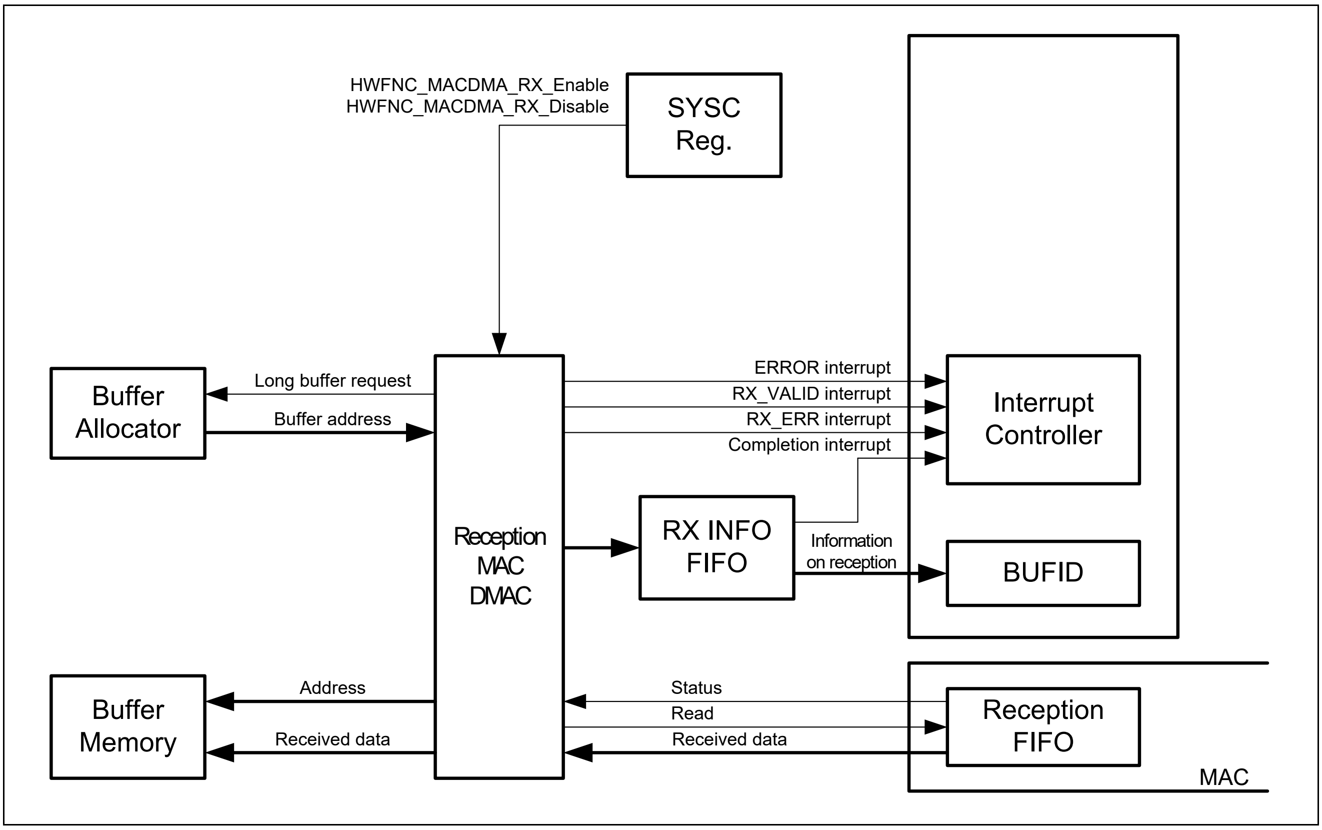


Figure 12.8 Outline of Processing by the Reception MACDMAC

(a) Description of the individual functions of the MAC DMA controller

- **Partial release of buffer space**

  The reception MACDMAC automatically releases an unused area that has no received data in the last buffer to have been acquired (buffer return function call). However, if the unused area is no larger than 128 bytes (one segment), buffer return does not proceed. Buffer return is a function call to release part of the secured buffer area and differs from the buffer release function call that releases the whole area of a secured buffer.

- **Full release of the buffer**

  If the following conditions are satisfied, the reception MACDMAC automatically releases the acquired buffer (calls the buffer release function).

  (1) The result of executing the function call for the buffer acquisition request was failure (the buffer has no unused area).

  (2) The result of analyzing the RX frame information is that the received frame is invalidated by HWFNC_MACDMA_RX_Control.

  (3) HWFNC_MACDMA_RX_Disable is executed under the following condition:
  - The number of received words is not greater than 4092 words

  In the above cases 1) and 2), all received frames are discarded and the buffer is released. In case 3), the received frames are not discarded (data resides in the MAC RX FIFO) but only the release of the buffer is executed, after which the reception MACDMAC is immediately disabled. In any of cases 1), 2), and 3), the result of reception is not written to the BUFID.

- **Generation of an error interrupt**

  An error interrupt is issued in response to detection of the reception MACDMAC having failed to continue operation for reception for some reason or data not having been received correctly. The source of an error interrupt can be checked by executing the hardware function call HWFNC_MACDMA_RX_Errstat.

  For details, see section 12.4.1.4(2)(c), List of hardware function calls.

- **Generation of reception completed interrupts**

  If the BUFID has information on the reception of one or more frames, the reception completed interrupt goes to its active level. The reception completed interrupt remains active as long as the BUFID register is not empty; that is, it has information on the reception of one or more frames.

  The reception completed interrupt is de-asserted when the BUFID is read and becomes empty.

- **Judging whether a received frame is valid or invalid.**

  Judgment of whether a received frame is valid or invalid leads to an RX_VALID (received frame normal) or RX_ERR (Ethernet reception frame error) interrupt being issued.
  Each interrupt has more than one source and the generation of interrupts is enabled for all sources in the initial state.
  A specified source can be disabled by executing HWFNC_MACDMA_RX_Control. The frame which corresponds to the disabled source is discarded by full release of the buffer.

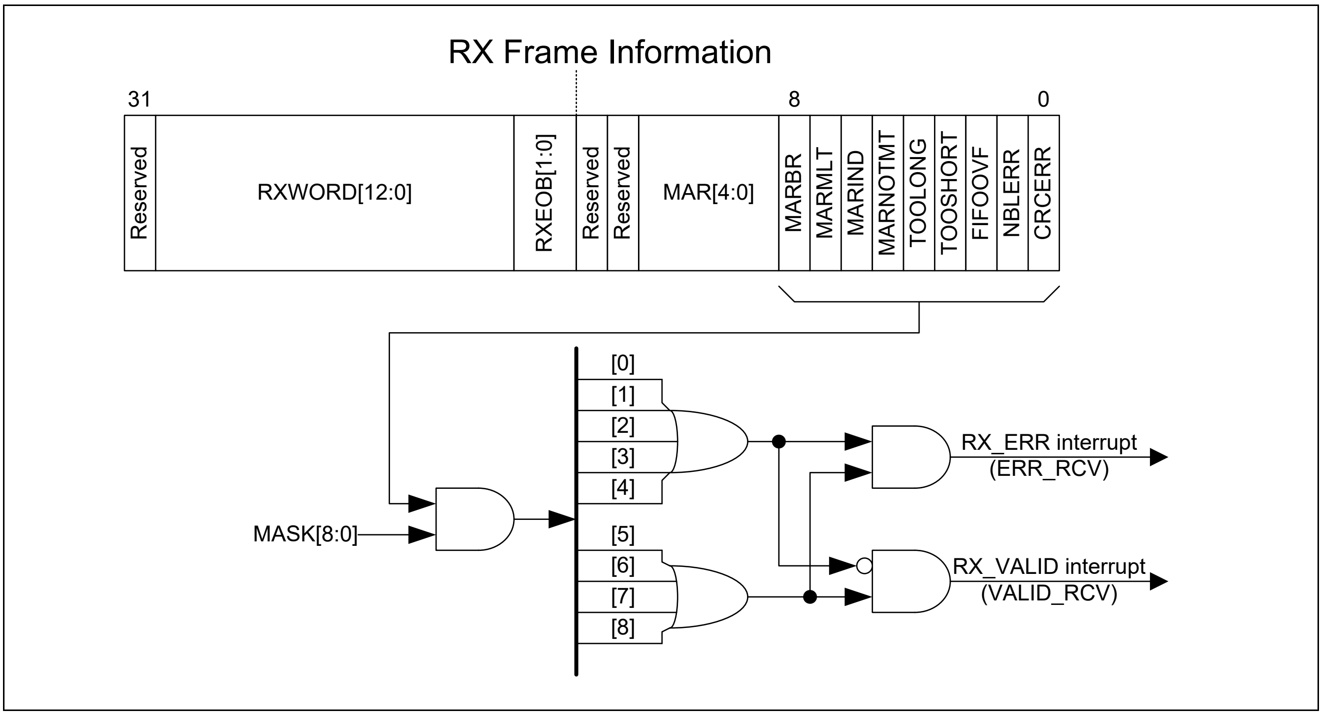


Figure 12.9 Conceptual Diagram of Judging Whether a Received Frame is Valid or Invalid

(b) Usage

- **Procedure for reading and releasing buffers**

  A buffer which has received data must always be released after use. An example of the procedure is given below.

  [Example of reading and releasing a buffer]

  (1) Read the BUFID register

  (2) Shift the bits [27:16] read from BUFID 16 bits to the right to obtain the number of received words.

  (3) The bits [15:0] read from the BUFID are bits [26:11] of the address where the acquired buffer starts. The individual bits of the address where the acquired buffer starts are configured as follows.

  [31:27]: 00001b

  [26:18]: Equivalent to the bits [15:7] in the BUFID

  [17:11]: Equivalent to the bits [6:0] in the BUFID

  [10: 0]: Always 0

  (4) After using the buffer, specify the start address as an argument and issue the buffer release function call to release the buffer.

- **Procedure for processing in response to an error interrupt**

  An example of the recommended procedure for processing in response to an error interrupt is given below. The value of R0[7:0] obtained by the HWFNC_MACDMA_RX_Errstat function call is hereafter called bits [7:0] of the result of reading the error state.

(1) Bit [3] of the result of reading the error state = 1 (a function call to forcibly end MACDMA RX has been executed)

a) If bit [0] of the result of reading the error state = 1, proceed to step (3).

b) If bits [2:0] of the result of reading the error state have the value 4 or 0, the interrupt source is the forced termination of reception while it was in progress and this does not represent a problem. Since the received frames are all discarded and the information is not written to the BUFID, nothing is done, so simply return to normal processing. The reception MAC FIFO may still have frame data that was received, but in such cases, the hardware automatically discards that data before the next round of reception starts.

(2) Bit [2] of the result of reading the error state = 1 (the size of the frame is at least 4096 words)

a) If bit [0] of the result of reading the error state = 1, proceed to step (3).

b) Received data are all stored. The start address is obtained by reading the BUFID.

c) Buffers that are no longer required are released according to the method described in "Procedure for reading and releasing buffers".

d) Return to normal processing.

(3) Bit [0] of the result of reading the error state = 1 (the remaining capacity of the buffer is insufficient)

a) If bit [2] of the result of reading the error state = 1 (the size of the received frame is at least 4096 words) is satisfied at the same time, the buffer capacity is considered temporarily insufficient, so nothing is done.

b) If the remaining capacity of the buffer is considered insufficient, the buffer is released to provide space.

c) Return to normal processing. Note that received frames may have been lost during this period.

(c) List of hardware function calls

The table below lists the hardware function calls.

If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.

If an error occurs while the hardware function call is running, an interrupt is generated.

Table 12.7 HWFNC_MACDMA_RX_Enable

Name	HWFNC_MACDMA_RX_Enable	
Function	Enables DMA for the reception MAC, that is, the transfer of data to the buffer memory from the MAC. As long as the reception DMAC is enabled, transfer starts automatically whenever the FIFO buffer within the MAC collects received frames. Since the DMAC executes Get Buffer at this time, the buffer memory is automatically acquired.	
Command register		
SYSC[15:0]	0x5101	
Argument registers		
R4[31:0]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Reserved	Always 0
Return value registers		
R0[0]	Result	0: Success 1: Invalid system call*
R0[28:1]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

Note: If this hardware function is called while it is not disabled (this function call is already being executed) or this hardware function is called while a buffer return or release operation is in progress after reception has been suspended, the result is an invalid system call.

Caution: The number of bytes to be transferred at a time is from 4 to 2048 bytes. Exceeding this range leads to the generation of an exception.

Table 12.8 HWFNC_MACDMA_RX_Disable

Name	HWFNC_MACDMA_RX_Disable	
Function	Disables DMA for the reception MAC. When forced reset is enabled, the data being received are discarded and information on reception is not stored in the BUFID register. At this time, the buffer is automatically released. When forced reset is disabled, the buffer is not automatically released.	
Command register		
SYSC[15:0]	0x5102	
Argument registers		
R4[0]	Forced reset	0: This function is disabled while reception is in progress. 1: If the reception DMAC is enabled, it is disabled even if reception is in progress (the reception DMAC is forcibly reset). Nothing is done if the reception DMAC is already disabled.
R4[31:1]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[0]	Result when R4[0] = 0	2’b00: Success 2’b01: Invalid system call (the buffer is in use or reception is suspended) 2’b10: The function cannot be disabled since reception is in progress. 2’b11: The function has already been disabled.
	Result when R4[0] = 1	2’b00: Success 2’b01: Invalid system call (the buffer is in use or reception is suspended)
R0[28:1]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

Table 12.9 HWFNC_MACDMA_RX_Control

Name	HWFNC_MACDMA_RX_Control	
Function	Controls enabling or disabling of the interrupt source corresponding to bits [8:0] of the received frame information.	
Command register		
SYSC[15:0]	0x510b	
Argument registers		
R4[8:0]	Interrupt source	Controls enabling or disabling of the interrupt source corresponding to each bit. 0: Interrupts disabled 1: Interrupts enabled (initial value)
R4[31:9]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[0]	Result	0: Success 1: Invalid system call
R0[28:1]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

Table 12.10 HWFNC_MACDMA_RX_Errstat

Name	HWFNC_MACDMA_RX_Errstat	
Function	Obtains error interrupt sources for the reception MACDMAC.	
Command register		
SYSC[15:0]	0x510d	
Argument registers		
R4[31:0]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[3:0]	Result	[0]: Buffer Get fails [1]: Always 0 [2]: The RX data size is over 4096 words (16 KB). [3]: HWFNC_MACDMA_RX_Disable is issued when forced reset is enabled.
R0[28:4]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

(3) DMA for the transmission MAC

(a) Usage

The transmission MACDMAC uses descriptors. The descriptors are located in the buffer memory. That is, the software must acquire a buffer for the descriptor by issuing a hardware function call before DMA can be set up. This buffer can be long or short. One buffer can hold multiple descriptors.

A transmission descriptor is shown in detail in Figure 12.10. Note that a descriptor must start on a 64-bit boundary. If it is not on a 64-bit boundary, the result of trying to use it is in an invalid system call.

A descriptor is formed in a succession of a 32-bit address and a 32-bit transfer byte count. Address 0xFFFF FFFF indicates the end of a descriptor. The address field of a descriptor indicates the transmission start address, and the byte count indicates the number of bytes to be transmitted from that address. The DMAC reads the first pair of address and byte count in a descriptor, and then writes the specified data to the transmission MAC FIFO. After that, the DMAC reads the next pair of address and byte count, and then writes the specified data to the transmission MAC FIFO. The DMAC continues this processing until it reads the end of the descriptor (0xFFFF FFFF).

Source start addresses in the descriptor can be specified in units of bytes. The size of data to be transmitted can be specified in units of bytes. If the data writing point in the transmission FIFO is not at a word boundary, the DMAC automatically inserts padding.

The transmission MACDMAC starts when a function call HWFNC_MACDMA_TX_Start is issued. When this function call is issued, the start address of the transmit descriptor must be specified in the argument register R4.

Note that if the address field is not 0xFFFF FFFF and 0 is specified in the descriptor byte count field, the DMAC ignores the address field and does not perform transmission. In this case, the DMAC reads the next descriptor.

If the value of an address field is incorrect (for example, the address is outside the buffer area) or the number of transfer bytes is incorrect (for example, continued access causes a buffer area overflow), a MACDMA transmission error interrupt occurs.

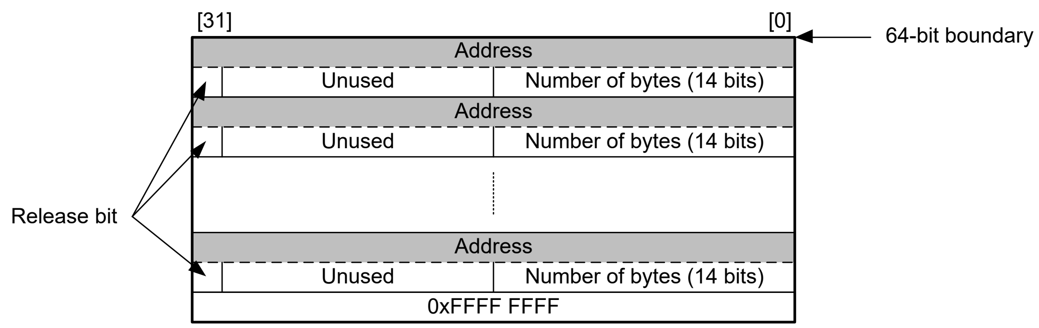


Figure 12.10 Transmission Descriptor

(b) Automatic release of the buffer

If the release bit of the transmission descriptor is 0, no buffer is released. If the release bit is 1, the transmission MACDMAC uses a buffer release function call to automatically release a buffer from the buffer area whose start address is indicated by the relevant descriptor after the completion of transmission.

(c) Example of operation

Figure 12.11 shows an example of operation for transmission by combining multiple buffers for use by the transmission MACDMAC.

Two independent buffers of buffer 1 and buffer 2 are combined for transmission by the transmission MACDMAC by allocating transmission descriptors at the consecutive 64-bit boundary addresses. The area labelled "Unused" means that the data end before the end of the segment (that is, it does not end at the 128-byte boundary). In transfer, the address where the data in a buffer start need not necessarily be the start of the buffer.

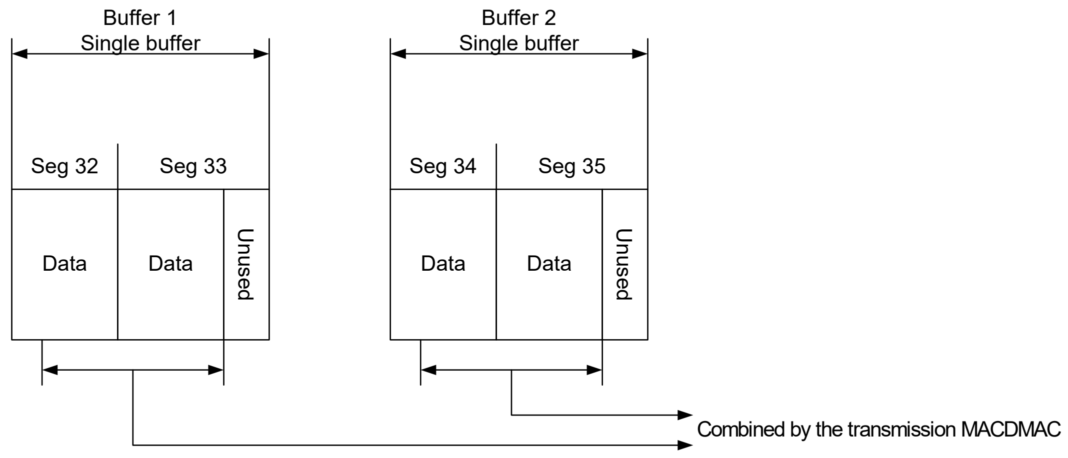


Figure 12.11 Example of Transmission as One Frame by Combining Multiple Buffers

(d) List of hardware function calls

The table below lists the hardware function calls.

If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.

If an error occurs while the hardware function call is running, an interrupt is generated.

Table 12.11 HWFNC_MACDMA_TX_Start

Name	HWFNC_MACDMA_TX_Start	
Function	Transfers data from the buffer memory to the Ethernet MAC. The address where the transmission descriptor starts is set in R4. When transfer ends, an interrupt is generated. The number of bytes to be transferred at a time is from 1 to 2048 bytes.	
Command register		
SYSC[15:0]	0x5100	
Argument registers		
R4[31:0]	Address of the descriptor	Address of the transmission descriptor
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[6:0]	Reserved	Always 0
R7[31:7]	Unused	
Return value registers		
R0[1:0]	Result	0: Success 1: Invalid system call
R0[28:2]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

Table 12.12 HWFNC_MACDMA_TX_Errstat

Name	HWFNC_MACDMA_TX_Errstat	
Function	Obtains error interrupt sources for the transmission MACDMAC.	
Command register		
SYSC[15:0]	0x510C	
Argument registers		
R4[31:0]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	[0]: Memory Access Violation • Access to the buffer that is not acquired • The number of transfer bytes is not correct • The start address of the descriptor is not on a 64-bit boundary. [1]: Memory Access Timeout • The start address of a transmission descriptor turns to be an end value (FFFF FFFFh) • Releasing the buffer automatically is failed
R0[28:2]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

12.4.1.5 Buffer RAM DMA Controller

(1) Functional overview

The buffer RAM DMA controller transfers data between the buffer RAM and data RAM or the buffer RAM and buffer RAM. It is used to transfer data for transmission by the MAC DMAC to the buffer RAM and to transfer data received by the MAC DMAC to the data RAM.

(2) DMA transfer

Control of the buffer RAM DMA controller for each form of transfer is described below.

(a) Transfer between the buffer RAM and the data RAM

Calling the HWFNC_Direct_Memory_Transfer hardware function starts transfer between the buffer RAM and data RAM. After calling the function, confirm its completion by reading bit 29 of the R0 register. At this time, DMA transfer has been completed.

(b) Replacing data in the buffer RAM or data RAM

By executing the hardware function HWFNC_Direct_Memory_Replace, an area in the buffer RAM or data RAM can be overwritten by a desired 32-bit data pattern.

The start and end of the area to be written must be on 128-bit boundaries so the amount of data written must be a multiple of 128 bits. After calling the function, confirm its completion by reading bit 29 of the R0 register. At this time, writing of the data pattern has been completed.

(c) Transfer between the buffer RAMs

By executing the hardware function HWFNC_INTBUFF_DMA_Start or HWFNC_INTBUFF_DMA_Start (descriptor), data can be transferred from the buffer RAM to the buffer RAM. After calling the function, confirm its completion by reading bit 29 of the R0 register. However, DMA transfer has not been completed at this time. Check the completion of DMA transfer by means of the InterBuffer DMA transfer complete interrupt.

(d) List of hardware function calls

The table below lists the hardware function calls.

If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.

Access to an access-prohibited area (an area other than the buffer RAM, etc.) while the hardware function call is running leads HWFNC_Direct_Memory_Transfer and HWFNC_Direct_Memory_Replace to return an exception to the return value register R0, whereas it leads HWFNC_INTBUFF_DMA_Start and HWFNC_INTBUFF_DMA_Start (Descriptor) to generate an interrupt.

Table 12.13 HWFNC_Direct_Memory_Transfer

Name	HWFNC_Direct_Memory_Transfer	
Function	Transfers data from the data RAM to the buffer RAM or from the buffer RAM to the data RAM. Data cannot be transferred from the buffer RAM to the buffer RAM. For transfer from the buffer RAM to the buffer RAM, use HWFNC_INTBUFF_DMA_Start (data transfer between the data RAMs is possible).	
Command register		
SYSC[15:0]	0x5211	
Argument registers		
R4[31:0]	Address where the source area for transfer starts	Specifies the address where the source area for transfer starts.
R5[31:0]	Address where the destination area for transfer starts	Specifies the address where the destination area for transfer starts.
R6[31:0]	Number of bytes for transfer	Specifies the number of bytes for transfer.
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	2'b00: Success 2'b01: Invalid system call (transfer between the buffer RAMs has been specified) 2'b10: An exception has occurred.
R0[28:2]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Address where the exception occurred	When an exception occurred, this is the address where it occurred. In other cases, all 0s.

Table 12.14 HWFNC_Direct_Memory_Replace

Name	HWFNC_Direct_Memory_Replace	
Function	Replaces the specified memory area in the data RAM or buffer RAM with a defined data pattern. The number of words to be written must be at least four. (A word unit is 32 bits)	
Command register		
SYSC[15:0]	0x5212	
Argument registers		
R4[31:0]	Pattern	Specifies the data pattern for writing.
R5[31:0]	Start address	Specifies the address where the destination area for writing starts.
R6[31:0]	Number of words	Specifies the number of words to be written.
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	2'b00: Success 2'b01: Invalid system call The set address was specified in byte units or the setting for the number of words to be transferred is three or fewer. 2'b10: An exception has occurred.
R0[28:2]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Address where the exception occurred	When an exception has occurred, this is the address where it occurred. In other cases, all 0s.

Table 12.15 HWFNC_INTBUFF_DMA_Start

Name	HWFNC_INTBUFF_DMA_Start	
Function	Transfers data in the buffer memory. The address where the source area for transfer starts is set in R4, the address where the destination area for transfer starts is set in R5, and the number of bytes for transfer is set in R6. When transfer ends, an interrupt is generated.	
Command register		
SYSC[15:0]	0x5104	
Argument registers		
R4[31:0]	Address where the source area for transfer starts	Specifies the address where the source area for transfer starts.
R5[31:0]	Address where the destination area for transfer starts	Specifies the address where the destination area for transfer starts.
R6[15:0]	Number of bytes for transfer	Specifies the number of bytes for transfer.
R6[31:16]	Unused	
R7[6:0]	Reserved	Always 0
R7[31:8]	Unused	
Return value registers		
R0[0]	Result	0: Success 1: Invalid system call
R0[28:1]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

Table 12.16 HWFNC_INTBUFF_DMA_Start (Descriptor)

Name	HWFNC_INTBUFF_DMA_Start (descriptor)	
Function	Transfers data in the buffer memory. When transfer ends, an interrupt is generated. This function gives a descriptor instead of an address and size as an argument.	
Command register		
SYSC[15:0]	0x5114	
Argument registers		
R4[31:0]	Address where the transfer source descriptor starts	Specifies the address where the transfer source descriptor starts.
R5[31:0]	Address where the transfer destination descriptor starts	Specifies the address where the transfer destination descriptor starts.
R6[31:0]	Unused	
R7[6:0]	Reserved	Always 0
R7[31:8]	Unused	
Return value registers		
R0[0]	Result	0: Success 1: Invalid system call
R0[28:1]	Unused	All 0s
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0s
R1[31:0]	Unused	All 0s

Cautions 1. The structure of the descriptor is the same as for the MACDMAC, but the function does not automatically release the buffer.

2. When specifying the size in the descriptor, the transfer source descriptor is given priority. When the sizes specified for the source and destination differ, the operation is as follows.

Specification of the size of the transfer source descriptor < Specification of the size of the transfer destination descriptor -> No problem

Specification of the size of the transfer source descriptor > Specification of the size of the transfer destination descriptor -> An exception may occur.

12.4.2 Interrupts

The interrupts that the Gigabit Ethernet MAC generates are described below.

Table 12.17 Interrupts Related to Operations for Transmission

Interrupt Name	Symbol	Conditions for Asserting and De-asserting Interrupts
TX FIFO underflow interrupt	INTETHTXFIFO	This interrupt is generated when the transmission size specified in the descriptor and transmission frame control information differ. At this time, transmission does not proceed. Modify the settings of the descriptor or the transmission frame information for retransmission. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
TX-FIFO error interrupt	INTETHTXFIFOERR	This interrupt is generated when information is further updated while the GMAC_TXID/GMAC_TXRESULT register is holding the maximum number of items of information (four). Take care, since the oldest of the retained information will have been overwritten when this error occurs. Reading the GMAC_TXID/GMAC_TXRESULT register until the value of the GMAC_TXFIFO.TRBFR bit becomes 0 leads to clearing of the retained information and restoring normal operation.
MACDMA transmission error interrupt	INTETHTXDERR	This interrupt is generated, when an error occurs while the transmission MAC DMA is operating. As there are several error sources, HWFNC_MACDMA_TX_Errstat is used to obtain the error source. Modify the settings of the transmission descriptor for retransmission. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
Ethernet MACDMA transmission complete interrupt	INTETHTXDMA	This interrupt is generated when DMA transfer from the buffer RAM to the transmission MAC FIFO is completed. At this time, DMA transfer has been completed but operations for communications by the MAC are not. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
Ethernet transmission complete interrupt	INTETHTXCMP	This interrupt occurs when operations for communications by the transmission MAC are completed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.

Table 12.18 Interrupts Related to Operations for Reception

Interrupt Name	Symbol	Conditions for Asserting and De-asserting Interrupts
Ethernet MACDMA reception complete interrupt	INTETHRXDMA	This interrupt is generated when operations by the reception MACDMAC end normally. It remains active until the BUFID register becomes empty of information on reception. The interrupt source is de-asserted when the BUFID is read and becomes empty.
MACDMA reception error interrupt	INTETHRXDERR	This interrupt indicates that an error has occurred while the reception MACDMAC was operating. There is more than one source for this error, and the precise source is obtained by issuing HWFNC_MACDMA_RX_Errstat. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
Received frame normal interrupt	INTMACDMARXFRM	This interrupt is generated when operations by the reception MACDMAC end normally and the received frame is normal. The interrupt source can be specified by referring to information on the received frame. It remains active until the BUFID register becomes empty of information on reception. The interrupt source is de-asserted when the BUFID is read and becomes empty.
Ethernet reception frame error interrupt	INTETHRXERR	This interrupt is generated when operations by the reception MACDMAC end normally and the received frame has an error. The interrupt source can be specified by referring to information on the received frame. It remains active until the BUFID register becomes empty of information on reception. The interrupt source is de-asserted when the BUFID is read and becomes empty.
RX FIFO overflow interrupt	INTETHRXFIFO	This interrupt is generated when data are received while the buffer does not have enough space, so the RX FIFO overflows. When this error occurs, received data may already have been discarded. Restore the system to the state where reception is possible by releasing the buffer, etc. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.

Table 12.19 Interrupts Related to Other Operations

Interrupt Name	Symbol	Conditions for Asserting and De-asserting Interrupts
Ethernet MII management access complete interrupt	INTETHMII	This interrupt is generated when reading from or writing to the MII management bus is completed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
Ethernet pause packet transmission complete interrupt	INTETHPAUSE	This interrupt is generated when the transmission of a pause packet is completed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
InterBuffer DMA transfer complete interrupt	INTBUFDMA	This interrupt is generated if DMA transfer between buffer RAMs is completed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
InterBuffer DMA transfer error interrupt	INTBUFDMAERR	This interrupt is generated if DMA access reaches to unassigned buffer area during transfer between buffer RAMs. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
Buffer RAM area access error	INTBRAMERR	This interrupt is generated, if the buffer that is not acquired by the CPU is accessed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.

12.4.3 Transmitting Ethernet Frames

This section explains processing for transmission of Ethernet frames. The Gigabit Ethernet MAC handles transmission according to the following flow.

1. Make initial settings (→ 12.4.1.1).
2. Acquire a TX buffer (→ 12.4.3.1).
3. Create TX frame control information (→ 12.4.3.2(1)).
4. Create Ethernet transmit frame data (→ 12.4.3.2(2)).
5. Create TX descriptors (→ 12.4.3.3).
6. Execute the DMAC activation command (→ 12.4.3.4).
7. The DMAC transfers data to the FIFO buffer in the MAC according to TX descriptors.
8. The MAC starts transmitting Ethernet frames according to the TX frame control information included in the transfer data.
9. The TX-completed interrupt occurs.
10. Processing for checking the status, etc. after the completion of the transmission (→ 12.4.3.5).
11. Release the TX buffer (optional)

The details of the above steps are described below.

12.4.3.1 Acquiring a Transmit Buffer

Set the hardware function call register as follows to acquire a transmit buffer.

Register	Value
SYSC	0x5000
R4	Size of the memory block to be secured (1 to 2048 bytes)
R5	0 (Unused)
R6	0 (Unused)
R7	0 (Unused)

In addition, the hardware function returns the value returned as follows.

Register	Value
R0	2'b0x and R0[29] = 1: Success 2'b10: Invalid system call 2'b11: The buffer is insufficient.
R1	Address where the secured memory block starts

12.4.3.2 Creating TX Data

Figure 12.12 shows the TX data format. A transmission descriptor points the start address of this frame.

In the Gigabit Ethernet MAC, the size of transmission frames and various controls are directed by appending 64-bit TX frame control information before the normal Ethernet frame data.

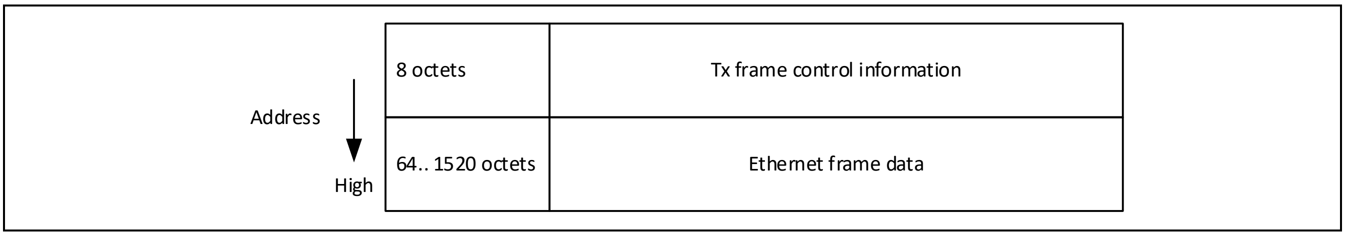


Figure 12.12 TX Data Format

Caution: Make sure that the TX data conforms to this format.

(1) TX frame control information

The table below describes each field of TX frame control information.

TX frame control information	31	30 – 18	17 – 16	15 – 11	10	9	8	7	6	5	4	3	2	1 – 0	
	Reserved	TX_WORD[12:0]	TX_EOB[1:0]	Reserved	Port1	Port0	Forced Forwarding	Transmit Timestamp	Reserved	TCPIP ACC OFF	ITAG	ICRC	APAD	Reserved	0
	Frame ID [31:0]														1

Figure 12.13 Example of TX Frame Control Information

Field Name	Description
TX_WORD[12:0]	The number of words of the Ethernet frame for transmission (a word unit is 32 bits).The number of valid bytes in the last word is directed by using TX_EOB[1:0].
TX_EOB[1:0]	Octet up to which the last word in this frame is valid. 00: 1 byte is valid. 01: 2 bytes are valid. 10: 3 bytes are valid. 11: 4 bytes are valid.
Port 1*1	Port 1 is used to enable forced forwarding of the Ethernet switch.
Port 0*1	Port 0 is used to enable forced forwarding of the Ethernet switch.
Forced Forwarding*1	Enables forced forwarding of the Ethernet switch When this function is enabled, a frame is output from the specified port regardless of the setting of the switch filter.
Transmit Timestamp*1	Enables timestamping of transmission frames when the Ethernet switch is in use.
TCPIP ACC OFF*2	1: Disables the TCPIP accelerator. 0: Enables the TCPIP accelerator
ITAG	Indicates that this frame has a VLAN Tag.
ICRC	Indicates that this frame already has a CRC attached to it. The APAD field is ignored if this bit is set.
APAD	Indicates that the frame is automatically padded if its length is shorter than 64 octets.
Frame ID[31:0]	An optional frame identifier is designated.

Note 1: These function are only available when insertion of a management tag is permitted by the Ethernet switch management TAG control register (ETHSWMTC). If insertion of a management tag is disabled, these fields are not valid.

2: Disable the TCPIP accelerator if the following frames are sent;

- **IPv6 frames without UDP or TCP packet**
- **IEEE802.3 + IEEE802.2 (LLC) frames**

In cases where TX_WORD [12:0] and TX_EOB [1:0] are combined into TX_LENGH [14:0] (15 bits), TX_LENGH [14:0] can be calculated from the following formula based on the Ethernet frame size (in bytes):

TCPIPACC Pad Size is 2 when TX TCPIPACC is enabled (GMAC_ACC.TTCPIPEN = 1) and 0 when it is disabled.

TX_LENGH [14:0] = (TX frame size – TCPIPACC Pad Size + 3) (bytes)

(2) Ethernet frame

The transmission Ethernet frame data format and the description of the fields are given below.

Field name	Description
Destination MAC Address	MAC address of the destination
Source MAC Address	MAC address of the source
Type / Length	Ethernet Type or length
VLAN Tag	Tag Protocol Identifier. This field is available if VLAN Tag is included.
VLAN Info	Tag Control Information. This field is available if VLAN Tag is included.
Frame Payload	Payload

(a) When TX TCPIP accelerator is enabled

If the TX TCPIP accelerator function is enabled (GMAC_ACC.TTCPIPEN = 1), Ethernet frame data requires 2-byte padding between the Type/Length field and Payload.

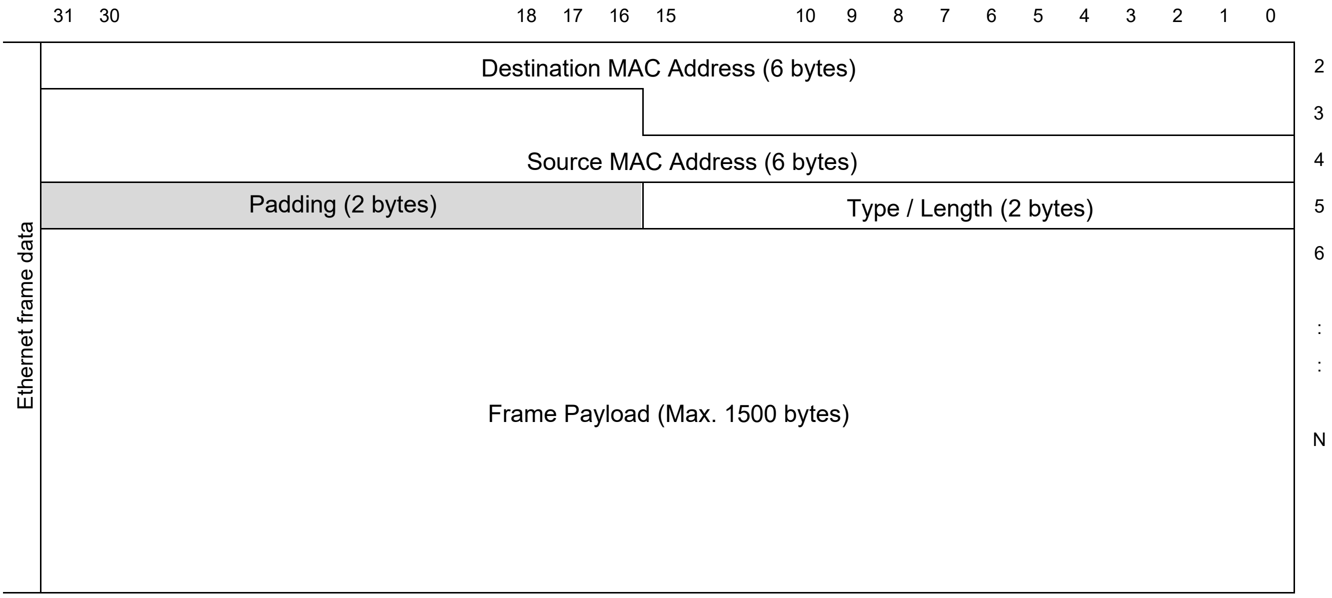


Figure 12.14 TX Ethernet Frame Data Format – TCPIPACC is enabled, without VLAN Tag

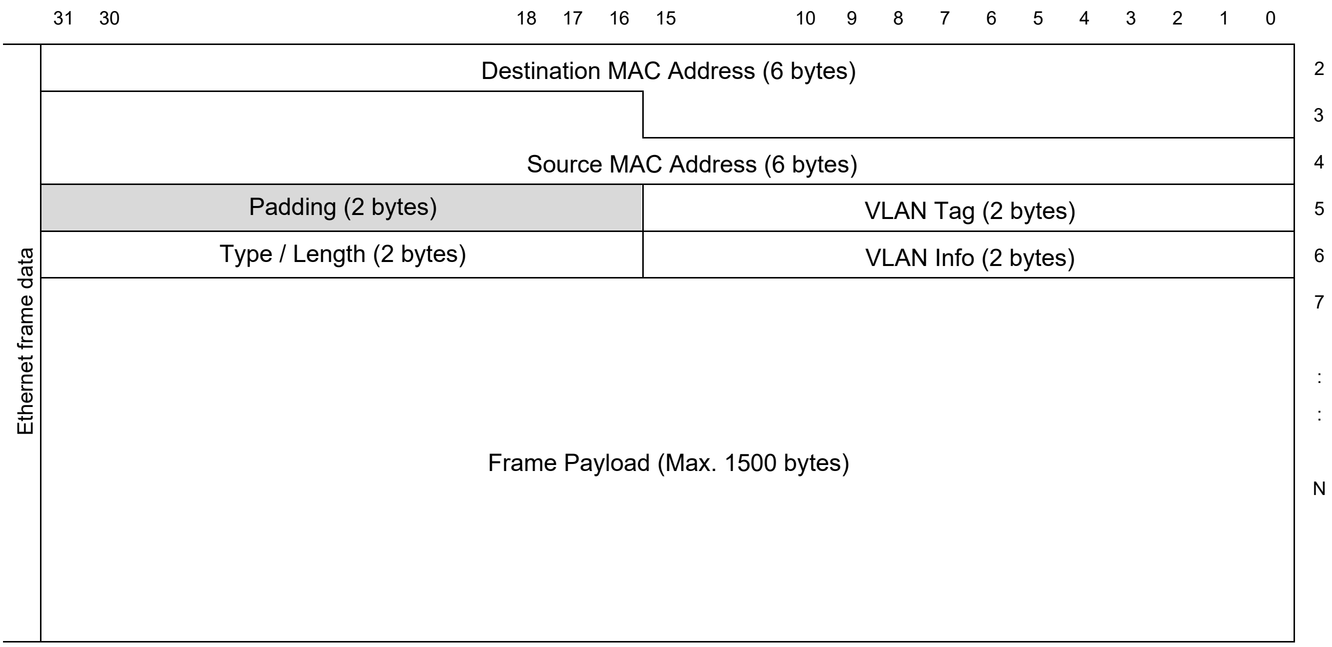


Figure 12.15 TX Ethernet Frame Data Format – TCPIPACC is enabled, with VLAN Tag

Caution: Padding (2 bytes) can be by any value.
Padding (2 bytes) is not included in the specified size of Ethernet frames (TX_WORD[12:0], TX_EOB[1:0]).

(b) When TX TCPIP accelerator is disabled

The Ethernet frame data formats when the TX TCPIP accelerator function is disabled (GMAC_ACC.TTCPIPEN = 0) are shown below.

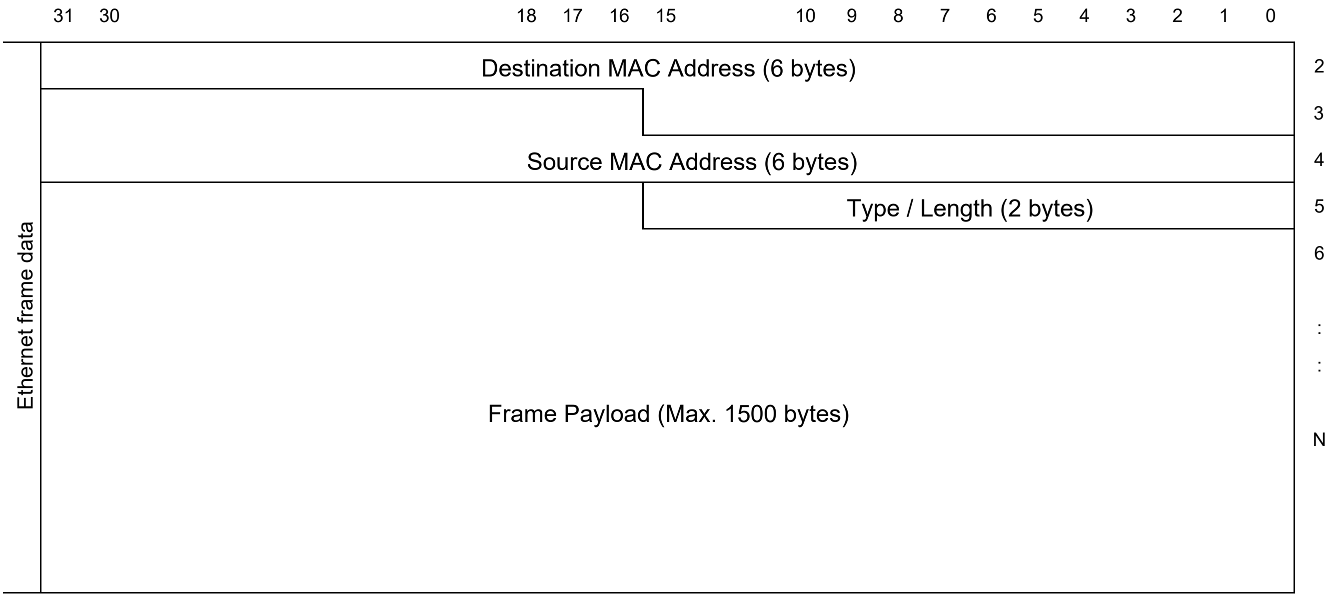


Figure 12.16 TX Ethernet Frame Data Format – TCPIPACC is disabled, without VLAN Tag

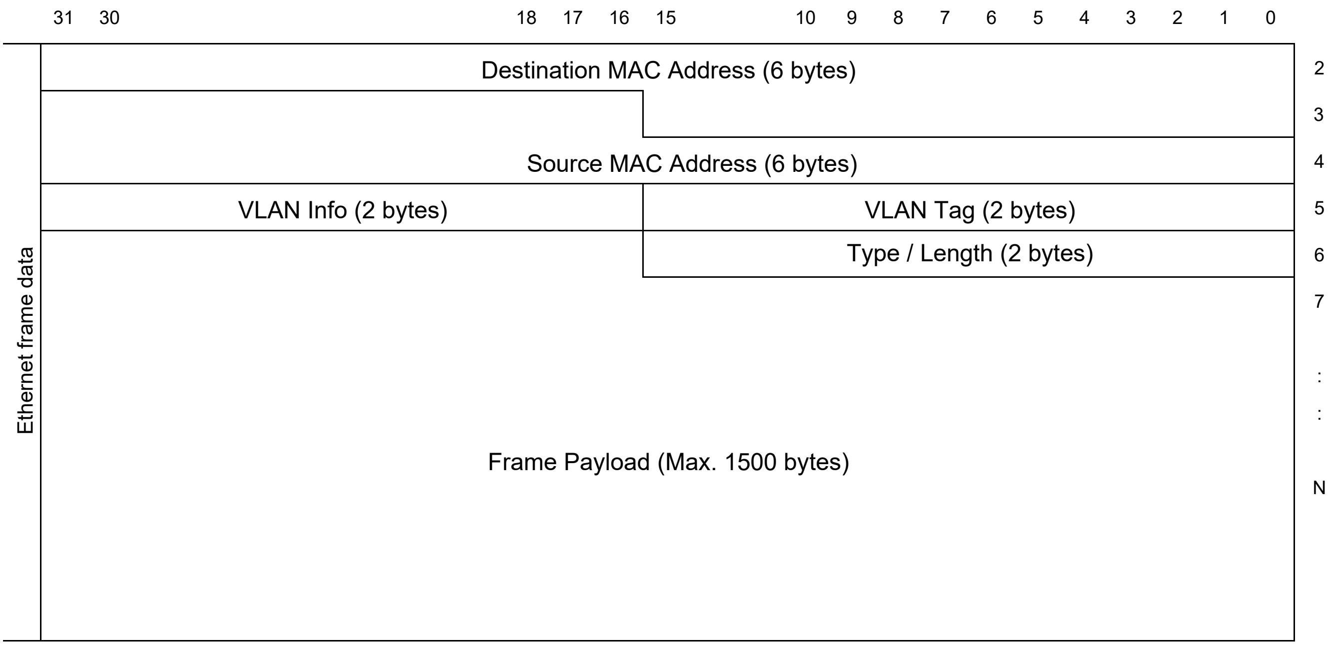


Figure 12.17 TX Ethernet Frame Data Format – TCPIPACC is disabled, with VLAN Tag

12.4.3.3 Creating TX Descriptors

The transmission MAC DMA controller uses the following descriptors.
After creating descriptors, activate the DMAC to start processing for transmission.

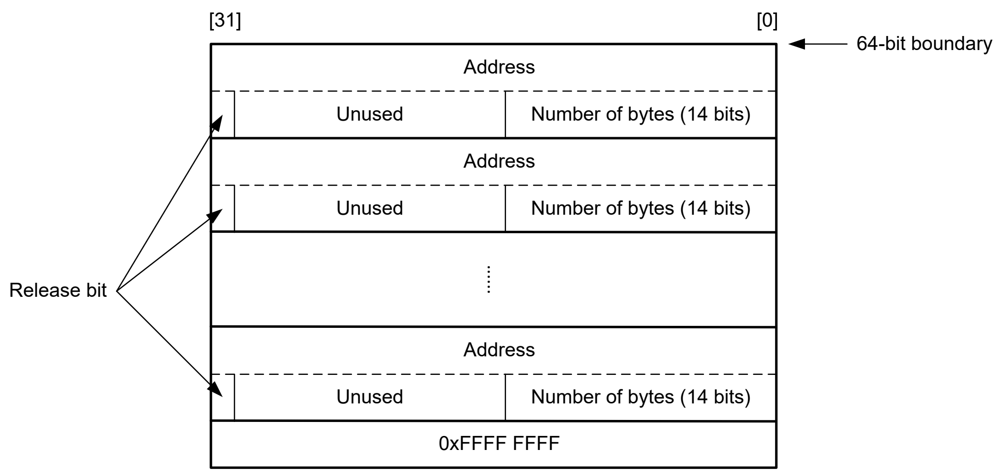


Figure 12.18 Structure of the TX Descriptor

A descriptor must start on a 64-bit boundary ([2:0] = 0). If it is not on a 64-bit boundary, an error code is returned in the return value register R0.

In the descriptor, the addresses and number of bytes for transfer are written to consecutive 32-bit segments. The address 0xFFFF FFFF indicates the end of the descriptors. The address field of a descriptor indicates the start address for transmission, and the number of bytes indicates how many bytes to forward from its address. The DMAC reads the address at the start of the descriptor and reads the number of bytes, which follows the address, and writes the designated data to the transmission MAC FIFO. Next, the DMAC reads the address written in the next descriptor and the number of bytes, which follows the address, and writes the designated data to the transmission MAC FIFO. This is repeated until 0xFFFF FFFF is read (indicating the end of the descriptors).

Addresses in descriptors (addresses where the source areas for transfer start) can be specified in byte units. The amount of data for transfer can be specified in byte units per transfer. If an address written to the transmission FIFO is not on a word boundary, the DMAC transfers data while aligning the data automatically.

The transmission MACDMA starts by issuing "start of transmission operation" as a hardware function call. The address where the transmission descriptor is to start must be specified in the R4 register when this function call is issued.

Furthermore, when the address field is not 0xFFFF FFFF and 0 is specified in the field for the number of bytes (14 bits) in the descriptor, that address field is ignored and is not forwarded. After that, the next descriptor will be read.

If the address field is invalid (outside the buffer area, etc.) or the number of bytes for transfer is invalid (such as leading to the buffer area overflowing during access), an error interrupt is generated.

When a release bit is 1, the buffer area which starts at the address indicated by the descriptor is automatically released (Buffer Release Function Call) by the transmission MACDMA after the end of transmission. When a release bit is 0, the buffer is not released.

12.4.3.4 Starting Transmission

The transmission DMAC is activated and transmission starts by setting a hardware function call register as follows.

Register	Value
SYSC	0x5100
R4	TX descriptor address
R5	0 (Unused)
R6	0 (Unused)
R7	Must be 0

In addition, the hardware function returns the value returned as follows.

Register	Value
R0	0: Success 1: Error (invalid calling)
R1	Fixed to 0

12.4.3.5 Completion of Transmission

The Ethernet MACDMA transmission complete interrupt occurs when DMA transfer has been completed, and the Ethernet transmission complete interrupt occurs when MAC transmission has been completed.

If the TX buffer which is already acquired is to be reused for the next transmission, acquisition of the TX buffer is not required.

12.4.4 Receiving Ethernet Frames

This section explains processing for reception of Ethernet frames. The Gigabit Ethernet MAC handles processing for reception according to the following flow.

1. Initial settings (→ 12.4.1.1)
2. Enabling the RX MAC (→ 12.4.4.1)
3. Activating the RX DMAC (→ 12.4.4.2)
4. Receiving a frame and acquiring the buffer (→ 12.4.4.3)
5. The reception completed interrupt occurs.
6. Acquiring the RX buffer information (→ 12.4.4.4)
7. Checking the status of frames (→ 12.4.4.5(1))
8. Acquiring the Ethernet frame data (→ 12.4.4.5(2))
9. Releasing the RX buffer

12.4.4.1 Enabling the RX MAC

Set 1 to the reception enable register (GMAC_RXMAC_ENA → 12.3.4.15) to enable the reception MAC.

12.4.4.2 Activating the RX DMAC

The reception DMA controller is activated by setting the hardware function call registers as follows.

Register	Value
SYSC	0x5101
R4	0 (Unused)
R5	0 (Unused)
R6	0 (Unused)
R7	Must be 0

In addition, the hardware function returns the value returned as follows.

Register	Value
R0	0: Success 1: Error (Invalid call)
R1	Fixed to 0

12.4.4.3 Receiving a Frame and Acquiring the Buffer

When a frame was received, the RX buffer is automatically acquired by hardware.

12.4.4.4 Acquiring the RX Buffer Information

After the completion of reception has been detected in response to the reception-completed interrupt, etc, read the RX buffer information register (BUFID) to acquire the address and size of the buffer which holds the received data.

After the address information has been acquired, read the buffer which holds data and acquire the RX frame information and Ethernet frame data. Refer to section 12.4.4.5 for the format of received data.

12.4.4.5 RX Data Format

In the reception of frames by the Gigabit Ethernet MAC, 64 bits of received frame information will be appended after the frame data. This information indicates the state of reception: size of the Ethernet frame, errors, and so on.

Since the received frame information starts on a 64-bit boundary, the amount of padding following the Ethernet frame varies with the frame size.

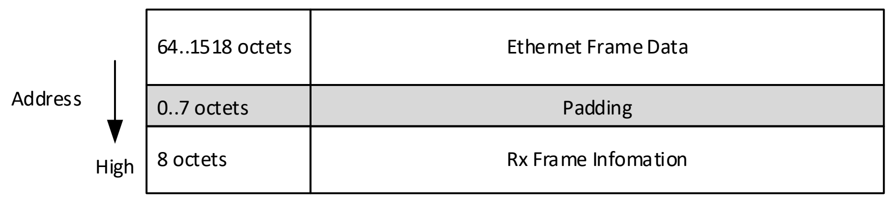


Figure 12.19 RX Data Format

(1) RX frame information

The descriptions of the fields of the received frame information are given below.

RX frame information (word address: N ~ (N+2))

Bits	Upper word	Bits	Lower word
31	Reserved (always 0)	31–16	SESSION_ID[15:0]
30–18	RX_WORD[12:0]	15	Reserved (always 0)
17–16	RX_EOB[1:0]	14	Reserved (always 0)
15	Reserved (always 0)	13	Reserved (always 0)
14	Reserved (always 0)	12	Reserved (always 0)
13–9	MAR[4:0]	11	Reserved (always 0)
8	MARBR	10–8	MARSTAT[2:0]
7	MARMLT	7	IPNG
6	MARIND	6	TCPNG
5	MARNOTMT	5	IPV6NG
4	TOOLONG	4	OUT_OF_LIST
3	TOOSHOTR	3	TYPEIP
2	FIFOOVF	2	MAACL
1	NBLERR	1	PPPOE
0	CRCERR	0	VTAG

Figure 12.20 RX frame information

Field Name	Description
SESSION_ID[15:0]	1: Session ID of PPPoE session stage
MARSTAT[2:0]	MARSTAT[2]: 1: Broadcast address MARSTAT[1]: 1: Multicast address MARSTAT[0]: 1: Individual address
IPNG*2	1: The checksum of the IPv4 Header does not match the calculation result of the TCPIP accelerator.
TCPNG*2	1: The checksum of the TCP or UDP header does not match the calculation result of the TCPIP accelerator.
IPV6NG*2	1: The IPv6 expansion header is Routing, Hop-by-Hop, or Destination Opt, and also the header length field is invalid.
OUT_OF_LIST*2	1: The protocol number not listed below was detected in the expansion header in case of IPv6. 0x06 (TCP header) 0x11 (UDP header) 0x00 (Hop-by-Hop) 0x3C (Destination Opt) 0x2C (Fragment) 0x2B (Routing) 0x3B (No next header) 0x32 (ESP header) 0x33 (AH header)
TYPEIP*2	1: IP packet
MAACL*2	1: 802.3 (LLC/SNAP) packet
PPPOE*2	1: PPPoE packet
VTAG*2	1: Packet with VLAN Tag
RX_WORD[12:0]	Number of words of Ethernet frame*1
RX_EOB[1:0]	Indicate valid bytes in the last word of this frame*1 00: 1 byte is valid 01: 2 bytes are valid 10: 3 bytes are valid 11: 4 bytes are valid
MAR[4:0]	MAR[4:1]: Unused (Fixed 0) MAR[0]: Reception of the destination address of the pause packet
MARBR	1: The received frame is broadcast address
MARMLT	1: The received frame is multicast address
MARIND	1: The received frame consists of packets at the address registered in the MAC address register.
MARNOTMT	1: The received frame is not the address for this station
TOOLONG	1: The received frame is a frame longer than the prescribed maximum frame length (1518 octets)
TOOSHORT	1: The received frame is a frame shorter than the prescribed minimum frame length (64 octets). Packets for which TOOSHORT becomes 1 are never received since a TOOSHORT packet is automatically discarded by this MAC.
FIFOOVF	1: The RX FIFO buffer overflows during frame reception. When this bit is set, received data may be invalid.
NBLERR	1: A word in the received frame has a code error, etc.
CRCERR	1: The received frame has a CRC error

Note1: The FCS of an Ethernet frame (4 bytes) and padding of the MAC header to be inserted by the RX TCPIP accelerator function (2 bytes) are also included in the number of received bytes.

2: These fields are invalid if TCPIP accelerator is disabled.

In cases where RX_WORD[12:0] is combined as the higher-order bits with RX_EOB[1:0] as lower-order bits to form RX_LENGTH[14:0], the number of bytes of the received frame is calculated from the following formula.

(Number of received bytes in the Ethernet frame) = RX_LENGTH [14:0] – 3

Examples:

- If RX data is 1 byte → RX_WORD = 0x1 RX_EOB = 0x0 → 4 – 3 = 1 (byte)
- If RX data is 8 bytes → RX_WORD = 0x2 RX_EOB = 0x3 → 11 – 3 = 8 (bytes)
- If RX data is 5 bytes → RX_WORD = 0x2 RX_EOB = 0x0 → 8 – 3 = 5 (bytes)
- If RX data is 9 bytes → RX_WORD = 0x3 RX_EOB = 0x0 → 12 – 3 = 9 (bytes)

(2) RX Ethernet frame

The data format of the received Ethernet frame is listed below.

Field Name	Description
Destination MAC Address	MAC address of the destination When insertion of a management tag is permitted by the Ethernet switch management TAG control register (ETHSWMTC), the management tag information is stored.
Source MAC Address	MAC address of the source
VLAN Tag	Tag Protocol Identifier. This field is available if VLAN Tag is included.
VLAN Info	Tag Control Information. This field is available if VLAN Tag is included.
Type / Length	Ethernet type or length
Frame Payload	Payload
FCS	Frame check sequence If the RX TCPIP accelerator function is enabled and the received packet has TCP/UDP, the FCS field is overwritten by the TCP/UDP checksum. This checksum can be used to calculate the total checksum of fragmented TCP/UDP packets.

When insertion of a management tag is permitted by the Ethernet switch management TAG control register (ETHSWMTC), a field of the destination MAC Address [47:0] is used as follows.

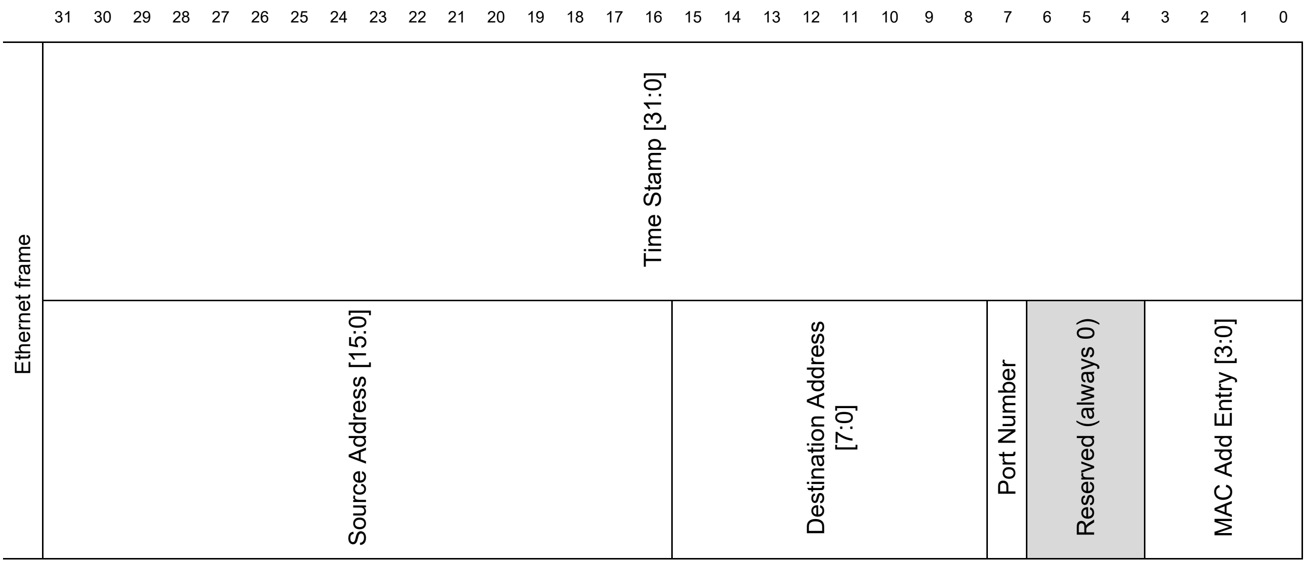


Figure 12.21 Destination MAC Address Field (when insertion of management tag is permitted)

Field Name	Description
Time Stamp [31:0]	The timestamp when the received frame passed a port
MAC Add Entry [3:0]	The index number of MAC address registers (GMAC_ADRnA, GMAC_ADRnB) matching the received frame. Example: value = 5 A destination address of the frame corresponds to the setting of GMAC_ADR5A and GMAC_ADR5B.
Port Number	Port with the received timestamp
Destination MAC Address	MAC address of the destination
Source MAC Address	MAC address of the source

Caution: If the AFILLTEREN bit of the GMAC_RXMODE register is set to 1, it is impossible to recover the destination MAC address because the MAC Add Entry field is invalid.

(a) When RX TCPIP accelerator is enabled and a frame has no TCP/UDP packet

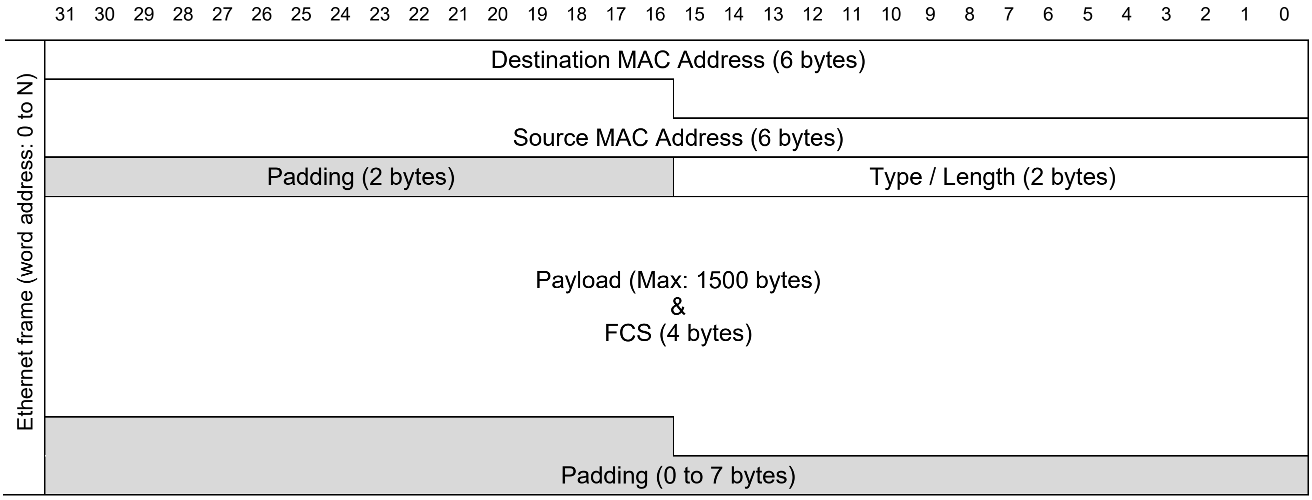


Figure 12.22 Format of Receive Ethernet Frame – TCPIPACC is enabled, without VLAN Tag, no TCP/UDP packets

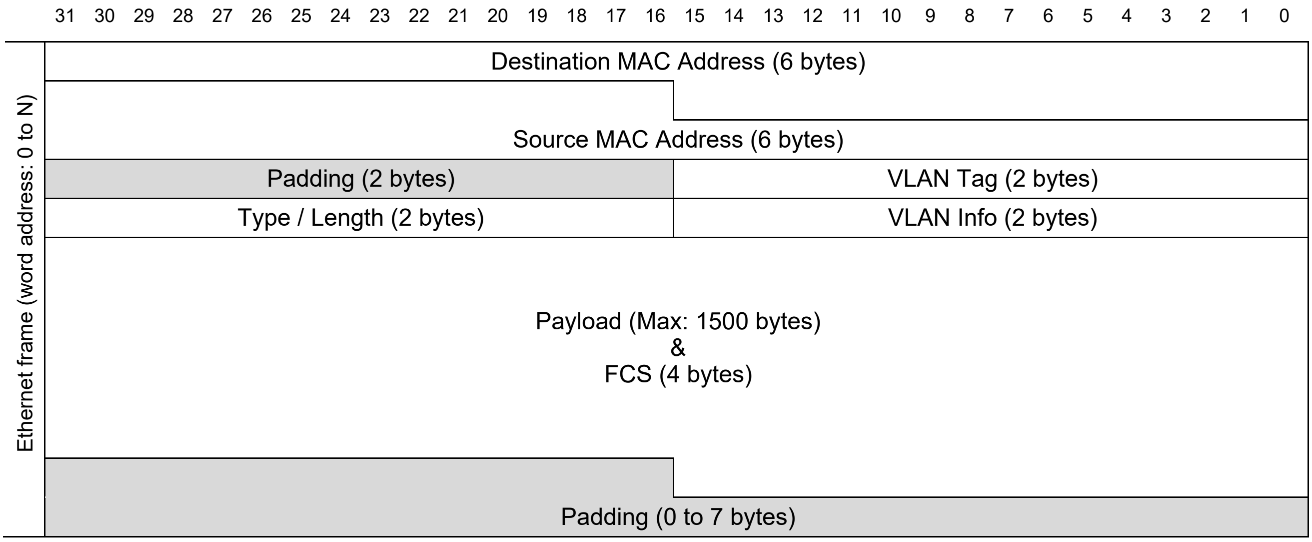


Figure 12.23 Format of Receive Ethernet Frame – TCPIPACC is enabled, with VLAN Tag, no TCP/UDP packets

(b) When RX TCPIP accelerator is enabled and a frame has TCP/UDP packets

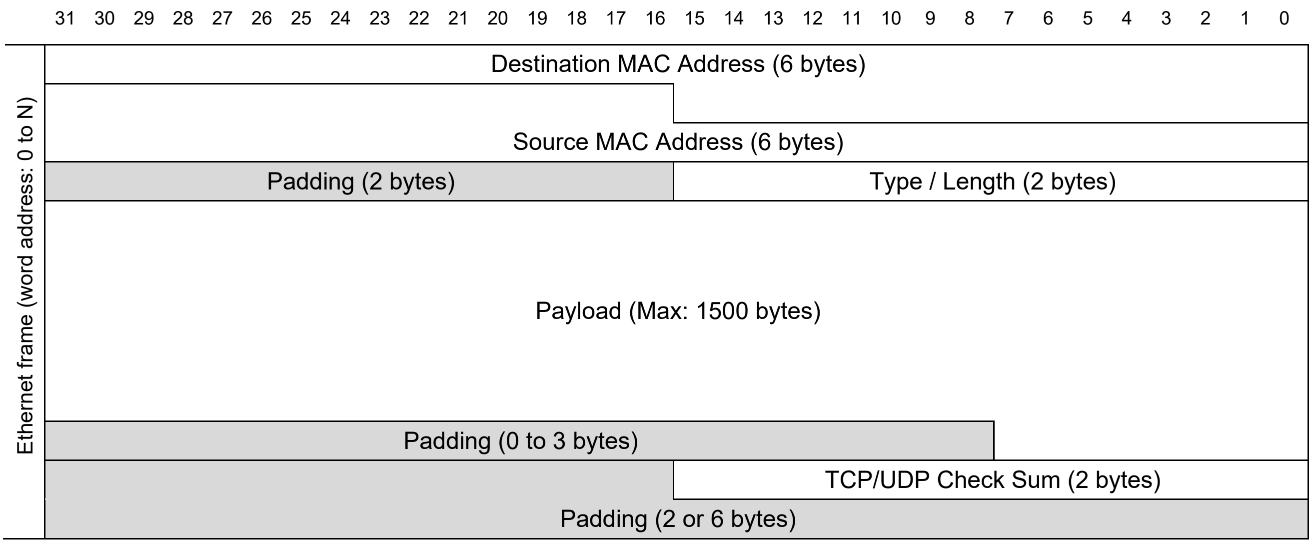


Figure 12.24 Format of Receive Ethernet Frame – TCPIPACC is enabled, without VLAN Tag, with TCP/UDP packets

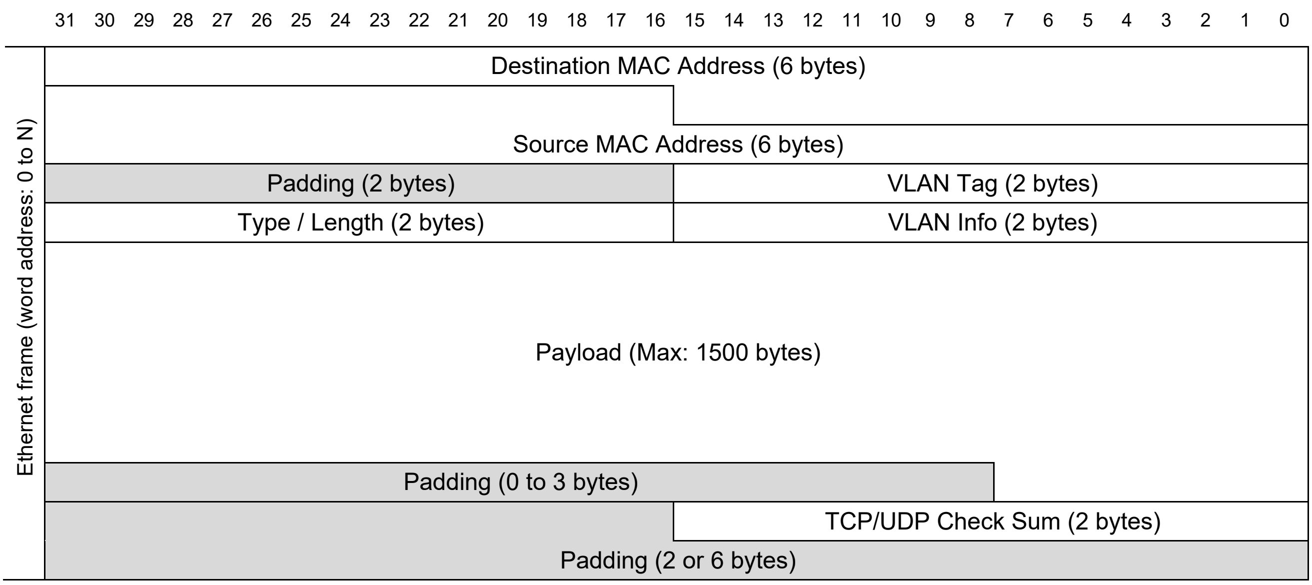


Figure 12.25 Format of Receive Ethernet Frame – TCPIPACC is enabled, with VLAN Tag, with TCP/UDP packets

(c) When RX TCPIP accelerator is disabled

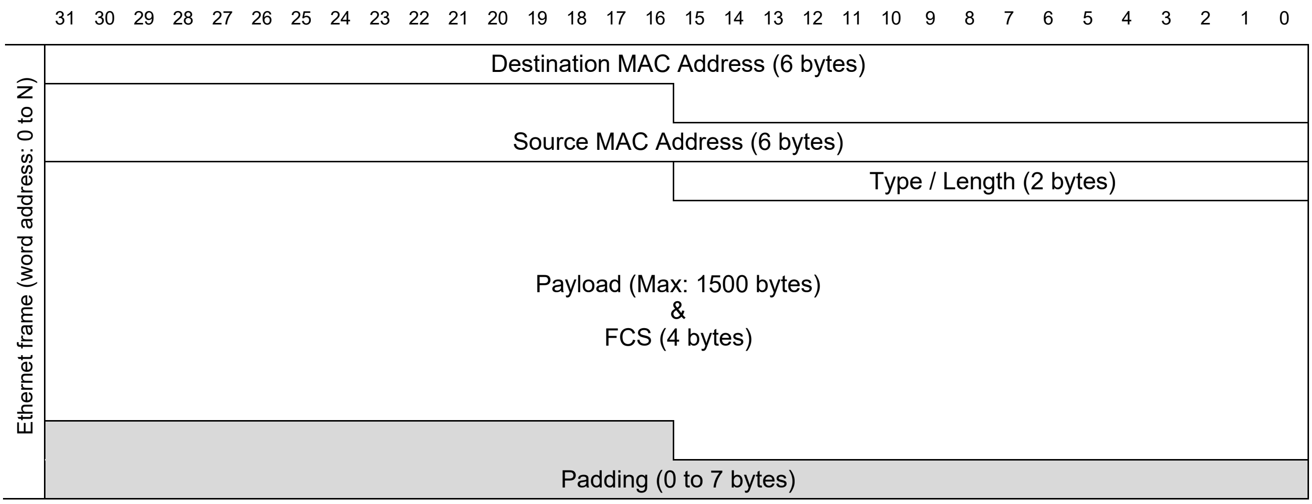


Figure 12.26 Format of Receive Ethernet Frame – TCPIPACC is disabled, without VLAN Tag

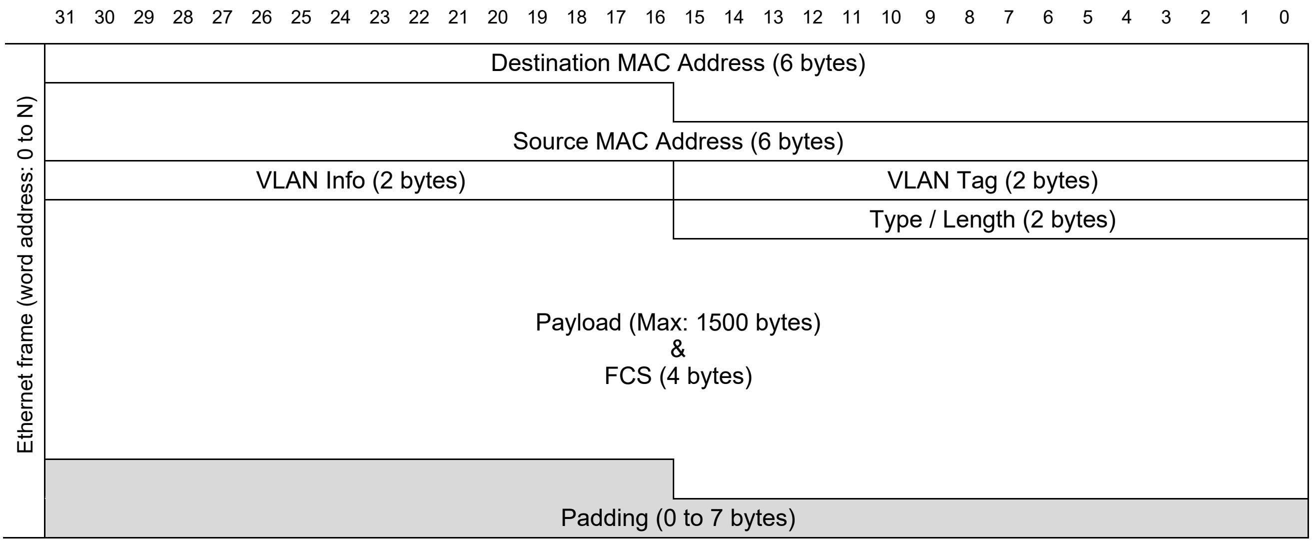


Figure 12.27 Format of Receive Ethernet Frame – TCPIPACC is disabled, with VLAN Tag

12.4.5 TCPIP Accelerator Function

If the TCPIP accelerator function is enabled, hardware can calculate the checksum for transmission or reception of packets. The following three protocols are targets for checksum calculation.

- IPv4 header checksum
- TCP checksum
- UDP checksum

This section explains how to use the TCPIP accelerator for transmission and reception.

12.4.5.1 Transmission Using the TCPIP Accelerator

When the TTCPIPEN bit of the GMAC_ACC register is set to 1, the TCPIP accelerator for transmission is enabled. If a packet including IPv4, TCP/IP, or UDP/IP is transmitted while the TCPIP accelerator is enabled, hardware automatically calculates the checksum and writes it to the checksum field in the packet. The TCPIP accelerator requires 2-byte padding in the MAC header.
In addition, if the TCPIP ACC OFF field of TX frame control information is set to 1, the TCPIP accelerator function is switched off for each packet.
Hardware does not calculate the TCP/UDP checksum of fragmented packets. The checksum should be calculated by software.

When the TTCPIPEN bit of the GMAC_ACC register is set to 0, the TCPIPP accelerator for transmission is disabled.

Table 12.20 GMAC_ACC Register Settings and Operation of the TX TCPIP Accelerator

GMAC_ACC.TTCPIPEN	TX Frame Control Information TCPIP ACC OFF	Checksum Calculation (TX)	Padding for TCPIPACC (TX)
0	0	Not available	Not required
0	1	Not available	Not required
1	0	Available	Required
1	1	Not available	Required

Remark: If the UDP checksum for transmission packets calculated by hardware is 0x0000, the checksum field of the UDP header is changed to 0xFFFF in this packet.

Caution: If the value of header length field in IPv4 header doesn’t match real length of the header, transmission might not complete and the status might not be able to return to normal operation. Be careful to set the correct value.

12.4.5.2 Reception Using the TCPIP Accelerator

If the RTCPIPEN bit of the GMAC_ACC register is set to 1, the RX TCPIP accelerator function for reception is enabled. If a packet including IPv4, TCP/IP, or UDP/IP is received while the RX TCPIP accelerator is enabled, hardware automatically calculates the checksum of the packet. If the result of calculation is not equal to the value of the checksum field in the packet, error information is stored in the IPNG or TCPNG field of RX frame information.
While RX TCPIPACC is enabled, 2-byte padding for TCPIPACC is inserted in the MAC header of the received frame.
If the RX TCPIP accelerator function is enabled and the received packet has TCP/UDP, the FCS field is overwritten by the TCP/UDP checksum. This checksum can be used to calculate the total checksum of fragmented TCP/UDP packets. But calculate the checksum of pseudo header by software in case of fragmented packets since it is not included in the checksum calculated by hardware.
If any field of IPNG, IPV6NG, or OUT_OF_LIST of RX frame information shows 1, hardware does not calculate the checksum for the received frame at that time. And also, if the IPv6 extension header includes the fragment, ESP, or AH protocol, TCP/UDP checksum calculation does not proceed.

If the RTCPIPACC bit of the GMAC_ACC register is set to 1, checksum calculation does not proceed but padding for TCPIPACC is inserted in the received frame.

If the RTCPIPEN bit of the GMAC_ACC register is set to 0, the RX the TCPIP accelerator function for reception is disabled. If this is the case, padding for TCPIPACC is not inserted in the received frame.

Table 12.21 GMAC_ACC Register Settings and Operation of the RX TCPIP Accelerator

GMAC_ACC. RTCPIPEN	GAMC_ACC. RTCPIPACC	Checksum Calculation (RX)	Padding for TCPIPACC (RX)	Checksum Calculated by Hardware Overwrites the FCS Field
0	0	No	No	No
0	1	No	No	No
1	0	Yes	Yes	Yes
1	1	No	Yes	No

Remark: If the UDP checksum field in the received packet is 0x0000, hardware does not check checksum validation. TCPNG field is 0.

12.5 Notes

Note the following when using the Gigabit Ethernet MAC.

12.5.1 Appending Padding to the MAC Header Section within the TX Frame

In the Gigabit Ethernet MAC, a transmission frame is normally composed of the 14-byte MAC header plus 2 bytes of padding so that the TCPIP accelerator handles the data.
However, the padding is not actually sent. Accordingly, note that it is not included in the data size of the frame for transmission.
Refer to section 12.4.5.1, Transmission Using the TCPIP Accelerator, for detail.

12.5.2 Erroneous Judgment about Checksum Validation at Specific Packet Reception

(1) Ethernet II frames and IEEE802.3 + IEEE802.2 (LLC + SNAP) frames

If frame with any of the following conditions is received, the IPNG and/or TCPNG field of RX frame information may be set to 1 despite the received packet is valid. In that case, check the checksum by software.

- IPv4 and checksum field value in the TCP header is 0x0000 or 0xFFFF
- IPv6, frame size is more than or equal to 60 bytes, payload size of TCP or UDP is 1 byte and the following date is not 0
- IPv6 and checksum of pseudo header used for TCP or UDP checksum calculation is more than or equal to 21 bits

(2) IEEE802.3 + IEEE802.2 (LLC) frames

If IEEE802.3 + IEEE802.2 (LLC) frame without SNAP is received, the TYPEIP and IPNG field or RX frame information may be set to 1 despite IP packet is not included. In that case, check by software if SNAP is included or not. If SNAP is not included, consider the received frame valid.

12.5.3 Error of Rx Frame Information at RX FIFO Overflow

If Rx TCPIP accelerator is enabled and Rx FIFO is overflowed, Rx frame information might include error as below.

- Error information which is related to the previous error frame is included in frame information of normal reception frame.
- Abnormal frame which caused Rx FIFO overflow is regarded as normal frame because of illegal value is included in frame information

Apply any of the following methods to avoid the issue

(A) Disable Rx TCPIP accelerator without padding insertion to MAC header. Specifically, clear bit0 of GMAC_ACC register.

(B) When Rx FIFO is overflowed, discard all frames left in Rx FIFO and Buffer RAM. Specifically, apply the following procedure;

(1) Disable Rx MAC.

(2) Discard all frames inside Rx FIFO.

(3) Discard all frames inside Buffer RAM.

(4) Enable Rx MAC.

(5) Discard at least one frame with BUFID VALID bit = 1. This is because FIFO empty state can be read even if the frame which caused FIFO overflow remains in the FIFO. Receive normal frame once and discard remained abnormal frame with it.

Figure 12.28 to Figure 12.31 are the flowcharts of workaround (B) described above.

- In case the hardware real-time OS is used

  Figure 12.28: Flowchart of RX FIFO overflow processing task

  Figure 12.29: Flowchart of Reception processing task

• Create overflow processing task with higher priority than one of reception processing task • Start overflow task by HWISR combined with FIFO overflow interrupt • Discard the abnormal frame remained in the FIFO by HWISR combined with reception interrupt

- In case the hardware real-time OS is not used

  Figure 12.30: Flowchart of RX FIFO overflow processing

  Figure 12.31: Flowchart of Reception processing

• The abnormal frame remained in the FIFO is discarded in reception processing. Discard valid data once when overflow return flag is set. • Overflow return flag is a global variable. • Overflow interrupt is disabled from reading BUFID till checking overflow return flag

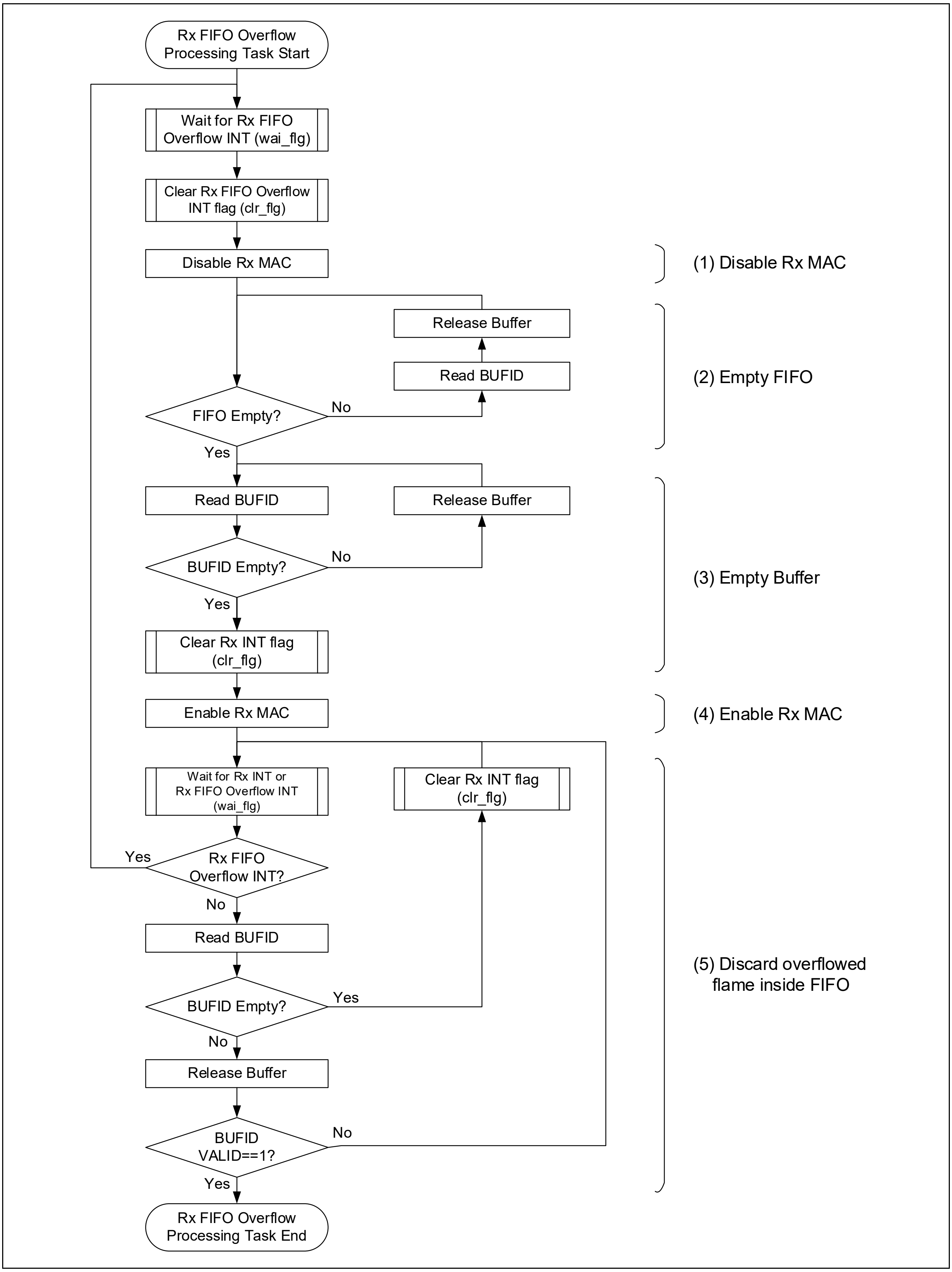


Figure 12.28 Flowchart of RX FIFO overflow processing task (In case the hardware real-time OS is used)

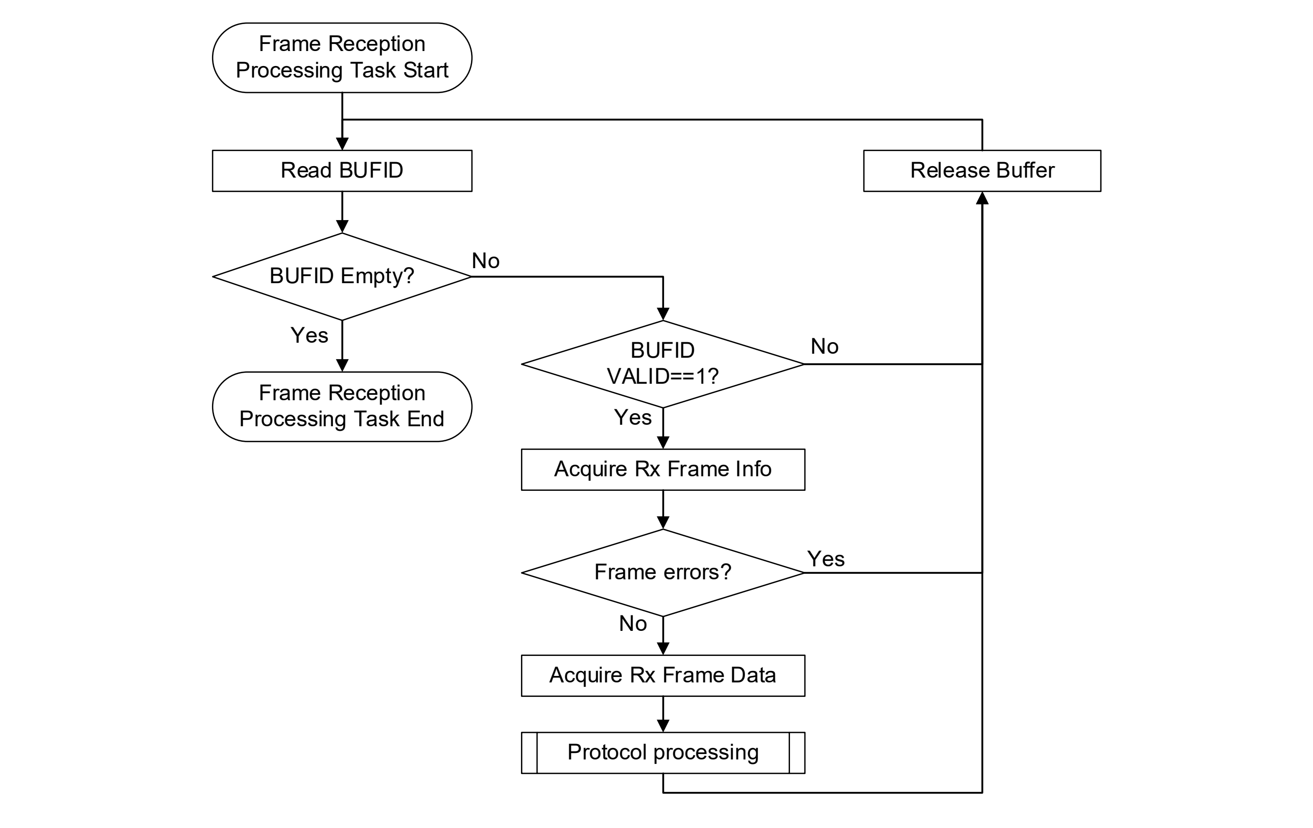


Figure 12.29 Flowchart of Reception processing task (In case the hardware real-time OS is used)

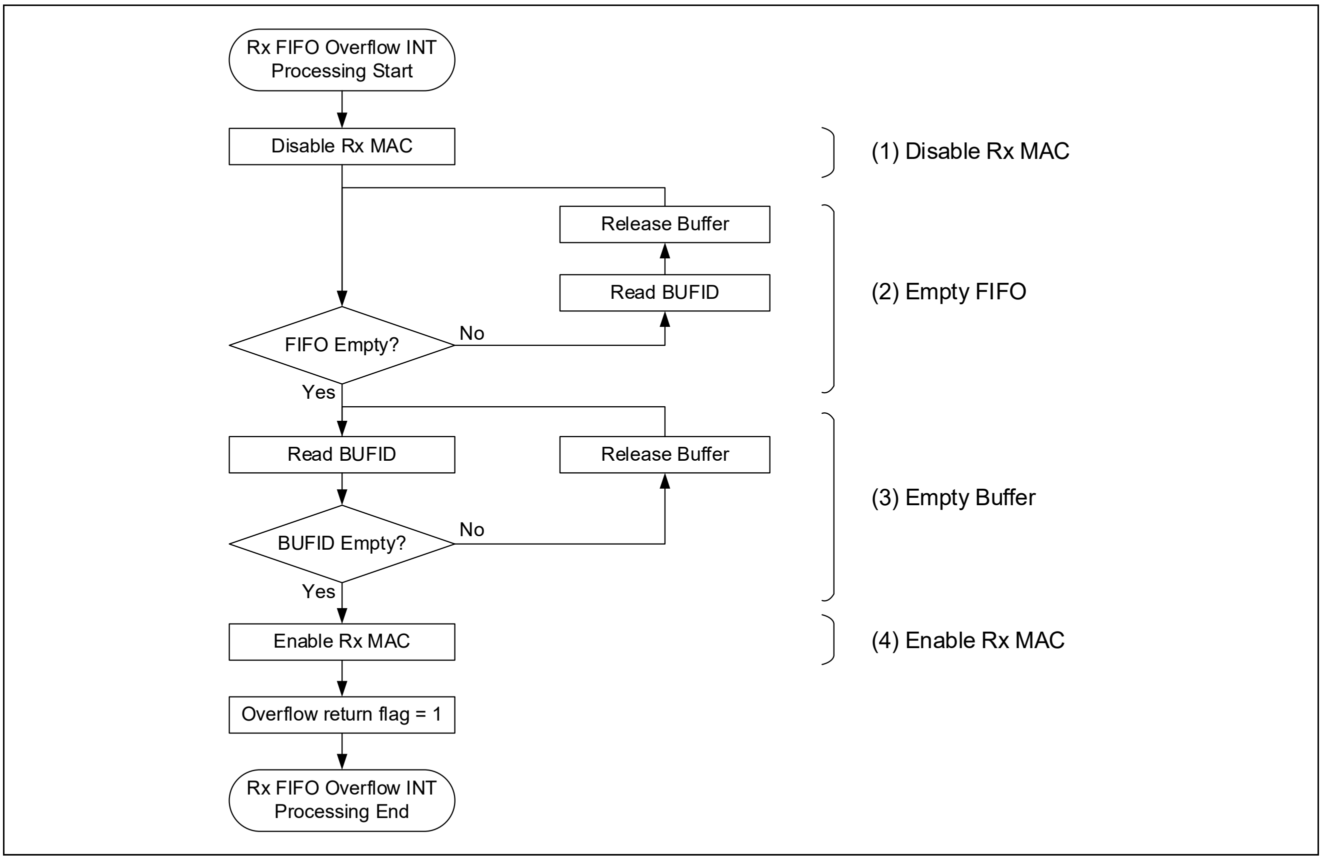


Figure 12.30 Flowchart of RX FIFO overflow INT processing (In case the hardware real-time OS is not used)

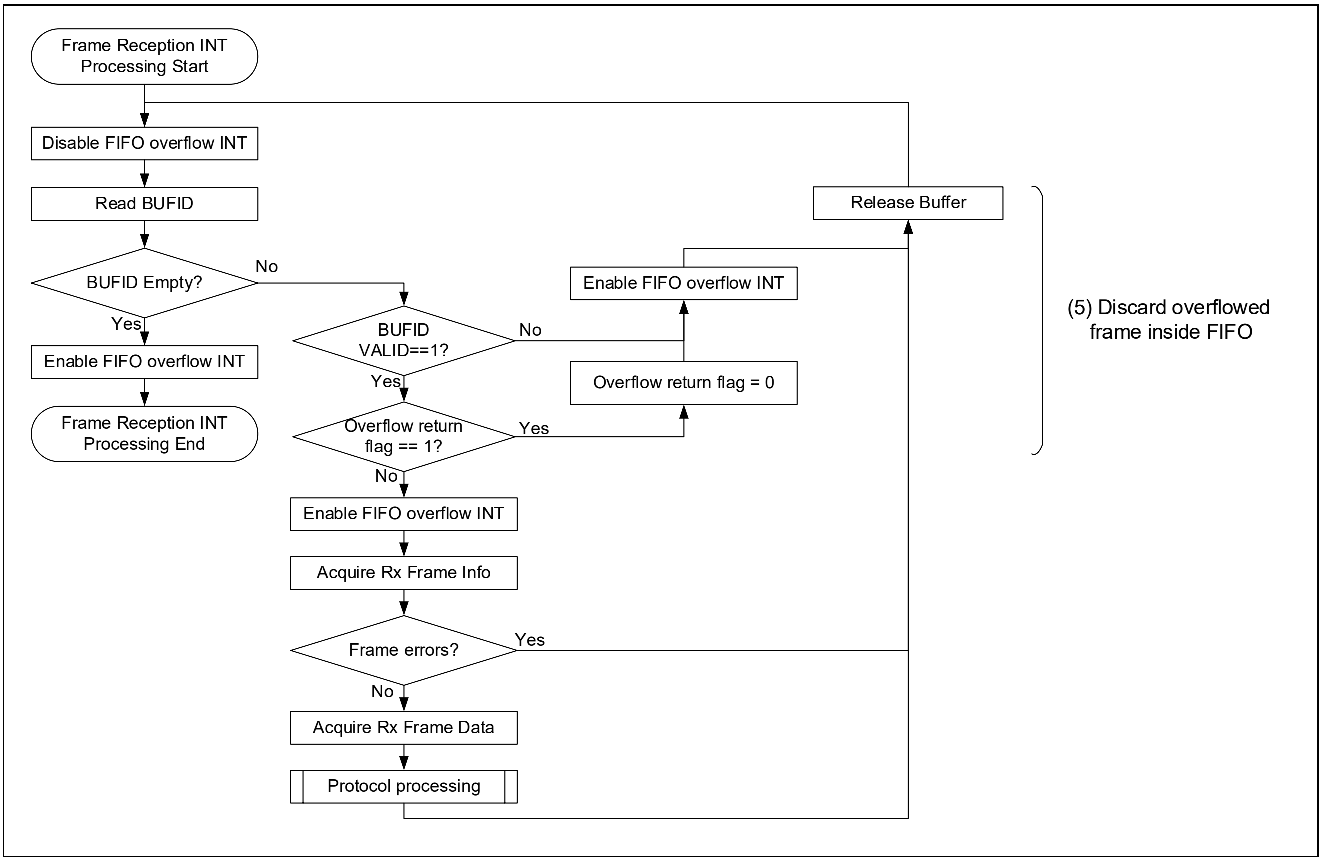


Figure 12.31 Flowchart of Reception processing (In case the hardware real-time OS is not used)

12.5.4 Error of Rx Frame Information at Reception of the Frame more than 64 Bytes with Padding

If Rx TCPIP accelerator is enabled and the frame meets all the following condition, it is possible that reception word size (RX_WORD[12:0]) in the frame information increases by 1 word (4 bytes) or decreases by 1 word compared with correct size. In case of decrease by 1 word, it is possible that RX_WORD indicates the size which causes lack of IP packet. IP packet itself is NOT lacked.

- Frame size including FCS is more than 64 bytes.
- TCP/IP or UDP/IP packet is included.
- Padding (Trailer) is included between IP packet and FCS.

Apply any of the following methods to avoid the issue

(A) Disable Rx TCPIP accelerator. Specifically, clear bit0 or set bit2 of GMAC_ACC register.

(B) To avoid lack of the IP packet, increase reception word size by 1 and transfer the size to protocol stack. In the protocol stack, payload data should be extracted based on size of Total Length field in IP header and the rest data should be discarded. Figure 12.32 is the flowchart of this workaround.

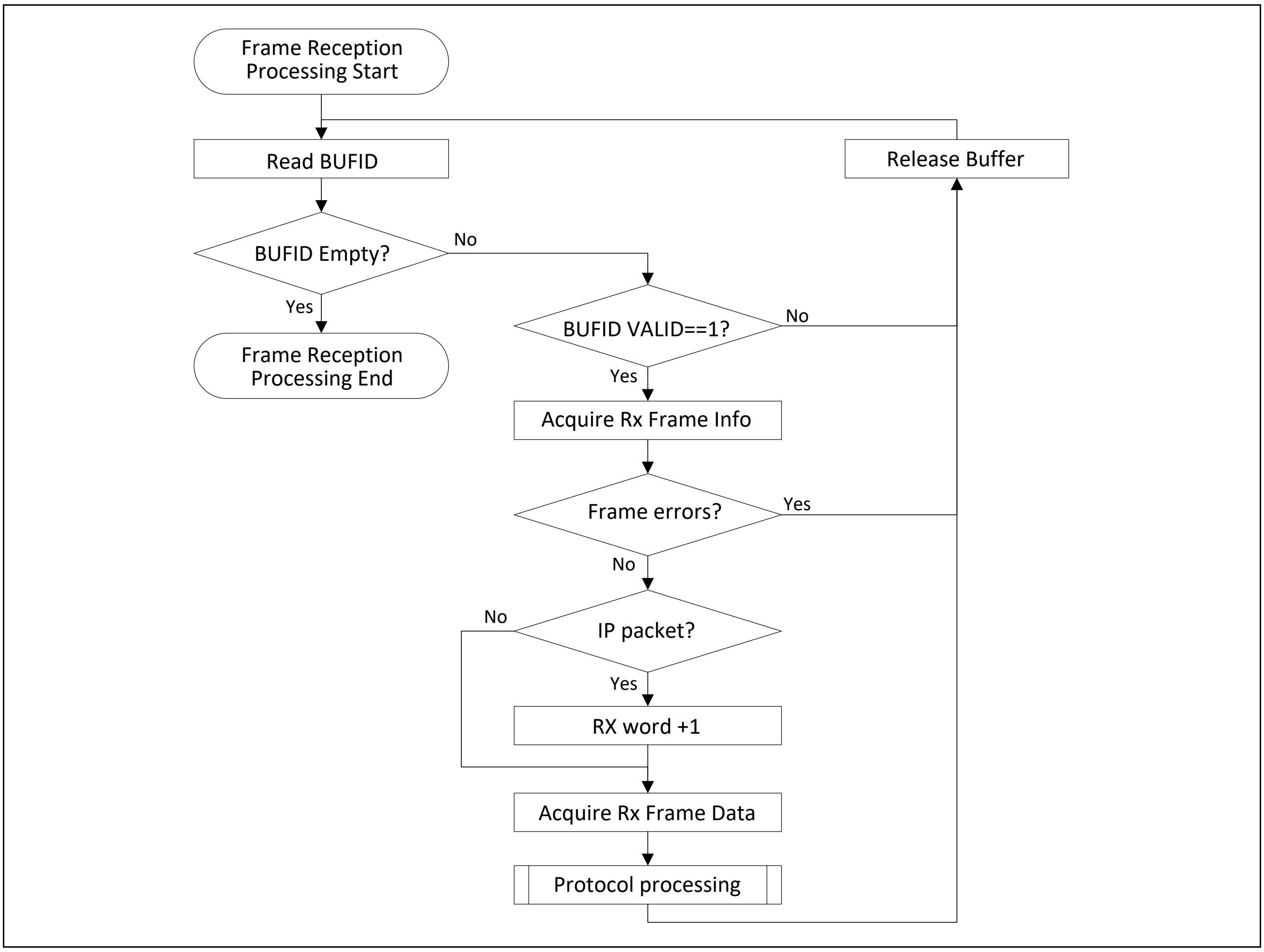


Figure 12.32 Flowchart of Reception processing

12.5.5 Transmitting Data in Cut-Through Mode

Setting the SF bit (b29) of the TX Mode register (GMAC_TXMODE) to 0 may lead to generation of an unexpected TX FIFO underflow interrupt. To avoid this, always set this bit to 1 (Store & Forward mode).

12.5.6 Jumbo Frames

This product does not support transmission and reception of frames exceeding 1,518 bytes, i.e. jumbo frames.

13. Ethernet Switch

This section describes the Ethernet switch of an R-IN32M4-CL3.

13.1 Overview

R-IN32M4-CL3 product incorporates an Ethernet switch, which enables building a linear or ring-type network topology without using a switching hub outside the chips themselves. Use of the Ethernet switch and mode settings are controlled by registers.

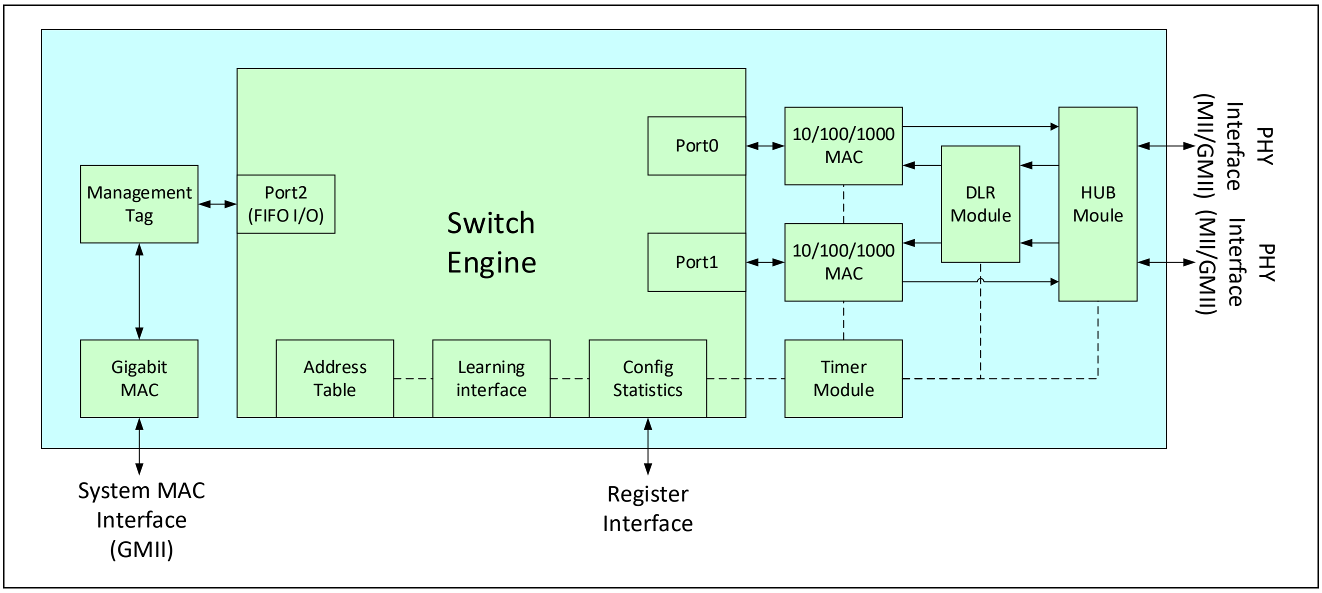


Figure 13.1 Overview of the Ethernet Switch

13.2 Features

The Ethernet switch of an R-IN32M4-CL3 has the following features:

- Two-port PHY interface
- For IEEE802.3
- 10 BASE-T, 100 BASE-TX
- 1000 BASE-T Note
- Full- and half-duplex communications Note
- Hardware switching, address table, and filtering
- Support for QoS, which allows classification of the priority of frames
- Priority control based on VLAN Priority (IEEE802.1q), which enables priorities to be re-assigned
- Classification and priority assignment based on Differentiated Services (DiffServ) Code Point Field of IP v. 4 and Class of Service (CoS) in IP v. 6
- Queue with four priority levels
- Multicasting and broadcasting
- VLAN frames
- Cut-through and hub features
- Device level ring (DLR)
- IEEE1588 timer module
- MII/GMII interface Note
- Generation of desired pulses for output by the timer counter of the switch

Note: Half-duplex communications are not supported for 1000 BASE-T.

Interrupt Signals of Ethernet Switch

Exception No.	Name	Interrupt Source	Connected to				
			NVIC	HW-RTOS	DMAC	Real-Time Port	Timer TAUJ2 /TAUD
54	INTETHSW	Ethernet SWITCH Timer interrupt	○	○	○	○	○
55	INTETHSWDLR	Ethernet SWITCH DLR interrupt	○	○	○	○	○
56	INTETHSWSYNC	Ethernet SWITCH SYNC interrupt	○	○	○	○	○

I/O Signals of Ethernet Switch

Pin Name	I/O	Function	Shared Port	Active
ETHSWSYNCOUT	O	Ethernet switch event output	P24	High

13.3 Control Registers

13.3.1 List of Registers

(1) Operating Mode Registers

Register Name	Symbol	Address
Ethernet switch management TAG control register	ETHSWMTC	BASE + 0680H
Ethernet switch operating mode setting register	ETHSWMD	BASE + 0684H

(2) Switch Configuration Registers

Register Name	Symbol	Address
• Configuration and Setting		
Port enable register	PORT_ENA	4007 0008H
Unicast default mask register	UCAST_DEFAULT_MASK	4007 000CH
Broadcast default mask register	BCAST_DEFAULT_MASK	4007 0014H
Multicast default mask register	MCAST_DEFAULT_MASK	4007 0018H
Input learning blocking register	INPUT_LEARN_BLOCK	4007 001CH
Management configuration register	MGMT_CONFIG	4007 0020H
Mode configuration register	MODE_CONFIG	4007 0024H
VLAN tag ID register	VLAN_TAG_ID	4007 0034H
• Output Queue Management		
Output queue management status register	OQMGR_STATUS	4007 0080H
Output queue minimum memory register	QMGR_MINCELLS	4007 0084H
Output queue minimum memory statistics register	QMGR_ST_MINCELLS	4007 0088H
Output queue congestion status register	QMGR_CGS_STAT	4007 008CH
Internal queue interface status register	QMGR_IFACE_STAT	4007 0090H
Queue weight register	QMGR_WEIGHTS	4007 0094H
• Per Port Configurations and Setting (n = 0 to 2)		
VLAN priority register n	VLAN_PRIORITYn	4007 0100H + 0004H*n
IP priority register n	IP_PRIORITYn	4007 0140H + 0004H*n
Priority configuration register n	PRIORITY_CFGn	4007 0180H + 0004H*n
• HUB Module Setting		
HUB control register	HUB_CONTROL	4007 01C0H
HUB frame count register	HUB_STATS	4007 01C4H
• Hub Reception Filter MAC Address Setting (n = 0 to 6)		
Hub input filter MAC address low register n	HUB_FLT_MACnlo	4007 01C8H + 0008H*n
Hub input filter MAC address high register n	HUB_FLT_MACnhi	4007 01CCH + 0008H*n
• Statistics registers		
Switch statistics registers	See section 13.3.3.22	See section 13.3.3.22

(3) Learning Interface Registers

Register Name	Symbol	Address
Learning record A register	LRN_REC_A	4007 0500H
Learning record B register	LRN_REC_B	4007 0504H
Learning data status register	LRN_STATUS	4007 0508H
Address table	ADR_TABLE	4007 4000H to 4007 47FCH

(4) MAC Port Registers

Register Name	Symbol	Address (n = 0, 1)
• Configuration and Setting		
Command configuration register n	COMMAND_CONFIGn	4007 8008H + 2000H*n
Maximum frame length register n (shared)	FRM_LENGTHn	4007 8014H + 2000H*n
FIFO buffer threshold register n (shared)	See section 13.3.5.3	See section 13.3.5.3
MAC status register n (shared)	MAC_STATUSn	4007 8058H + 2000H*n
Transmit IPG length register n (shared)	TX_IPG_LENGTHn	4007 805CH + 2000H*n
• Statistic Counters		
MAC RX/TX statistic counters	See section 13.3.5.6	See section 13.3.5.6

(5) Timer Module Registers

(1/2)

Register Name	Symbol	Address
• Configuration and Setting		
Timer module configuration register	TSM_CONFIG	4007 C004H
Interrupt status/ACK register	TSM_IRQ_STAT_ACK	4007 C008H
• Transmit Timestamp (n = 0, 1)		
Port timestamp control/status register n	PORTn_CTRL	4007 C020H + 0008H*n
Port timestamp register n	PORTn_TIME	4007 C024H + 0008H*n
• Timer Settings		
Timer control register	ATIME_CTRL	4007 C120H
Timer nanosecond register	ATIME	4007 C124H
Timer offset correction register	ATIME_OFFSET	4007 C128H
Timer periodic event generation register	ATIME_EVT_PERIOD	4007 C12CH
Timer drift correction register	ATIME_CORR	4007 C130H
Timer increment register	ATIME_INC	4007 C134H
Timer second register	ATIME_SEC	4007 C138H
Timer offset correction count register	ATIME_OFFS_CORR	4007 C13CH

(2/2)

Register Name	Symbol	Address
Timer output enable register	SWTMEN	BASE + 1100H
Timer seconds start setting register L	SWTMSTSECL	BASE + 1110H
Timer seconds start setting register H	SWTMSTSECH	BASE + 1114H
Timer nanoseconds start setting register L	SWTMSTNSL	BASE + 1118H
Timer nanoseconds start setting register H	SWTMSTNSH	BASE + 111CH
Timer seconds period setting register L	SWTMPSECL	BASE + 1120H
Timer seconds period setting register H	SWTMPSECH	BASE + 1124H
Timer nanoseconds period setting register L	SWTMPNSL	BASE + 1128H
Timer nanoseconds period setting register H	SWTMPNSH	BASE + 112CH
Timer pulse width setting register	SWTMWTH	BASE + 1130H
Timer maximum counter value register L	SWTMMAXPL	BASE + 1134H
Timer maximum counter value register H	SWTMMAXPH	BASE + 1138H
Timer seconds time hold register	SWTMLATSEC	BASE + 1140H
Timer nanoseconds time hold register	SWTMLATNS	BASE + 1144H

(6) DLR Module Registers

Register Name	Symbol	Address
• Configuration and Setting		
DLR control register	DLR_CONTROL	4007 E000H
DLR status register	DLR_STATUS	4007 E004H
DLR Ethernet type register	DLR_ETH_TYP	4007 E008H
DLR interrupt control register	DLR_IRQ_CTRL	4007 E00CH
DLR interrupt status/ACK register	DLR_IRQ_STAT_ACK	4007 E010H
DLR local MAC address low register	LOC_MAClo	4007 E014H
DLR local MAC address high register	LOC_MAChi	4007 E018H
• Beacon Frame Parameters		
DLR supervisor MAC address low register	SUPR_MAClo	4007 E020H
DLR supervisor MAC address high register	SUPR_MAChi	4007 E024H
DLR ring status/VLAN register	STATE_VLAN	4007 E028H
DLR beacon timeout timer register	BEC_TMOUT	4007 E02CH
DLR beacon interval register	BEC_INTRVL	4007 E030H
DLR supervisor IP address register	SUPR_IPADR	4007 E034H
DLR sub type/protocol version register	ETH_STYP_VER	4007 E038H
DLR beacon invalid timeout timer register	INV_TMOUT	4007 E03CH
DLR sequence ID register	SEQ_ID	4007 E040H
• DLR statistics counters		
DLR MAC statistics counters	See section 13.3.7.17	See section 13.3.7.17

13.3.2 Operating Mode Setting Registers

13.3.2.1 Ethernet Switch Management TAG Control Register (ETHSWMTC)

This register is used to specify management tag information when Ethernet switching is used.

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

Caution: This register is only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

Address: BASE + 0680H
Initial value: 0000 E001H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETHSWMTC	SWTAGEN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETHSWMTC	SWTAGTYP15	SWTAGTYP14	SWTAGTYP13	SWTAGTYP12	SWTAGTYP11	SWTAGTYP10	SWTAGTYP9	SWTAGTYP8	SWTAGTYP7	SWTAGTYP6	SWTAGTYP5	SWTAGTYP4	SWTAGTYP3	SWTAGTYP2	SWTAGTYP1	SWTAGTYP0
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31	SWTAGEN	Inserting management tags into frames 0: Disabled 1: Enabled
15 to 0	SWTAGTYP15 to SWTAGTYP0	Specify the Ethernet type to be set for management tags. The initial value is E001H.

13.3.2.2 Ethernet Switch Operating Mode Setting Register (ETHSWMD)

This register is used to specify the operating mode when Ethernet switching is used.

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

Caution: This register is only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETHSWMD	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	P1HDMODE	P1GIGMODE	P0HDMODE	P0GIGMODE
R/W	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	R/W	R/W	R/W	R/W

Address BASE + 0684H
Initial value 0000 0000H

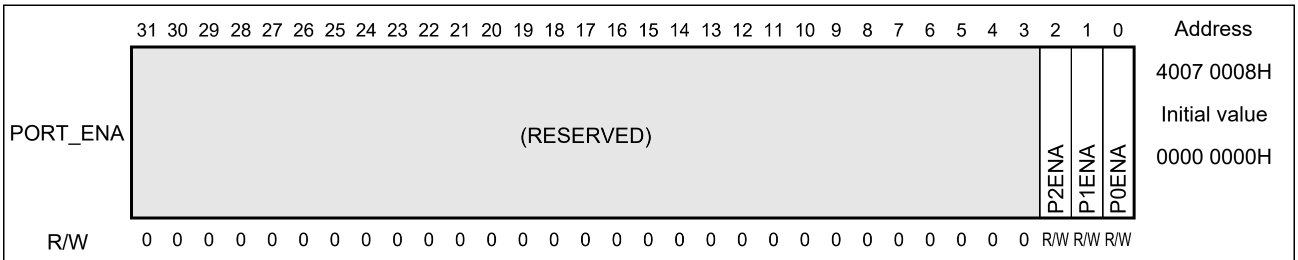
Bit Position	Bit Name	Description
3	P1HDMODE	Specifies the transfer mode of port 1. 0: Full-duplex 1: Half-duplex
2	P1GIGMODE	Specifies the transfer rate of port 1. 0: 10/100 Mbps 1: 1000 Mbps
1	P0HDMODE	Specifies the transfer mode of port 0. 0: Full-duplex 1: Half-duplex
0	P0GIGMODE	Specifies the transfer rate of port 0. 0: 10/100 Mbps 1: 1000 Mbps

13.3.3 Switch Configuration Registers

13.3.3.1 Port Enable Register (PORT_ENA)

This register enables or disables each port of the Ethernet switch. A disabled port does not transmit frames, but it can receive frames.

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



Bit Position	Bit Name	Description
31 to 3	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
2	P2ENA	For enabling the internal interface (port 2) 0: Disabled 1: Enabled
1	P1ENA	For enabling port 1 0: Disabled 1: Enabled
0	P0ENA	For enabling port 0 0: Disabled 1: Enabled

13.3.3.2 Unicast Default Mask Register (UCAST_DEFAULT_MASK)

This register is used to make settings for transfer of unicast frames for each port of the Ethernet switch. If the destination addresses for frames are unicast, the frames are transferred to the port where masking is enabled even if the addresses are not found in the address table.

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

UCAST_DEFAULT_MASK — Address 4007 000CH, Initial value 0000 0007H

Bit	31 to 3	2	1	0
UCAST_DEFAULT_MASK	(RESERVED)	P2UCASTDM	P1UCASTDM	P0UCASTDM
R/W	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	R/W	R/W	R/W

Bit Position	Bit Name	Description
31 to 3	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
2	P2UCASTDM	For setting masking as the default in unicasting for the internal interface (port 2) Masking is not required in many cases because the internal interface will generally not need to receive unnecessary unicast traffic. If masking is not to be set, either statically setting the local device's unicast address in the table of the switch or letting the switch learn dynamically from the past results of transfer will be required. 0: Disabled 1: Enabled
1	P1UCASTDM	For setting masking as the default in unicasting for port 1 0: Disabled 1: Enabled
0	P0UCASTDM	For setting masking as the default in unicasting for port 0 0: Disabled 1: Enabled

13.3.3.3 Broadcast Default Mask Register (BCAST_DEFAULT_MASK)

This register is used to make settings for transfer of broadcast frames for each port of the Ethernet switch. If the destination addresses for frames are broadcast, the frames are transferred to the port where masking is enabled.

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

BCAST_DEFAULT_MASK — Address 4007 0014H, Initial value 0000 0000H

Bit	31 to 3	2	1	0
Name	(RESERVED)	P2BCASTDM	P1BCASTDM	P0BCASTDM
R/W	0 (each bit)	R/W	R/W	R/W

Bit Position	Bit Name	Description
31 to 3	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
2	P2BCASTDM	For setting masking as the default in broadcasting for the internal interface (port 2) 0: Disabled 1: Enabled
1	P1BCASTDM	For setting masking as the default in broadcasting for port 1 0: Disabled 1: Enabled
0	P0BCASTDM	For setting masking as the default in broadcasting for port 0 0: Disabled 1: Enabled

13.3.3.4 Multicast Default Mask Register (MCAST_DEFAULT_MASK)

This register is used to make settings for transfer of multicast frames for each port of the Ethernet switch. If the destination addresses for frames are multicast, the frames are transferred to the port where masking is enabled.

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

MCAST_DEFAULT_MASK — Address 4007 0018H, Initial value 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCAST_ DEFAULT_ MASK	(RESERVED)																													P2MCASTDM	P1MCASTDM	P0MCASTDM
R/W	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	R/W	R/W	R/W

Bit Position	Bit Name	Description
31 to 3	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
2	P2MCASTDM	For setting masking as the default in multicasting for the internal interface (port 2) 0: Disabled 1: Enabled
1	P1MCASTDM	For setting masking as the default in multicasting for port 1 0: Disabled 1: Enabled
0	P0MCASTDM	For setting masking as the default in multicasting for port 0 0: Disabled 1: Enabled

13.3.3.5 Input Learning Blocking Register (INPUT_LEARN_BLOCK)

This register sets address learning and frame blocking for each port of the Ethernet switch.

If address learning is disabled (by setting the corresponding bit to 1), only bridge protocol data unit (BPDU) frames are subject to learning and learning does not apply to any other frames.

If frame blocking is enabled (by setting the corresponding bit to 1), only BPDU frames are received, but all other frames are discarded at the port which received them, and are not transferred to another port.

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

INPUT_LERAN_BLOCK — Address 4007 001CH, Initial value 0000 0000H

Bit	31 to 19	18	17	16	15 to 3	2	1	0
INPUT_LERAN_BLOCK	(RESERVED)	P2LEARNDIS	P1LEARNDIS	P0LEARNDIS	(RESERVED)	P2BLOCKEN	P1BLOCKEN	P0BLOCKEN
R/W	0 0 0 0 0 0 0 0 0 0 0 0 0	R/W	R/W	R/W	0 0 0 0 0 0 0 0 0 0 0 0 0	R/W	R/W	R/W

Bit Position	Bit Name	Description
31 to 19	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
18	P2LEARNDIS	For setting address learning for the internal interface (port 2) 0: Enabled 1: Disabled
17	P1LEARNDIS	For setting address learning for port 1 0: Enabled 1: Disabled
16	P0LEARNDIS	For setting address learning for port 0 0: Enabled 1: Disabled
15 to 3	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
2	P2BLOCKEN	For setting input port blocking for the internal interface (port 2) 0: Disabled 1: Enabled
1	P1BLOCKEN	For setting input port blocking for port 1 0: Disabled 1: Enabled
0	P0BLOCKEN	For setting input port blocking for port 0 0: Disabled 1: Enabled

13.3.3.6 Management Configuration Register (MGMT_CONFIG)

This register configures the bridge management port of the Ethernet switch. It makes settings for the management port for transfer of BPDU frames and its operation. In this LSI chip, the internal interface port (port 2) must be set as a management port.

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

Address 4007 0020H

Initial value 0000 0000H

MGMT_CONFIG	31 to 18	17	16	15 to 13	12 to 8	7	6	5	4 to 2	1, 0
Bit	(RESERVED)	P1PORTMASK	P0PORTMASK	PRIORITY	(RESERVED)	DISCARD	ENABLE	MSGTRANS	(RESERVED)	PORT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Description
31 to 18	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
17	P1PORTMASK Note 1	Enables masking of the transfer of management frames to port 1 from the management port. When this bit is set to 1, only BPDU frames are forcibly transferred to port 1. This setting has no effect on other frames. 0: Disabled 1: Enabled
16	P0PORTMASK Note 1	Enables masking of the transfer of management frames to port 0 from the management port. When this bit is set to 1, only BPDU frames are forcibly transferred to port 0. This setting has no effect on other frames. 0: Disabled 1: Enabled
15 to 13	PRIORITY	Sets the priority of the transfer of management frames. When you want to transmit management frames earlier than normal frames, these bits can be used to raise their priority in the output queue.
12 to 8	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
7	DISCARD	Sets BPDU frames to be discarded. When this bit is set to 1, BPDU frames are always discarded. If the setting of the ENABLE bit is 1, be sure to set this bit to 0. 0: Disabled 1: Enabled
6	ENABLE	Sets the transfer of BPDU frames to the management port. 0: The frames are discarded If the setting of the DISCARD bit is 1. 1: All BPDU frames are exclusively transferred to the management port.
5	MSGTRANS	This bit is set to 1 when messages are transferred to another port from the management port. Write 0 to this bit to reset it.
4 to 2	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
1, 0	PORT	These bits set the port which operates as a management port. In this LSI chip, the internal interface port (port 2) must be set as a management port, so be sure to set these bits to 2'b10.

Note 1. Transfer of BPDU frames is given priority over each frame to be forcibly transferred by the management TAG. Accordingly, when the management TAG is used for forcible transfer, set PORTMASK to 0.

13.3.3.7 Mode Configuration Register (MODE_CONFIG)

This register is used to reset the statistics counter in the Ethernet switch.

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

MODE_CONFIG — Address 4007 0024H, Initial value 0000 0000H

Bit	31	30 to 0
Name	STATSRESET	(RESERVED)
R/W	R/W	R/W

Bit Position	Bit Name	Description
31	STATSRESET	This bit is for resetting the statistics counter in the Ethernet switch (4007 0300H to 4007 0324H). When 1 is written to this bit, the statistics counter is reset. After that, this bit is also cleared to 0.
30 to 0	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.

13.3.3.8 VLAN Tag ID Register (VLAN_TAG_ID)

This register sets the VLAN tag ID for the identification of VLAN tagged frames. IEEE802.1Q defines the VLAN tag ID as 0x8100. Since the initial value of this register is 0x8100, do not write a new value.

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

VLAN_TAG_ID — Address 4007 0034H, Initial value 0000 8100H

Bit	31 to 16	15 to 0
Name	(RESERVED)	VLANTAGID
R/W	R/W	R/W

Bit Position	Bit Name	Description
30 to 16	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
15 to 0	VLANTAGID	These bits define the VLAN tag ID for the identification of VLAN tagged frames. Since the VLAN tag ID is defined as 0x8100 in IEEE8021.Q, leave the value at the initial value.

13.3.3.9 Output Queue Management Status Register (OQMGR_STATUS)

This register indicates the state of the output queue of the Ethernet switch. The latched values of bits 1 and 3 can be cleared by writing any value to the register.

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

OQMGR_STATUS	31 to 16	15 to 7	6	5 to 4	3	2	1	0
	CELLAVILABLE	(RESERVED)	DEQUEGRANT	(RESERVED)	MEMFULL_LT	MEMFULL	NOCELL	BUSYINIT
R/W	R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W	R/W R/W R/W R/W R/W R/W R/W R/W R/W	R/W	R/W R/W	R/W	R/W	R/W	R/W

Address 4007 0080H

Initial value 0020 004AH

Bit Position	Bit Name	Description
31 to 16	CELLAVILABLE	These bits indicate the number of the currently available memory cells in realtime.
15 to 7	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
6	DEQUEGRANT	This bit indicates whether the input data are currently de-queued. This bit is usually set to 1 and it is cleared to 0 when the memory becomes full.
5 to 4	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
3	MEMFULL_LT	This bit indicates the latched value of the result of MEMFULL. Even if the MEMFULL bit is returned to 0 from 1, this bit retains the value 1.
2	MEMFULL	This bit indicates whether the memory is currently full. This bit is set to 1 if the number of memory cells is fewer than the minimum memory size set in the output queue minimum memory register (QMGR_MINCELLS). Unlike the NOCELL bit, this does not indicate an error, and the memory controller continues to operate normally. However, the switch stops operation of the input ports to prevent overflowing of the memory.
1	NOCELL	This bit is set to 1 when the remaining number of memory cells of the output queue becomes 0. Since this bit is always set to 1 after a reset, write any value to the register on completion of initialization (when the BUSYINIT bit becomes 0) to clear it. This bit becoming 1 during operation indicates a fatal error. It does not become 1 as long as the hardware is operating correctly. If this bit becomes 1 for some reason, apply a reset. Also set the output queue minimum memory register (QMGR_MINCELLS) to a greater value.
0	BUSYINIT	This bit being set to 1 indicates the state that the memory controller in the Ethernet switch has initialized the memory. This bit is set to 1 after a reset and cleared to 0 after initialization by the memory controller has been completed. Do not enable the Ethernet switch until initialization by the memory controller is completed.

13.3.3.10 Output Queue Minimum Memory Register (QMGR_MINCELLS)

This register sets the minimum amount of memory to be secured for the output queue of the Ethernet switch. The setting must allow a sufficient margin to avoid underflows of the memory.

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

	31 to 5	4 to 0	
QMGR_MINCELLS	(RESERVED)	MINCELLS	Address 4007 0084H Initial value 0000 0009H
R/W	R/W (each bit)	R/W (each bit)	

Bit Position	Bit Name	Description
31 to 5	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
4 to 0	MINCELLS	These bits set the threshold number of vacant cells in the memory for the output queue. If fewer vacant cells than this value are available, the switch stops operation of the input ports and discards received frames. Set the size for a margin allowing at least one full-sized frame to be secured. Note that a memory underflow is a fatal error and requires a reset. Each cell has 256 bytes and the initial setting is 9 (2.3 Kbytes).

13.3.3.11 Output Queue Minimum Memory Statistics Register (QMGR_ST_MINCELLS)

This register indicates the minimum number of vacant cells in the memory for the output queue of the Ethernet switch.

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

	31 to 5	4 to 0	
QMGR_ST_MINCELLS	(RESERVED)	STMINCELLS	Address 4007 0088H Initial value 0000 0000H
R/W	R/W (each bit)	R/W (each bit)	

Bit Position	Bit Name	Description
31 to 5	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
4 to 0	STMINCELLS	This bit indicates the number of vacant cells when the number of vacant cells of the output queue memory reaches the smallest number while the Ethernet switch is operating. The value is reset when any value is written to the register.

13.3.3.12 Output Queue Congestion Status Register (QMGR_CGS_STAT)

This register indicates the state of congestion of (concentration of access to) each port of the Ethernet switch.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QMGR_CGS_STAT	(RESERVED)																													P2CGS	P1CGS	P0CGS
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address 4007 008CH
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 3	(RESERVED)	Reserved. Ignore reading.
2	P2CGS	This bit indicates the congested state of the queue of the internal interface port (port 2). 0: Vacant 1: Congested
1	P1CGS	This bit indicates the congested state of the queue of port 1. 0: Vacant 1: Congested
0	P0CGS	This bit indicates the congested state of the queue of port 0. 0: Vacant 1: Congested

13.3.3.13 Internal Queue Interface Status Register (QMGR_IFACE_STAT)

This register indicates the state of the Rx and Tx FIFO buffers for each port of the Ethernet switch. It represents the result of the handshaking of signals within the switch.

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

QMGR_IFACE_STAT — Address 4007 0090H, Initial value 0000 0007H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QMGR_IFACE_STAT	(RESERVED)													P2RXFIFOAV	P1RXFIFOAV	P0RXFIFOAV	(RESERVED)													P2TXFIFOST	P1TXFIFOST	P0TXFIFOST
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Description
31 to 19	(RESERVED)	Reserved. Ignore reading.
18	P2RXFIFOAV	Indicates whether RX FIFO data for the internal interface port (port 2) are available 0: Not available 1: Available
17	P1RXFIFOAV	Indicates whether RX FIFO data for port 1 are available 0: Not available 1: Available
16	P0RXFIFOAV	Indicates whether RX FIFO data for port 0 are available 0: Not available 1: Available
15 to 3	Reserved	Ignore reading.
2	P2TXFIFOST	Indicates whether the TX FIFO buffer for the internal interface port (port 2) is ready. 0: Not ready 1: Ready
1	P1TXFIFOST	Indicates whether the TX FIFO buffer for port 1 is ready. 0: Not ready 1: Ready
0	P0TXFIFOST	Indicates whether the TX FIFO buffer for port 0 is ready. 0: Not ready 1: Ready

13.3.3.14 Queue Weight Register (QMGR_WEIGHTS)

This register sets the weight (priority) for the output queue of the Ethernet switch. Each port has four queues and a weight can be set for each queue. The set weights are common to all ports.

If the weight of all queues is set to 0, the priority takes the form of an absolute priority based on the queue number. That is, queue 3 has the highest priority.

If you are not using this absolute priority, set the weights for queues 0, 1, 2, and 3 to 1, 2, 4, and 8 in general.

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

	31 to 29	28 to 24	23 to 21	20 to 16	15 to 13	12 to 8	7 to 5	4 to 0	
QMGR_WEIGHTS	(RESERVED)	QUEUE3	(RESERVED)	QUEUE2	(RESERVED)	QUEUE1	(RESERVED)	QUEUE0	Address 4007 0094H Initial value 0000 0000H
R/W	R/W R/W R/W	R/W R/W R/W R/W R/W	R/W R/W R/W	R/W R/W R/W R/W R/W	R/W R/W R/W	R/W 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 Position	Bit Name	Description
31 to 29	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
28 to 24	QUEUE3	Set the weight for queue 3. Valid values per queue are between 0 and 12 (although 5 bits are writable per queue, the full range of values cannot be used).
23 to 21	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
20 to 16	QUEUE2	Set the weight for queue 2. Valid values per queue are between 0 and 12 (although 5 bits are writable per queue, the full range of values cannot be used).
15 to 13	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
12 to 8	QUEUE1	Set the weight for queue1. Valid values per queue are between 0 and 12 (although 5 bits are writable per queue, the full range of values cannot be used).
7 to 5	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
4 to 0	QUEUE0	Set the weight for queue 0. Valid values per queue are between 0 and 12 (although 5 bits are writable per queue, the full range of values cannot be used).

13.3.3.15 VLAN Priority Register n (VLAN_PRIORITYn)

The Ethernet switch has a programmable priority lookup table with eight entries for each port of the switch. The priority included in the three higher-order bits of the first octet of the VLAN tag is used as an index for the lookup table and the priority can be re-mapped.

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

Caution: The range of the values that can be set is 0 to 3 for the given priority. Always write 0 to the third bit.

	31 to 24	23 to 21	20 to 18	17 to 15	14 to 12	11 to 9	8 to 6	5 to 3	2 to 0	Address
VLAN_PRIORITYn	(RESERVED)	PRIORITY7	PRIORITY6	PRIORITY5	PRIORITY4	PRIORITY3	PRIORITY2	PRIORITY1	PRIORITY0	4007 0100H + 0004H*n Initial value 0000 0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31 to 24	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
23 to 21	PRIORITY7	Set the priority to be set for priority 7 of the VLAN tag of an input frame.
20 to 18	PRIORITY6	Set the priority to be set for priority 6 of the VLAN tag of an input frame.
17 to 15	PRIORITY5	Set the priority to be set for priority 5 of the VLAN tag of an input frame.
14 to 12	PRIORITY4	Set the priority to be set for priority 4 of the VLAN tag of an input frame.
11 to 9	PRIORITY3	Set the priority to be set for priority 3 of the VLAN tag of an input frame.
8 to 6	PRIORITY2	Set the priority to be set for priority 2 of the VLAN tag of an input frame.
5 to 3	PRIORITY1	Set the priority to be set for priority 1 of the VLAN tag of an input frame.
2 to 0	PRIORITY0	Set the priority to be set for priority 0 of the VLAN tag of an input frame.

Remark: n = 0 to 2
n = 0: Port 0, n = 1: Port 1, n = 2: Internal interface port (port 2)

13.3.3.16 IP Priority Register (IP_PRIORITYn)

The Ethernet switch has a CoS (class of service) table of IPv4 and IPv6 for each port of the switch. On the IPv4 CoS table, the 6-bit DiffServ field included in frames is used as an index for the lookup table and the 2-bit priority information can be set. On the IPv6 CoS table, the 8-bit CoS field included in frames is used as an index and the 2-bit priority information can be set. This register is used to set and refer to the CoS table. The CoS table can be set by writing to this register and the table can be referenced by reading from the register.

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

IP_PRIORITYn	31	30 to 11	10 to 9	8	7 to 0
Field	READ	(RESERVED)	PRIORITY	IPV6SELECT	ADDRESS
R/W	R/W	R/W (each bit)	R/W R/W	R/W	R/W (each bit)

Address: 4007 0140H + 0004H*n
Initial value: 0000 0000H

Bit Position	Bit Name	Description
31	READ	This bit is for switching operations for reading from and writing to the CoS table. When this register is written: 0: The priority information is written to the CoS table. 1: No priority information is written to the CoS table (no change). When this register is read: The priority information on the address to which writing most recently proceeded is always read from the CoS table. To read the priority information from the CoS table, follow the procedure below. - Set this bit to 1 and set the IPV6SELECT and ADDRESS bits to desired values to perform write operation (fix the address to be read). - Perform read operation.
30 to 11	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
10 to 9	PRIORITY	When writing, the priority information for writing to the CoS table is set. When read, the priority information written to the CoS table can be read. The address index for the CoS table to be read is the address to which writing most recently proceeded.
8	IPV6SELECT	This bit indicates that the CoS table of IPv6 is selected. 0: The CoS table of IPv4 is accessed. The valid range of the ADDRESS bits is from 0 to 63. 1: The CoS table of IPv6 is accessed. The valid range of the ADDRESS bits is from 0 to 255.
7 to 0	ADDRESS	These bits indicate the address as an index for the CoS table. The IPv4 priority table has 6 bits (64 entries). The IPv6 table has 8 bits (256 entries).

Remark: n = 0 to 2
n = 0: Port 0, n = 1: Port 1, n = 2: Internal interface port (port 2)

13.3.3.17 PRIORITY Configuration Register (PRIORITY_CFGn)

This register sets which priority field in frames is used to reallocate received frames according to the priority of queues for each port within the switch. When priority fields of multiple types are enabled, processing for reallocation of the priority proceeds in order of IP priority (DiffServ or CoS), then VLAN priority, and then the default priority.

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

	31 to 7	6 to 4	3, 2	1	0	
PRIORITY_CFGn	(RESERVED)	DEFAULTPRI	(RESERVED)	IPEN	VLANEN	Address 4007 0180H + 0004H*n Initial value 0000 0000H
R/W	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31 to 7	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
6 to 4	DEFAULTPRI	These bits set the default priority of frames to be received through port n. This priority is used when neither IP priority nor VLAN priority is used. The range of allowable settings is from 0 to 3. Always write 0 to the third bit.
3, 2	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
1	IPEN	This bit enables the IP priority for frames received through port n. 0: The IP priority is not used. The IP Diffserv/COS fields of frames is ignored 1: The IP DiffServ/COS field of frames is used to determine the priority within the switch. The priority is determined in accord with the settings of the IP_PRIORITYn register for the port through which frames were received.
0	VLANEN	This bit enables the VLAN priority for frames received through port n. 0: The VLAN priority is not used. The VLAN tag priority field of frames is ignored. 1: The VLAN tag priority field of frames is used to determine the priority within the switch. The priority is determined in accord with the settings of the VLAN_PRIORITYn register for the port through which frames were received.

Remark: n = 0 to 2
n = 0: Port 0, n = 1: Port 1, n = 2: Internal interface port (port 2)

13.3.3.18 HUB Control Register (HUB_CONTROL)

This register selects the hub operation. Enabling the hub allows cut-through transfer at high speed.

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

HUB_CONTROL	31 to 8	7 to 4	3	2	1	0
Field	(RESERVED)	HUBIPG	BROCAFILEN	DIR1TO0EN	DIR0TO1EN	HUBEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 4007 01C0H

Initial value 0000 00A0H

Bit Position	Bit Name	Description
31 to 8	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
7 to 4	HUBIPG	These bits set the size (number of octets) of the inter-packet gaps (IPGs) to be inserted between frames when the hub transmits consecutive frames. Set the value of the gap to be actually inserted minus two. In Ethernet transfer by default, the IPG is 12 octets, so set 10 in this register. The valid settings are from 6 to 13. Note: The hub must be disabled if any port is operating as half-duplex.
3	BROCAFILEN	This bit enables or disables the broadcast filter. When set, the hub will not transfer any broadcast frames (usually the bit should be set). 0: Disabled 1: Enabled
2	DIR1TO0EN	This bit enables or disables transfer from port 1 to port 0 through the hub. When set, all traffics received by port 1 are transferred to port 0 through the hub while the hub is active. 0: Disabled 1: Enabled
1	DIR0TO1EN	This bit enables or disables transfer from port 0 to port 1 through the hub. When set, all traffics received by port 0 are transferred to port 1 through the hub while the hub is active. 0: Disabled 1: Enabled
0	HUBEN	This bit enables or disables the hub. 0: Disabled 1: Enabled Note: When this bit is set to 0 and the FORCEFOW bit in the HUB_FLT_MACnhi register is set to the disabled setting, the hub module is reset. Note: Even if the hub is disabled, forcible transfer that can be enabled by using the individual HUB_FLT_MACnhi registers is not disabled.

Remark: The DIR1TO0EN and DIR0TO1EN bits can be enabled at the same time. That is, bidirectional, simultaneous transfer is possible.

13.3.3.19 HUB status register (HUB_STATS)

This register indicates the number of frames transferred through the hub from one port to another port. When the setting for transfer is disabled (i.e., the DIR1TO0EN or DIR0TO1EN bit of the HUB_CONTROL register is 0), the counter of the corresponding channel is cleared.

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

HUB_STATS	31 to 16	15 to 0
Field	NUM1TO0	NUM0TO1
R?W	R R R R R R R R R R R R R R R R	R R R R R R R R R R R R R R R R

Address 4007 01C4H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 16	NUM1TO0	Number of frames transferred through the hub from port 1 to port 0 (when the DIR1TO0EN bit of the HUB_CONTROL register is 1).
15 to 0	NUM0TO1	Number of frames transferred through the hub from port 0 to port 1 (when the DIR1TO1EN bit of the HUB_CONTROL register is 1).

13.3.3.20 MAC Address Low Register for HUB Input Filter (HUB_FLT_MACnlo)

This register sets the MAC address to be filtered by the hub. The first four octets of the MAC address is set in the HUB_FLT_MACnlo register and the remaining two octets are set in the HUB_FLT_MACnhi register. Up to seven octets of the MAC address can be set. If any of the set MAC addresses matches the destination address of the received frame, that frame is not transferred via the hub. The setting of the MAC address if this register is not to be used must be 0.

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

	31–24	23–16	15–8	7–0	
HUB_FLT_MACnlo	MACADD4n	MACADD3n	MACADD2n	MACADD1n	Address 4007 01C8H+ 0008H*n Initial value 0000 0000H Note 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	R/W R/W R/W R/W 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 Position	Bit Name	Description
31 to 24	MACADD4n	4th byte of MAC address n
23 to 16	MACADD3n	3rd byte of MAC address n
15 to 8	MACADD2n	2nd byte of MAC address n
7 to 0	MACADD1n	1st byte of MAC address n

Remark: n = 0 to 6

Note. n = 0 to 5: The initial value is 0000 0000H.

n = 6: The initial value is 006C 2101H. The initial value means that the destination address of the beacon frame is set. When the DLR function is used, this register must hold the destination address of the beacon frame.

13.3.3.21 MAC Address High Register for HUB Input Filter (HUB_FLT_MACnhi)

This register sets the MAC address to be filtered by the hub. The first four octets of the MAC address is set in the HUB_FLT_MACnlo register and the remaining two octets are set in the HUB_FLT_MACnhi register. Up to seven octets of the MAC address can be set. If any of the set MAC addresses matches the destination address of the received frame, that frame is not transferred via the hub. The setting of the MAC address if this register is not to be used must be 0 and the MASKCOMP bit must be set to 0xFF.

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

	31–25	24	23–16	15–8	7–0	Address
HUB_FLT_MACnhi	(RESERVED)	FORCEFOW	MASKCOMP	MACADD6n	MACADD5n	4007 01CCH+ 0008H*n
R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000 0000H Note

Bit Position	Bit Name	Description
31 to 25	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
24	FORCEFOW	This bit enables or disables forcible transfer. 0: Disabled 1: Enabled When forcible transfer is enabled, the frame with a matching MAC address is not filtered and is transferred via the hub. On the other hand, this frame is not transferred to the switch. That is, the switch does not receive this frame since it is forcibly transferred before being processed by the MAC within the switch. However, the DLR module can receive this module. Note that forcible transfer is enabled even while the hub module is disabled (i.e., the HUBEN bit of the HUB_CONTROL register is set to 0).
23 to 16	MASKCOMP	These eight bits are masking bits for the last byte (sixth byte) of the MAC address (MACADD6n). Only those bits of the sixth byte for which the setting of the corresponding bit of MASKCOMP is 1 are targets for comparison in filtering. Bits of MACADD6n for which the MASKCOMP setting is 0 are not compared. In actual processing, the logical AND of the last byte of a received MAC address and these bits is taken and the result is compared with the MACADD6n bits. Accordingly, set the MACADD6n bits corresponding to bits with the setting 0 in MASKCOMP to 0.
15 to 8	MACADD6n	These bits set the 6th byte of MAC address n. When the MASKCOMP bit is set to a value other than 0xFF, these bits corresponding to bits with the setting 0 must be set to 0.
7 to 0	MACADD5n	These bits set the 5th byte of MAC address n.

Remark: n = 0 to 6

Note. n = 0 to 5: The initial value is 0000 0000H.

n = 6: The initial value is 01FF 0100H. The initial value means that forced transfer is enabled and the destination address of the beacon frame is set. When the DLR function is used, this register must hold the destination address of the beacon frame.

13.3.3.22 Switch Statistics Registers

These registers hold the statistics of the frame processed by the Ethernet switch.

All registers are 32-bit, read-only and the initial value is 0000 0000H.

Address	Symbol	Description
4007 0300H	TOTAL_BYT_FRM	The total number of bytes of received frames which were processed by the switch and have not been discarded (sum of bytes in the frames counted by TOTAL_FRM)
4007 0304H	TOTAL_BYT_DISC	The total number of bytes of received frames which were processed by the switch but have been discarded (sum of the bytes in the frames counted by TOTAL_DISC)
4007 0308H	TOTAL_FRM	The number of received frames which were processed by the switch and have not been discarded
4007 030CH	TOTAL_DISC	The number of received frames which were processed by the switch but have been discarded
4007 0310H + 0008H*n	ODISCn	The number of frames for transmission which have been discarded at port n due to congestion in the output queue.
4007 0314H + 0008H*n	IDISC_BLOCKEDn	The number of received frames which have been discarded at port n after learning since it is configured in blocking mode.

Remark: n = 0 to 2
n = 0: Port 0, n = 1: Port 1, n = 2: Internal interface port (port 2)

13.3.4 Learning Interface Registers

The source address and port information which the Ethernet switch has learned can be obtained through the learning interface. The information is used to construct a lookup table. The information can be obtained from the two registers, but the LRN_REC_A register must be read before the LRN_REC_B register.

If the next learning information is available after access to the LRN_REC_B register, that information is set in the LRN_REC_A and LRN_REC_B registers from the FIFO buffer.

- Access These registers are readable in 32-bit units.

13.3.4.1 Learning Record A Register (LRN_REC_A)

LRN_REC_A	31 to 24	23 to 16	15 to 8	7 to 0
	SRCADD4n	SRCADD3n	SRCADD2n	SRCADD1n
R/W	R R R R R R R R	R R R R R R R R	R R R R R R R R	R R R R R R R R

Address 4007 0500H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 24	SRCADD4n	4th octet of the source MAC address
23 to 16	SRCADD3n	3rd octet of the source MAC address
15 to 8	SRCADD2n	2nd octet of the source MAC address
7 to 0	SRCADD1n	1st octet of the source MAC address

13.3.4.2 Learning Record B Register (LRN_REC_B)

LRN_REC_B	31 to 28	27 to 24	23 to 16	15 to 8	7 to 0
	(RESERVED)	PORT	HASH	SRCADD6	SRCADD5
R/W	R R R R	R R R R	R R R R R R R R	R R R R R R R R	R R R R R R R R

Address 4007 0504H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 28	(RESERVED)	Reserved. Ignore reading.
27 to 24	PORT	Port number
23 to 16	HASH	Hash code for the source MAC address
15 to 8	SRCADD6	6th octet of the source MAC address
7 to 0	SRCADD5	5th octet of the source MAC address

13.3.4.3 Learning Data Status Register (LRN_STATUS)

This register indicates whether the values of the LRN_REC_A and LRN_REC_B registers are valid.

- Access This register is readable in 32-bit units

Register	Bits 31 to 1	Bit 0	
LRN_STATUS	(RESERVED)	LERNAVAL	Address 4007 0508H Initial value 0000 0000H
R/W	R (each bit)	R	

Bit Position	Bit Name	Description
31 to 1	(RESERVED)	Reserved. Ignore reading.
0	LERNAVAL	Learning data valid status 0: Invalid 1: Valid

13.3.4.4 Address Table (ADR_TABLE)

The address table consists of blocks of 256 entries. Each block has eight records, each of which contains 64-bit information. A 64-bit record contains the 48-bit MAC address, information required for transfer to proceed, priority information, and a timestamp. The hash code calculated from the MAC address refers to the start address of a block of eight entries. For details of the address table, see sections 13.4.1.4(3) and 13.5.3.

13.3.5 Mac Port Registers

These registers are for the MACs of ports 0 and 1. Ports 0 and 1 share most of the registers (the exceptions being the command configuration registers and statistics registers). These registers are mapped to address ranges set for each port and are read and written at the addresses in those ranges. Shared registers are indicated by the word “shared” following their names in the headings.

13.3.5.1 Command Configuration Register n (COMMAND_CONFIGn)

These registers are used to set and reset the MAC.

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

COMMAND_CONFIGn — Address 4007 8008H+ 2000H*n — Initial value 0000 0010H

Bit	31	30 to 27	26	25	24	23	22 to 15	14	13	12 to 10	9	8 to 6	5	4	3 to 2	1	0
Name	CNTRESET	(RESERVED)	RXERRDISC	(RESERVED)	NOLGTHCHK	CNTRLREMEN	(RESERVED)	(RESERVED)	SWRESET	(RESERVED)	(RESERVED)	(RESERVED)	(RESERVED)	(RESERVED)	(RESERVED)	RXENA	TXENA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R/W	R	R	R/W	R/W	R/W

Bit Position	Bit Name	Description
31	CNTRESET	Self-Clearing Counter Reset Command When 1 is written to this bit, all the statistic counters are cleared to 0. After that, this bit is automatically returned to 0. Note: These registers are not shared by MAC0 and MAC1, but this bit is an exception: Writing 1 to this bit in either the register for MAC0 or MAC1 leads to clearing of the statistics counters of both MACs.
30 to 27	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
26	RXERRDISC	Receive Error Frame Discard Enable 0: Errored frames are transferred to another port with RX_ER asserted (for debugging). 1: Any frame received in error is discarded in the Core and not transferred to another port. Note: In this LSI chip, always write 1 to this bit.
25	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.
24	NOLGTHCHK	Payload Length Check Disable 0: Enabled (for debugging) 1: Disabled Note: In this LSI chip, always write 1 to this bit.
23	CNTRLREMEN	MAC Control Frame Enable 0: MAC control frames with any opcode other than 0x0001 are discarded. 1: MAC control frames with any opcode other than 0x0001 are received and transferred to another port.
22 to 14	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
13	SWRESET	Self-Clearing Software Reset Command When 1 is written to this bit, transfer of the MAC is disabled and the receive FIFO buffer is cleared. This bit is automatically returned to 0 on completion of the software reset sequence. Note: This bit is automatically returned to 0 only when the clock signal on the line side of the both MACs is being supplied. If the clock signal on the line side is not supplied, write 0 to this bit to clear it to 0.

Bit Position	Bit Name	Description
12 to 5	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
4	(RESERVED)	Reserved. When writing to this bit, write 1. Ignore reading.
3 to 2	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
1	RXENA	This bit enables or disables the MAC receive path. 0: Disabled 1: Enabled This bit is cleared by a software reset.
0	TXENA	Enables or disables the MAC transmit path. 0: Disabled 1: Enabled This bit is cleared by a software reset.

Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

13.3.5.2 Maximum Frame Length Register n (FRM_LENGTHn) (Shared)

These registers set the maximum frame lengths. They are used to check the frame length in the MAC reception circuit. The initial value is 1522, which allows the acceptance of frames with a single VLAN tag. To provide flexibility in handling tags, the value can be changed to around 1536 in initialization. The maximum setting is 1700.

- Access These registers can be read or written in 32-bit units.

Register	31 to 14	13 to 0	Address / Initial value
FRM_LENGTHn	(RESERVED)	FRMLEN	Address 4007 8014H+ 2000H*n Initial value 0000 05F2H
R/W	R/W (all bits 31 to 14)	R/W (all bits 13 to 0)	

Bit Position	Bit Name	Description
31 to 14	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
13 to 0	FRMLEN	Maximum Frame Length

Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

13.3.5.3 FIFO Buffer Threshold Register n (Shared)

These registers set the threshold of the FIFO buffer of the MAC and manage overflow and underflow. Basically, there is no need to change the initial value.

Address	Symbol	Initial Value	R/W	Description
4007 801CH + 2000H*n	RX_SECTION_EMPTYn	0000 0000H	R	This is the threshold to indicate that the receive FIFO buffer is nearly full. This value is generally used to control the transmission of pause frames, but they are not generated if the setting is 0. In this LSI chip, the value cannot be changed from 0.
4007 8020H + 2000H*n	RX_SECTION_FULLn	0000 0000H	RW	This is the threshold to indicate that there are enough entries to read from the reception FIFO buffer. When the setting is 0, store-and-forward is used. In this LSI chip, the setting should always be 0.
4007 8024H + 2000H*n	TX_SECTION_EMPTYn	0000 0048H	RW	This is the threshold to indicate that the transmit FIFO buffer is nearly full.
4007 8028H + 2000H*n	TX_SECTION_FULLn	0000 0014H	RW	This is the threshold to indicate that there are enough entries to start transmission of frames from the transmission FIFO buffer.
4007 802CH + 2000H*n	RX_ALMOST_EMPTYn	0000 0008H	R	This is the threshold for the number of entries yet to be read before the reception FIFO buffer is empty. The value is used to stop the FIFO buffer from underflowing. In this LSI chip, the value cannot be changed.
4007 8030H + 2000H*n	RX_ALMOST_FULLn	0000 0005H	R	This is the threshold for the number of entries yet to be written before the reception FIFO buffer is full. The value is used to stop the FIFO buffer from overflowing. In this LSI chip, the value cannot be changed.
4007 8034H + 2000H*n	TX_ALMOST_EMPTYn	0000 0004H	R	This is the threshold for the number of entries yet to be read before the transmission FIFO buffer is empty. The value is used to stop the FIFO buffer from underflowing. In this LSI chip, the value cannot be changed.
4007 8038H + 2000H*n	TX_ALMOST_FULLn	0000 0010H	R	This is the threshold for the number of entries yet to be written before the transmission FIFO buffer is full. The value is used to stop the FIFO buffer from overflowing. In this LSI chip, the value cannot be changed.

Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

13.3.5.4 MAC Status Register (MAC_STATUSn) (Shared)

These registers indicate the communications settings for the MAC.

- Access These registers can be read or written in 32-bit units.

MAC_STATUSn

Address: 4007 8058H + 2000H*n

Initial value: 0000 0000H

Bit	31 to 15	14	13	12	11	10	9	8	7 to 0
Name	(RESERVED)	HDPP1	(RESERVED)	SPEEDP1	(RESERVED)	HDPP0	(RESERVED)	SPEEDP0	(RESERVED)
R/W	R	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Description
31 to 15	(RESERVED)	Reserved. Ignore reading.
14	HDPP1	Setting of Duplex for MAC port 1 0: Full duplex 1: Half duplex
13	(RESERVED)	Reserved. Ignore reading.
12	SPEEDP1	Setting of the link speed for MAC port 1. 0: 10 or 100 Mbps 1: 1 Gbps
11	(RESERVED)	Reserved. Ignore reading.
10	HDPP0	Setting of Duplex for MAC port 0 0: Full duplex 1: Half duplex
9	(RESERVED)	Reserved. Ignore reading.
8	SPEEDP0	Setting of the link speed for MAC port 0. 0: 10 or 100 Mbps 1: 1 Gbps
7 to 0	(RESERVED)	Reserved. Ignore reading.

Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

13.3.5.5 Transmit IPG Length Register n (TX_IPG_LENGTHn) (Shared)

These registers set the inter-packet gap (IPG) in transmission.

- Access These registers can be read or written in 32-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TX_IPG_ LENGTHn	(RESERVED)																											TXIPGLEN					4007 805CH+ 2000H*n Initial value 0000 000CH
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31 to 5	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
4 to 0	TXIPGLEN	These bits set a value of the inter-packet gap (IPG) (byte times) in transmission. The range of allowable settings is from 8 to 27. If a value below 8 or above 27 is programmed, it is set to 12.

Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

13.3.5.6 MAC RX/TX Statistic Counters

These registers hold the statistics of the frame processed by the Ethernet switch for each port.

All registers are 32-bit, read-only and the initial value is 0000 0000H.

- Access These registers can be read or written in 32-bit units.

(1) MAC RX Statistic Counters

(1/2)

Address	Symbol	Description
4007 8100H + 2000H*n	etherStatsOctets_n	Total number of octets in frames which have been received through port n (including normal and abnormal frames)
4007 8104H + 2000H*n	OctetsOK_n	Total number of octets in normal frames which have been received through port n. It is an alternative to ifInOctets of the MIB counter.
4007 8108H + 2000H*n	aAlignmentErrors_n	Number of frames received through port n in which a start-of-frame delimiter (SFD) was not detected in the frame even though RX_DV has been de-asserted.
4007 810CH + 2000H*n	aPAUSEMACCtrlFrames_n	Number of normal pause frames received through port n
4007 8110H + 2000H*n	FramesOK_n	Number of normal frames received through port n
4007 8114H + 2000H*n	CRCErrors_n	Number of frames received through port n which have an abnormal CRC but are of normal length.
4007 8118H + 2000H*n	VLANOK_n	Number of frames received through port n which have a normal VLAN tag
4007 811cH + 2000H*n	ifInErrors_n	Number of frames which had any of the following errors in reception through port n: - FIFO overflow - CRC error - Payload length error - Jabber or oversized error - PHY errors (RX_ER asserted)
4007 8120H + 2000H*n	ifInUcastPkts_n	Number of normal unicast frames received through port n
4007 8124H + 2000H*n	ifInMulticastPkts_n	Number of normal multicast frames received through port n
4007 8128H + 2000H*n	ifInBroadcastPkts_n	Number of normal broadcast frames received through port n
4007 812CH + 2000H*n	etherStatsDropEvents_n	Number of frames for which reception through port n was impossible due to insufficient FIFO capacity.
4007 8130H + 2000H*n	etherStatsPkts_n	All frames received through port n (including normal and abnormal frames)
4007 8134H + 2000H*n	etherStatsUndersizePkts_n	Number of frames received through port n which have 64 or fewer bytes and a normal CRC. However, frames with 24 or fewer bytes are not included.

(2/2)

Address	Symbol	Description
4007 8138H + 2000H*n	etherStatsPkts64Octets_n	Number of frames received through port n which have a length of 64 bytes
4007 813CH + 2000H*n	etherStatsPkts65to127Octets_n	Number of frames received through port n which have a length of at least 65 bytes and up to 127 bytes.
4007 8140H + 2000H*n	etherStatsPkts128to255Octets_n	Number of frames received through port n which have a length of at least 128 bytes and up to 255 bytes
4007 8144H + 2000H*n	etherStatsPkts256to511Octets_n	Number of frames received through port n which have a length of at least 256 bytes and up to 511 bytes
4007 8148H + 2000H*n	etherStatsPkts512to1023Octets_n	Number of frames received through port n which have a length of at least 512 bytes and up to 1023 bytes
4007 814CH + 2000H*n	etherStatsPkts1024to1518Octets_n	Number of frames received through port n which have a length of at least 1024 bytes and up to 1518 bytes
4007 8150H + 2000H*n	etherStatsPkts1519toMax_n	Number of frames received through port n which have a length of at least 1519 bytes and up to the value of the maximum frame length register (FRM_LENTGHn)
4007 8154H + 2000H*n	etherStatsOversizePkts_n	Number of frames received through port n which have a length exceeding the value of the maximum frame length register (FRM_LENGTHn) and a normal CRC
4007 8158H + 2000H*n	etherStatsJabbers_n	Number of frames received through port n which have a length exceeding the value of the maximum frame length register (FRM_LENGTHn) and an abnormal CRC
4007 815CH + 2000H*n	etherStatsFragments_n	Number of frames received through port n which have 64 or fewer bytes and an abnormal CRC. However, frames with 24 or fewer bytes are not included. DLR beacon frames are also counted.
4007 8160H + 2000H*n	aMACControlFramesReceived_n	Number of normal frames received through port n which have 0x8808 as type
4007 8164H + 2000H*n	aFrameTooLong_n	Number of frames received through port n which have a length exceeding the value of the maximum frame length register (FRM_LENGTHn) (including normal and abnormal frames)
4007 816CH + 2000H*n	StackedVLANOK_n	Number of normal frames received through port n which have a stacked VLAN tag

(2) MAC TX Statistic Counters

(1/2)

Address	Symbol	Description
4007 8180H + 2000H*n	TXetherStatsOctets_n	Total number of octets in frames which have been received through port n (including normal and abnormal frames)
4007 8184H + 2000H*n	TxOctetsOK_n	Total number of octets in normal frames only transmitted through port n
4007 818CH + 2000H*n	TXaPAUSEMACCtrlFrames_n	Number of normal pause frames transmitted through port n
4007 8190H + 2000H*n	TxFramesOK_n	Number of normal frames transmitted through port n
4007 8194H + 2000H*n	TxCRCErrors_n	Number of frames transmitted through port n which have an abnormal CRC but are of normal length
4007 8198H + 2000H*n	TxVLANOK_n	Number of frames transmitted through port n which have a normal VLAN tag
4007 819CH + 2000H*n	ifOutErrors_n	Number of frames which had any of the following errors in transmission through port n: - TX_ER - Frame length error
4007 81A0H + 2000H*n	ifUcastPkts_n	Number of normal unicast frames transmitted through port n
4007 81A4H + 2000H*n	ifMulticastPkts_n	Number of normal multicast frames transmitted through port n
4007 81A8H + 2000H*n	ifBroadcastPkts_n	Number of normal broadcast frames transmitted through port n
4007 81ACH + 2000H*n	TXetherStatsDropEvents_n	Number of frames of insufficient size transmitted through port n. Such frames are due to insufficient FIFO capacity or collisions during half-duplex communications.
4007 81B0H + 2000H*n	TXetherStatsPkts_n	Number of all frames transmitted through port n (including normal and abnormal frames)
4007 81B4H + 2000H*n	TXetherStatsUndersizePkts_n	Number of frames transmitted through port n which have 64 or fewer bytes and a normal CRC (basically such frames are not generated)
4007 81B8H + 2000H*n	TXetherStatsPkts64Octets_n	Number of frames transmitted through port n which have a length of 64 bytes
4007 81BCH + 2000H*n	TXetherStatsPkts65to127Octets_n	Number of frames transmitted through port n which have a length of at least 65 bytes and up to 127 bytes.
4007 81C0H + 2000H*n	TXetherStatsPkts128to255Octets_n	Number of frames transmitted through port n which have a length of at least 128 bytes and up to 255 bytes
4007 81C4H + 2000H*n	TXetherStatsPkts256to511Octets_n	Number of frames transmitted through port n which have a length of at least 256 bytes and up to 511 bytes
4007 81C8H + 2000H*n	TXetherStatsPkts512to1023Octets_n	Number of frames transmitted through port n which have a length of at least 512 bytes and up to 1023 bytes

(2/2)

Address	Symbol	Description
4007 81CCH + 2000H*n	TXetherStatsPkts1024to1518Octets_n	Number of frames transmitted through port n which have a length of at least 1024 bytes and up to 1518 bytes
4007 81D0H + 2000H*n	TXetherStatsPkts1519toMax_n	Number of frames transmitted through port n which have a length of at least 1519 bytes and up to the value of the maximum frame length register (FRM_LENTGHn)
4007 81D4H + 2000H*n	TXetherStatsOversizePkts_n	Number of frames transmitted through port n which have a length exceeding the value of the maximum frame length register (FRM_LENGTHn) and a normal CRC
4007 81D8H + 2000H*n	TXetherStatsJabbers_n	Number of frames transmitted through port n which have a length exceeding the value of the maximum frame length register (FRM_LENGTHn) and an abnormal CRC
4007 81DCH + 2000H*n	TXetherStatsFragments_n	Number of frames transmitted through port n which have 64 or fewer bytes and for which the error signal was asserted.
4007 81E0H + 2000H*n	aMACControlFrames_n	Number of normal frames transmitted through port n which have 0x8808 as type
4007 81E4H + 2000H*n	TXaFrameTooLong_n	Number of frames transmitted through port n which have a length exceeding the value of the maximum frame length register (FRM_LENGTHn) (including normal and abnormal frames)
4007 81ECH + 2000H*n	aMultipleCollisions_n	Number of frames which have been successfully transmitted through port n after several collisions. It is only valid in half-duplex communications.
4007 81F0H + 2000H*n	aSingleCollisions_n	Number of frames which have been successfully transmitted through port n after a single collision. It is only valid in half-duplex communications.
4007 81F4H + 2000H*n	aLateCollisions_n	Number of frames transmitted in error through port n due to a late collision. It is only valid in half-duplex communications.
4007 81F8H + 2000H*n	aExcessCollisions_n	Number of frames which were discarded from port n due to excessive collisions (16 failures in transmission). It is only valid in half-duplex communications.

Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

13.3.6 Timer Module Registers

The switch incorporates a timer module for use in timestamping. The timer module registers indicate the settings and states of the timer module.

13.3.6.1 Timer Module Configuration Register (TSM_CONFIG)

This register controls generation of interrupts in response to the event that was generated in timestamping.

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

TSM_CONFIG	31 to 14	13	12	11 to 5	4	3	2	1	0
Bit name	(RESERVED)	IRQTXENAP1	IRQTXENAP0	(RESERVED)	IRQTEST	IRQTIMOVER	IRQEVTPERD	IRQEVTOFF	IRQENA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: 4007 C004H

Initial value: 0000 0000H

Bit Position	Bit Name	Description
31 to 14	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
13	IRQTXENAP1	This bit enables or disables generation of interrupts in response to capturing a transmission timestamp by port 1. 0: Disabled 1: Enabled (an interrupt is generated) When this bit is set to 1, an interrupt is generated when a new timestamp is stored in the transmission timestamp register for the port.
12	IRQTXENAP0	This bit enables or disables generation of interrupts in response to capturing a transmission timestamp by port 0. 0: Disabled 1: Enabled (an interrupt is generated) When this bit is set to 1, an interrupt is generated when a new timestamp is stored in the transmission timestamp register for the port.
11 to 5	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
4	IRQTEST	This bit controls generation of interrupts for testing the software. 0: Normal operation 1: An interrupt is generated at the same time as writing.
3	IRQTIMOVER	This bit controls generation of interrupts when the timer overflows. 0: An interrupt is not generated. 1: An interrupt is generated.
2	IRQEVTPERD	This bit controls generation of interrupts when the ns timer reaches one second. 0: An interrupt is not generated. 1: An interrupt is generated.
1	IRQEVTOFF	This bit controls generation of interrupts when offset correction for the timer is completed. 0: An interrupt is not generated. 1: An interrupt is generated.
0	IRQENA	This bit enables or disables timer interrupts. If this bit is not set to 1, interrupts are not generated even if another bit is set to 1. 0: Disabled 1: Enabled

13.3.6.2 Interrupt Status/ACK Register (TSM_IRQ_STAT_ACK)

This register is used for checking the state of interrupts from the timers and other sources and acknowledging the interrupts.

The state is confirmed by reading the value of this register. A value of 1 means that the interrupt has been generated and 0 means that the interrupt has not been generated.

Writing 1 to this register leads to acknowledging and clearing of the interrupt. At the same time, the value of the corresponding bit is cleared.

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

TSM_IRQ_STAT_ACK — Address 4007 C008H, Initial value 0000 0000H

Bit	31 to 14	13	12	11 to 5	4	3	2	1	0
TSM_IRQ_STAT_ACK	(RESERVED)	IRQTXP1	IRQTXP0	(RESERVED)	IRQTEST	IRQTIMOVER	IRQEVTPERD	IRQEVTOFF	IRQENA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Description
31 to 14	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
13	IRQTXP1	Indicates that the transmission timestamp capture interrupt for port 1 has been generated.
12	IRQTXP0	Indicates that the transmission timestamp capture interrupt for port 0 has been generated.
11 to 5	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
4	IRQTEST	Indicates that the interrupt for testing the software has been generated.
3	IRQTIMOVER	Indicates that the timer overflow interrupt has been generated.
2	IRQEVTPERD	Indicates that an interrupt has been generated in response to the ns timer reaching one second.
1	IRQEVTOFF	Indicates that an interrupt has been generated when the correction by the addition of an offset to the value of the timer.
0	IRQENA	Indicates that an interrupt from the timer has been generated.

13.3.6.3 Port Timestamp Control/Status Register (PORTn_CTRL)

This register sets the method of storing timestamps acquired by port n in the timestamp register and indicates the state of the stored timestamp.

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

	31 to 3	2	1	0	Address
PORTn_CTRL	(RESERVED)	TSKEEP	TSOVR	TSVALID	4007 C020H+ 0008H*n
R/W	R/W	R/W	R/W	R/W	Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 3	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
2	TSKEEP	This bit sets which timestamp the timestamp register is to hold when new timestamps are received. 0: A new timestamp is written. 1: The last timestamp is retained. When this bit is set to 1, the stored timestamp is retained. The timestamp is processed by software and new timestamps are ignored until the value of the TSVALID bit becomes 0.
1	TSOVR	This bit indicates whether a new timestamp has been written over the last timestamp to have been stored. 0: A new timestamp has not been written. 1: A new timestamp has been written. When a valid timestamp is stored (TSVALID = 1), this bit is set to 1 if another timestamp is received before the previous value has been read by software. This bit is cleared by writing any value to it. The TSKEEP bit being set to 1 indicates that a new stamp has been ignored even though it was received while the TSVALID bit was 1.
0	TSVALID	This bit indicates the state of the stored timestamp. 0: The timestamp is not valid. 1: The timestamp is valid. This bit is cleared by writing any value to it.

Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

13.3.6.4 Port Timestamp Register (PORTn_TIME)

This register holds timestamps acquired by port n.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
PORTn_TIME	TSREG																																4007 C024H+ 0008H*n Initial value 0000 0000H
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	

Bit Position	Bit Name	Description
31 to 0	TSREG[31:0]	Timestamps acquired by port n

Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

13.3.6.5 Timer Control Register (ATIME_CTRL)

This register is used to set timer interrupt events and control timers.

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

ATIME_CTRL — Address 4007 C120H, Initial value 0000 0000H

Bit	31 to 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	(RESERVED)	PLUS1	CAPTR	(RESERVED)	RST	(RESERVED)	(RESERVED)	(RESERVED)	EVTPERIRST	EVTPERIENA	(RESERVED)	EVTOFFENA	(RESERVED)	TMENA
R/W	R/W	R/W	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 Position	Bit Name	Description
31 to 13	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
12	PLUS1	Writing 1 to this bit increments the timer counter by 1. When this is completed, this bit is cleared.
11	CAPTR	When 1 is written to this bit, the current timer value is captured. When this is completed, this bit is cleared and the current time can be read from the ATIME and ATIME_SEC registers.
10	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.
9	RST	Writing 1 to this bit resets the timer to zero. This does not affect the counter enable signal. When the counter is enabled, writing 1 to this bit resets the timer to zero and counting continues from this value.
8 to 6	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.
5	EVTPERIRST	This bit sets a reset for the periodic event timer. 0: The timer counts up until its value wraps around. 1: When the timer counter value reaches one second, the timer is reset to zero.
4	EVTPERIENA	This bit enables or disables periodic events in one-second units. 0: Periodic events are not generated. 1: Periodic events in one-second units are generated. If generation of interrupts is set by the TSM_CONFIG register, a periodic event interrupt is also generated in response to the switch interrupt. Note: The value of the timer period should be set beforehand.
3	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.

Bit Position	Bit Name	Description
2	EVTOFFENA	This bit enables or disables offset-correction events. 0: Offset correction does not proceed. 1: Offset correction proceeds. When offset correction is completed, an offset-correction interrupt is generated and this bit is cleared to 0 if generation of interrupts has been set by the TSM_CONFIG register. To proceed with offset correction again, this bit must be set to 1 again. Note: The timer offset value should be set beforehand.
1	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.
0	TMENA	0: The timer stops at the current value. 1: The timer starts counting up

Cautions 1. Bits 12, 11, and 9 are for issuing commands and are used to start up the directly corresponding events. When the command bits are set, retaining the values of the other bits in the register is not necessary (that is, read-modify-write operations are not required). When the command is completed, the corresponding bit is cleared to 0. If any of the command bits is not set to 0, the values of the other bits will be ignored.

2. The timer value cannot be captured accurately while the timer is stopped. The captured value is invalid.

13.3.6.6 Timer Nanoseconds Register (ATIME)

This register represents the value of the nanoseconds timer. The time setting of the nanoseconds timer and the captured time can be acquired.

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

Register	Bits 31 to 0	Address / Initial value
ATIME	TMR	Address 4007 C124H Initial value 0000 0000H
RW	RW (all bits 31 to 0)	

Bit Position	Bit Name	Description
31 to 0	TMR[31:0]	These bits represent the setting of the nanoseconds timer and the value read. When this register is written: The time is set in the timer. When this register is read: The last captured value is returned. To read the current value, the capture command (CAPTR bit) in the ATIME_CTRL register must be issued. The value is in units of nanoseconds (ns).

Caution: The seconds value must be programmed into the ATIME_SEC register before writing to this register.

13.3.6.7 Timer Offset Correction Register (ATIME_OFFSET)

This register sets the value of an offset for use in timer correction. There are two methods of offset correction and the value to be set in this register will differ with the method.

Since offset correction starts by writing to this register, set the value in the ATIME_OFFS_CORR register beforehand to fix the method of offset correction.

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

ATIME_OFFSET — Address 4007 C128H, Initial value 0000 0000H

Bits	31 to 0
Field	OFFSET
RW	RW (all bits)

Bit Position	Bit Name	Description
31 to 0	OFFSET	These are for setting the value of an offset for use in timer correction. • If ATIME_OFFS_CORR is zero Write a value for the offset. The value is in units of nanoseconds. When a value is written to this register, the written value is immediately added to the current timer value. When the value written is negative, the value is subtracted from that of the timer. • If ATIME_OFFS_CORR is non-zero Write a value for the number of times offset correction is to be applied. When a value is written to this register, offset correction starts, but the written value is not immediately added to the current timer value. Each time the timer value is corrected by the offset, the value of this register is decremented by 1. Offset correction and decrementing of the register value are repeated until the value becomes 0, and offset correction ends when the value becomes 0 and the register is then reset to 0. The offset interval and offset value are specified by the ATIME_OFFS_CORR and ATIME_INC registers respectively.

13.3.6.8 Generate Timer Periodic Event Register (ATIME_EVT_PERIOD)

This register sets the period for generating periodic events. Each time the nanoseconds timer has reached this time, the period event occurs and the nanoseconds timer restarts. The value is in units of nanoseconds (nsec). The initial value is 10^9 [nsec] = 1 [sec].

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

ATIME_EVT_PERIOD	Bits 31 to 0	Address
	TIMPEREVET	4007 C12CH
RW	RW (bits 31 to 0)	Initial value 3B9A CA00H

Bit Position	Bit Name	Description
31 to 0	TIMPEREVET	Period for generating periodic events. Set 3B9A CA00H (1 sec).

Caution: The value of periodic events is fixed to one second and cannot be changed. If changed, the timer will not operate normally.

13.3.6.9 Timer Drift Correction Register (ATIME_CORR)

This register sets the correction period over which correction for drift is applied as a number of clock cycles. Use the ATIME_INC register to specify the amount of correction.

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

ATIME_CORR	Bit 31	Bits 30 to 0	Address
	(RESERVED)	DRIFCORVAL	4007 C130H
RW	RW	RW (bits 30 to 0)	Initial value 0000 0000H

Bit Position	Bit Name	Description
31	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.
30 to 0	DRIFCORVAL	Period over which correction for drift is applied. Set the value as a number of clock cycles of 25 MHz (8 ns).

Caution: The correction value is the inverse of the difference between the frequencies (ppm) of the master and slave oscillators. The value is in units of clock cycles, not in units of nanoseconds.

13.3.6.10 Timer Increment Register (ATIME_INC)

This register sets the amount of correction for used in correction of the timer value by the offset and correction for drift.

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

ATIME_INC	31 to 23	22 to 16	15	14 to 8	7	6 to 0
Field	(RESERVED)	OFFSCORRINC	(RESERVED)	CORRINC	(RESERVED)	CLKPERD
R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 4007 C134H

Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 23	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
22 to 16	OFFSCORRINC	These bits set the amount of correction for use in offset correction. This value is added every clock cycle set in the ATIME_OFFS_CORR register.
15	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.
14 to 8	CORRINC	These bits set the amount of correction of clock cycles for use in correction for drift. This value is added every correction period set in the ATIME_CORR register. The value of these bits being lower than the value of the CLKPERD bits makes counting by the timer slower. This value being higher than the value of the CLKPERD bits makes counting by the timer faster.
7	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.
6 to 0	CLKPERD	These bits set the period in nanoseconds of the clock signal that provides the timing for acquiring timestamps. Be sure to set 8 ns (001000b). In this LSI chip, the timer operates at 125 MHz and the value cannot be changed to any other value. The timer will be incremented by the time corresponding to the value of these bits every clock cycle.

13.3.6.11 Timer Seconds Register (ATIME_SEC)

This register represents the value of the seconds timer. The time settings of the seconds timer and the time captured by the timer can be acquired. The seconds timer will be incremented when the nanoseconds timer reaches 10^9 [nsec].

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

ATIME_SEC	Bits 31 to 0	Address
	SECTIM	4007 C138H
		Initial value 0000 0000H
RW	RW (all bits 31 to 0)	

Bit Position	Bit Name	Description
31 to 0	SECTIM	These bits represent the setting of the seconds timer and the value read. When this register is written: The time is set in the timer. However, this register is not updated when it is written, but the value is stored in a local register. When the ATIME register is written, the value of the local register is loaded to this register. When this register is read: The last captured value is returned. The capture command (CAPTR bit) of the ATIME_CTRL register must be issued before reading the current value. The value is in units of seconds.

13.3.6.12 Timer Offset Correction Count Register (ATIME_OFFS_CORR)

This register sets the interval at which offset correction is applied. It is used in combination with the ATIME_OFFSET register, and dispersing the changes in time due to adding the offset over longer periods may avoid dramatic changes in the time and suppress jitter.

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

ATIME_OFFS_CORR	Bits 31 to 0	Address
	OFFCOR	4007 C13CH
		Initial value 0000 0000H
RW	RW (all bits 31 to 0)	

Bit Position	Bit Name	Description
31 to 0	OFFCOR	These bits set the interval at which offset correction is applied. The value is specified in the number of 125 MHz clock cycles

13.3.6.13 Timer Output Enable Register (SWTMEN)

This register is for enabling the time synchronization timer pulse output signal (ETHSWSYNCOUT signal).

Caution: Be sure to set the registers below before enabling the output with this register.
SWTMSTSECL, SWTMSTSECH, SWTMSTNSL, SWTMSTNSH,
SWTMPSECL, SWTMPSECH, SWTMPNSL, SWTMPNSH,
SWTMWTH, SWTMMAXPL, SWTMMAXPH

- Access This register can be read or written in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address: BASE + 1100H
SWTMEN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OUTEN	Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

Bit Position	Bit Name	Description
0	OUTEN	This bit enables the output of the ETHSWSYNCOUT signal. 0: Output disabled 1: Output enabled

13.3.6.14 Timer Seconds Start Setting Registers (SWTMSTSECL/H)

These registers set the start time for the output of the ETHSWSYNCOUT signal seconds.

- Access These registers can be read or written in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0	Address
SWTM STSECL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMSTSEC15–0	BASE + 1110H, Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	
SWTM STSECH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMSTSEC31–16	BASE + 1114H, Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

Bit Position	Bit Name	Description
15 to 0	TMSTSEC31–0	These bits set the start time for the output of the ETHSWSYNCOUT signal in seconds.

13.3.6.15 Timer Nanoseconds Start Setting Registers (SWTMSTNSL/H)

These registers set the start time for the output of the ETHSWSYNCOUT signal in nanoseconds.

- Access These registers can be read or written in 32- or 16-bit units.

SWTMSTNSL — Address: BASE + 1118H, Initial value: 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
SWTMSTNSL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMSTNS15–0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W

SWTMSTNSH — Address: BASE + 111CH, Initial value: 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
SWTMSTNSH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMSTNS31–16
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W

Bit Position	Bit Name	Description
15 to 0	TMSTNS31–0	These bits set the start time for the output of the ETHSWSYNCOUT signal in nanoseconds.

13.3.6.16 Timer Seconds Period Setting Register (SWTMPSECL/H)

These registers set the period of the ETHSWSYNCOUT signal output in seconds.

- Access These registers can be read or written in 32- or 16-bit units.

SWTMPSECL — Address: BASE + 1120H, Initial value: 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
SWTMPSECL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMPSEC15–0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W

SWTMPSECH — Address: BASE + 1124H, Initial value: 0000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
SWTMPSECH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMPSEC31–16
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W

Bit Position	Bit Name	Description
15 to 0	TMPSEC31–0	These bits set the period of the ETHSWSYNCOUT signal output in seconds.

13.3.6.17 Timer Nanoseconds Period Setting Registers (SWTMPNSL/H)

These registers set the period for the output of the ETHSWSYNCOUT signal in nanoseconds. These registers must be set to a value corresponding to the division by the value for one second set in the ATIME_EVT_PREIOD register.

Cautions 1. Be sure to set these registers before enabling the output of the ETHSWSYNCOUT signal.
2. The setting of these registers must be at least 16 ns (10H).

- Access These registers can be read or written in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0	Address
SWTM PNSL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMPNS15–0	BASE + 1128H Initial value 0000 CA00H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0	Address
SWTM PNSH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMPNS31–16	BASE + 112CH Initial value 0000 3B9AH
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

Bit Position	Bit Name	Description
15 to 0	TMPNS31–0	These bits set the period for the output of the ETHSWSYNCOUT signal in nanoseconds. The value must be a multiple of 8 ns.

13.3.6.18 Timer Pulse Width Setting Register (SWTMWTH)

This register sets the pulse width for the output ETHSWSYNCOUT signal. When the ETHSWSYNCOUT signal is connected as an interrupt signal, leave the value at the initial value. When the signal is used as an external signal, set an appropriate width.

Caution: Be sure to set this register before enabling the output of the ETHSWSYNCOUT signal.

- Access These registers can be read or written in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0	Address
SWTM WTH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMWTH15–0	BASE + 1130H Initial value 0000 0003H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

Bit Position	Bit Name	Description
15 to 0	TMWTH15–0	These bits set the pulse width for the output ETHSWSYNCOUT signal. The value must be a multiple of 8 ns. The ETHSWSYNCOUT signal is fixed to 0 while this register is set to 0H.

13.3.6.19 Timer Maximum Counter Value Registers (SWTMMAXPL/H)

These registers set the maximum value (1 second) of the counter. Set these registers to the same value as the ATIME_EVT_PERIOD register.

- Access These registers can be read or written in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0	Address
SWTM MAXPL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMMAXP15–0	BASE + 1134H Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0	Address
SWTM MAXPH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TMMAXP31–16	BASE + 1138H Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

Bit Position	Bit Name	Description
15 to 0	TMMAXP31–0	These bits set the value of the ATIME_EVT_PERIOD register (3B9A CA00H). TMMAXP31–16: 3B9AH TMMAXP15–0: CA00H

13.3.6.20 Timer Seconds Time Hold Register (SWTMLATSEC)

This register holds the seconds counter value of the IEEE1588 timer of the switch on rising edges of the ETHSWSYNCOUT signal. The value of this register is updated on every rising edge of the ETHSWSYNCOUT signal.

- Access This register is only readable in 32-bit units.

	31–0	Address
SWTM LATSEC	TMLATSEC31–0	BASE + 1140H Initial value 0000 0000H
R/W	R	

Bit Position	Bit Name	Description
31 to 0	TMLATSEC31–0	These bits latch the seconds counter value on rising edges of the ETHSWSYNCOUT signal.

13.3.6.21 Timer Nanoseconds Time Hold Register (SWTMLATNS)

This register holds the nanoseconds counter value of the IEEE1588 timer of the switch on rising edges of the ETHSWSYNCOUT signal. The value of this register is updated on every rising edge of the ETHSWSYNCOUT signal.

- Access This register is only readable in 32-bit units.

	31 to 0	
		Address
SWTM LATNS	TMLATNS31–0	BASE + 1144H
		Initial value
		0000 0000H
R/W	R	

Bit Position	Bit Name	Description
31 to 0	TMLATNS31–0	These bits latch the nanoseconds counter value on rising edges of the ETHSWSYNCOUT signal.

13.3.7 DLR Module Registers

13.3.7.1 DLR Control Register (DLR_CONTROL)

This register is used to make settings for the DLR operation.

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

	31 to 16	15 to 8	7 to 5	4	3 to 1	0
DLR_CONTROL	(RESERVED)	CYCMCLK	(RESERVED)	BECTIMOUT	(RESERVED)	DLRENA
R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 4007 E000H

Initial value 0000 3200H

Bit Position	Bit Name	Description
31 to 16	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
15 to 8	CYCMCLK	Number of cycles required per second. Since the DLR module of this LSI chip operates at 100 MHz, always set these bits to 0x64. Leave the value at the initial value.
7 to 5	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
4	BECTIMOUT	This bit is used to select ignoring beacon frames which have invalid timeout timer values. The local device will ignore and not acquire the parameters in beacon frames having values for the timeout timer that are not within the range from 200 microseconds to 500 milliseconds. If the timeout timer value is invalid, the INV_TMOUT register always acquires that value irrespective of the setting of this bit. Ignored frames will be transferred through the hub normally. 0: Not ignored 1: Ignored
3 to 1	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
0	DLRENA	This bit enables or disables the DLR module. 0: Disabled 1: Enabled

13.3.7.2 DLR Status Register (DLR_STATUS)

This register indicates the state of the DLR ring node.

- Access This register is only readable in 32-bit units.

	Bit	31 to 24	23 to 18	17	16	15 to 8	7 to 2	1	0
DLR_STATUS		NETTOPGY	(RESERVED)	LINSTAP1	LINSTAP0	CURRSTA	(RESERVED)	BEAREV1	BEAREV0
	R/W	R	R	R	R	R	R	R	R

Address 4007 E004H

Initial value 0003 0000H

Bit Position	Bit Name	Description
31 to 24	NETTOPGY	These bits indicate the current network topology. 0: Linear topology when the local node is in the idle state 1: Ring topology when local node is not in the idle state
23 to 18	(RESERVED)	Reserved. Ignore reading.
17	LINSTAP1	Indicates the linked state of port 1. 0: Port 1 link is down 1: Port 1 link is up
16	LINSTAP0	Indicates the linked state of port 0. 0: Port 0 link is down 1: Port 0 link is up
15 to 8	CURRSTA	These bits indicate the current state of the local node. 0x0: IDLE_STATE 0x1: NORMAL_STATE 0x2: FAULT_STATE Others: Not used
7 to 2	(RESERVED)	Reserved. Ignore reading.
1	BEAREV1	Indicates that a beacon frame has been received from the active supervisor through port 1. 0: No beacon frame has been received 1: A beacon frame has been received
0	BEAREV0	Indicates that a beacon frame has been received from the active supervisor through port 0. 0: No beacon frame has been received 1: A beacon frame has been received

13.3.7.3 DLR Ethernet Type Register (DLR_ETH_TYP)

This register defines the Ethernet type for detecting DLR frames. This value is compared with the type field of received frames for detection of DLR frames.

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

	31 to 16	15 to 0	
DLR_ETH_TYP	(RESERVED)	ETHTYPDLR	Address 4007 E008H Initial value 0000 80E1H
R/W	R/W (each bit)	R/W (each bit)	

Bit Position	Bit Name	Description
31 to 16	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
15 to 0	ETHTYPDLR	These bits set the Ethernet type for detection of DLR frames.

13.3.7.4 DLR Interrupt Control Register (DLR_IRQ_CTRL)

This register controls the generation of interrupts by DLR.

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

DLR_IRQ_CTRL — Address 4007 E00CH, Initial value 0000 0000H

Bit	31	30	29	28–16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ATOMICAND	ATOMICOR	(RESERVED)	(RESERVED)	IRQFRMDSP1	IRQFRMDSP0	IRQBECENA1	IRQBECENA0	IRQINVTMREN	IRQIPADDREN	IRQSUPIGENA	IRQLINKENA1	IRQLINKENA0	IRQSUPENA	IRQBECENA1	IRQBECENA0	IRQSTOPP1	IRQSTOPP0	IRQFLUENA	IRQCHNGENA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31	ATOMICAND	When this register is written, the logical AND of the setting of this bit and the enable setting bit of this register is taken and the result is written to it. 0: All bits are cleared to 0. 1: Normal write operation
30	ATOMICOR	When this register is written, the logical OR of the setting of this bit and the enable setting bit of this register is taken and the result is written to it. 0: Normal write operation 1: All bits are cleared to 1.
29 to 16	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
15	IRQFRMDSP1	This bit controls the generation of interrupts when frames are discarded when the local address matches the transmission source address at port 1. 0: No interrupt is generated. 1: An interrupt is generated.
14	IRQFRMDSP0	This bit controls the generation of interrupts when frames are discarded when the local address matches the transmission source address at port 0. 0: No interrupt is generated. 1: An interrupt is generated.
13	IRQBECENA1	This bit controls the generation of interrupts when beacon frames are detected at port 0. 0: No interrupt is generated. 1: An interrupt is generated.
12	IRQBECENA0	This bit controls the generation of interrupts when beacon frames are detected at port 1. 0: No interrupt is generated. 1: An interrupt is generated.
11	IRQINVTMREN	This bit controls the generation of interrupts when frames having values for the beacon timeout timer that are not within the specified range are detected. 0: No interrupt is generated. 1: An interrupt is generated.
10	IRQIPADDREN	This bit controls the generation of interrupts when the IP address in beacon frames output by the ring supervisor has been changed. 0: No interrupt is generated. 1: An interrupt is generated.
9	IRQSUPIGENA	This bit controls the generation of interrupts in response to the detection of beacon frames having a MAC address associated with a priority equal to or less than that of the current ring supervisor. 0: No interrupt is generated. 1: An interrupt is generated.

Bit Position	Bit Name	Description
8	IRQLINKENA1	This bit controls the generation of interrupts in response to a change in the linked state of port 1. 0: No interrupt is generated. 1: An interrupt is generated.
7	IRQLINKENA0	This bit controls the generation of interrupts in response to a change in the linked state of port 0. 0: No interrupt is generated. 1: An interrupt is generated.
6	IRQSUPENA	This bit controls the generation of interrupts in response to the change of the ring supervisor. 0: No interrupt is generated. 1: An interrupt is generated.
5	IRQBECENA1	This bit controls the generation of interrupts when the beacon timeout timer reaches the timeout time on port 1. 0: No interrupt is generated. 1: An interrupt is generated.
4	IRQBECENA0	This bit controls the generation of interrupts when the beacon timeout timer reaches the timeout time on port 0. 0: No interrupt is generated. 1: An interrupt is generated.
3	IRQSTOPP1	This bit controls the generation of interrupts when operation of the neighbor check timeout timer must be stopped for port 1. 0: No interrupt is generated. 1: An interrupt is generated.
2	IRQSTOPP0	This bit controls the generation of interrupts when operation of the neighbor check timeout timer must be stopped for port 0. 0: No interrupt is generated. 1: An interrupt is generated.
1	IRQFLUENA	This bit controls the generation of interrupts when the local MAC address must be erased from the learning table. 0: No interrupt is generated. 1: An interrupt is generated.
0	IRQCHNGENA	This bit controls the generation of interrupts when the state of the local beacon based DLR ring node has been changed. 0: No interrupt is generated. 1: An interrupt is generated. Note: The interrupt service routine must reload the parameters of the beacon frame before clearing the bit.

13.3.7.5 DLR Interrupt Status/Acknowledge Register (DLR_IRQ_STAT_ACK)

This register is used for checking the state of DLR interrupts and acknowledging the interrupts.

The state is confirmed by reading the value of this register. A value of 1 means that the event has been generated and 0 means that the event has not been generated.

Writing 1 to this register leads to acknowledging and clearing of the interrupt. At the same time, the value of the corresponding bit is cleared.

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

DLR_IRQ_STAT_ACK — Address 4007 E010H, Initial value 0000 0180H

Bit	31 to 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLR_IRQ_STAT_ACK	(RESERVED)	FRMDSP1	FRMDSP0	BECENA1	BECENA0	INVTMR	IPCHANEVET	SUPIGNBEC	LINKSTAP1	LINKSTAP0	SUPRCHAG	BECTMRP1	BECTMRP0	STOPNBCHK1	STOPNBCHK0	FLUEVENT	STACHANGE
R/W	R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31 to 16	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
15	FRMDISP1	This bit indicates that frames have been discarded when the local address matches the transmission source address at port 1.
14	FRMDISP0	This bit indicates that frames have been discarded when the local address matches the transmission source address at port 0.
13	BECFRAP1	This bit indicates that beacon frames have been detected at port 1.
12	BECFRAP0	This bit indicates that beacon frames have been detected at port 0.
11	INVTMR	This bit indicates that frames having values for the beacon timeout timer that are not within the specified range have been detected.
10	IPCHANEVET	This bit indicates that the IP address in beacon frames output by the ring supervisor has been changed.
9	SUPIGNBEC	This bit indicates the detection of beacon frames having a MAC address associated with a priority equal to or less than that of the current ring supervisor.
8	LINKSTAP1	This bit indicates that the linked state of port 1 has been changed.
7	LINKSTAP0	This bit indicates that the linked state of port 0 has been changed.
6	SUPRCHAG	This bit indicates that the ring supervisor has been changed.
5	BECTMRP1	This bit indicates that the beacon timeout timer has reached the timeout time on port 1.
4	BECTMRP0	This bit indicates that the beacon timeout timer has reached the timeout time on port 0.
3	STOPNBCHK1	This bit indicates that operation of the neighbor check timeout timer must be stopped for port 1.
2	STOPNBCHK0	This bit indicates that operation of the neighbor check timeout timer must be stopped for port 0.
1	FLUEVENT	This bit indicates that the local MAC address must be erased from the learning table.
0	STACHANGE	This bit indicates that the state of the local beacon based DLR ring node has been changed.

Caution: When any event described in these bits occurs, the corresponding bit is latched to 1, regardless of the DLR_IRQ_CONTROL register setting.

13.3.7.6 DLR Local MAC Address Low Register (LOC_MAClo)

This register specifies the local MAC address for use in the loop filter. Set the first four octets of the MAC address in the LOC_MAClo register and the remaining two octets in the LOC_MAChi register.

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

LOC_ MAClo	31–24	23–16	15–8	7–0	
	MACADD4	MACADD3	MACADD2	MACADD1	Address 4007 E014H Initial value 0000 0000H
R/W	R/W R/W R/W R/W R/W R/W R/W R/W	R/W R/W R/W R/W R/W R/W R/W R/W	R/W R/W R/W R/W 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 Position	Bit Name	Description
31 to 24	MACADD4	Set the 4th byte of the local MAC address.
23 to 16	MACADD3	Set the 3rd byte of the local MAC address.
15 to 8	MACADD2	Set the 2nd byte of the local MAC address.
7 to 0	MACADD1	Set the first byte of the local MAC address.

13.3.7.7 DLR Local MAC Address High Register (LOC_MAChi)

This register specifies the local MAC address for use in the loop filter. Set the first four octets of the MAC address in the LOC_MAClo register and the remaining two octets in the LOC_MAChi register.

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

LOC_ MAChi	31–16	15–8	7–0	
	(RESERVED)	MACADD6	MACADD5	Address 4007 E018H Initial value 0000 0000H
R/W	R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W	R/W R/W R/W R/W 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 Position	Bit Name	Description
31 to 16	(RESERVED)	Reserved. When writing to these bits, write 0. Ignore reading.
15 to 8	MACADD6	Set the 6th byte of the local MAC address.
7 to 0	MACADD5	Set the 5th byte of the local MAC address.

13.3.7.8 DLR Supervisor MAC Address Low Register (SUPR_MAClo)

This register indicates the first four octets of the MAC addresses of the active ring supervisors extracted from the destination address fields of beacon frames.

- Access This register is only readable in 32-bit units.

	31–24	23–16	15–8	7–0
SUPR_ MAClo	MACADD4	MACADD3	MACADD2	MACADD1
R/W	R R R R R R R R	R R R R R R R R	R R R R R R R R	R R R R R R R R

Address 4007 E020H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 24	MACADD4	Indicate the 4th byte of the MAC address of the active ring supervisor.
23 to 16	MACADD3	Indicate the 3rd byte of the MAC address of the active ring supervisor.
15 to 8	MACADD2	Indicate the 2nd byte of the MAC address of the active ring supervisor.
7 to 0	MACADD1	Indicate the first byte of the MAC address of the active ring supervisor.

13.3.7.9 DLR Supervisor MAC Address High Register (SUPR_MAChi)

This register indicates the last two octets of the MAC addresses of the active ring supervisors extracted from the destination address fields of beacon frames. It also indicates the order of priority for the supervisors.

- Access This register is only readable in 32-bit units.

	31–24	23–16	15–8	7–0
SUPR_ MAChi	(RESERVED)	SUPRPRE	MACADD6	MACADD5
R/W	R R R R R R R R	R R R R R R R R	R R R R R R R R	R R R R R R R R

Address 4007 E024H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 16	(RESERVED)	Reserved. Ignore reading.
23 to 16	SUPRPRE	Indicate the order of priority for the ring supervisors.
15 to 8	MACADD6	Indicate the 6th byte of the MAC address of the active ring supervisor.
7 to 0	MACADD5	Indicate the 5th byte of the MAC address of the active ring supervisor.

13.3.7.10 DLR Ring Status/VLAN Register (STATE_VLAN)

This register indicates the state of the device level ring (DLR) and VLAN ID. These are extracted from the ring state field and VLAN control information fields of beacon frames.

- Access This register is only readable in 32-bit units.

	31 to 16	15 to 9	8	7 to 0	Address
STATE_VLAN	VLANCL	(RESERVED)	VLANVALID	RINGSTATE	4007 E028H Initial value 0000 0000H
R/W	R R R R R R R R R R R R R R R R	R R R R R R R	R	R R R R R R R R	RO

Bit Position	Bit Name	Description
31 to 16	VLANCI	Indicate the control field of the 802.1Q VLAN tag, which is the value of the VLAN control information field of beacon frames. This bit contains a VLAN ID of the DLR.
15 to 9	(RESERVED)	Reserved. Ignore reading.
8	VLANVALID	Indicates that VLAN is valid. In this case, the VLANCI bit contains a valid VLAN ID. 0: Invalid 1: Valid
7 to 0	RINGSTATE	Indicate the state of the device level ring (DLR), which is the value of the ring state field of beacon frames. 0x1: RING_NORMAL_STATE 0x2: RING_FAULT_STATE Others: Not used

13.3.7.11 DLR Beacon Timeout Register (BEC_TMOUT)

This register indicates the timeout timer value of beacon frames. This is extracted from the beacon timeout field of beacon frames.

- Access This register is only readable in 32-bit units.

BEC_ TMOUT	31–0
	BECTIMOUT
R/W	R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R

Address 4007 E02CH
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 0	BECTIMOUT	These bits indicate the beacon timeout timer value. The value is in units of microseconds (μsec). While bit 4 (BECTIMOUT bit) of the DLR_CONTROL register is set to 1, invalid timeout timer values are not written to this register. State transition is not also taken into consideration. A normal expected value is within the range from 200 microseconds to 500 milliseconds. A typical value is 1960 microseconds.

13.3.7.12 DLR Beacon Interval Register (BEC_INTRVL)

This register indicates the interval of beacon frames. This is extracted from the beacon interval field of beacon frames.

- Access This register is only readable in 32-bit units.

BEC_ INTRVL	31–0
	BECINTVAL
R/W	R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R

Address 4007 E030H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 0	BECINTVAL	These bits indicate the interval of beacon frames. The value is in units of microseconds (μsec). An expected value is within the range from 100 microseconds to 100 milliseconds. A typical value is 400 microseconds.

13.3.7.13 DLR Supervisor IP Address Register (SUPR_IPADR)

This register indicates the IP address of the ring supervisor. This is extracted from the source IP address field of beacon frames.

- Access This register is only readable in 32-bit units.

Register	31 to 0	Address
SUPR_IPADR	SPVIP	4007 E034H
R/W	R (all bits)	Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 0	SPVIP	These bits indicate the IP address of the ring supervisor. A value of 0x0 is received when the supervisor has no IP address.

13.3.7.14 DLR Sub Type/Protocol Version Register (ETH_STYP_VER)

This register indicates information of DLR messages. This is extracted from the corresponding field in beacon frames.

- Access This register is only readable in 32-bit units.

Register	31 to 24	23 to 16	15 to 8	7 to 0	Address
ETH_STYP_VER	(RESERVED)	SOURP	DLRRINGVER	DLRRINGTPY	4007 E038H
R/W	R	R	R	R	Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 24	(RESERVED)	Reserved. Ignore reading.
23 to 16	SOURP	These bits indicate the transmission source port of DLR messages. The expected value is 0x00.
15 to 8	DLRRINGVER	These bits indicate the ring protocol version of DLR messages. The expected value is 0x01.
7 to 0	DLRRINGTPY	These bits indicate the ring subtype of DLR messages. The expected value is 0x02.

13.3.7.15 DLR Beacon Timeout Timer Register (INV_TMOUT)

This register indicates the timeout timer value beyond the specified range. When beacon frames having a value for the timeout timer that is not within the specified range are received, that timeout timer value is extracted and stored in this register.

- Access This register is only readable in 32-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4007 E03CH Initial value 0000 0000H
INV_ TMOUT	INVBECTMOUT																																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit Position	Bit Name	Description
31 to 0	INVBECTMOUT	These bits indicate the timeout timer value of frames having a value for the timeout timer that is not within the specified range. The value is in units of microseconds (μsec). The scope of beacon timeout timer values to be indicated in this register is those beyond the range from 200 microseconds to 500 milliseconds. Every time frames having a value for the beacon timeout timer that is not within the specified range, this register is always updated to a new value. While bit 11 of the IRQ_STAT_ACK register is set to 1, values contained in this register are valid.

13.3.7.16 DLR Sequence ID Register (SEQ_ID)

This register indicates the sequence ID of beacon frames. This is extracted from the sequence ID field of beacon frames.

- Access This register is only readable in 32-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4007 E040H Initial value 0000 0000H
SEQ_ID	SEQID																																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit Position	Bit Name	Description
31 to 0	SEQID	These bits indicate the sequence ID of the last beacon frame received through port 0 or port 1. If the frame is ignored due to an invalid timeout timer value, that sequence ID is not stored in this register.

13.3.7.17 DLR MAC Statistics Counters

These registers hold statistics of beacon frames processed by the DLR module.

All registers are 32-bit, read only and the initial value is 0000 0000H.

Address	Symbol	Description
4007 E060H + 0010H*n	RX_STATn	Number of beacon frames received through port n. Beacon frames matching the destination address, Ether type, DLR frame type, and CRC are counted. In the case of a mismatch, frames are not counted. The counters are cleared if the DLR module is disabled.
4007 E064H + 0010H*n	RX_ERR_STATn	Number of beacon frames with CRC error which have been received through port n. Beacon frames matching the destination address, Ether type, DLR frame type but having a CRC error are counted. The counters are cleared if the DLR module is disabled.
4007 E068H + 0010H*n	TX_STATn	Number of beacon frames transferred from port n to port m through the hub. The counters are cleared if the DLR module is disabled.

Remark: n = 0, 1
n = 0: m = 1; n = 1: m = 0

13.4 Function details

13.4.1 Switching Engine

13.4.1.1 Overview

The Ethernet switch implements the following functions:

- Input frame parsing and priority extraction
- Output port(s) resolution
- Frame queuing
- Output queue scheduling

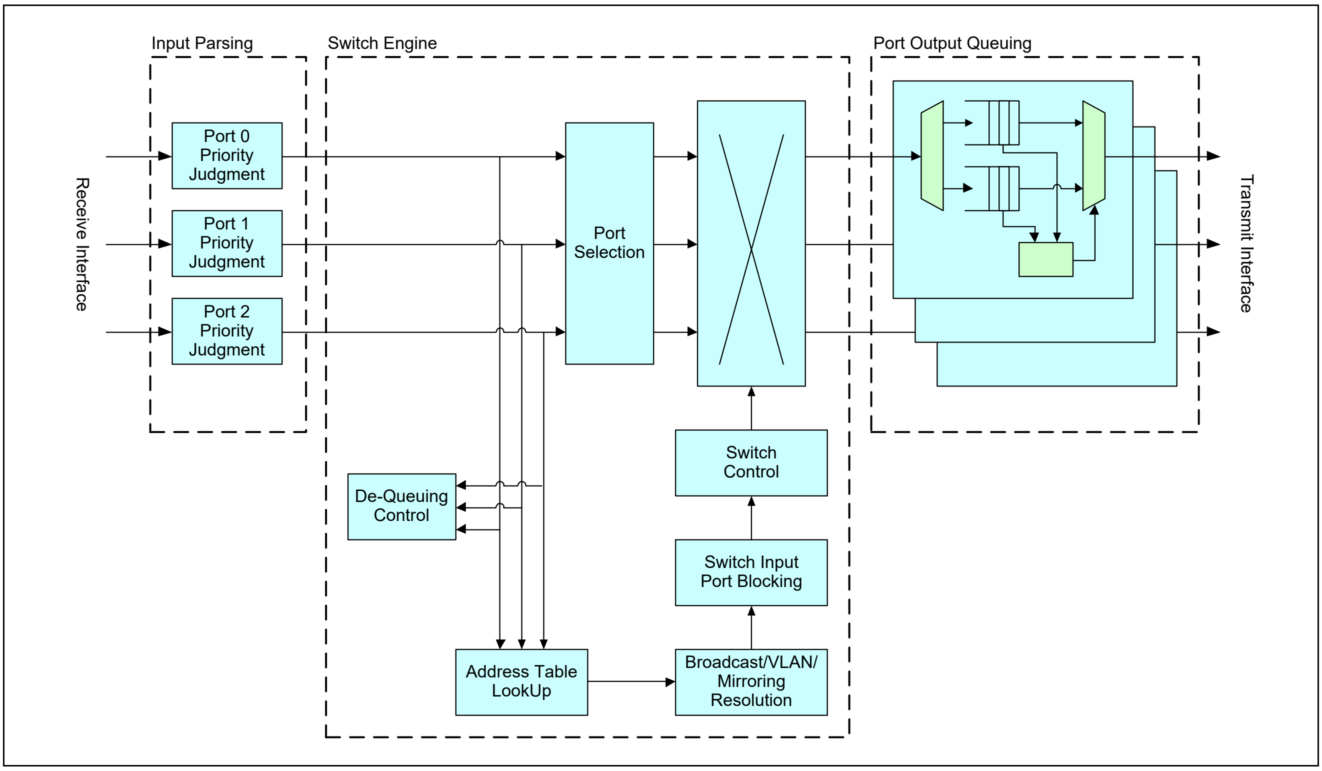


Figure 13.2 Switching Engine Overview

13.4.1.2 Frame Classification and Priority Resolution

(1) Overview

When a frame is received at the input port, the type of frame is judged and several items of information such as the MAC address, VLAN tag, and IP header are extracted from the frame.

Frames are classified with up to eight levels of priority (in the case of VLAN frames), and the priority can be remapped as desired to determine the priority for output. Frames are stored in the corresponding queues at the output port. If a frame has a higher priority than that of the output queue allocated to the port, the frame is stored in the highest priority queue.

(2) VLAN Priority Look-Up

Each port has a programmable priority table with eight entries. The VLAN_PRIORITYn register contains the mapping of the priority for port n (n = 0 to 2) and the final priority can be mapped in the 3 bits of each VLAN priority field.

The index to the mapping field consists of the three-bit priority field of the of the VLAN tag, i.e. bits 7 to 5 of the first octet. The LSB is bit 5 and MSB is bit 7. The destination for mapping has four levels, with the value 0 being the lowest and 3 being the highest.

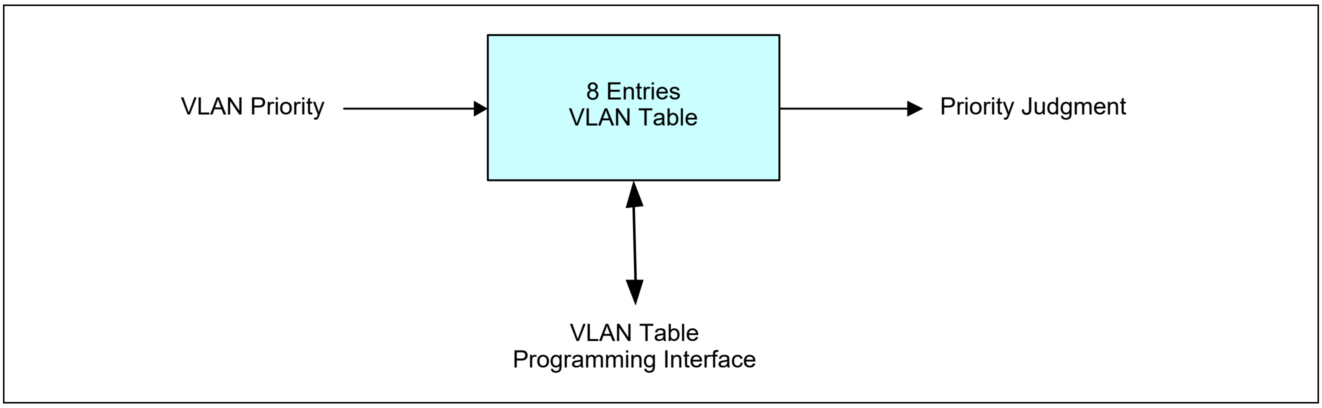


Figure 13.3 VLAN Priority Table Overview

(3) Ipv4 and Ipv6 Priority Look Up (Optional Function)

As an optional (synthesis) function, the switch can classify both Ipv4 and Ipv6 frames: A lookup table with 64 entries is implemented per port to classify the IPv4 frames and a lookup table with 256 entries is implemented per port to classify IPv6 frames. The IP_PRIORITYn is used to set up lookup tables.

The value of the 6-bit DiffServ field from the IPv4 CoS (Class of Service) table entry is input to the table, which returns a 2-bit priority value.

The value of the 8-bit Class of Service field from the Ipv6 COS table entry is input to the table, which returns a 2-bit priority value.

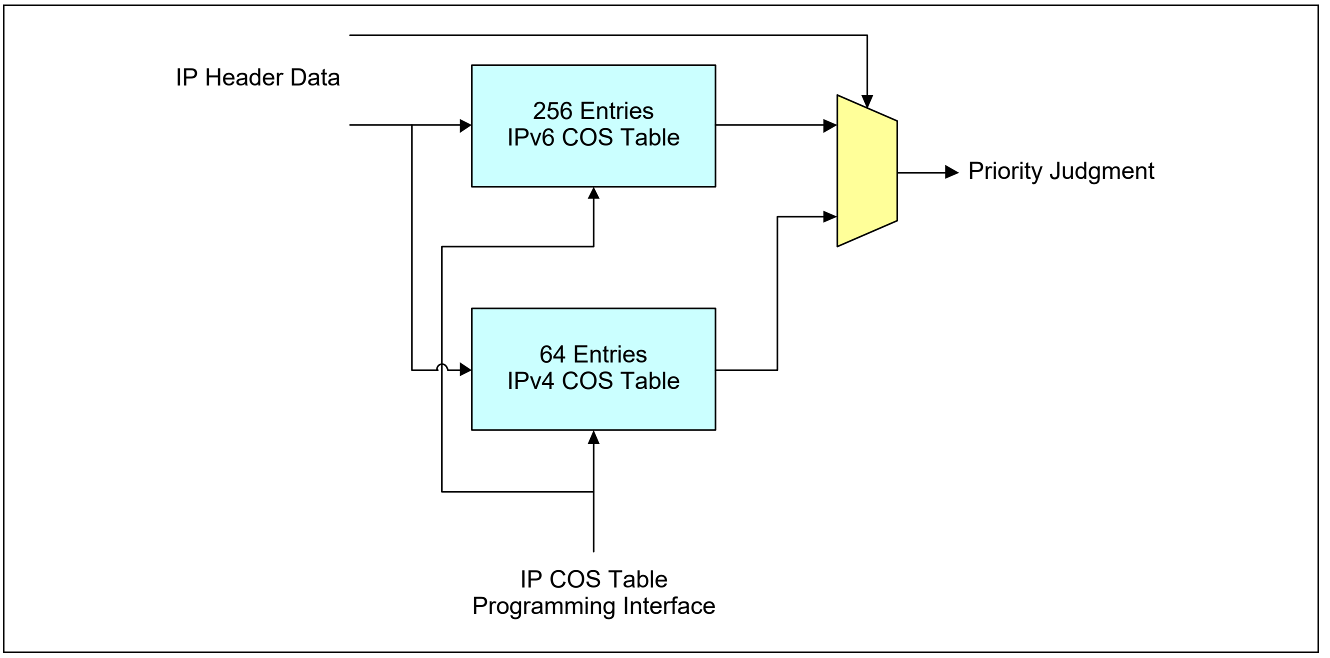


Figure 13.4 IP COS Tables Overview

(4) Determination of Priority

Programming the PRIORITY_CFGn registers allows independent settings for how the levels of priority are determined for packets arriving at each port. The PRIORITY_CFGn registers are used to enable or disable the classification of priority based on the VLAN or IP priority field or by the MAC address.

The priority is determined according to the following rules. The processing differs according to which classification is enabled and which field is found within the frame.

- If determination of priority from the IP priority is enabled and an IP header is found, the priority is mapped by using the IP_PRIORITYn register.
- If the above is not applicable, determination of priority from the VLAN priority is enabled, and a VLAN tag is found, the priority is mapped by using the VLAN_PRIORITYn register.
- Furthermore, if none of the above is satisfied, the default priority as specified in the PRIORITY_CFG register for the port where the frame was received is used.

13.4.1.3 Input Port Selection

The port selection circuit constantly polls all input ports to check if they have available data. If one has data, that port is selected and a frame is read from the port. After reading of a frame, another port is selected even if the port which was read has further data.

In other words, applications running on a FIFO input interface like that of the MAC cannot consecutively transmit frames to the switch. After one frame is transmitted, the sender must wait for the port to be selected again.

13.4.1.4 Layer 2 Look Up Engine

(1) Overview

A hash code is calculated using the frame destination MAC address. It is used as an entry (address) to a table, which contains MAC addresses with destination port number and validity information for each hash value.

As each hash code can represent more than one MAC address, space for up to eight MAC address entries (8-entry block) is allocated in the memory from the location to which the hash code points, and the entries are searched linearly.

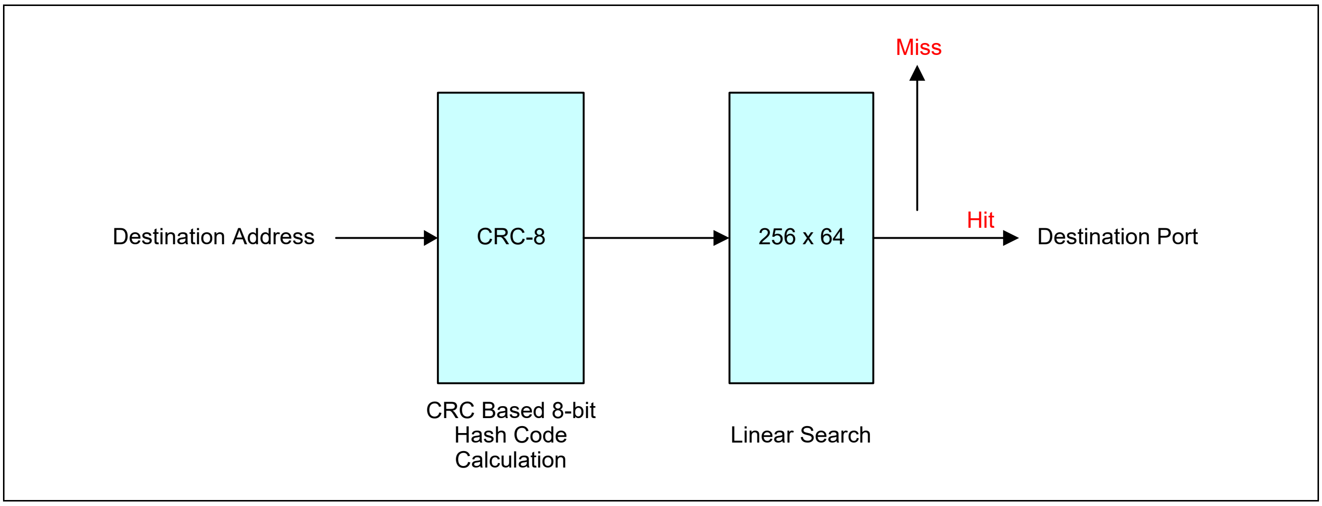


Figure 13.5 Port Look-Up Overview

(2) Hash Code

For a MAC address table with 2048 entries, an 8-bit hash value is calculated from the least significant 24 bits (or all 48-bits) of the MAC address. The hash code is using a CRC-8: $x^8 + x^2 + x + 1$ (0x07)

An 8-bit CRC is also used for smaller address tables with up to 256 entries. In this case, every hash code directly points to one entry in the memory, and the blocks of 8 entries overlap each other.

Caution: The size of the address table is fixed at 256 entries.

(3) Address Table

The address table consists of multiple blocks. Each block has eight records, which contain 64 bits of information each. Each record contains a 48-bit MAC address, information required for transfer, and priority or time stamp information. The address where the block of 8 entries starts is the hash value calculated from the MAC address. Two types of record are defined.

- Dynamic Record:
  A dynamic entry consists of a MAC address together with a 10-bit timestamp and destination port number. These entries are created by a function for learning from received frames to enable the transfer of frames to particular ports. Dynamic entries are deleted by an aging function if they are not updated.
- Static Multiport/Priority Record:
  Switch management can also write static entries in the address table. Along with MAC addresses, these can include priority levels as well as specifications of multiple destination ports for transfer (by using port bit masks). The MAC addresses can be unicast or multicast. These records can be used to e.g. specify the ports to participate in a specific multicast domain or to assign priority based on the MAC address to a frame. The aging and learning functions are not applied to static records.

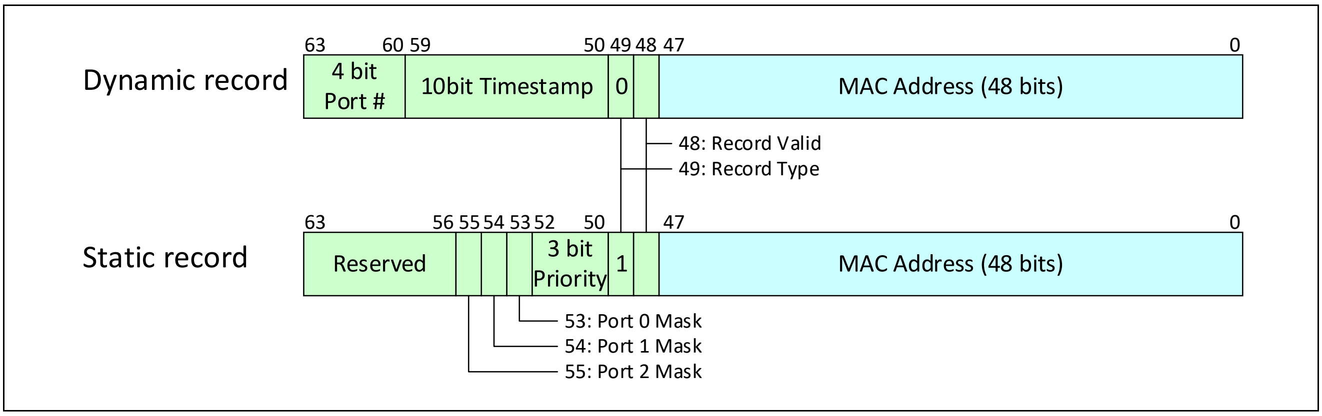


Figure 13.6 Record Types of Address Memory

Bit 49 of the records decides which type of record is found in the table:

- If the value is 0, the entry is interpreted as dynamic, and this bit is followed by a 10-bit timestamp and 4-bit port number.
- If the value is 1, the entry is interpreted as static, and this bit is followed by a 3-bit priority field and a 3-bit port bit mask. For the port bit mask, bits 53, 54, and 55 represent port 0, port 1, and the internal port (port 2) respectively. Frames will be transferred to all ports that have a 1 in the port bit mask. Frames are not transferred again to the source port for transmission, even if the port bit mask is 1.

13.4.1.5 Learning Interface

The learning interface provides the software with the information required to construct a lookup table. The interface has a FIFO buffer for storage of multiple entries.

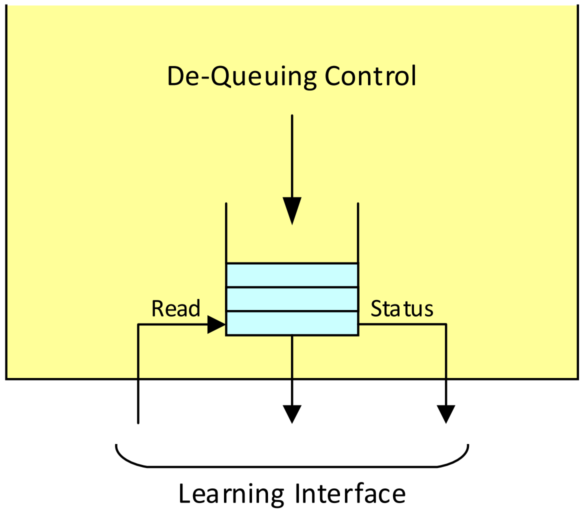


Figure 13.7 Learning Interface Overview

Two 32-bit records (record A and record B) are written to the FIFO buffer for each frame received by the switch. Record A is written first followed by record B.

Record A contains the source MAC address of the frame and record B contains the 8-bit hash code calculated from that address, and the port number at the source. The first octet of the MAC address is bits 7 to 0 of record A and the sixth octet is bits 15 to 8 of record B.

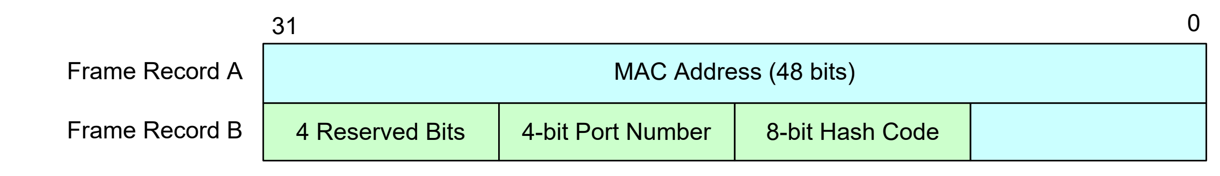


Figure 13.8 Record Formats

Software can read these records by using the LRN_REC_A and LRN_REC_B registers.

13.4.1.6 Frame Transfer Processing

(1) Overview

When a frame is processed, its 48-bit source and destination MAC addresses are extracted. The address table is searched for the destination MAC address. The following rules apply in the order from top to bottom:

- If the destination address is found, the frame is transferred to the port(s) specified by the address table entry.
- If the above is not satisfied and the destination address is unicast, the frame is transferred to all ports specified by the UCAST_DEFAULT_MASK register.
- If the above is not satisfied and the destination address is broadcast, the frame is transferred to all ports specified by the BCAST_DEFAULT_MASK register.
- If the above is not satisfied and the destination address is multicast, the frame is transferred to all ports specified by the MCAST_DEFAULT_MASK register.
- If none of the above conditions are satisfied, the frame is transferred to all ports specified by the BCAST_DEFAULT_MASK register.

The address table can hold static entries. Registering multicast addresses in static entries is also possible. Accordingly, the specified multicast addresses can also be transferred by using static entries instead of the setting of the MCAST_DEFAULT_MASK register.

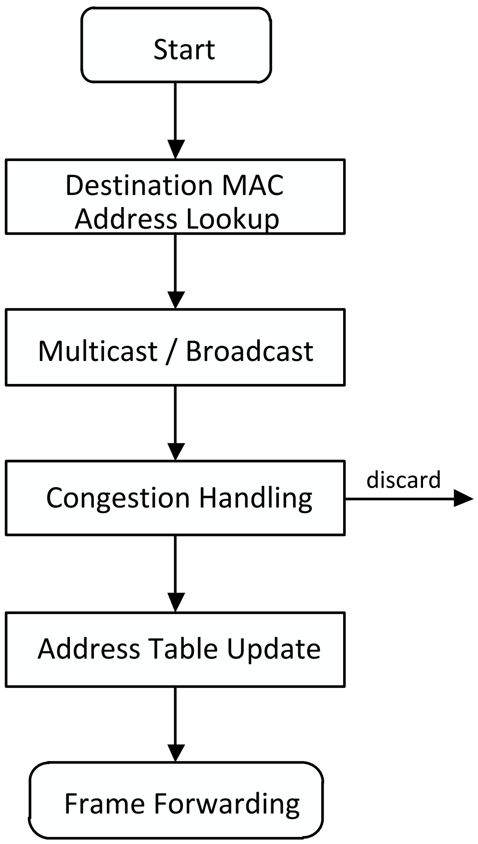


Figure 13.9 Overview of Processing for Frame Transfer

(2) Processing to Handle Congestion

(a) Overview

Processing to handle congestion is used whenever an output port is not available but sending of data to that port is required. An output port is defined to be “available” if the port is enabled by register PORT_ENA and the corresponding output queue has enough room to store a full-sized frame.

This processing determines whether the frame should be processed further or discarded according to the following rules:

(b) Unique destination (one input to one output)

If the output port is enabled and can accept a frame, the frame will be transferred normally. In any other case, the frame will be discarded.

(c) Multiple destinations (flooding)

After broadcast, multicast, or flooding processing, frames must be transferred to multiple output ports.

- If there are output disabled ports, all disabled ports are removed from the list of outputs.
- If any of the output ports cannot accept a frame due to output congestion (as indicated by the output queue management for the port), that port is removed from the list of outputs.

If no output port is left in the list of outputs after the removal, the frame is read from the input and then discarded. The frame discard counter (ODISCn) corresponding to that port is incremented.

(3) Bridge Protocol Frame Processing

To implement bridge control protocols like the Spanning Tree protocol, the following controls are performed by protocol frame processing:

(a) Input port blocking

Input port blocking is used to avoid transfer of frames after address learning. This can be enabled or disabled by using the INPUT_LEARN_BLOCK register. If a frame is received through the port which should be blocked and that frame is not a bridge protocol frame, the frame will be discarded and will not be transferred to any output port.

(b) Disabling input port learning

To reduce the load of software processing, a port can be configured to be out of the scope of learning by using the INPUT_LEARN_BLOCK register. When learning is disabled for a port, source addresses of received frames are not extracted for that port, except for those of BPDU frames. The source addresses of BPDU frames are always extracted and transferred to the learning interface.

(c) Transfer to management port (internal port)

If bit 6 of the MGMT_CONFIG register is enabled, bridge protocol frames are always transferred to the management port, independent of any address lookup or other transfer processing.

Bridge protocol frames are identified by its destination address being any of the following:

- 01-80-c2-00-00-00 to 01-80-c2-00-00-0F (Spanning Tree, IEEE 802.1d)
- 01-80-c2-00-00-10 (Bridge Management Address, 802.1d)
- 01-80-c2-00-00-20 to 01-80-c2-00-00-2F (Generic Attribute Registration Protocol, 802.1d)

(d) Transfer of management frames

If the management port (internal port) transmits frames, they are transferred according to the port mask settings of bits 17 and 16 of the MGMT_CONFIG register. A handshaking mechanism is implemented (bit 5 of the MGMT_CONFIG register) and the port mask settings can be changed for management frames in units of frames.

(4) Forcible Transfer

The switch is capable of forcibly transferring frames to specific ports by disabling the method of transfer determined by transfer processing. This function is generally used for management frames. Multicast addresses are used for management frames, but they need only be transferred to specific output ports.

Depending on the implementation of the switch application, either of the following is used.

- When a BPDU is transferred, the port mask defined in the MGMT_CONFIG register can be used. The application must set the register before transferring the BPDU frame to the switch. After that, if bit 5 indicating the completion of transmission of the BPDU frame is set, the port mask setting can be cleared.
- Forcible transfer can be set in units of frames by using a management tag that can be used between the internal port and Ethernet switch. This method is preferred since it eliminates the need for any handshaking that requires use of the MGMT_CONFIG register.

The difference between the above two transfer methods is that only BPDU frames are transferred in the former, while the latter allows the forcible transfer of all frames.

Note: When the management tag is used for forcible transfer, bits 17 and 16 of the MGMT_CONFIG register must always be set to 0. The setting of MGMT_CONFIG is given priority and the management tag setting will be overwritten.

13.4.1.7 Output Frame Queuing

(1) Overview

A shared memory architecture to store frames of desired size for multiple output ports is adopted for the memory controller.

Each output port can have queues with up to four priority levels. The memory controller has a single input port (write port) and multiple output ports (multiple read ports) which can handle the virtual duplication of frames.

The memory is divided into small cells for efficiently sharing of the available memory area among small and large frames. Therefore, even the storage of small frames does not leave a large unused area.

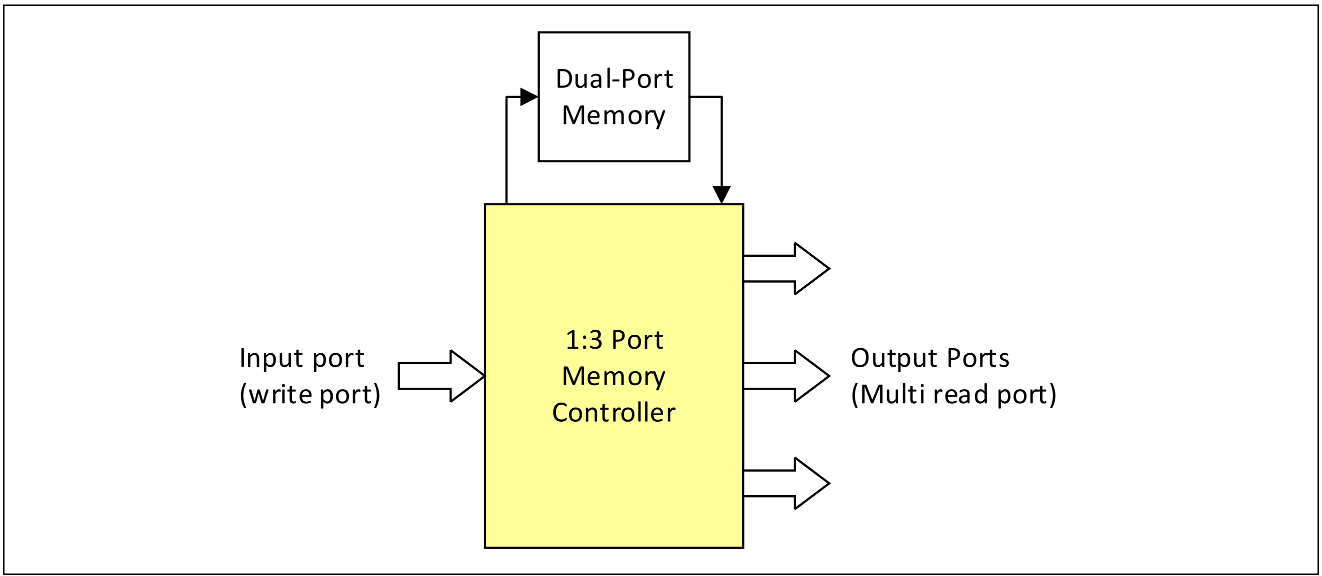


Figure 13.10 Overview of Output Port Memory Controller

(2) Functions

- Memory controller with one write port and multiple read ports
- Shared memory (8KB) partitioned into 256-byte cells
- When writing to memory from an input port, simultaneous writing to multiple destinations is possible (virtual frame duplication).
- Multiple read ports (output ports) for time-divided multiple outputs from memory to achieve output to all output ports in a rapid sequence.
- Queues with 4 priority levels for each output port
- Congestion information for backpressure and overflow protection available
- Memory status statistics available

(3) Implementation

The memory manager implements 8 Kbytes of shared memory for all queues on output ports 0 and 1. Port 2 (internal port) has a single FIFO queue, which operates independently from the shared memory. Therefore, an internal port being congested (the software is not reading fast enough) does not affect transfer between ports 0 and 1.

13.4.2 Hub Module Supporting Cut-Through

The Ethernet switch has a hub module which supports cut-through. Use of this module allows high-speed transfer of frames without using the switch engine between ports 0 and 1.

The hub module operates at the level of the MII between the MAC and Ethernet PHY. Operation of the hub module for packets from both ports 0 and 1 and also for one port only is possible. If operation for one port is enabled, cut-through transfer is used in one direction and store-and-forward transfer is used in the other direction. The settings for the direction of operation of the hub module and the enabling or disabling of operation itself can be controlled by software.

If the hub module is enabled, transfer of all received frames to the opposite port is immediate and proceeds before they are completely received (cut-through transfer). Filters can be configured to avoid cut-through transfer of certain specific management frames that must be routed through the switch with normal store-and-forward behavior.

13.4.2.1 Operating in Normal Switch Mode

In normal switching mode, the MAC interfaces are directly connected to the Ethernet PHY interfaces and data are directly transferred by the switch. The switch engine is responsible for transferring all frames in between the individual ports.

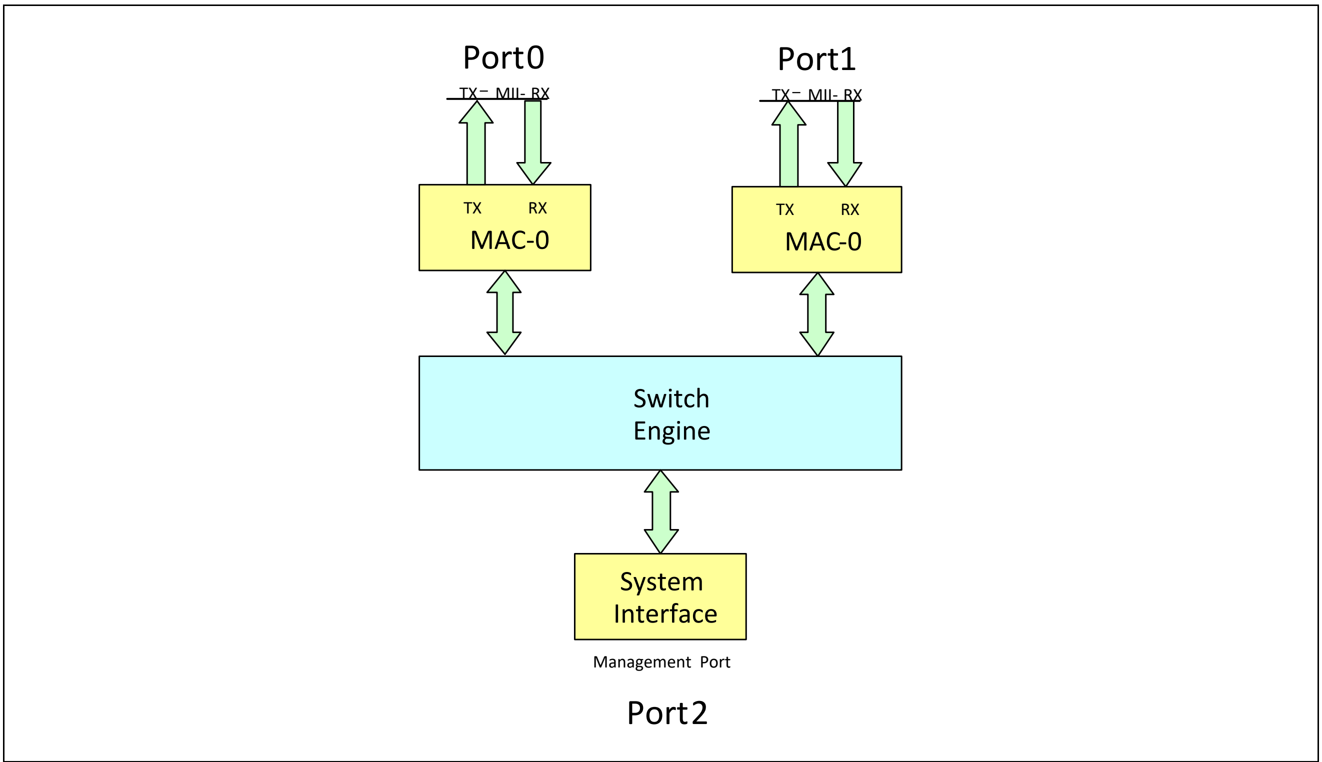


Figure 13.11 Normal Switch Mode Operation

13.4.2.2 Operation of the Hub in the Direction from Port 0 to Port 1

When operation of the hub in the direction from port 0 to port 1 is enabled, the data paths are changed at the PHY interface level as follows.

- The reception PHY interface of port 0 is connected to the reception interfaces of both the switch and the hub. The hub will send received frames to the transmission interface of port 1. A filtering mechanism is implemented to avoid duplication of frames.
- The switch can transmit frames to port 0 normally and will continue to receive all traffics from port 0.
- The switch will receive frames from port 1 normally.
- When the switch transmits frames to port 1, frame duplication must be avoided. If a frame has already been transferred through the hub or is waiting in the transmission queue, repetition of handling of the frame must be avoided. The address filter table is also used for this purpose.
- IEEE 1588 frames must not pass through the hub to ensure proper operation of the protocol (to update the correction field).

Even when the hub is enabled, received frames are sent to the switch and the method of transferring each frame is decided within the switch. However, a frame that is also to be transferred by the switch to the same port as one to which the hub has already sent it will be discarded. In other words, data transfer to the internal port (port 2) may occur. A FIFO buffer is used for arbitration of frames both the switch and hub are attempting to transmit.

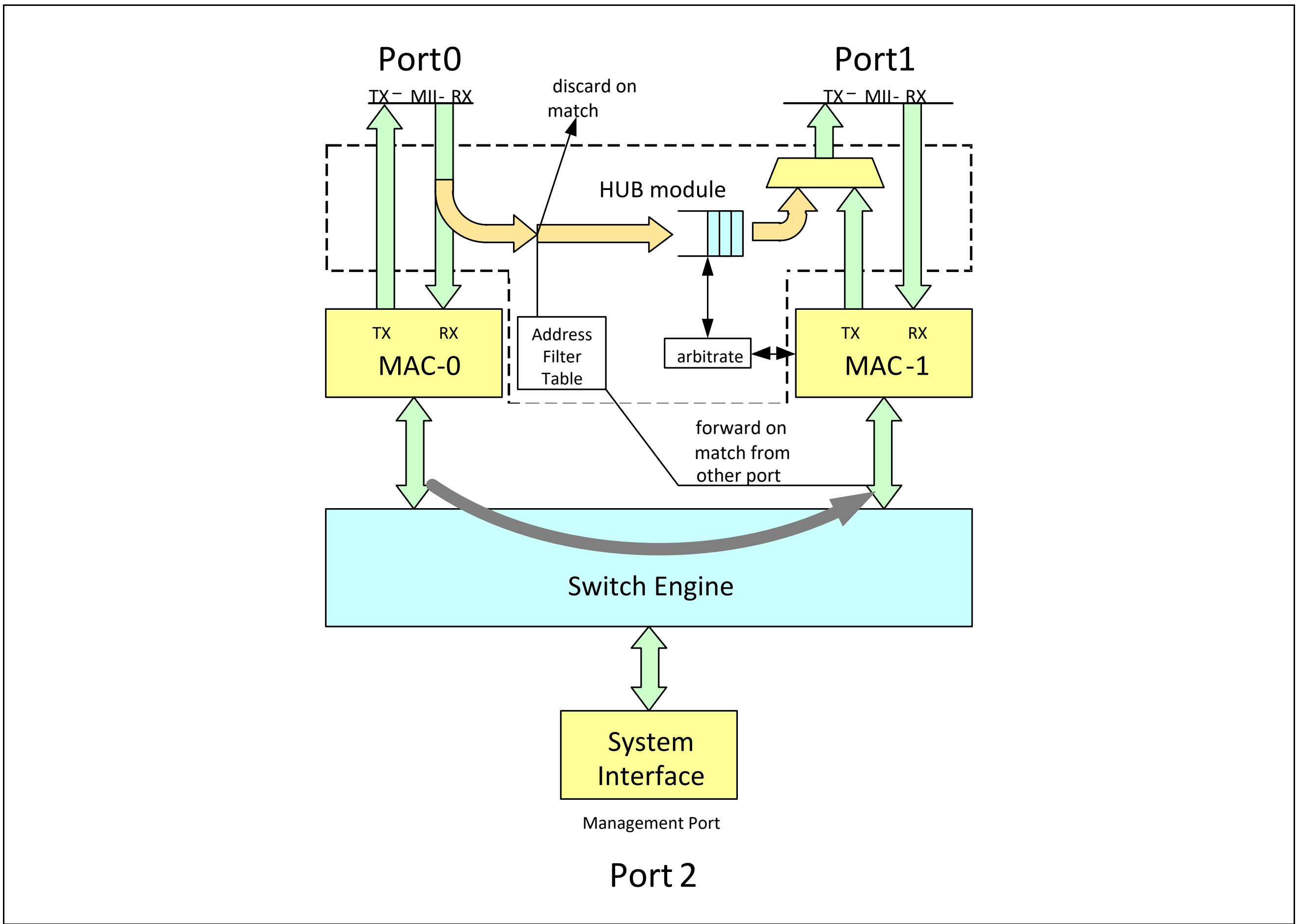


Figure 13.12 Operation of the Hub when Transfer from Port 0 to Port 1 is Enabled

13.4.2.3 Operation of the Hub in the Direction from Port 1 to Port 0

When operation of the hub in the direction from port 1 to port 0 is enabled, the data paths are changed at the PHY interface level to receive data from port 1 and transmit it directly to port 0. Transmission from both the switch and hub is arbitrated at port 1. Operation is the same as that in the direction from port 0 to port 1.

Each direction can be enabled separately and also both directions can be enabled at the same time.

13.4.2.4 Hub Reception Filtering

When the hub mode is enabled, the receive interface of the hub module must not transfer the following frames through the hub:

- Frames having a unicast MAC destination address matching the local system's unicast MAC address.
- IEEE 1588 frames.
- Any local management frames (e.g. MAC pause frames) that are not expected to propagate through a switch

The hub receive filter operates on MAC destination addresses. Up to 7 MAC addresses can be registered for filtering. Furthermore, it is used to handle masking of the last byte of addresses to expand the range of the addresses that can be filtered, or in the opposite way, for forcible transfer instead of for filtering. (see the description of registers HUB_FLT_MACnlo and HUB_FLT_MACnhi).

Transfer operation differs between the hub and switch depending on the filter setting.

Table 13.1 Operation of the Hub and Switch by Filter Setting

Forcible Transfer		Disable		Enable Note 1	
Address Match/Mismatch		Match	Mismatch	Match	Mismatch
Hub enabled	Hub	Not transferred	Cut-through	Cut-through	Filtering with forcible transfer disabled
	Switch	Store and forward Note 2	Not transferred	— Note 3	
Hub disabled	Hub	Not transferred	Not transferred	Cut-through	
	Switch	Store and forward Note 2	Store and forward Note 2	— Note 3	

Note 1: Use of beacon frames of DLR is assumed.

Note 2: This is a case when transfer proceeds between the PHY ports. Transfer may not proceed depending on the address table and default mask settings.

Note 3: A frame will be discarded before entering the switch.

- When forcible transfer of frames is disabled

  If the destination address of the received frame matches the address registered in the filter, the hub does not transfer that frame to another port. However, it transfers that frame to another port if transfer proceeds between port 0 and port 1 within the switch.

  If the destination address of the received frame does not match the address registered in the filter, the nub transfer the frame to another port. On the other hand, the switch does not transfer the frame to another port. This prevents duplication of frames.

- When forcible transfer of frames is enabled

  If the destination address of the received frame matches the address registered in the filter, the hub always transfers the frame to another port even while the hub is disabled. Forcible transfer is generally used for beacon frames. If the DLR function is enabled, beacon frames can be processed by using the DLR module, but they are discarded before entering the switch module. This is to prevent frame duplication.

The management port (port 2) is not affected by any frame filtering and will always receive frames from both MAC ports.

The filter addresses must include the local system unicast addresses as well as destination addresses (multicast addresses) of IEEE 1588 frames and of frames that should not be transferred through the hub while forcible transfer is disabled. The following tables give examples of relevant addresses. For details, see the respective specifications.

Table 13.2 PTPv2 Multicast Domains: Layer 2

Name	MAC Address Mapping
Normal messages	01-1b-19-00-00-00
Peer delay messages	01-80-c2-00-00-0e

Table 13.3 PTP Multicast Domains: UDP/IP

Name	IP Address	MAC Address Mapping
Default PTP domain	224.0.1.129	01-00-5e-00-01-81
Alternate PTP domain1	224.0.1.130	01-00-5e-00-01-82
Alternate PTP domain2	224.0.1.131	01-00-5e-00-01-83
Alternate PTP domain3	224.0.1.132	01-00-5e-00-01-84

Table 13.4 Management Frame Domains

Name	IP Address	MAC Address Mapping
Generic Switch Management	224.0.0.0	01-00-5e-00-00-00
IGMP	224.0.0.1	01-00-5e-00-00-01

Table 13.5 Switch Management Frame Domains

Name	MAC Address mapping
Spanning Tree, IEEE 802.1d	01-80-c2-00-00-00 to 01-80-c2-00-00-0F
Bridge Management Address, 802.1d	01-80-c2-00-00-10
GARP	01-80-c2-00-00-20 to 01-80-c2-00-00-2F
MAC Layer Control Frames (Pause)	01-80-c2-00-00-01

Table 13.6 DLR Multicast Domains

Name	MAC Address mapping
Beacon Frame	01-21-6C-00-00-01
Neighbor Check Request, Neighbor Check Response, Sign ON	01-21-6C-00-00-02
Announce, Locate Fault	01-21-6C-00-00-03

Based on the above, initial settings must be made for the hub module to include at least the addresses listed in Table 13.7. The address and mask values are programmed by using the HUB_FLT_MACnlo/hi registers.
The first byte of the MAC address must be set in bits 7 to 0 of the HUB_FLT_MACnlo register. The logical AND of the mask value and the last byte of the address of the received frame is taken, and the result is compared with the set address.

The forcible transfer bit should only be set to 1 when frames must always be transferred via the hub. Forcible transfer operates regardless of the enabled or disabled setting of the hub. That is, when the hub module is disabled, only specified frames can be transferred in a cut-through fashion.

If a request is not issued by the application, broadcast frames must not be transferred through the hub. There is no need to input broadcast addresses to the filter table. Filtering can be enabled by using the corresponding control bit in the HUB_CONTROL register.

Table 13.7 Typical Hub MAC Filter Setup

MAC Address	Mask	Forcible Transfer	Notes
01-80-c2-00-00-00	0xC0	0	Filters all frames in range 01-80-c2-00-00-{00..3F} The settings of the HUB_FLT_MACnlo/hi registers would be: HUB_FLT_MACnlo = 00C2 8001H HUB_FLT_MACnhi = 00C0 0000H
01-1b-19-00-00-00	0xFF	0	Filters only this address (PTPv2)
01-00-5e-00-01-80	0xF8	0	Filters 01-00-5e-00-01-{80..87}(224.0.1.{128..135})
01-00-5e-00-00-00	0xFC	0	Filters 01-00-5e-00-00-{00..03} (224.0.0.{0..3})
<local node unicast address>	0xFF	0	Should be entered to avoid unnecessary transfer of frames that are directed to the node only.
01-21-6C-00-00-01	0xFF	1	Beacon frames should be forcibly transferred through the hub. The settings of the HUB_FLT_MAC6lo/hi registers would be: HUB_FLT_MAC6lo = 006C 2101H HUB_FLT_MAC6hi = 01FF 0100H

13.4.2.5 Forcible Transfer by the Hub Module

The forcible transfer bit (bit 24 of the HUB_FLT_MACnhi register) can be set for each entry of the filter. This bit changes operation of the hub module to forcibly transferring frames instead of filtering them. If the addresses match and the forced transfer bit for that address entry is set, frames are transferred via the hub in a cut-through fashion. On the other hand, frames to be transferred to the MAC and switch are discarded before the MAC and switch. Forcible transfer always proceeds independently of the hub enable control bit (bit 0 of the HUB_COTNROL register).

Since frames are discarded before they are loaded to the switch in this operating mode, the forcibly transferred frame cannot be processed by the switch. Accordingly, there is no address learning from such frames. Also, these frames cannot be transferred to a local application via port 2. This is different from normal hub operation. In normal hub operation, all frames are loaded to the switch but they are only discarded at the port through which they have been transferred to avoid frame duplication on the line side of the port.

The DLR module, which is described in the next section, can receive forcibly transferred frames normally. This is because this module is located before the MAC and switch and is not affected by frames being discarded. Accordingly, forcible transfer is intended to be used for beacon frames of the DLR. The load on the application can be reduced by using the DLR module to process beacon frames.

13.4.2.6 Loop Filtering

The hub module has a loop filter, which is used to discard frames with specific source addresses at the reception port. This prevents such frames from passing through the hub or switch. This functionality is generally required by applications where connection is in a ring. In this case, frames from the local node may reach the local node again after they have passed through the ring, so if the loop filter discards a frame, that frame has no further processing by the hub or switch, and can be completely removed from the network.

The MAC address of the local node to be processed by the loop filter can be configured with the LOC_MAClo/hi registers of the DLR module.

13.4.3 DLR Module

The device level ring (DLR) module offers beacon frame processing on the reception paths of ports 0 and 1 of the switch core as a beacon node.

The DLR module is inserted between the HUB module and the switch module.
The DLR module detects beacon frames on the reception paths from both external ports and discards them before they enter the switch module. The DLR module analyzes all beacon frame parameters and stores them in local registers to allow access by software.

The DLR module can issue an interrupt to notify the CPU of any change in the state of the ring indicated by a beacon node. This allows parameters in received beacon frames to be read at any time.

Statistics counters to count the number of transferred beacon frames are also implemented.

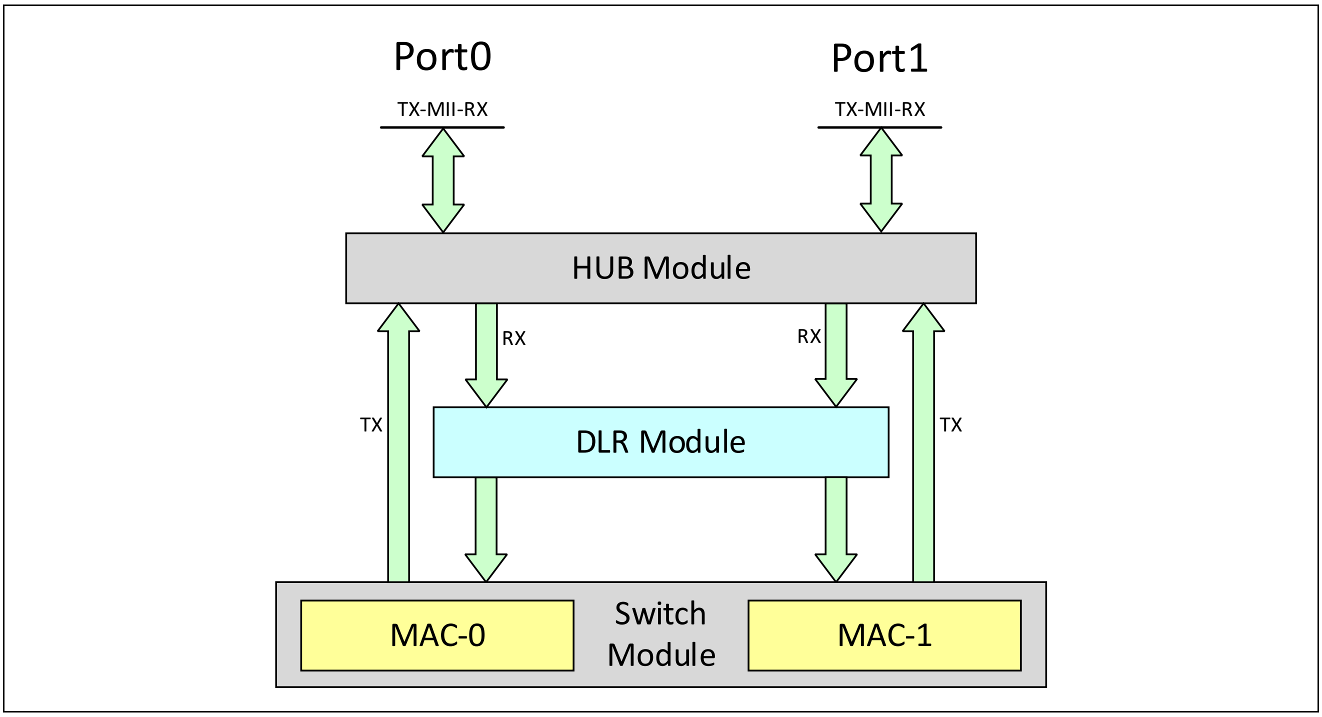


Figure 13.13 Connection between the Hub Module and the DLR Module

13.4.3.1 Beacon Frame Format

Within a DLR network, the active ring supervisor transmits a beacon frame through both of its Ethernet ports per beacon interval (400 microseconds by default). DLR frames are using the frame format of 802.1Q. Frames are transmitted with the highest priority (7). A beacon frame is 64 bytes of DLR frame, excluding the preamble and the SFD, and it consists of the following fields:

Frame length (all rows)

Common DLR Protocol Header (PREAMBLE through Ring Protocol Version (0x01))

7 octets	PREAMBLE
1 octet	SFD
6 octets	DESTINATION ADDRESS
6 octets	SOURCE ADDRESS
2 octets	VLAN Tag (0x8100)
2 octets	VLAN info (0xE00 + VLAN ID)
2 octets	Ring Ether Type (0x80E1)
1 octet	Ring Sub Type (0x02)
1 octet	Ring Protocol Version (0x01)
1 octet	Frame Type (0x01)
1 octet	Source Port (0x00)
4 octets	Source IP Address (0x00 if source has no IP address)
4 octets	Sequence ID
1 octet	Ring State
1 octet	Supervisor precedence
4 octets	Beacon Interval
4 octets	Beacon Timeout in microseconds
20 octets	Reserved
4 octets	Frame Check Sequence

Figure 13.14 Beacon Frame Format

The DLR module processes beacon frames and stores beacon frame parameters in local registers to allow access by software. The following table shows the beacon frame fields and relevant register names to store the values for the ring node.

Table 13.8 Definitions of Beacon Frame Fields

Term	Description	Register Name
Destination address	The destination MAC address of beacon frames is a fixed multicast address of 01-21-6C-00-00-01. This is an exclusive MAC address used only for beacon frames. Cut-through transfer proceeds based on matching with this address.	—
Source address	Source MAC address of the supervisor. 48-bit addresses are stored in the two registers.	SUPR_MAClo/hi
VLAN tag	DLR messages contain 2 octets of the VLAN tag (0x8100) after the source MAC address according to 802.1Q.	—
VLAN information	16-bit information fields contain the priority field and the VLAN_ID. The VLAN ID is configured by the ring supervisor and received by the ring nodes. The default value of the VLAN ID is 0 when there is no VLAN ID available. The default VLAN ID does not need to be changed unless a commercially-available switch is used within the ring.	Bits 31 to 16 of STATE_VLAN. However, when bit 8 is set to 1, the value is valid.
Ring Ether type	Ether type for DLR frames is 0x80E1.	—
Ring sub type	The value of ring sub type for the DLR messages is always 0x02.	Bits 7 to 0 of ETH_STYP_VER
Ring protocol version	Protocol version of DLR messages	Bits 15 to 8 of ETH_STYP_VER
Frame type	The value of frame type of beacon frames is always 0x01.	—
Source port	The value of the source port for beacon frames is always 0x0.	Bits 23 to 16 of ETH_STYP_VER
Source IP address	IP address of the supervisor. The default value of the IP address is 0 if there is no IP address available.	SUPR_IPADR
Sequence ID	Sequence identification number of frames	SEQ_ID
Ring state	State of the ring network transmitted by the ring supervisor.	Bits 7 to 0 of STATE_VLAN
Supervisor priority	The ring supervisor priority value contains the value of priority assigned to the ring supervisor. When multiple supervisors are enabled, a supervisor with the highest priority can be selected. The ring supervisor's priority value can be any value within the range from 0 to 255, with numerically higher values indicating higher precedence.	Bits 23 to 16 of SUPR_MAChi
Beacon interval	Interval at which the ring supervisor sends beacon frames. The setting is in units of microseconds. Valid values are within the range from at least 100 microseconds up to 100 milliseconds. A typical value is 400 microseconds.	BEC_INTRVL
Beacon timeout	When a timeout for beacon frames is detected, this indicates the time over which to wait before performing appropriate processing in units of microseconds has elapsed. Valid values are within the range from at least 200 microseconds up to 500 milliseconds. A typical value is 1960 microseconds.	BEC_TMOUT
Frame check sequence	CRC value for frames	—

13.4.3.2 Functional Description of Ring Node

Beacon frames are detected and analyzed by the DLR module so that, if the CPU is for a ring node, it is not burdened with the processing of beacon frames. If a beacon node indicates any change in the ring state (configuration), this is conveyed to the CPU through an interrupt.

In addition, parameters in received beacon frames can be read at a desired time. Statistics counters are also implemented to check the number of transferred beacon frames.

(1) Initial Settings

The procedure for setting up the DLR module is as follows:

- Set the lower-order 4 bytes of the beacon destination address (006C 2101H) in the HUB_FLT_MAC6lo register. This value is the initial value of this register.
- Set the higher-order 2 bytes of the beacon destination address and the setting to enable forcible transfer (01FF 0100H) in the HUB_FLT_MAC6hi register. This value is the initial value of this register. Note the setting of the mask bit is 0xFF.
- Set the lower-order 4 bytes of the unicast address of the local device in the LOC_MAClo register of the DLR module which is used by the loop filter.
- Set the higher-order 2 bytes of the unicast address of the local device in the LOC_MAChi register of the DLR module which is used by the loop filter.
- Set the DLR Ethernet frame type value of 0x80E1 in the DLR_ETH_TYP register. This value is the initial value of this register.
- Enable the DLR module through the DLR_CONTROL register. Also set the number of clock cycles required for counting one microsecond in this register. The DLR module of this LSI chip operates at 100 MHz, so always set 0x64 in this register. The setting must be changed from the initial value.
- Use the DLR_IRQ_CTRL register to enable or disable desired interrupt sources as required by the software.

(2) Start Up

At start-up, the ring node is placed in the idle state and assumes that the network is in a linear topology. The current state of the local ring node and the values of the other status bits are stored in the DLR_STATUS register and can be accessed by software.

When a beacon frame holding an invalid timer value is received while bit 4 of the DLR_CONTROL register is set for ignoring invalid timer values, that frame will be ignored. On the other hand, invalid timer values are stored in the INV_TMOUT register regardless of the setting of bit 4 of the DLR_CONTROL register. Setting bit 11 of the DLR_IRQ_CTRL to 1 also allows generation of interrupts.

When a beacon frame is received through either port, the ring node is placed in the idle state and assumes that the network is in a linear topology. When bit 1 of the DLR_IRQ_CTRL register is set to 1, an interrupt is generated and the CPU is notified that the MAC address learning table requires flushing and that a state transition has occurred. All parameters of the ring supervisor are stored in the register and can be accessed by software. However, the following parameters are only stored during transitions from the idle state to the fault state.

- Supervisor's MAC address: Stored in register SUPR_MAClo or SUPR_MAChi.
- Supervisor's priority value: Stored in register SUPR_MAChi
- VLAN ID: Stored in register STATE_VLAN
- Beacon timeout timer value: Stored in register BEC_TMOUT

The IP address of the supervisor is accepted to change at any time. The new IP address will always replace the old one. An interrupt indicating a change of the IP address is generated by setting bit 10 of the DLR_IRQ_CTRL register.

If a beacon frame is received from a supervisor which has a higher priority than the current supervisor or from another supervisor with the same priority which has a higher MAC address, parameters of the new beacon frame will replace all old values. An interrupt indicating the change of the supervisor is generated by setting bit 6 of the DLR_IRQ_CTRL register. The ring node will stay in the fault state.

If a beacon frame is received from a supervisor which has a lower priority than the current supervisor or from another supervisor with the same priority which has a lower MAC address, that beacon frame will be ignored. An interrupt indicating that the beacon frame has been ignored is generated by setting bit 9 of the DLR_IRQ_CTRL register. The ring node will stay in the fault state.

The ring supervisor is not expected to change parameters in beacon frames. If parameters need to be changed, the supervisor stops transmitting beacon frames for at least two beacon timeout periods before transmitting beacon frames with new parameters.

If the local node returns to the idle state when the beacon timeout timer reaches the timeout time on both ports, an interrupt is generated by setting bits 4 and 5 of the DLR_IRQ_CTRL register. The current interrupt state is accessible by software. Since a beacon timeout has occurred on both ports, erasure of the MAC address learning table and changing the state of the DLR_IRQ_STAT_ACK register are required.

If beacon frames are received through both ports and a beacon frame with the ring state field set to RING_STATE_NORMAL is received from the active ring supervisor through either of the ports, the local node enters the normal state. The interrupt status bit indicates the change of state. As a result, the unicast MAC address learning table must be erased. In addition, if the software has set the neighbor check timeout timer running, it must be stopped.

Note: The neighbor check timeout timer for neighbor check processing (100 milliseconds) should be implemented by the software. The software can use bit 3 or 2 of the DLR_IRQ_STAT_ACK register to stop the timer.

(3) Fault Detection

Any of the following events shall cause a transition of the ring node from NORMAL_STATE to another state.

- Reception of a beacon frame with the state parameter set to RING_FAULT_STATE.
  The DLR_IRQ_STAT_ACK register indicates that bit 0 is set and the node state has been changed.
  An interrupt is also generated if generation of interrupts is enabled.
- Reception of a beacon frame with a different MAC address from the currently active ring supervisor or from a supervisor with a higher priority.
  In addition to the change of the state, bit 6 of the DLR_IRQ_STAT_ACK register is set, indicating the change of the supervisor.
- A beacon frame could not be received through both ports during the period specified by the beacon timeout time value.
  The node enters the idle state. Furthermore, bits 5 and 4 of the DLR_IRQ_STAT_ACK register are set, indicating that the beacon timeout timer has reached the timeout time on both ports.
- A beacon frame could not be received through either of the ports during the period specified by the beacon timeout time value.
  The node enters the fault state. Furthermore, bit 5 or 4 of the DLR_IRQ_STAT_ACK register is set, indicating that the beacon timeout timer has reached the timeout time on that port.

(4) Error Handling

The DLR node module is capable of handling the following error conditions:

- A CRC error being detected in beacon frames
  When a CRC error is detected in beacon frames, the DLR node does not process these frames but discard them before they enter the switch. Parameters in beacon frames which have a CRC error are not stored in the registers. On the other hand, since the CRC is not checked in the hub, the beacon frame will be transferred through the hub even if it has a CRC error. Beacon frames which have a CRC error are counted by the statistics counters RX_ERR_STAT0/1.
- The timeout timer value of beacon frames being outside the valid range
  The valid range of the timeout timer value of beacon frames is from 200 microseconds to 500 milliseconds. If beacon frames from the supervisor have an invalid beacon timeout value, they will be ignored and discarded before they enter the switch if bit 4 of the DLR_CONTROL register is set. Regardless of this setting, frames which have an invalid beacon timeout value are always detected and any invalid timeout value is stored in the INV_TMOUT register. When bit 11 of the DLR_IRQ_CTRL register is set, an interrupt is also generated.

This document mainly describes the DLR module incorporated in this LSI chip. For details of the DLR module, refer to the specification of ODVA.

13.4.4 IEEE 1588 Timer & Control Module

13.4.4.1 Overview

The timer & control module (TSM) has an adjustable timer for use with the Precision Time Protocol (PTP) defined by the IEEE 1588 standard. This allows synchronization of the local time of the timer with a remote master clock. However, this requires software compliant with the PTP or a similar protocol.

In addition, the module provides a reference time for timestamps of all frames acquired at the MAC/PHY interfaces of the external ports. The timestamps enable use of a time synchronization protocol such as the PTP to synchronize distributed clocks in the network with a common master clock.

13.4.4.2 IEEE 1588 Message Formats

(1) Transport Encapsulation

Datagrams for the Precision Time Protocol (PTP) are encapsulated in Ethernet frames by using the UDP/IP transport mechanism. In PTP v2, as well as UDP/IP, the PTP data may be directly transported in layer 2 Ethernet frames. Generally, multicast addresses are used for the efficient distribution of messages for synchronization.

- UDP/IP

  The 1588 messages (v1 and v2) can be transported by using UDP/IP multicast messages. The following IP multicast groups are defined for PTP. The table also shows MAC layer multicast address mapping according to RFC 1112.

Table 13.9 UDP/IP Multicast Domains

Name	IP Address	MAC Address Mapping
Default PTP domain	224.0.1.129	01-00-5e-00-01-81
Alternate PTP domain1	224.0.1.130	01-00-5e-00-01-82
Alternate PTP domain2	224.0.1.131	01-00-5e-00-01-83
Alternate PTP domain3	224.0.1.132	01-00-5e-00-01-84

Table 13.10 UDP Port Numbers

Message Type	UDP Port	Note
event	319	Used for SYNC and DELAY_REQUEST messages.
general	320	Used for the other messages (e.g. follow-up, delay-response)

- Native Ethernet (Layer 2)

  As previously stated, in addition to the usage of UDP/IP frames, IEEE 1588 version 2 defines a native Ethernet frame format. The frames are identified by the value 0x88F7 in the EtherType field. The payload of the Ethernet frame immediately contains the PTP datagram, starting with the PTP v2 header.

  Additions to PTP v2 include a peer delay mechanism. This allows measurement of delays between individual point-to-point links along a path over multiple nodes. The following multicast domains are also defined in PTP v2.

Table 13.11 PTPv2 Multicast Domains

Name	MAC Address
Normal messages	01-1b-19-00-00-00
Peer delay messages	01-80-c2-00-00-0e

(2) PTP Header

All PTP frames contain a common header, which includes the protocol version number as well as the type of message. The type of message further defines the contents of the message. All multi-octet fields are transmitted in big-endian order. The last 4 bits of the version field are at the same position in both PTP v1 and PTP v2 headers. Accordingly, the version can be correctly identified by checking the first 2 bytes of a message.

Note: Consult the IEEE 1588 standard for more details on the meanings of the contents of PTP frames. This document only covers some of the relevant information that will be useful in understanding the terminology. PTPv1 refers to version 1 of the IEEE 1588 standard while PTPv2 refers to version 2 of this standard.

- PTPv1 Header

Table 13.12 Common PTPv1 Message Header

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
versionPTP = 0x0001								2	0
versionNetwork								2	2
subdomain								16	4
messageType								1	20
sourceCommunicationTechnology								1	21
sourceUuid								6	22
sourcePortId								2	28
sequenceId								2	30
control								1	32
0x00								1	33
flags								2	34
reserved								4	36

The type of message is encoded in the messageType and control fields as listed in the table below:

Table 13.13 PTPv1 Message Type Identification

messageType	control	Message Name	Message
0x01	0	SYNC	event message
0x01	1	DELAY_REQ	event message
0x02	2	FOLLOW_UP	general message
0x02	3	DELAY_RESP	general message
0x02	4	MANAGEMENT	general message
other	other		reserved

- PTPv2 Header

Table 13.14 Common PTPv2 Message Header

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
ransportSpecific				messageId				1	0
reserved				versionPTP = 0x2				1	1
messageLength								2	2
domainNumber								1	4
reserved								1	5
flags								2	6
correctionField								8	8
reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
control								1	32
logMeanMessageInterval								1	33

The type of message is encoded in the messageId field as listed in the table below:

Table 13.15 PTPv2 Message Type Identification

messageId	Message Name	Message
0x0	SYNC	event message
0x1	DELAY_REQ	event message
0x2	PATH_DELAY_REQ	event message
0x3	PATH_DELAY_RESP	event message
0x4 – 0x7		reserved
0x8	FOLLOW_UP	general message
0x9	DELAY_RESP	general message
0xa	PATH_DELAY_FOLLOW_UP	general message
0xb	ANNOUNCE	general message
0xc	SIGNALING	general message
0xd	MANAGEMENT	general message

The PTPv2 flags field contains the details on the type of message, especially if one-step or two-step implementations are used. The flags field consists of two octets with the following meanings for the bits. Reserved bits are set to 0.

Table 13.16 PTPv2 Message Flags Field Definitions

Octet Offset	Bit	Name	Description
6 (first)	0	ALTERNATE_MASTER	See IEEE 1588 Clause 17.4.
	1	TWO_STEP	0: One-step clock 1: Two-step clock
	2	UNICAST	0: Multicast Addresses 1: Unicast Addresses
	3, 4	reserved	
	5	profile specific	
	6	profile specific	
	7	reserved	

Note: Please refer to the IEEE 1588 specification for details on frame formats and fields.

13.4.4.3 Adjustable Timer Module

(1) Overview

The adjustable timer module (TSM) has a free-running counter (FRC), which is used to generate timestamps for received and transmitted frames. The FRC of this LSI chip runs at 125 MHz, for a time resolution of 8 ns.

A dedicated time correction circuit can be used to adjust the timer for synchronization with a remote master and provide a time-synchronized reference to the local system.

The switch has two timers: a nanoseconds timer and a seconds timer. The nanoseconds timer reaching 10^9 leads to the generation of an interrupt.

This LSI chip is capable of generating time-synchronized pulse signals with desired cycles based on the current time values of the timers. It can also provide reference times to external systems.

(2) Timer Module Configuration

The adjustable timer consists of a programmable counter/accumulator and two correction counters. The periods of these counters and the values of their increments can be freely set. This allows fine tuning of the timer.

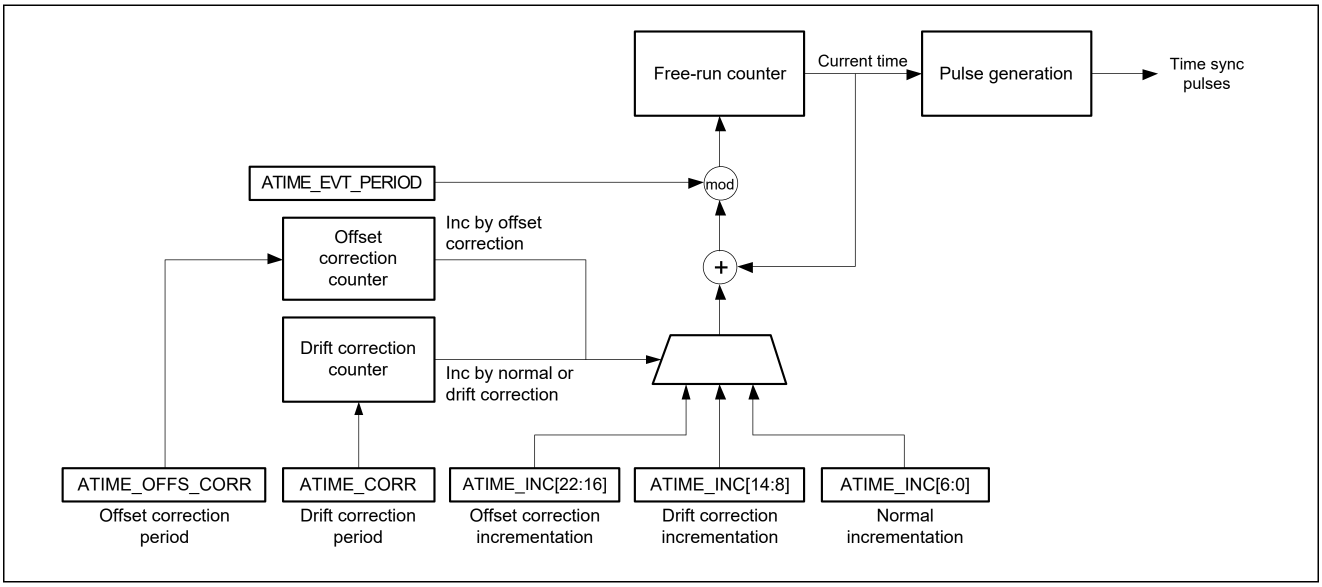


Figure 13.15 Configuration of Adjustable Timer

(3) Normal Timer Operation

The free-running counter (timer) continues to produce the current time. The constant value defined in bits 6 to 0 of the ATIME_INC register is added to the current time in each clock cycle. To achieve the correct time, do not set these bits to a value other than 001000b, which represents 8 (ns).

The period set in the ATIME_EVT_PERIOD register represents the modulus and is used in cycling the counter. Always set the value to 10^9. This allows the use of timestamps in nanoseconds.

(4) Drift Correction

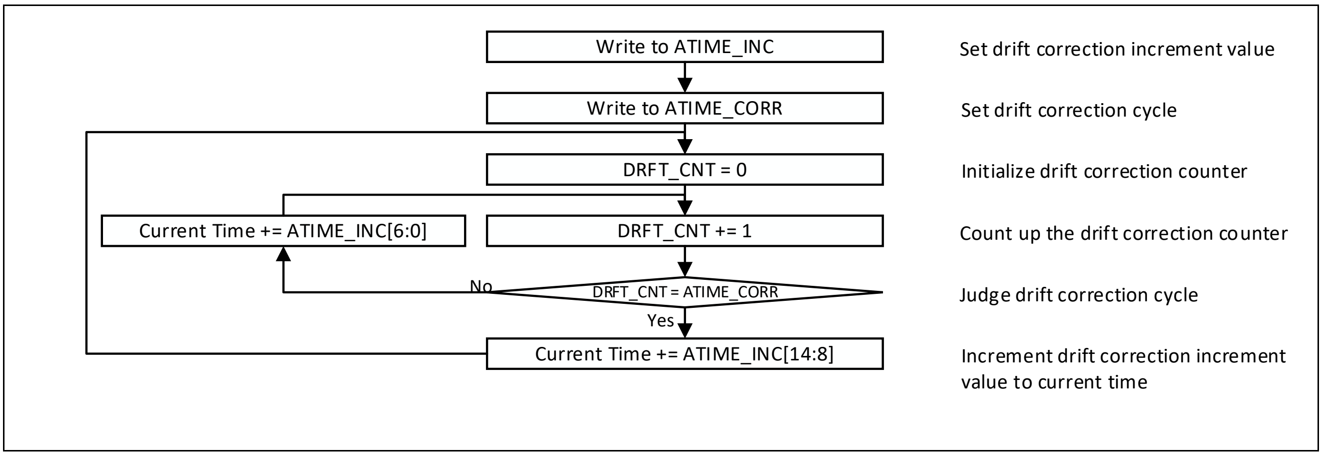


Figure 13.16 Drift Correction

The drift correction counter operates fully independently of the free-running counter (timer) and is incremented by 1 with each clock cycle. When it reaches the value set in the ATIME_CORR register, it is restarted and instructs the free-running counter to be incremented once by the correcting value, instead of the normal value. The normal and correction increments are set in the ATIME_INC register. To speed up the timer, set the correcting increment to a greater value than the normal increment. To slow down the timer, set the correcting increment to a smaller value than the normal increment. The correction counter does not define the amount of correction, but the interval of how many clock cycles at which correction proceeds. This allows very fine correction with a low jitter in units of 1 ns independently of the selected clock frequency.

(5) Offset Correction

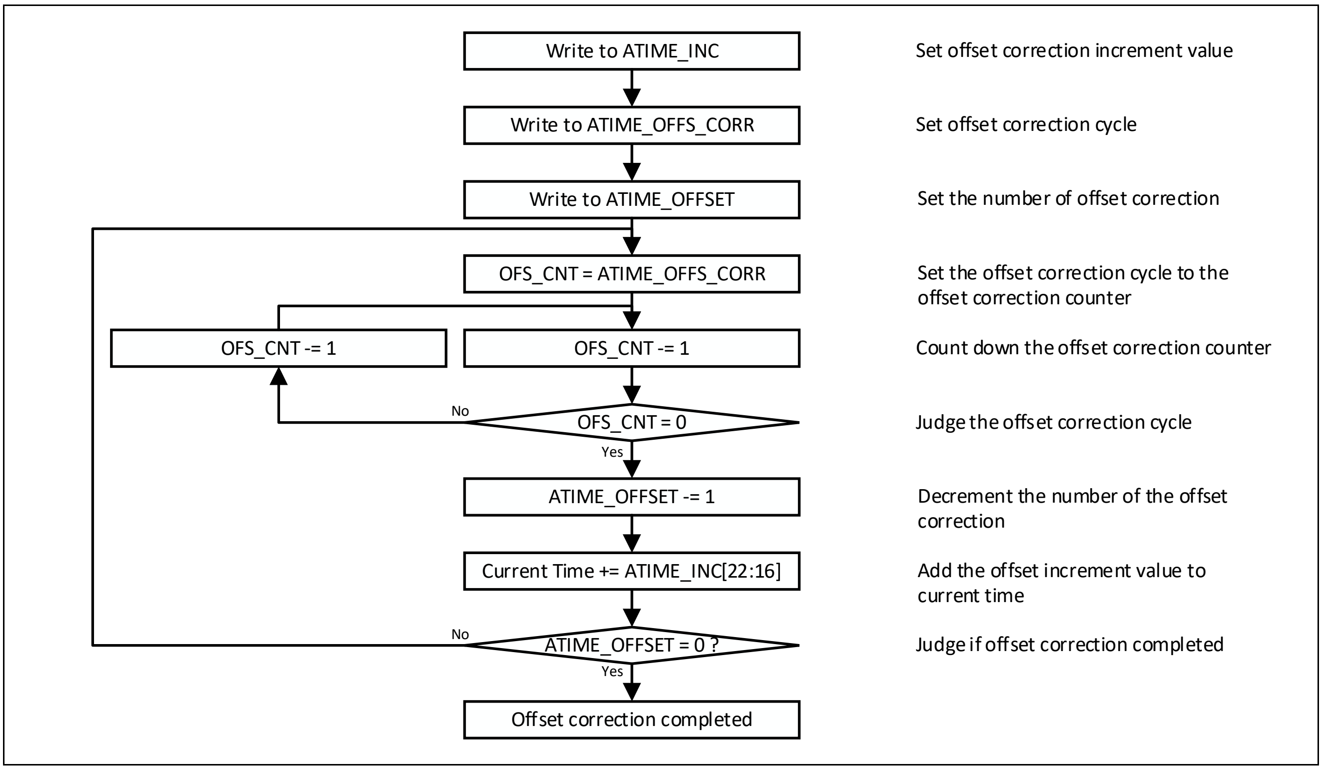


Figure 13.17 Offset Correction (when ATIME_OFFS_CORR is not zero)

The offset correction counter operates fully independently of the free-running counter (timer). When the offset correction counter is loaded with a value, it is decremented by 1 with each clock cycle. The value written to the ATIME_OFFS_CORR register is loaded to the counter. The timer does not start until an offset correction has been written to the ATIME_OFFSET register (i.e., a value must be written to the ATIME_OFF_CORR register before writing to the ATIME_OFFSET register).

When a value is written to the ATIME_OFFSET register, the offset correction counter is loaded with the value of the ATIME_OFFS_CORR register and starts counting. When the value counted reaches zero, it decrements the ATIME_OFFSET value by 1 and increments the timer by the offset value defined in bits 22 to 16 of the ATIME_INC register. If this does not cause the ATIME_OFFSET value to become zero, the counter reloads the value of the ATIME_OFFS_CORR register and repeats the procedure until the ATIME_OFFSET value does become zero. After the value becomes 0, further correction does not proceed.

With this correction, it is possible to shift the timer to another time without causing sudden large changes in the time. When the offset correction has been completed, the ATIME_OFFSET register becomes zero and the offset event interrupt can be triggered if this is required.

Alternatively, instead of applying offset correction over time by using the offset correction timer, it is possible to immediately change the current time by the offset amount. This leads to the value of the timer jumping to the time *current-time* + *offset*. This is achieved by setting the ATIME_OFFS_CORR register to zero and then writing the offset to the ATIME_OFFSET register. The timer offset value can be positive or negative.

(6) Generation of Pulse Signals

This LSI chip is capable of generating pulse signals with desired cycles based on the output value of the current time of the timers.

Table 13.17 Parameters for Generation of Pulse Signals

Parameter	Related Registers	Description
Enabling operation for pulse generation	SWTMEN	This register is for enabling or stopping the output of pulses.
Pulse output start time	SWTMSTSECL/H SWTMSTNSL/H	These registers are for specifying the time pulse output starts in units of seconds and nanoseconds. When the SWTMEN register is set to 1 to enable the generation of pulses after specifying the time output starts, pulses start to be output when the current time exceeds the specified time for output to start. Pulses are not output if operation for pulse generation is enabled later than specified time.
Pulse cycle	SWTMPSECL/H SWTMPNSL/H	These registers are for specifying the cycle for the output of pulses in units of seconds and nanoseconds. The SWTMPNSL and H registers must be set to a value corresponding to the division by a positive integer of the value for one second set in the ATIME_EVT_PERIOD register. In addition, specify a value that is a multiple of 8. The value must be set before enabling operation for pulse generation.
Pulse width	SWTMWTH	This register is for specifying the width at high level in units of nanoseconds of the pulse signal to be output. Specify a multiple of 8 ns. If the specified pulse width is greater than the pulse cycle, the output will be fixed to the high level. If the pulse width is set to 0, pulses are not generated and the output will be fixed to the low level. The value must be set before enabling operation for pulse generation.
Maximum counter value	SWTMMAXPL/H	This register is for specifying the maximum value of the nanoseconds counter. Set the SWTMMAXPL and H registers to the same value as the ATIME_EVT_PERIOD register (one second: SWTMMAXPH = 0000 3B9AH, SWTMMAXPL = 0000 CA00H). The value must be set before enabling operation for pulse generation.
Rise time retention	SWTMLATSEC SWTMLATNS	These registers hold the time when the pulse output signal last rose. The registers are updated every time the pulse signal rises.

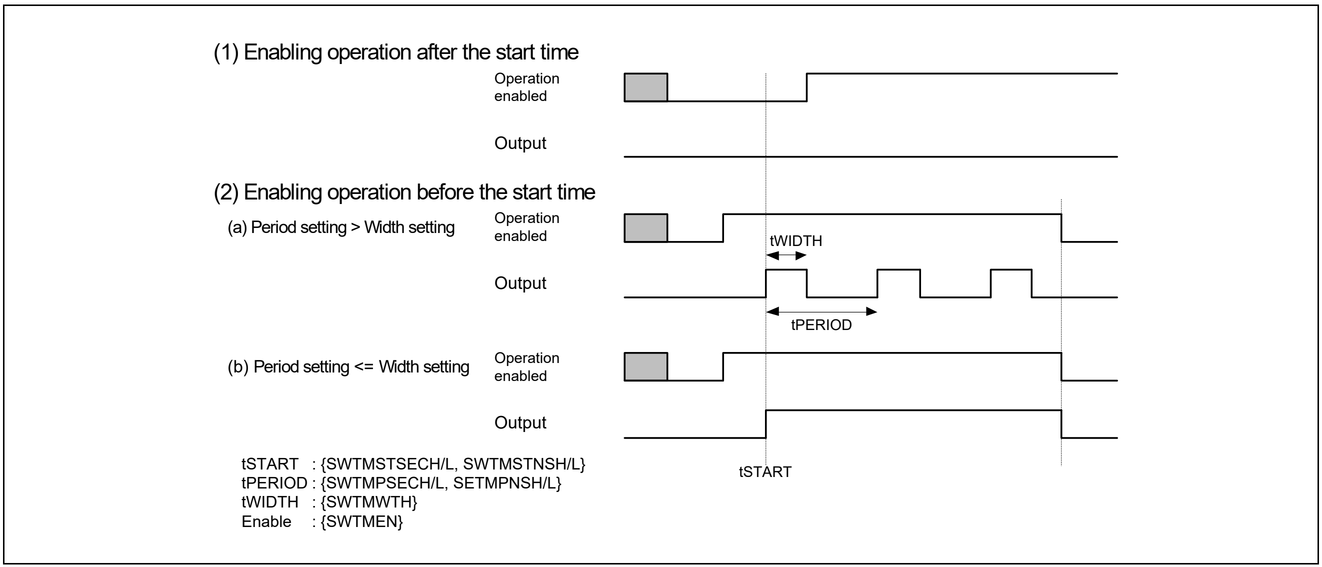


Figure 13.18 Timing Chart of Pulse Signal Generation

13.4.4.4 Timestamp Processing

(1) Reception Timestamp Processing

When a frame is received through port 0 or port 1, the timestamp based on the current time from the timer is captured when the start of frame delimiter (SFD) is detected at the PHY interface. The timestamp is transferred together with the frame in the switching module and can be accessed by the internal port (management port) of the switch. Use of the captured timestamp allows implementation of a protocol e.g. the Precision-Time-Protocol (PTP) in its application software. The timestamp information is encapsulated in in the frames as the dedicated tag.

(2) Transmission Timestamp Processing

When a frame is transmitted to the PHY through port 0 or port 1, the timestamp is also captured. The outgoing timestamp can be stored in the port specific timestamp register (PORTn_TIME) for each port. The internal port adds special control information to each frame to limit frames for capturing the outgoing timestamp. Timestamps can only be captured for specified event frames, not for all frames.

13.4.4.5 Support for Transparent Clocks

(1) Overview

The hardware implements the necessary functions to implement so-called transparent clocks (TC) for the end-to-end variant.

(2) Implementation of Correction Field Update

The correction field within outgoing Layer 2 PTP frames (i.e. frames with type 0x88F7) can be updated automatically. PTP messages within UDP/IP frames are not automatically updated.

The module for updating the correction field only processes event messages. To detect event messages, frames with the message type field found within the PTP header (type < 4) are extracted. This means that follow-up frames which are not event frames are not processed. Therefore, any correction field detected in the corresponding SYNC frame will be updated automatically. This allows supporting correction field update by one-step as well as two-step master and slave nodes.

For an end-to-end implementation, the correction field of SYNC and DELAY_REQ messages is updated only with the transient time (output time - input time).

Correction field updates occur only on frames that are exchanged between port 0 and port 1. Any frame transmitted from and to the internal port will not be modified.

13.4.5 Management Port (Internal Port) Specific Frame Tagging

Information related to frames such as control and timestamp information needs to be delivered between the Ethernet switch and internal Ethernet MAC. Such information can be appended to frames as a management tag. The frames with the tag can be transmitted between the Ethernet switch and internal Ethernet MAC. The frames with the tag are only used for transfers between the Ethernet switch and internal Ethernet MAC and once accepted on the receiving side, the information in the tag is acquired. Then the tag is removed.

13.4.5.1 Format of Management Tag

The additional control information and timestamp information are added into a frame right after the frame source address field as a frame type tag (programmable with a given value). The tag is added to the position before any other tag (VLAN tag), if exists. The tag includes the following information:

- ControlTag: Identifier indicating that the additional control data are present within the frame (defined by the ETHSWMTC register). The size is 2 octets.
- ControlData: Control information of the frame. The size is 2 octets.
- ControlData2: Specifies timestamp information on reception and transmit port on transmission. The size is 4 octets.

The original frame follows the ControlData2. For example, any VLAN tags will be found after ControlData2.

7 octets	PREAMBLE	
1 octet	SFD	
6 octets	DESTINATION ADDRESS	
6 octets	SOURCE ADDRESS	
2 octets	ControlTag	insert control information
2 octets	ControlData	
4 octets	ControlData2 (timestamp, portmask)	
2 octet	type/length	original frame contents
0..1500/9000 octets	PAYLOAD DATA	
0..42 octets	PAD	
4 octets	FRAME CHECK SEQUENCE	

Figure 13.19 Format of Frame with Management Tag in Internal Port

Once a tag is added to a frame, the CRC is recalculated and a new CRC replaces the original CRC received with the frame.

The first octet of ControlData is the more significant byte (bits 15:8) and the 2nd octet of ControlData is the less significant byte (bits 7:0). The first octet of ControlData2 is the more significant byte (bits 31:24) and the 4th octet of ControlData2 is the less significant byte (7:0).

13.4.5.2 Processing for Transmission (from the switch to the internal Ethernet MAC)

When the switch transmits a frame to the internal Ethernet MAC, the following information is added into all frames if tagging is enabled through bit 31 in register ETHSWMTC.

Table 13.18 Management Frame Tag (in transfer from the switch to the internal Ethernet MAC)

Field	Bit	Description
ControlData	0	Indicates the number of the external port through which the frame was received. 0: Port 0 1: Port 1
	15 to 1	Reserved
ControlData2	31 to 0	Indicate the received timestamp of the frame. The 32-bit nanoseconds value indicating the time when the frame start (SFD) was detected on the port where the frame was received.

13.4.5.3 Processing for Reception (from the internal Ethernet MAC to the switch)

When the internal Ethernet MAC transfers a frame to the switch, the internal Ethernet MAC adds the management tag which includes the following information to all frames if tagging is enabled through bit 31 in register ETHSWMTC. Once the switch receives the frame with a management tag, it removes the tag from the frame after acquiring the tag contents.

Table 13.19 Management Frame Tag (in transfer from the internal Ethernet MAC to the switch)

Field	Bit	Description
ControlData	0	Specifies forcible transfer. 0: Forcible transfer is disabled. Normal transfer processing proceeds. 1: Forcible transfer is enabled. Frames are transferred to all ports defined in bits 1 and 0 of ControlData2.
	1, 2	Reserved
	3	Specifies the frame for outgoing timestamping. When set, the frame transmit timestamp will be latched into the corresponding port's transmit timestamp register PORTn_TIME, when the frame is transmitted.
	15 to 4	Reserved
ControlData2	1, 0	Set a destination port mask. Relevant only if ControlData forcible transfer bit (bit 0) is set. Specify the port to which the frames are transferred. Simultaneous forcible transfer to multiple ports is possible. Bit 0 is for port 0 and bit 1 is for port 1. Each bit can be set as follows: 0: Frames are not forcibly transferred to the corresponding port. 1: Frames are forcibly transferred to the corresponding port.
	31 to 2	Reserved

13.4.5.4 Management Tag Settings

Insertion and removal of the management tag to and from frames are enabled with the ETHSWMTC register. When enabled, the tag will be automatically inserted to frames that are transferred to the internal Ethernet MAC from the switch. The tag will also be inserted to frames that are transferred to the switch from the internal Ethernet MAC. If the switch finds any tag, it acquires the tag information and removes the tag for normal transmission processing.

Cautions 1. The tag identifier must be configured to a value (e.g. initial value e001h) which is not used in the network.

2. For handling management tags in the internal Ethernet MAC, see Section 12, Gigabit Ethernet MAC.

13.5 Overview of Control Software

13.5.1 Overview

The Ethernet switch is the hardware to forward frames between ports. During the forwarding processing, the MAC destination address is searched and frames that requires special specific forwarding such as BPDU are filtered.

Software must initialize the switch and executes tasks to operate the switch. The minimum required task is management of the learning table.

The software that operates the IEEE1588 timestamp and the DLR is necessary, when they are used. In addition, the higher protocol such as spanning tree needs to be implemented as required.

This section describes the most basic procedures required for switch initialization and learning table management to operate the switch.

13.5.2 Switch Initialization

Follow the procedure below to make initial settings for the Ethernet switch.

- Clear the address table.
- Configure the management (internal) port.
- Enable all switch ports.
- Enable the MACs on ports.
- Configure the hub module.
- Configure the timer module.
- Configure the DLR module.

The following table lists examples of initial settings at least required for the operation of the switch. As to the timer module, correction is not performed, the DLR module is also disabled. For these, set appropriate values in higher-level protocols such as PTP and DLR.

Table 13.20 Examples of Initial Settings of the Address Table

Address	Register	Example Setting	Description
4007 4000H to 4007 47FC (4-byte units)	ADR_TABLE	0000 0000H	Initialize all entries to 0 in the address table.
4007 4000H + Hash value of Unicast MAC address × 8H	ADR_TABLE	0403 0201H	Set a unicast address as a static entry. The example settings are when the MAC address is 01-02-03-04-05-06. The priority level is 0 and only port 2 is masked. These settings are not required when set dynamically.
Address above +4H	ADR_TABLE	0083 0605H	

Table 13.21 Examples of Initial Settings of the Switch Engine

Address	Register	Example Setting	Description
4001 0680H	ETHSWMTC	0000 E001H	Does not use a management tag. Set the register to 8000 E001H to use it. Release the protection by using the system protect command register when writing to this register.
4001 0684H	ETHSWMD	0000 0000H	Set the mode of 10/100Mbps full-duplex. Release the protection by using the system protect command register when writing to this register.
4007 000CH	UCAST_DEFAULT_MASK	0000 0007H	Used to mask transfer of unknown unicast frames. When an unknown unicast frame is received at any port, it is transferred to all ports set by this mask. When the address table has been initialized, the management (internal) port can be removed from the list. This prevents unnecessary transfer of unicast frames to the local system. However, this requires the local device's unicast address to have been set in the address table (i.e. either set during initialization statically, or change the mask setting after at least one frame was sent from the local system and dynamically activate the learning function).
4007 0014H	BCAST_DEFAULT_MASK	0000 0007H	Defines all ports where a broadcast frame will be forwarded to.
4007 0018H	MCAST_DEFAULT_MASK	0000 0007H	Defines all ports where a multicast frame will be forwarded to, if the address is not found in the address table.
4007 0020H	MGMT_CONFIG	0000 0042H	Enables reception of BPDU frames (bit 6 = 1) to transfer them to the management port (port 2) If management frames should be discarded, bit 7 should be set to 1.
4007 0100H 4007 0104H 4007 0108H	VLAN_PRIORITY0 VLAN_PRIORITY1 VLAN_PRIORITY2	006D B688H	Map VLAN priority into the 4 queues available for each port. In this setting, priorities 0 to 3 are mapped into queues 0 to 3 and priorities 4 to 7 all into queue 3.
4007 0180H 4007 0184H 4007 0188H	PRIORITY_CFG0 PRIORITY_CFG1 PRIORITY_CFG2	0000 0001H	Enable mapping of the output queue by VLAN priority classification for each port and set default port priority to 0.
4007 0080H	OQMGR_STATUS	0000 0000H	Enables the output queue. Since bit 1 is set to 1 during the initialization of the memory cell, if bit 1 becomes 0, the register should be cleared to 0.
4007 0088H	QMGR_ST_MINCELLS	0000 0000H	Clearing of the minimum value for free memory space
4007 0094H	QMGR_WEIGHTS	0804 0201H	Sets the weight on the output queue.
4007 0008H	PORT_ENA	0000 0007H	Enables all ports of the switch.

Table 13.22 Examples of Initial Settings of the MAC

Address	Register	Example Setting	Description
4007 801CH 4007 A01CH	RX_SECTION_EMPTY0 RX_SECTION_EMPTY1	0000 0000H	The value cannot be changed.
4007 8020H 4007 A020H	RX_SECTION_FULL0 RX_SECTION_FULL1	0000 0000H	The value cannot be changed.
4007 8024H 4007 A024H	TX_SECTION_EMPTY0 TX_SECTION_EMPTY1	0000 0048H	The MAC has 128-stage FIFO buffers. If an entry to the transmit FIFO buffer is above the threshold, the transfer of data from internal to the transmit FIFO buffer stops. This setting is a threshold to prevent TX overflow. Set a value of at least 65.
4007 8028H 4007 A028H	TX_SECTION_FULL0 TX_SECTION_FULL1	0000 0014H	Set the number of entries required for the transmit FIFO buffer to start transmission. Set a value of at least 17.
4007 802CH 4007 A02CH	RX_ALMOST_EMPTY0 RX_ALMOST_EMPTY1	0000 0008H	The value cannot be changed.
4007 8030H 4007 A030H	RX_ALMOST_FULL0 RX_ALMOST_FULL1	0000 0005H	The value cannot be changed.
4007 8034H 4007 A034H	TX_ALMOST_EMPTY0 TX_ALMOST_EMPTY1	0000 0004H	The value cannot be changed.
4007 8038H 4007 A038H	TX_ALMOST_FULL0 TX_ALMOST_FULL1	0000 0010H	The value cannot be changed.
4007 8014H 4007 A014H	FRM_LENGTH0 FRM_LENGTH1	0000 05F2H	Set the maximum allowable value of the received frame size. The example setting is 1522, sufficient for 1 VLAN tagged frame. The value can also be set to around 1536 to allow a margin.
4007 8008H 4007 A008H	COMMAND_CONFIG0 COMMAND_CONFIG1	0580 0013H	Enable the transmission and reception by the MAC.

Table 13.23 Initial Settings of the Hub

Address	Register	Example Setting	Description
4007 01C8H	HUB_FLT_MAC0lo	00C2 8001H	Example settings of switch management frames such as spanning tree. For filtering the MAC address of 01-80-c2-00-00-{00.3F}.
4007 01CCH	HUB_FLT_MAC0hi	00C0 0000H	
4007 01D0H	HUB_FLT_MAC1lo	0019 1B01H	Example settings of normal messages of PTPv2.
4007 01D4H	HUB_FLT_MAC1hi	00FF 0000H	For filtering the MAC address of 01-1b-19-00-00-00.
4007 01D8H	HUB_FLT_MAC2lo	005E 0001H	Example settings of UDP/IP messages of PTP.
4007 01DCH	HUB_FLT_MAC2hi	00F8 8001H	For filtering the MAC address of 01-00-5e-00-01-{80..87}.
4007 01E0H	HUB_FLT_MAC3lo	005E 0001H	Example settings of management frames.
4007 01E4H	HUB_FLT_MAC3hi	00FC 0000H	For filtering the MAC address of 01-00-5e-00-00-{00..03}.
4007 01E8H	HUB_FLT_MAC4lo	0403 0201H	Set unicast addresses.
4007 01ECH	HUB_FLT_MAC4hi	00FF 0605H	The example settings are for 01-02-03-04-05-06.
4007 01F0H	HUB_FLT_MAC5lo	0000 0000H	Example settings when the hub is not used.
4007 01F4H	HUB_FLT_MAC5hi	00FF 0000H	
4007 01F8H	HUB_FLT_MAC6lo	006C 2101H	Example settings of beacon frames of the DLR.
4007 01FCH	HUB_FLT_MAC6hi	01FF 0100H	For forcible transfer of frames of the MAC address of 01-21-6C-00-00-01.
4007 01C0H	HUB_CONTROL	0000 00AFH	Enables the hub. Set it to 0000 00A0H when not using the hub.

Table 13.24 Examples of Initial Settings of the Timer Module

Address	Register	Example Setting	Description
4007 C004H	TSM_CONFIG	0000 300BH	Enables one second arrival interrupt (bit 2) of the nanosecond-timer and interrupt generation except for test interrupt (bit 4).
4007 C008H	TSM_IRQ_STAT_ACK	0000 301FH	Clears all interrupts.
4007 C138H	ATIM_SEC	0000 0000H	Initializes the timer. This register should be set before ATIME.
4007 C124H	ATIME	0000 0000H	Initializes the timer.
4007 C12CH	ATIME_EVT_PERIOD	3B9A CA00H	Sets 1 second.
4007 C134H	ATIME_INC	0000 0808H	Sets the clock period. Correction is not applied.
4007 C130H	ATIME_CORR	0000 0000H	Drift correction is not applied.
4007 C120H	ATIME_CTRL	0000 00A1H	Starts the timer. Correction is not applied.
4007 C020H 4007 C028H	PORT0_CTRL PORT1_CTRL	0000 0000H	Clear the timestamp control and status registers.

Table 13.25 Examples of Initial Settings of the DLR Module

Address	Register	Initial setting	Description
4007 E000H	DLR_CONTROL	0000 6400H	Sets the clock period of the timeout timer. The DLR is disabled.
4007 E008H	DLR_ETH_TYP	0000 80E1H	Sets Ether type of DLR frames.
4007 E00CH	DLR_IRQ_CTRL	0000 0000H	Disables the generation of DLR interrupts.
4007 E010H	DLR_IRQ_STAT_ACK	0000 FFFFH	Clears all interrupts.
4007 E014H	LOC_MAClo	0403 0201H	Set unicast addresses. The example settings are for 01-02-03-04-05-06.
4007 E018H	LOC_MAChi	0000 0605H	

13.5.3 Address Table Setting

13.5.3.1 Definition of Block Entry of Address Table

When the Ethernet switch receives frames, it searches the address table to find the destination port(s) the frame should be forwarded to. Software is not involved in the forwarding process and all frame processing is performed in hardware. Software, however, takes care of the address table initialization and management. This software task does not require a high priority. However, during operations, a low priority software task is required to continually check for learning data and add MAC addresses to the table or delete old entries when they are out of use for a longer time.

The hardware operates on hash values for an immediate search for the address table. A hash value is used directly as starting address to the address table to search for entries. The next 8 entries starting address are targeted for a linear search to find the MAC address. This is a system called “block entry”.

When the address table is small, the individual per-hash blocks of 8 entries do overlap. The hardware however does not distinguish and will always search all 8 entries starting with the first entry that is pointed to by the hash value. This allows efficient storage in a smaller table without the need to reduce the per hash entries available in a block.

The following figure shows the principle of the address table layout. The software designer needs to understand this when writing the learning and aging functions.

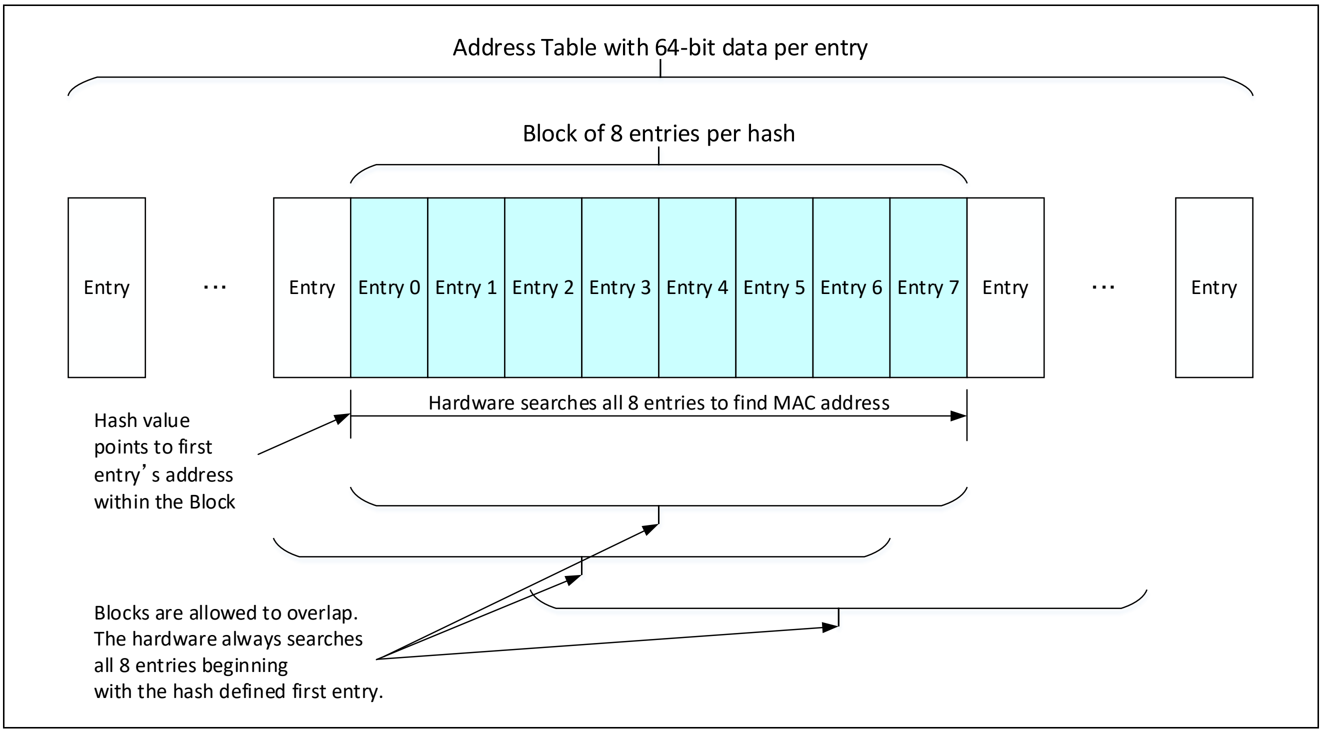


Figure 13.20 Entries of Address Table and Definition of Hash Block

13.5.3.2 Address Learning

The address table is used to identify the ports through which frames must be transmitted. The hardware automatically looks up the address table when it receives a frame to determine its destination. The software is responsible for keeping the address table updated and inserting the forwarding information that is then used by the hardware.

Control of learning by software is a low-priority background task, which continually inspects the learning data (i.e. retrieves source addresses and port numbers of received frames) and updates the address table whenever it finds a new address.

Learning proceeds through the following steps.

- Read data from the learning interface (via registers LRN_REC_A/B): The data records include a hash value, which is used as the start address where the entries in the address table should be found.
- The 8 entries from the hash-generated start address are searched and the aging time is updated if the entry is already in the table (or the port number is updated if it has been changed).
- If the entry is not found in any of the 8 entries in the address table, the entry is a new entry that must be added. Adding a new entry is either done into an unused position of the 8 entries, or overwriting a current entry (e.g. random, or the oldest).

The following figure shows the individual steps in learning and how an address table control function should be implemented. Implement the address learning task with reference to this flowchart.

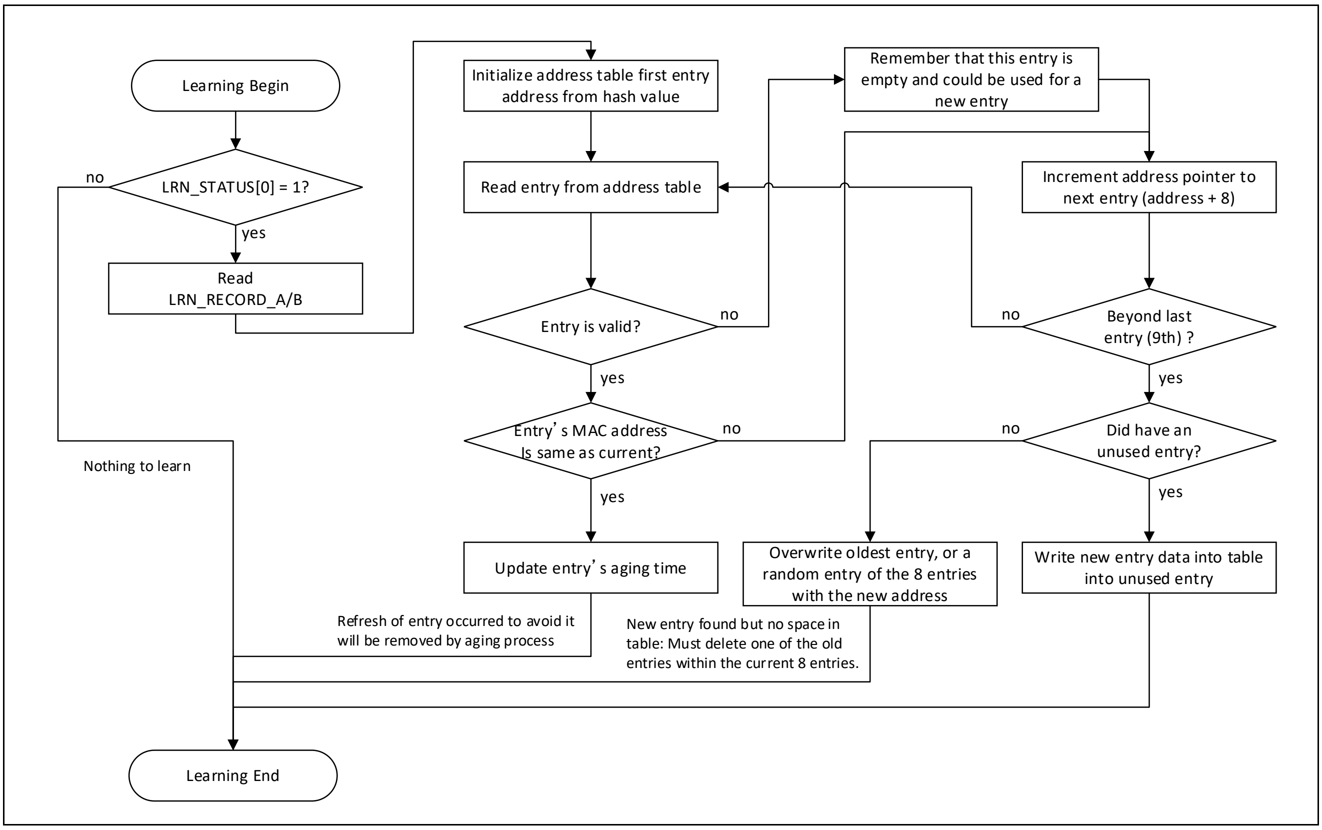


Figure 13.21 Address Learning Flow

14. Asynchronous SRAM Memory Controller (ROM/RAM)

The asynchronous SRAM memory controller is connectable to external paged ROM, ROM, and SRAM through a 16- or 32-bit bus. It is also connectable to peripheral devices compliant with the SRAM interface.

The pin functions for the asynchronous SRAM memory controller are multiplexed with those for the synchronous burst access memory controller and the external MCU interface, and the asynchronous controller can be used when the low level is applied to both the MEMCSEL and MEMIFSEL pins.

When both the BOOT0 and BOOT1 pins are at the low level, booting is from the memory connected to CSZ0.

Caution: Do not change the setting of the operating mode setting pins such as the MEMIFSEL and MEMCSEL pins during operation. Fix the setting before release from the reset state.

14.1 Overview

○ 32- or 16-bit data bus

○ Static memory control

- Four SRAM controller channels (channel 0 has a page ROM controller)
- SRAM and external I/O connection
- Page ROM connection (CSZ0 only)
- Programmable wait
  - Address setup wait
  - Data wait
  - Write recovery wait
  - Idle wait

○ Write strobe and byte enable are multiplexed

Caution: The memory controllers of an R-IN32M4-CL3 do not support an 8-bit bus width.

14.2 Features

(1) Static memory control

The memory controllers of an R-IN32M4-CL3 control the static memory (SRAM, I/O, or page ROM) connected to CSZ0 to CSZ3. Note that the page ROM can only be connected to CSZ0.

(a) SRAM and external I/O connection

The main features of the SRAM and external I/O connection are as follows.

- Minimum read cycle pattern of 4 cycles of BUSCLK
- Minimum write cycle pattern of 5 cycles of BUSCLK
- An address setup wait of up to 15 BUSCLK cycles can be inserted by setting the relevant register.
- A data wait of up to 15 BUSCLK cycles can be inserted by setting the relevant register.
- A write recovery wait of up to 15 BUSCLK cycles can be inserted by setting the relevant register.
- An idle wait of up to 16 BUSCLK cycles can be inserted by setting the relevant register.
- A data wait can be inserted by using external pin input.

(b) Page ROM connection

The main features of page ROM connection are as follows.

- The page ROM can only be connected to CSZ0.
- Minimum read cycle pattern of 3 cycles of BUSCLK
- On-page access judgment
- The address comparison bit width can be changed by setting the relevant register.
- An address setup wait of up to 15 BUSCLK cycles can be inserted by setting the relevant register.
- A data wait of up to 15 BUSCLK cycles can be inserted by setting the relevant register.
- An idle wait of up to 16 BUSCLK cycles can be inserted by setting the relevant register.
- A data wait can be inserted by using external pin input.
- If a write cycle is requested for an area where the page ROM is allocated, an SRAM write cycle is started.
- A write recovery wait of up to 15 BUSCLK cycles can be inserted by setting the relevant register.

Caution: On-page access to paged ROM is judged for each fixed-length burst.
In a fixed-length burst transfer over the AHB, off-page access proceeds in the first read cycle and on-page access proceeds in the second and subsequent read cycles.
In a single transfer or undefined length burst transfer over the AHB, on-page access does not proceed. The minimum number of cycles for off-page access is 3 cycles of BUSCLK.

Remark: The frequency of the BUSCLK is the same as that of the HCLK.

(2) Endian

The memory controllers of an R-IN32M4-CL3 always operate in little endian mode.

14.3 Bus Control

Operating an R-IN32M4-CL3 requires setting the bus control registers.

14.3.1 Overview of Registers

Table 14.1 Overview of Bus Control Registers

Register Name	Symbol	Address
Bus size control register	BSC	400A 2004H
Static memory control registers 0 to 3	SMC0 to SMC3	400A 2008H to 400A 2014H
Page ROM control register	PRC	400A 2018H
Write enable switching register	WREN	BASE + 0100H

14.3.2 Bus Size Control Register (BSC)

The BSC register sets the data bus width for the memory to be accessed for each chip select signal.

The SBS3 to SBS0 bits correspond to the chip select output pins (CSZ3 to CSZ0).

The initial value of the BSC register differs depending on the input level of the BUS32EN pin.

- Access This register can be read or written in 32-bit units.
  Be sure to set 0 to bits 31 to 16, 15, 13, 11, and 9. Be sure to set 1 to bits 14, 12, 10, 8, 6, 4, 2, and 0. Note 2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
BSC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	SBS3	1	SBS2	1	SBS1	1	SBS0	1	400A 2004H Initial value Note 1
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	R/W	1	R/W	1	R/W	1	R/W	1	

Bit Position	Bit Name	Description
31 to 15, 13, 11, 9	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
14, 12, 10, 8, 6, 4, 2, 0	—	Reserved. When writing to these bits, write 1. When read, 1 is returned.
7, 5, 3, 1	SBS3–SBS0	Sets the data bus width for each chip select output pin (CSZ3–CSZ0). 0: 16 bits 1: 32 bits

Notes 1. The external bus size changes as follows by the input of a signal to the BUS32EN pin.

BUS32EN	External bus size at startup	BSC register	A1 pin operation	D16–D31 pin operation
0	16 bits	0000 5555H	A1	Not used
1	32 bits	0000 FFFFH	Low level output	D16–D31

2. Do not overwrite a bit fixed to 1 or 0 with any other value. If this is done, correct operation is not guaranteed.

14.3.3 Static Memory Control Registers 0 to 3 (SMC0 to SMC3)

The SMC0 to SMC3 registers set a specific wait for each chip select output pin (CSZ0 to CSZ3).
One wait is equivalent to one BUSCLK cycle. The frequency of the BUSCLK is the same as that of the internal system bus (AHB) clock, HCLK.

- Access These registers can be read or written in 32-bit units.
  Be sure to set 0 to bits 31 to 16.

n (n= 0 to 3) in the bit names IWn, WWn, DWn, and ACn corresponds to the area numbers of channels 0 to 3.

Cautions 1. Do not set 0000 0000H in the SMC0 to SMC3 registers. Make sure that the total of the IW (idle wait), DW (data wait), and AC (address setup wait) times is at least 1.
2. Do not write to the SMC0 to SMC3 registers for unused channels.

(1/3)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMCn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IWn3	IWn2	IWn1	IWn0	WWn3	WWn2	WWn1	WWn0	DWn3	DWn2	DWn1	DWn0	ACn3	ACn2	ACn1	ACn0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: 400A 2008H+ n×4H
Initial value: 0000 FFFFH

Bit Position	Bit Name	Description
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 12	IWn3–IWn0	Sets an idle wait for each CSZn. The settings of these bits apply only to an idle wait after read access. An idle wait is the minimum cycle from de-assertion of CSZn (CSZn: L → H) to assertion of the next bus cycle. An idle wait is inserted in cases such as when a read cycle has a long data float time and a bus fight occurs in the subsequent write cycle.

IWn3	IWn2	IWn1	IWn0	Number of idle wait cycles for CSZn (IWn+1)
0	0	0	0	1 cycle of BUSCLK
0	0	0	1	2 cycles of BUSCLK
0	0	1	0	3 cycles of BUSCLK
⋮				⋮
1	1	0	1	14 cycles of BUSCLK
1	1	1	0	15 cycles of BUSCLK
1	1	1	1	16 cycles of BUSCLK (initial value)

Caution: One idle wait is inserted immediately after read access or write access. The setting of the IWn3 to IWn0 bits reflects the number of idle wait cycles after read access. The number of idle wait cycles after write access is always 1.

Remarks 1. n = 0 to 3
2. BUSCLK = HCLK

(2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>11 to 8</td><td>WWn3–WWn0</td><td>Set a write recovery wait for each CSZn.
A write recovery wait is the cycle from de-assertion of WRSTBZ and WRZn (WRZn: L → H) to de-assertion of CSZn (CSZn: L → H).
A write recovery wait is inserted in cases such as when the chip is used for a low-speed device that requires an interval between write operations
<table>
<tr><th>WWn3</th><th>WWn2</th><th>WWn1</th><th>WWn0</th><th>Number of write recovery wait cycles for CSZn</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td rowspan="2">1 cycle of BUSCLK</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 cycles of BUSCLK</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 cycles of BUSCLK</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 cycles of BUSCLK</td></tr>
<tr><td colspan="4">⋮</td><td>⋮</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13 cycles of BUSCLK</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14 cycles of BUSCLK</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15 cycles of BUSCLK (initial value)</td></tr>
</table>
Caution: The number of write recovery wait cycles cannot be set to 0 cycles of BUSCLK.
A write recovery wait of 1 BUSCLK cycle is always inserted.</td></tr>
<tr><td>7 to 4</td><td>DWn3–DWn0</td><td>Set a data wait for each CSZn.
In the case of no wait, RDZ and WRZn having a width of 1 cycle of BUSCLK are extended by the number of wait cycles set for the data wait.
<table>
<tr><th>DWn3</th><th>DWn2</th><th>DWn1</th><th>DWn0</th><th>Number of data wait cycles for CSZn</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0 (Setting prohibited in case of Page ROM)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 cycle of BUSCLK</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 cycles of BUSCLK</td></tr>
<tr><td colspan="4">⋮</td><td>⋮</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13 cycles of BUSCLK</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14 cycles of BUSCLK</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15 cycles of BUSCLK (initial value)</td></tr>
</table>
Caution: When the page ROM is used, set the DW03–DW00 bits in the SMC0 register to at least 0001B (number of wait cycles = 1).
The DW03–DW00 bits in the SMC0 register are used for off-page access when the page ROM is used.</td></tr>
</table>

Remarks 1. n = 0 to 3

2. BUSCLK = HCLK

(3/3)

Bit Position	Bit Name	Description
3 to 0	ACn3–ACn0	Sets an address setup wait for each CSZn. An address setup wait is the cycle from the time when CSZn is asserted (CSZn: H → L: the same timing as the address change point) until RDZ, WRSTBZ, or WRZn is asserted (REZ/WEZ: H → L). An address setup wait is inserted as necessary for access to a device that requires a setup time for an address or chip select signal for a read/write strobe.

ACn3	ACn2	ACn1	ACn0	Number of address setup waits for CSZn
0	0	0	0	0 (for reading) or 1 cycle of BUSCLK (for writing) Note
0	0	0	1	1 cycle of BUSCLK
0	0	1	0	2 cycles of BUSCLK
⋮				⋮
1	1	0	1	13 cycles of BUSCLK
1	1	1	0	14 cycles of BUSCLK
1	1	1	1	15 cycles of BUSCLK (initial value)

Note: An address setup wait of one cycle of BUSCLK is always inserted for a write operation.

Caution: The address setup wait set by this register is also inserted in on-page access to the page ROM.

Remarks 1. n = 0 to 3

2. BUSCLK = HCLK

14.3.4 Page ROM Control Register (PRC)

The register specifies the type of memory connected to chip select output 0 (CSZ0) and sets the addresses to be masked (not to be compared) out of the addresses (A3 to A6) and the number of wait cycles for BUSCLK according to the configuration of the page ROM to be connected when page ROM is selected and the number of bits that can be read consecutively. The frequency of BUSCLK is the same as that of the internal system bus (AHB) clock HCLK.

The static memory control register 0 (SMC0) is used to set wait cycles for off-page access. The number of wait cycles for off-page access should be at least 1.

- Access This register can be read or written in 32-bit units.
  Be sure to set 0 to bits 27 to 20 and 15 to 1.

Cautions 1. The page ROM can only be used in on-page ROM mode when it is connected to CSZ0.
2. Be sure to set the number of wait cycles for off-page access to at least 1.

Address 400A 2018H, Initial value F000 0000H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRC	PRW3	PRW2	PRW1	PRW0	0	0	0	0	0	0	0	0	MA6	MA5	MA4	MA3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST
R/W	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W

Bit Position	Bit Name	Description
31 to 28	PRW3–PRW0	Set the number of data wait cycles for on-page access to the page ROM of CSZ0.
19 to 16	MA6–MA3	Set the mask bits for comparison of addresses.
0	ST	Specifies the type of memory connected to chip select output 0 (CSZ0). 0 : SRAM, I/O device (initial value) 1 : Page ROM

PRW3–PRW0:

PRW3	PRW2	PRW1	PRW0	Number of wait cycles for on-page access of CSZ0
0	0	0	0	Setting prohibited
0	0	0	1	1 cycle of BUSCLK
0	0	1	0	2 cycles of BUSCLK
0	0	1	1	3 cycles of BUSCLK
⋮				⋮
1	1	0	1	13 cycles of BUSCLK
1	1	1	0	14 cycles of BUSCLK
1	1	1	1	15 cycles of BUSCLK (initial value)

MA6–MA3:

MA6	MA5	MA4	MA3	Page size of page ROM to be connected
0	0	0	0	32 bits × 2, 16 bits × 4 (initial value)
0	0	0	1	32 bits × 4, 16 bits × 8
0	0	1	1	32 bits × 8, 16 bits × 16
0	1	1	1	32 bits × 16, 16 bits × 32
1	1	1	1	32 bits × 32, 16 bits × 64
Other than the above				Setting prohibited (If set, correct operation is not guaranteed).

Sets the addresses to be masked (not to be compared) out of the addresses (A3 to A6) according to the configuration of the page ROM to be connected and the number of bits that can be read consecutively by using the page ROM control register (PRC).

The figure below shows an example of address mask control when the two page ROMs of 512 Kwords × 16 bits are connected.

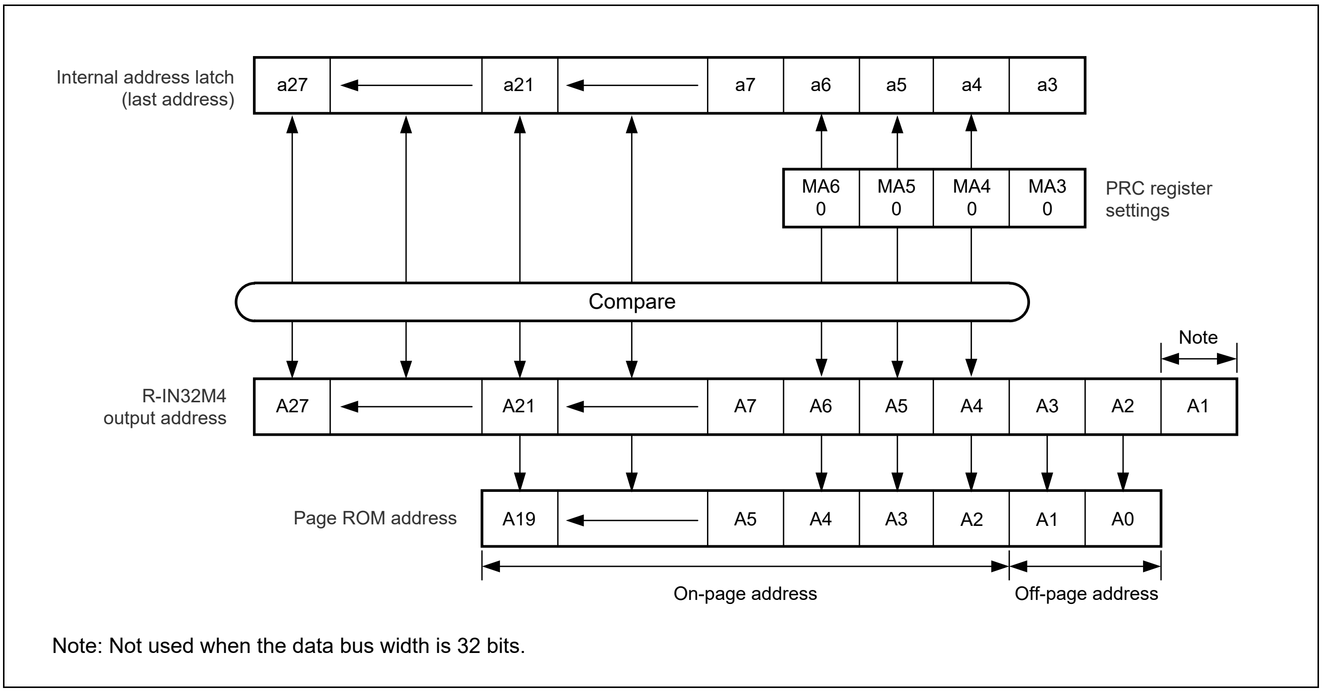


Figure 14.1 Example of Control Using the MA6 to MA3 Bits of the PRC Register

(1) On-page access judgment

On-page access is judged for transfer of the second or subsequent word during a fixed length burst transfer. Off-page access is performed in transfer of a first word, single transfer, or burst transfer of an undefined length.

Caution: If an access request is made that requires a greater width for data transfer than the width of the paged ROM data bus, the access to the page ROM is divided and on-page access is judged separately for access to the low- and high-order parts. When access is divided, on-page access is judged per transfer. In the case of a burst transfer of undefined length which can be replaced by consecutive single transfers, on-page access is judged for the low- and high-order parts on a word-by-word basis. On-page access is not judged between words; access is always off-page.

14.3.5 Write Enable Switching Register (WREN)

This register selects WRZ0 to WRZ3 or BENZ0 to BENZ3 for the BENZ0 to BENZ3 pin function.

The WREN register can be read or written in 32-bit units. The register is set to 0000 0001H by a reset and the BENZ0 to BENZ3 pins operate as WRZ0 to WRZ3.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
																																	BASE + 0100H
WREN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WREN	Initial value 0000 0001H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

Bit Position	Bit Name	Description
31 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	WREN	Selects the operation of the BENZ0 to BENZ3 pins of the external memory interface. 0: Operates as BENZ0 to BENZ3. 1: Operates as WRZ0 to WRZ3.

Caution: If 1 is written to an unused bit, correct operation is not guaranteed.
This register is only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)".
No special sequence is required for reading the register.

14.4 Memory Connection Examples

14.4.1 SRAM Connection Example

An example of connection with SRAM is shown below.

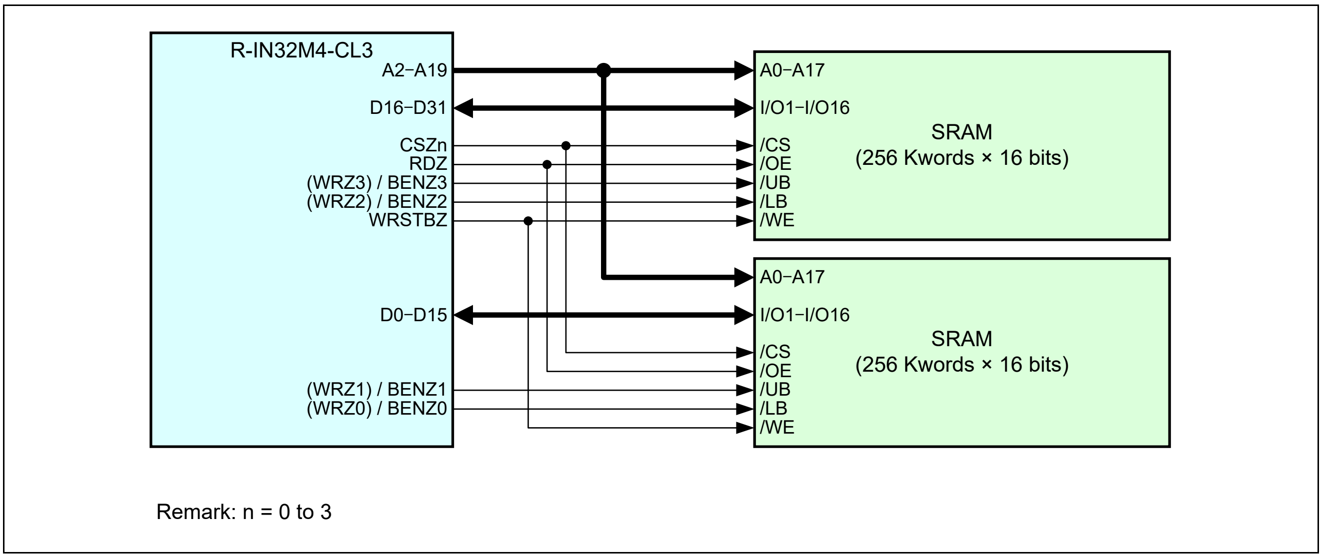


Figure 14.2 Example of Connection with 32-Bit SRAM

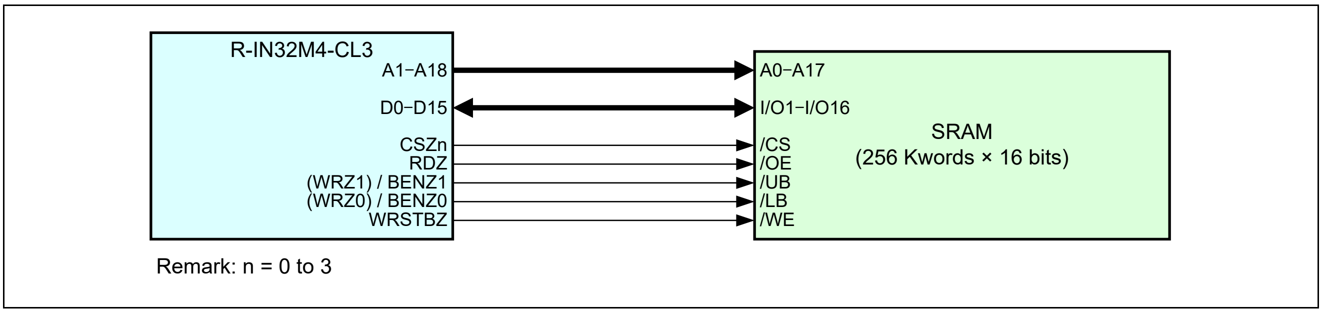


Figure 14.3 Example of Connection with 16-Bit SRAM

14.4.2 Page ROM Connection Example

An example of connection with page ROM is shown below.

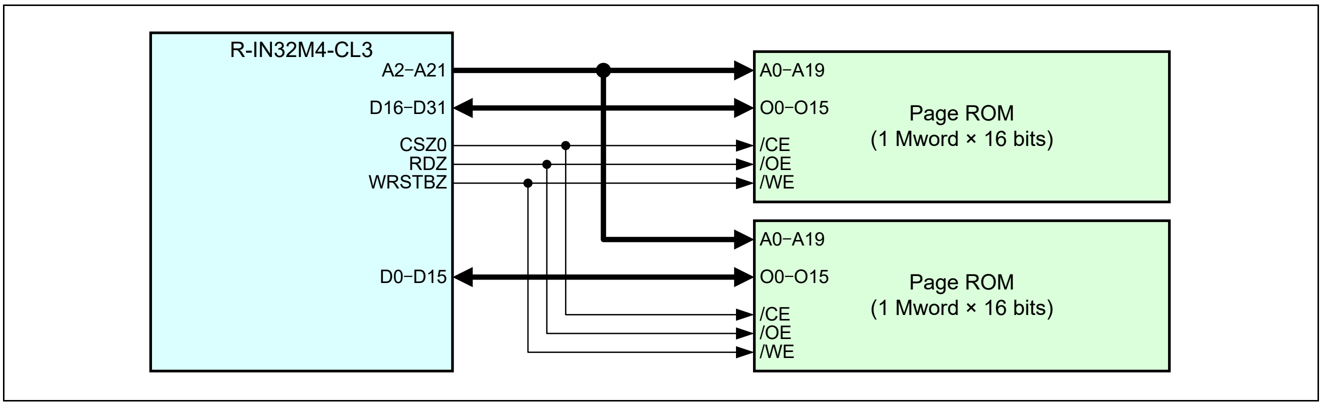


Figure 14.4 Example of Connection with 32-Bit Page ROM

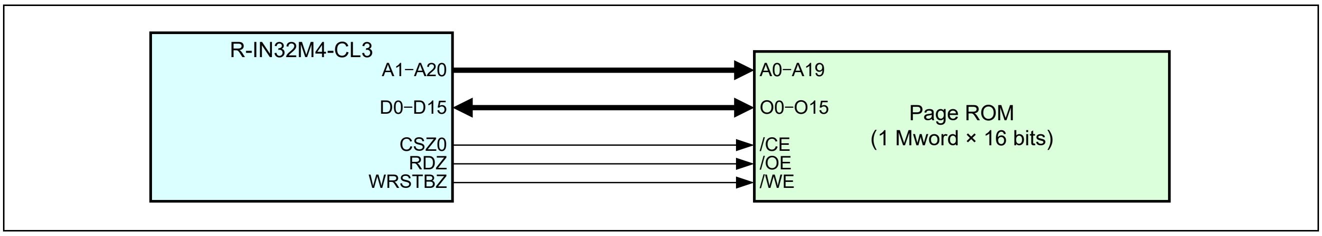


Figure 14.5 Example of Connection with 16-Bit Page ROM

Caution: On-page mode of the page ROM is only available when it is connected to CSZ0.

14.5 Procedure for Setting the Control Registers

The procedure for setting the control registers is described below using an example of connecting the page ROM and SRAM to the CSZ0 and CSZ1 areas respectively.

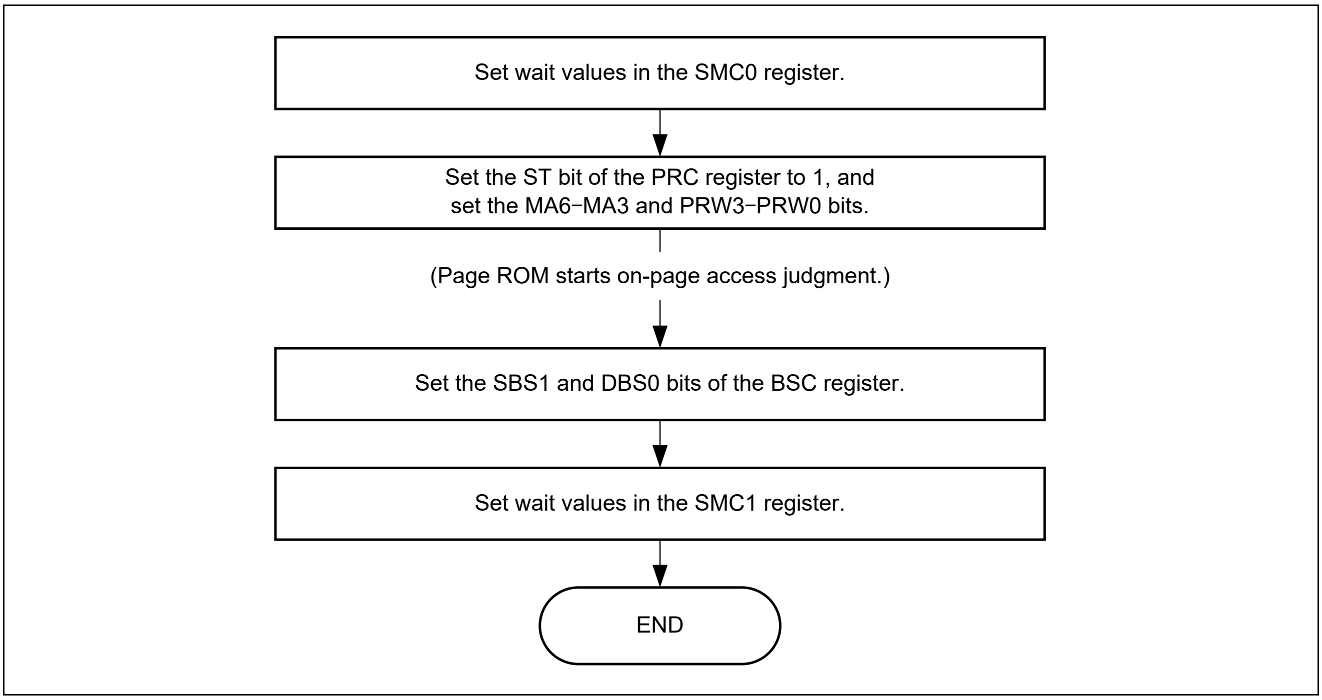


Figure 14.6 Example Procedure for Setting the Control Registers of the Memory Controller

14.6 External Wait Function

When a low-speed device or system is connected to the memory controller of an R-IN32M4-CL3, wait states can be inserted in bus cycles by using an external wait pin (WAITZ).

The WAITZ pin makes asynchronous input to the BUSCLK signal possible. The WAITZ signal is taken in through two flip-flop stages (on the falling and rising edges of BUSCLK). Therefore, sampling proceeds on the falling edge of BUSCLK and release from waiting is after 1.5 cycles of BUSCLK.

For this reason, to use an external wait function, at least 1 cycle of BUSCLK is required as the sum of the address setting wait (ACn3–ACn0 of an SMCn register) and data wait (DWn3–DWn0 of an SMCn register).

In an actual design, it is difficult to place WAITZ at the low level by the falling edge of the next cycle of BUSCLK from the time at which BCYSTZ and CSZn become active, so at least 2 cycles of BUSCLK are required as the sum of the address setting wait (ACn3–ACn0 of an SMCn register) and data wait (DWn3–DWn0 of an SMCn register).

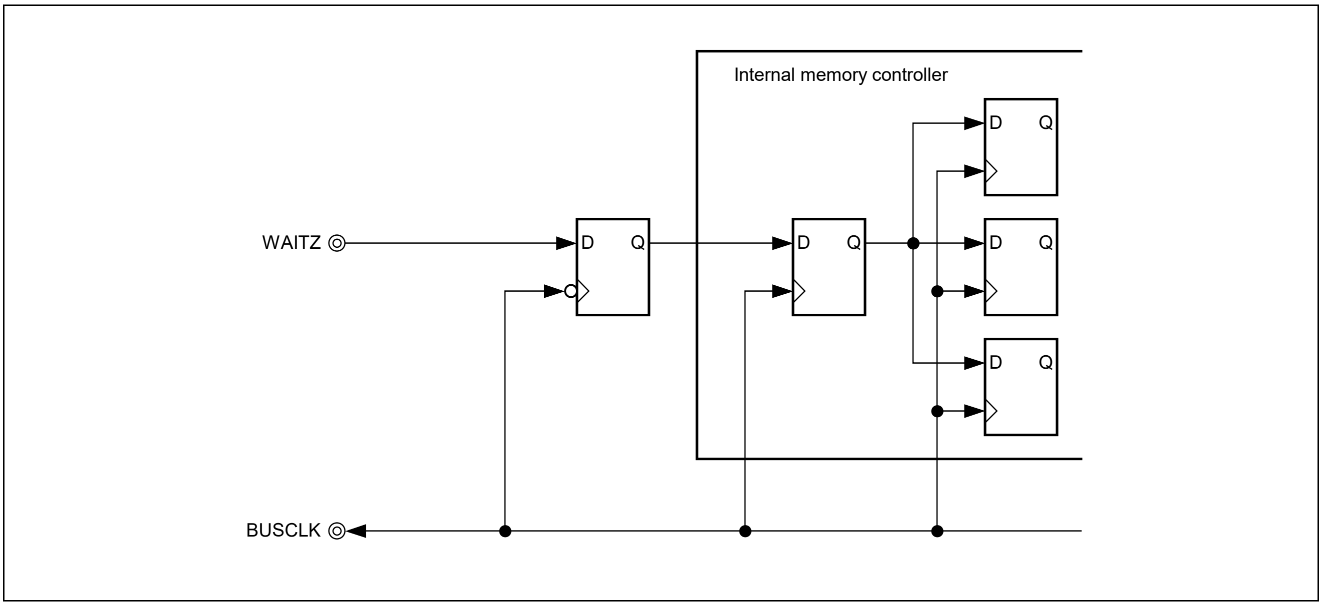


Figure 14.7 Configuration of the WAITZ Signal Sampling Circuit

14.7 Memory Access Timing Examples

The memory access timing examples are listed below.

Table 14.2 Memory Access Timing Examples

Figure Number	Memory Type	Access Condition	Page
Figure 14.8	SRAM	Reading without wait states	14-15
Figure 14.9	SRAM	Reading with wait states	14-16
Figure 14.10	SRAM	Reading with external wait insertion	14-17
Figure 14.11	SRAM	Writing without wait states	14-18
Figure 14.12	SRAM	Writing with wait states	14-19
Figure 14.13	SRAM	Writing with external wait insertion	14-20
Figure 14.14	Page ROM	Reading for a single transfer	14-21
Figure 14.15	Page ROM	Reading for a four-word burst transfer	14-22

BSC: SBS3–SBS0 = 1111B (32 bits), SMCn: IWn3–IWn0 = 0000B (1 wait cycle),
DWn3–DWn0 = 0000B (no wait), ACn3–ACn0 = 0000B (no wait)

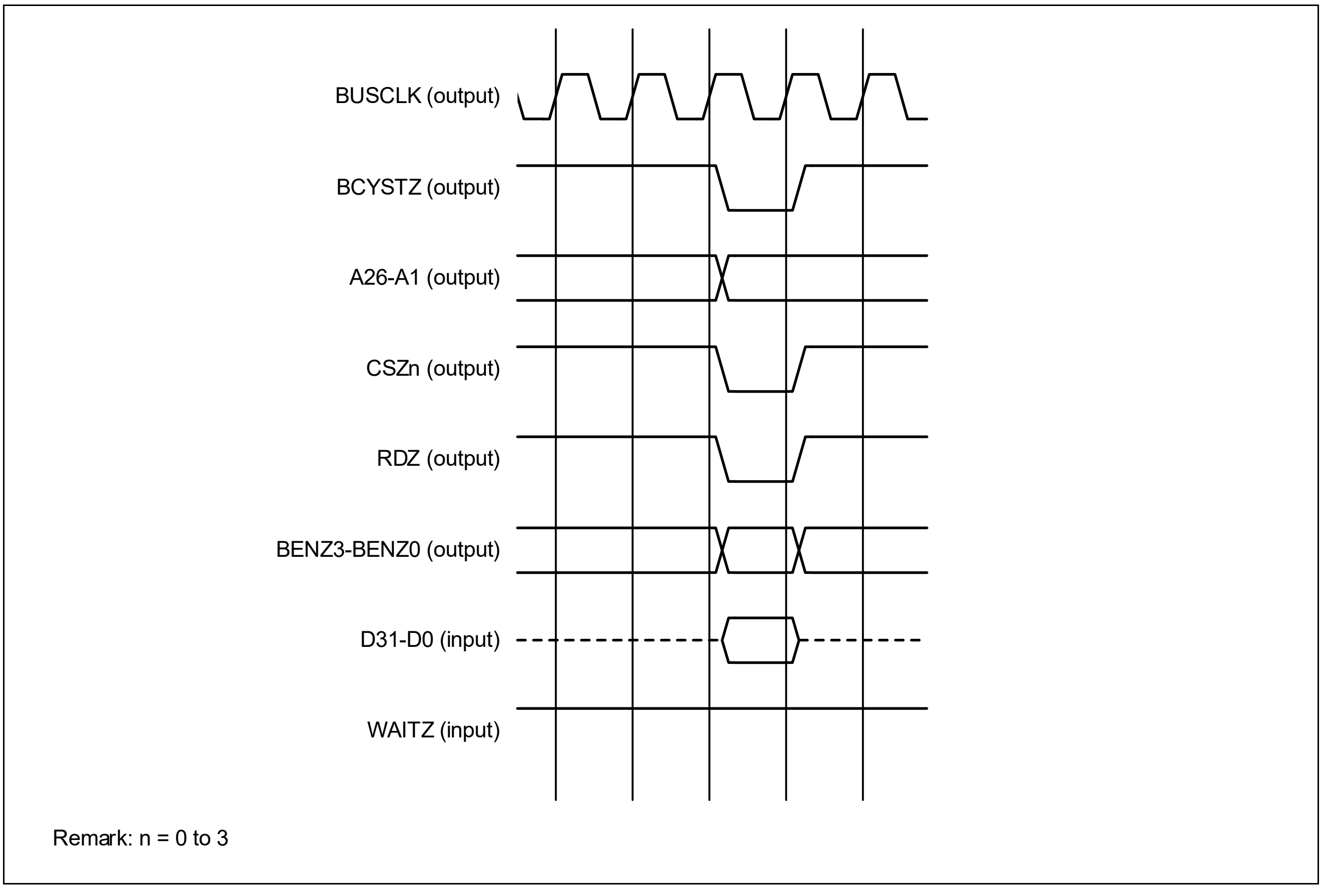


Figure 14.8 SRAM Read Cycles

BSC: SBS3–SBS0 = 1111B (32 bits), SMCn: IWn3–IWn0 = 0001B (2 wait cycles),
DWn3–DWn0 = 0001B (1 wait cycle), ACn3–ACn0 = 0001B (1 wait cycle)

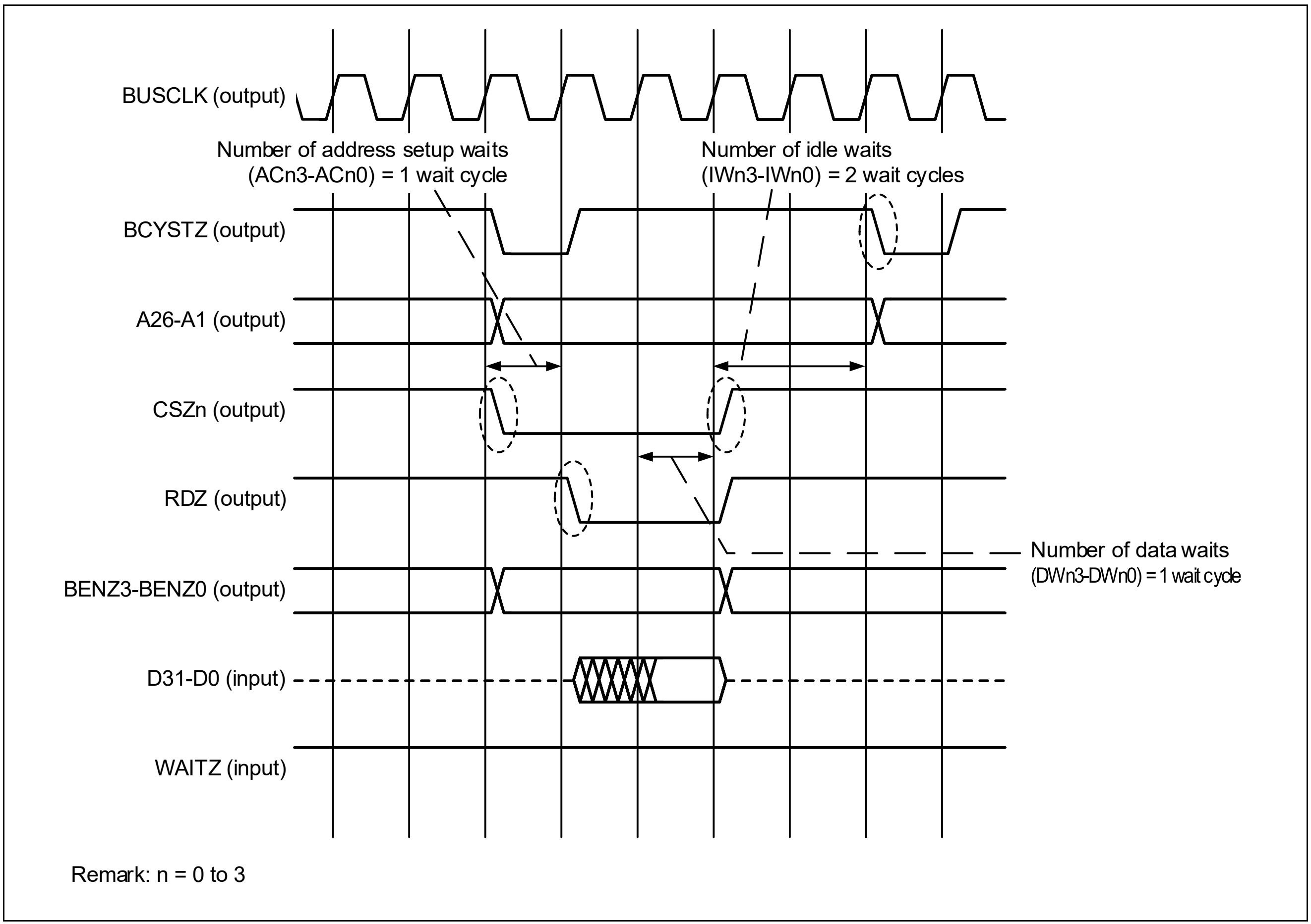
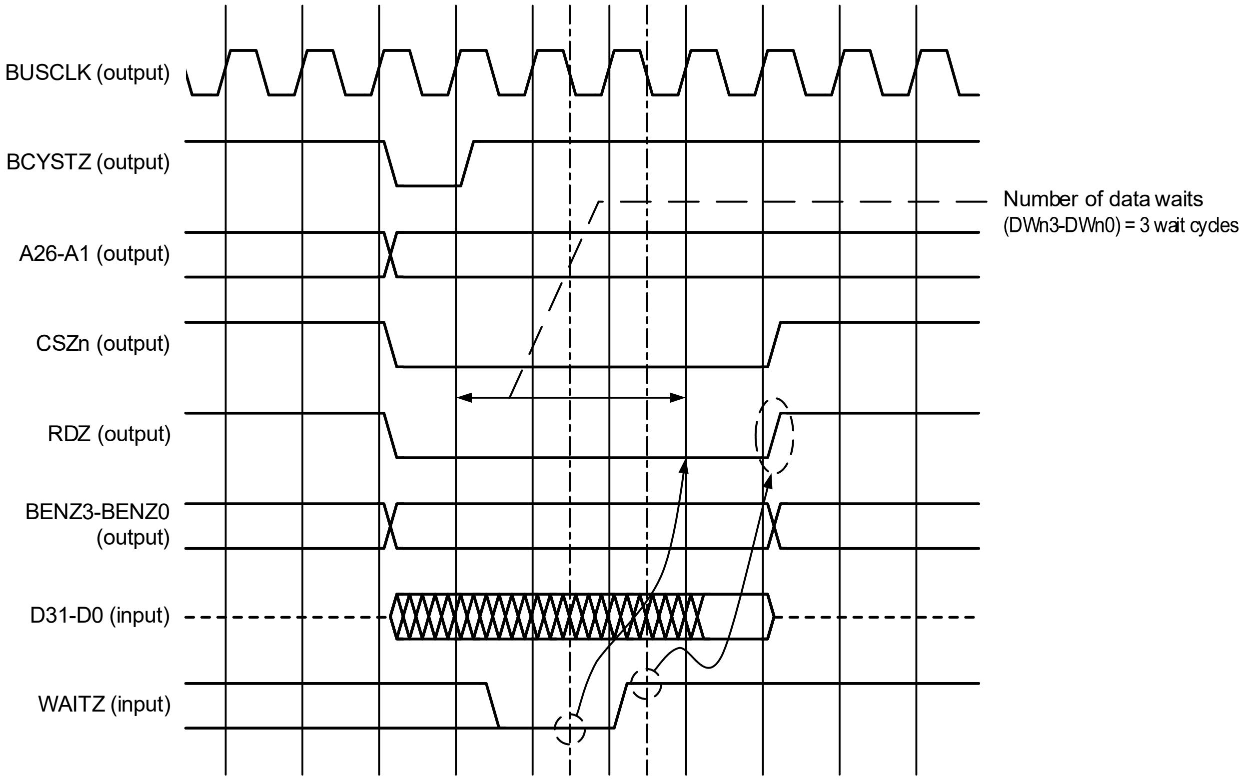


Figure 14.9 SRAM Read Cycles (with Wait Settings)

BSC: SBS3–SBS0 = 1111B (32 bits), SMCn: IWn3–IWn0 = 0000B (1 wait cycle),
DWn3–DWn0 = 0011B (3 wait cycles), ACn3–ACn0 = 0000B (no wait)



Note: To enable the external wait pin (WAITZ), the number of data waits (DWn3-DWn0) must to be set to at least 1. Setting of at least 2 wait cycles is recommended.

Remark: n = 0 to 3

Figure 14.10 SRAM Read Cycles (External Wait Insertion)

BSC: SBS3–SBS0 = 1111B (32 bits), SMCn: WWn3–WWn0 = 0000B/0001B (1 wait cycle),
DWn3–DWn0 = 0000B (no wait), ACn3–ACn0 = 0000B/0001B (1 wait cycle)

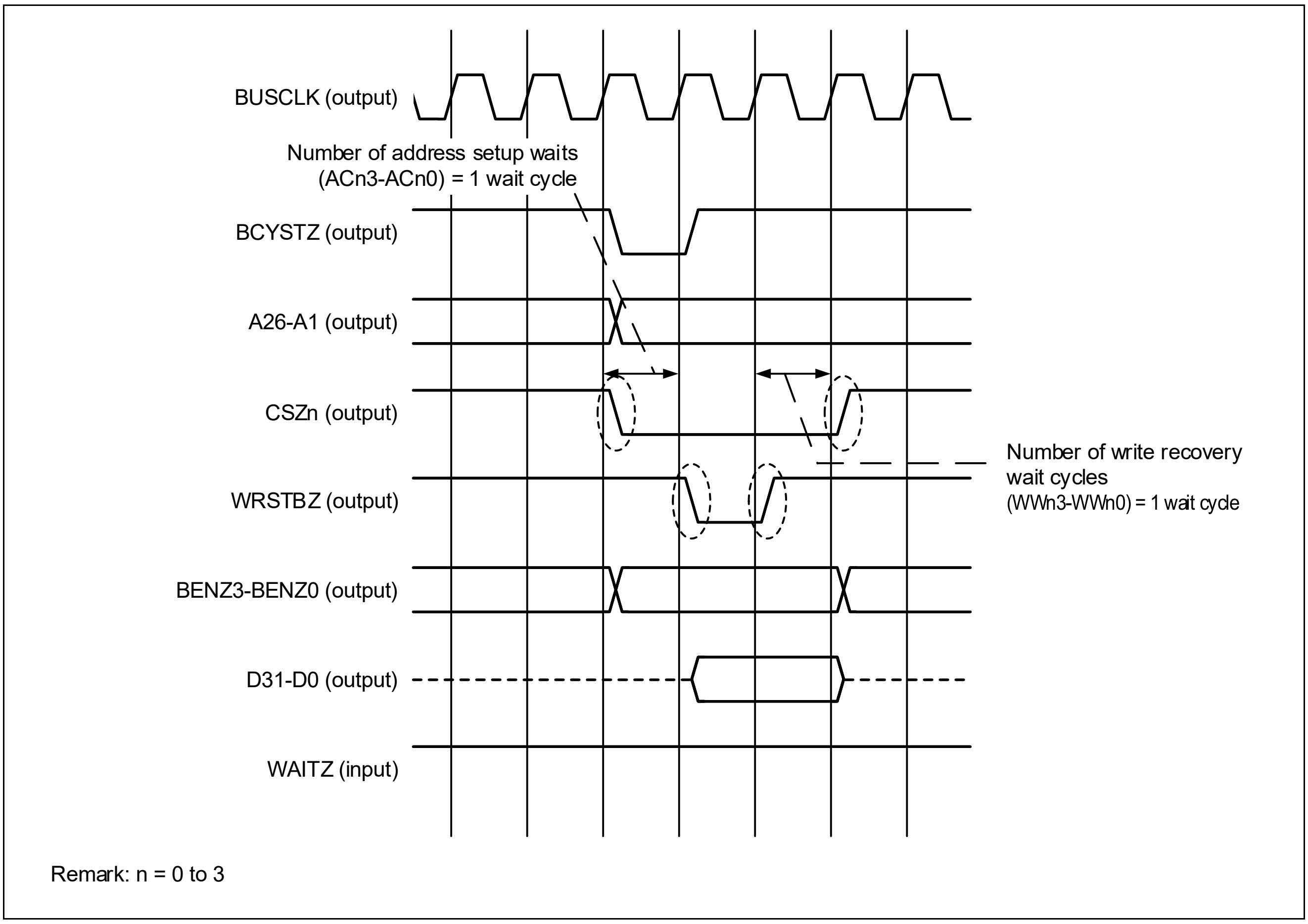


Figure 14.11 SRAM Write Cycles (with No Wait)

BSC: SBS3–SBS0 = 1111B (32 bits), SMCn: WWn3–WWn0 = 0010B (2 wait cycles),
DWn3–DWn0 = 0001B (1 wait cycle), ACn3–ACn0 = 0010B (2 wait cycles)

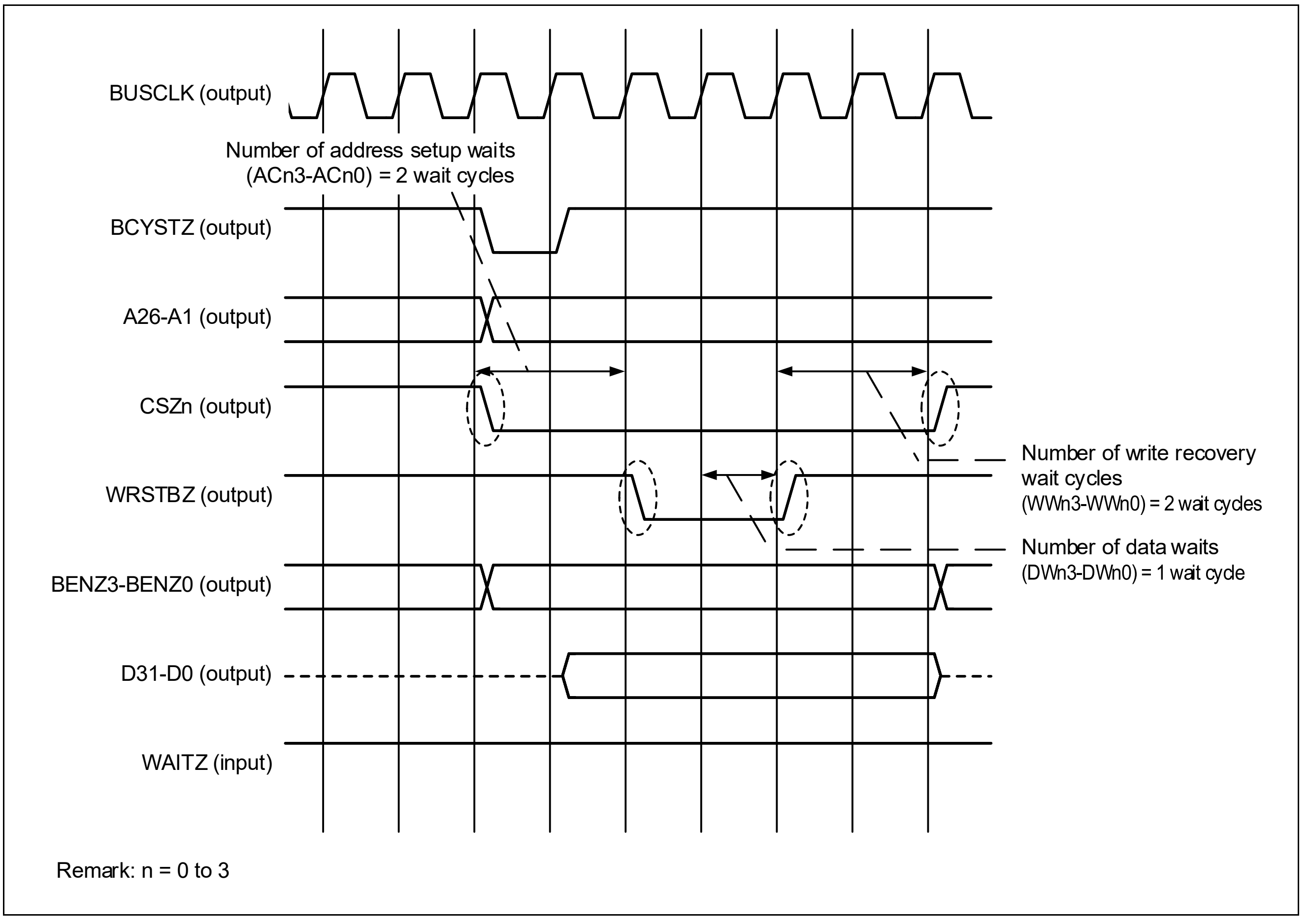
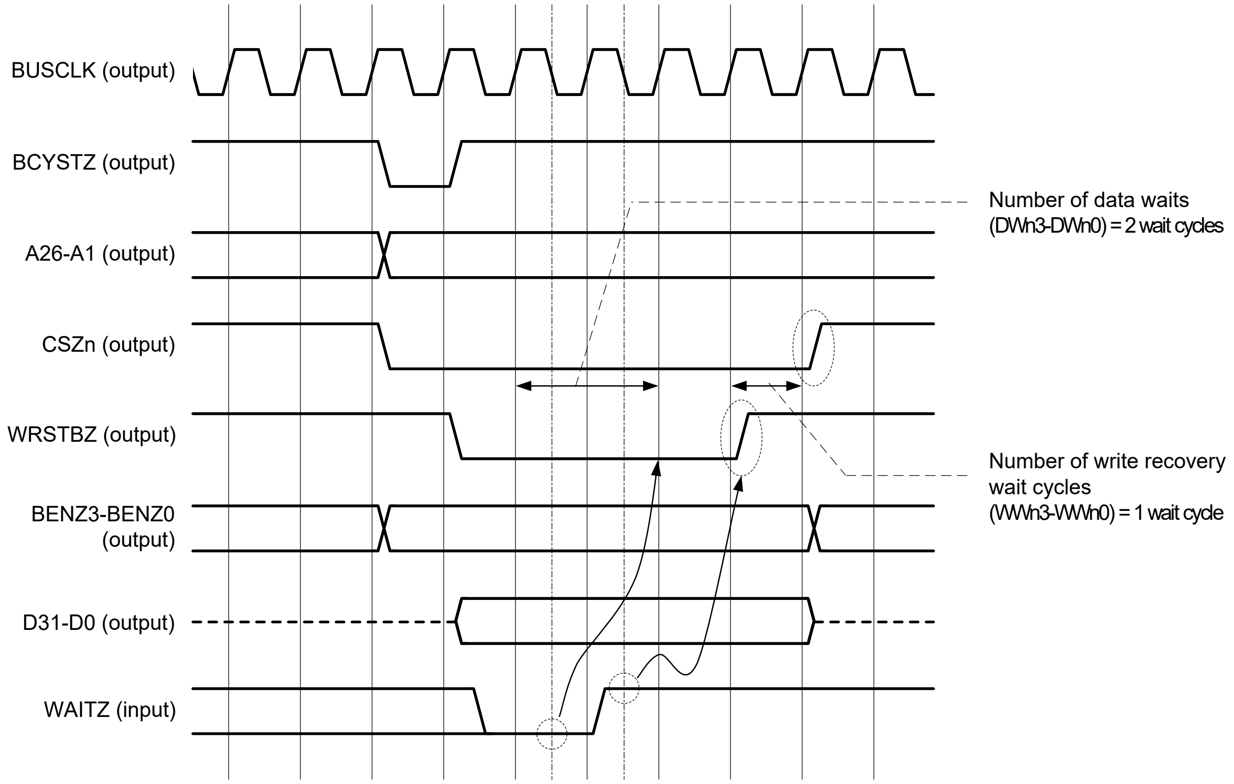


Figure 14.12 SRAM Write Cycles (with Wait States)

BSC: SBS3–SBS0 = 1111B (32 bits), SMCn: WWn3–WWn0 = 0000B/0001B (1 wait cycle),
DWn3–DWn0 = 0010B (2 wait cycles), ACn3–ACn0 = 0000B (no wait)



Note: To enable the external wait pin (WAITZ), the number of data waits (DWn3-DWn0) must to be set to at least 1. Setting of at least 2 wait cycles is recommended.

Remark: n = 0 to 3

Figure 14.13 SRAM Write Cycles (External Wait Insertion)

BSC: SBS3–SBS0 = 1111B (32 bits), SMC0: DW03–DW00 = 0001B (1 wait cycle)

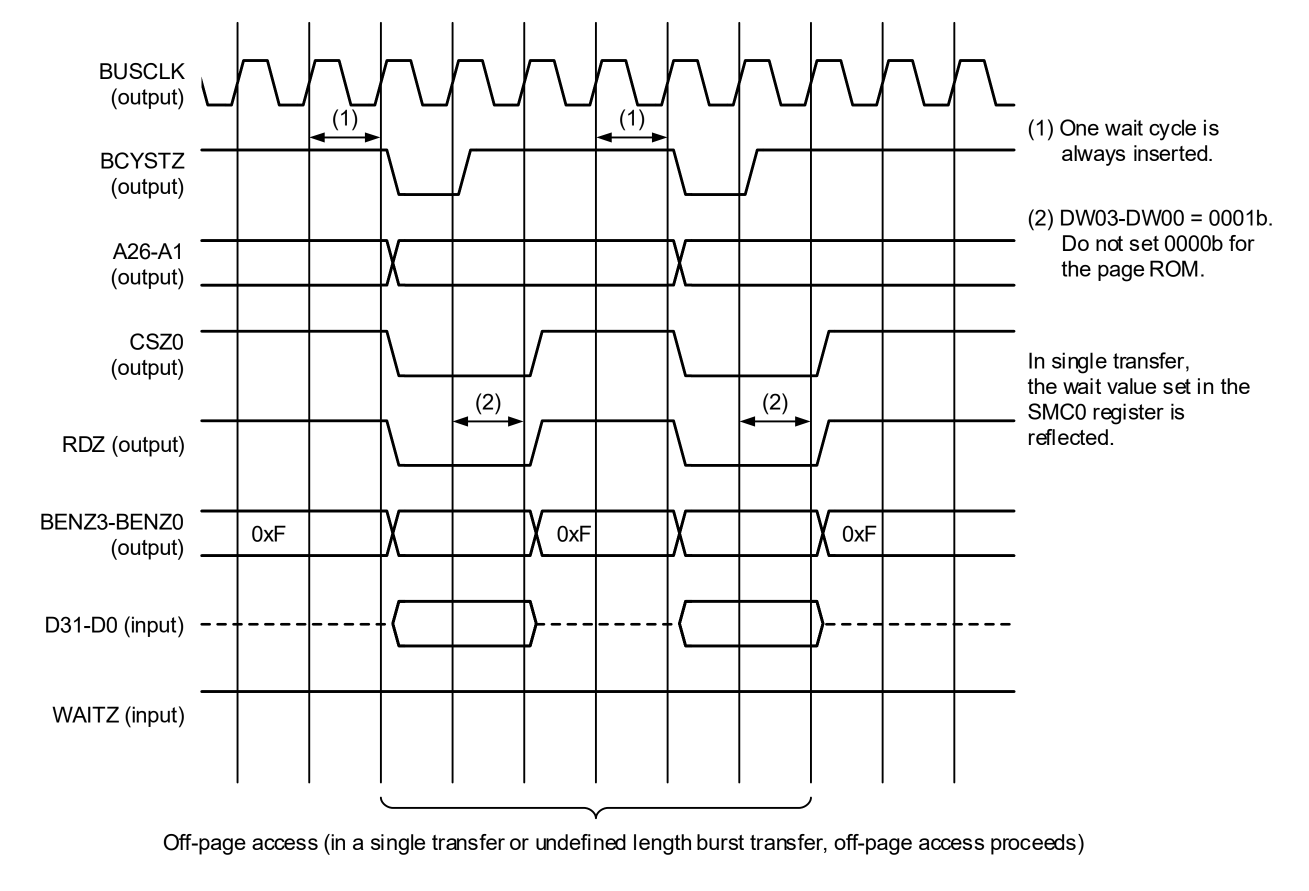


Figure 14.14 Page ROM Read Cycles (Single Transfer)

BSC: SBS3–SBS0 = 1111B (32 bits), SMC0 :IW03–IW00 = 0001B (2 wait cycles),
DW03–DW00 = 0001B (1 wait cycle), AC03–AC00 = 0001B (1 wait cycle),
PRC: PRW3–PRW0 = 0001B (1 wait cycle)

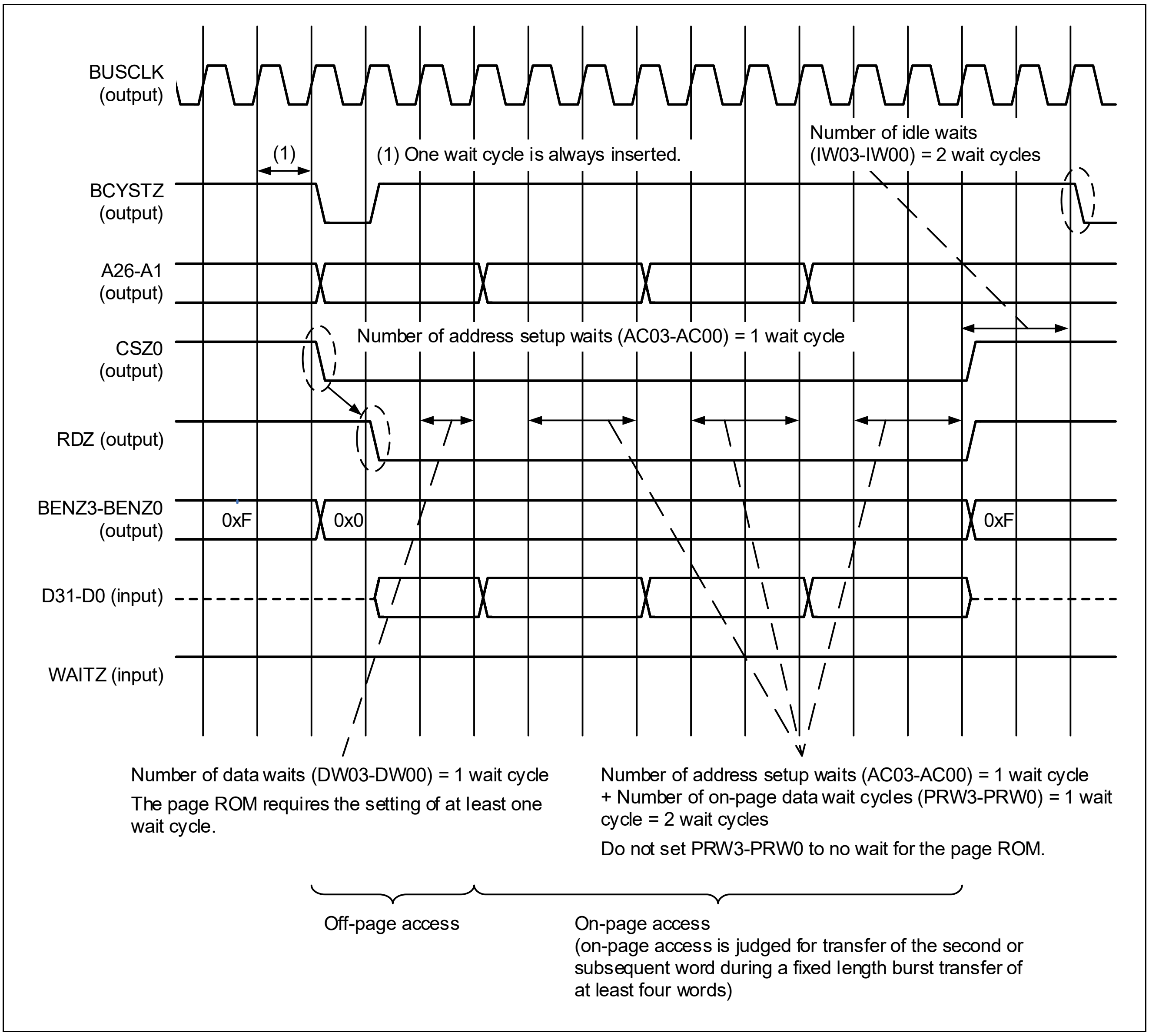


Figure 14.15 Page ROM Read Cycles (Four Burst Transfer)

14.8 Notes

1. In the 17 mm square package, the CSZ3 external memory area is not available.

15. Synchronous Burst Access Memory Controller

The synchronous burst access memory controller can be used to connect external paged ROM, ROM, SRAM, PSRAM, NOR-flash, and peripheral devices with an interface similar to the SRAM interface via the 32- or 16-bit bus.

Setting the ADMUXMODE pin to the high level selects multiplexing of the address signals with the data pins.

The synchronous burst access memory controller and asynchronous SRAM memory controller share pins with the external MCU interface. The synchronous burst access memory controller is selected when the MEMCSEL pin is at the high level and the MEMIFSEL pin is at the low level.

The CPU is booted from the memory connected to CSZ0 when both the BOOT0 and BOOT1 pins are at the low level.

Caution: Do not change the setting of the operating mode setting pins such as the MEMIFSEL and MEMCSEL pins during operation. Fix the setting before release from the reset state.

15.1 Features

- Memory controller for the page ROM, ROM, SRAM (synchronous, asynchronous), PSRAM, and NOR-Flash
- 32/16-bit data bus
- Address/data multiplexing feature

Remark: Page access is only possible for asynchronous access in separate bus mode.

- Static memory control
  - SRAM (synchronous, asynchronous), external I/O connection
  - Four chip select signals (CSZ0 to CSZ3) can be used.
    CSZ0: 1000_0000H to 13FF_FFFFH (64 MB)
    CSZ1: 1400_0000H to 17FF_FFFFH (64 MB)
    CSZ2: 1800_0000H to 1BFF_FFFFH (64 MB)
    CSZ3: 1C00_0000H to 1FFF_FFFFH (64 MB)
  - Programmable wait
  - Memory access frequency setting (1/2 to 1/6 the frequency of 100 MHz)
  - Up to four wait signals (WAITZ, WAITZ1 to WAITZ3) can be used.
  - Up to 16 bursts can be transferred.

Remark: Chip select areas can be assigned to the area between addresses 1000 0000H and 1FFF_FFFFH by using the SMADSEL register. (Specifiable in 16 MB units)

- Wait signal control
  - Up to four wait signals (WAITZ, WAITZ1 to 3) can be input.
  - The active level of the wait signal can be changed.
- BUSCLK signal masking
  - Output the BUSCLK signal only while the CSZ0 to CSZ3 signal is active.
- Write enable control
  - Keep the WRZ0 to WRZ3 signal active while the CSZ0 to CSZ3 signal is active.
- Control of data read timing: Read data and wait signals
  - Read data and the wait signals (WAITZ, WAITZ1 to WAITZ3) are fetched at the rising edge of BUSCLK.
  - Read data and the wait signals (WAITZ, WAITZ1 to WAITZ3) are fetched at the falling edge of BUSCLK.

15.2 Control Registers

When using the synchronous burst access memory controller, specify the operating mode by using the SMC operating mode setting register.

Caution: Access to these registers is prohibited when the synchronous burst access memory controller is not used.

Table 15.1 Synchronous Burst Access Memory Controller Control Registers

Register name	Symbol	Address
Wait signals select register	WAITZSEL	BASE + 0108H
Synchronous burst access memory controller area select register 0	SMADSEL0	BASE + 0110H
Synchronous burst access memory controller area select register 1	SMADSEL1	BASE + 0114H
Synchronous burst access memory controller area select register 2	SMADSEL2	BASE + 0118H
Synchronous burst access memory controller area select register 3	SMADSEL3	BASE + 011CH
Bus clock division setting register	BCLKSEL	BASE + 0120H
Synchronous burst access memory controller operation mode setting register	SMC352MD	BASE + 0124H
Synchronous burst access memory controller direct command register	DIRECT_CMD	400A 8010H
Synchronous burst access memory controller cycle setting register	SET_CYCLES	400A 8014H
Synchronous burst access memory controller mode setting register	SET_OPMODE	400A 8018H
Synchronous burst access memory controller refresh setting register	REF_PERIOD0	400A 8020H
Synchronous burst access memory controller CSZ0 cycle register	SRAM_CYCLES0_0	400A 8100H
Synchronous burst access memory controller CSZ0 mode register	OPMODE0_0	400A 8104H
Synchronous burst access memory controller CSZ1 cycle register	SRAM_CYCLES0_1	400A 8120H
Synchronous burst access memory controller CSZ1 mode register	OPMODE0_1	400A 8124H
Synchronous burst access memory controller CSZ2 cycle register	SRAM_CYCLES0_2	400A 8140H
Synchronous burst access memory controller CSZ2 mode register	OPMODE0_2	400A 8144H
Synchronous burst access memory controller CSZ3 cycle register	SRAM_CYCLES0_3	400A 8160H
Synchronous burst access memory controller CSZ3 mode register	OPMODE0_3	400A 8164H

15.2.1 Wait Signals Selection Register (WAITZSEL)

This register is used to enable or disable the signals input from the WAITZ pin and the WAITZ1 to WAITZ3 pins to the CSZ0 to CSZ3 areas.

Caution: This register is only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)".
No special sequence is required for reading the register.

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 27 to 16 to "0".

WAITZSEL — Address: BASE + 0108H, Initial value: 0000 000FH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WAITZSEL	ESWT3	ESWT2	ESWT1	ESWT0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAITZSEL	WSEL33	WSEL32	WSEL31	WSEL30	WSEL23	WSEL22	WSEL21	WSEL20	WSEL13	WSEL12	WSEL11	WSEL10	WSEL03	WSEL02	WSEL01	WSEL00
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Function
31 to 28	ESWT3 to ESWT0	Select the active level of the wait input signals (WAITZ, WAITZ1 to WAITZ3). 0: Active low 1: Active high
27 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 12	WSEL3n	Specify whether to enable the WAITZ3 input signal for each chip select area. 0000: Use the WAITZ3 pin as the WAIT pin xxx1: Enable input from the wait pin for access to the CSZ0 area. xx1x: Enable input from the wait pin for access to the CSZ1 area. x1xx: Enable input from the wait pin for access to the CSZ2 area. 1xxx: Enable input from the wait pin for access to the CSZ3 area.
11 to 8	WSEL2n	Specify whether to enable the WAITZ2 input signal for each chip select area. 0000: Use the WAITZ2 pin as the WAIT pin xxx1: Enable input from the wait pin for access to the CSZ0 area. xx1x: Enable input from the wait pin for access to the CSZ1 area. x1xx: Enable input from the wait pin for access to the CSZ2 area. 1xxx: Enable input from the wait pin for access to the CSZ3 area.
7 to 4	WSEL1n	Specify whether to enable the WAITZ1 input signal for each chip select area. 0000: Use the WAITZ1 pin as the WAIT pin xxx1: Enable input from the wait pin for access to the CSZ0 area. xx1x: Enable input from the wait pin for access to the CSZ1 area. x1xx: Enable input from the wait pin for access to the CSZ2 area. 1xxx: Enable input from the wait pin for access to the CSZ3 area.
3 to 0	WSEL0n	Specify whether to enable the WAITZ input signal for each chip select area. 0000: Use the WAITZ pin as the WAIT pin xxx1: Enable input from the wait pin for access to the CSZ0 area. xx1x: Enable input from the wait pin for access to the CSZ1 area. x1xx: Enable input from the wait pin for access to the CSZ2 area. 1xxx: Enable input from the wait pin for access to the CSZ3 area.

Remark: n = 0 to 3

15.2.2 Synchronous Burst Access Memory Controller Area Select Registers (SMADSEL0 to SMADSEL3)

These registers are used to specify the allocation of the CSZ0 to CSZ areas. Before changing the initial value, be sure to copy the program to an area other than the external memory area.

Caution: This register is only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

- Access These registers can be read or written in 32-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
SMADSEL0	0	0	0	1	SMCS0BASE3–0				0	0	0	0	0	0	0	0	BASE + 0110H
R/W	0	0	0	1	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	
	1				CSZ0_BASE_Address				0				0				

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
	0	0	0	0	0	0	0	0	1	1	1	1	SMCS0SIZE3–0				1000 00FCH
R/W	0	0	0	0	0	0	0	0	1	1	1	1	R/W	R/W	R/W	R/W	
	0				0				F				CSZ0_Size				

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
SMADSEL1	0	0	0	1	SMCS1BASE3–0				0	0	0	0	0	0	0	0	BASE + 0114H
R/W	0	0	0	1	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	
	1				CSZ1_BASE_Address				0				0				

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
	0	0	0	0	0	0	0	0	1	1	1	1	SMCS1SIZE3–0				1400 00FCH
R/W	0	0	0	0	0	0	0	0	1	1	1	1	R/W	R/W	R/W	R/W	
	0				0				F				CSZ1_Size				

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
SMADSEL2	0	0	0	1	SMCS2BASE3–0				0	0	0	0	0	0	0	0	BASE + 0118H
R/W	0	0	0	1	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	
	1				CSZ2_BASE_Address				0				0				

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
	0	0	0	0	0	0	0	0	1	1	1	1	SMCS2SIZE3–0				1800 00FCH
R/W	0	0	0	0	0	0	0	0	1	1	1	1	R/W	R/W	R/W	R/W	
	0				0				F				CSZ2_Size				

Caution: When setting these registers, only do so while the external memory area (1000 0000H to 1FFF FFFFH) is not accessed. Store programs in another area before running them.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
SMADSEL3	0	0	0	1	SMCS3BASE3–0				0	0	0	0	0	0	0	0	BASE + 011CH
R/W	0	0	0	1	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	
	1				CSZ3_BASE_Address				0				0				

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
	0	0	0	0	0	0	0	0	1	1	1	1	SMCS3SIZE3–0				1C00 00FCH
R/W	0	0	0	0	0	0	0	0	1	1	1	1	R/W	R/W	R/W	R/W	
	0				0				F				CSZ3_Size				

Bit Position	Bit Name	Function
27 to 24	SMCSnBASE3 to SMCSnBASE0	Specify the base address of the CSZn area used by the external memory interface.
3 to 0	SMCSnSIZE3 to SMCSnSIZE0	Specify the size of the CSZn area used by the external memory interface. 0000: 256 MB (Setting prohibited) 1000: 128 MB 1100: 64 MB 1110: 32 MB 1111: 16 MB Other than above: Setting prohibited

Cautions 1. The total size of all CSZn areas is 256 MB.

2. The specifiable address space is from 1000 0000H to 1FFF FFFFH.

3. The CSZn areas must not overlap. Specify base addresses and sizes such that the CSZ areas do not overlap.

4. When setting these registers, only do so while the external memory area (1000 0000H to 1FFF FFFFH) is not accessed. Store programs in another area before running them.

Remarks 1. Example of address area calculation

Base address ([31:24]) = access address [31:24] and size value [7:0]

If the CSZ1 area is allocated from addresses 1300 0000H to 13FF FFFFH

SMADSEL1: 1300_00FFH

If the CSZ1 area is allocated from addresses 1800 0000H to 1FFF FFFFH

SMADSEL1: 1800_00F8H

2. n = 0 to 3

15.2.3 Bus Clock Division Setting Register (BCLKSEL)

This register is used to frequency-divide the internal bus clock and BUSCLK pin (100 MHz) when the synchronous burst access memory controller is used. The division ratio ranges from divided by 2 to divided by 6.

Cautions 1. This register is only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)".
No special sequence is required for reading the register.

2. When setting this register, only do so while the external memory area (1000 0000H to 1FFF FFFFH) is not accessed. Store programs in another area before running them.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCLKSEL	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	BCLK2	BCLK1	BCLK0
R/W	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	R/W	R/W	R/W

Address: BASE + 0120H
Initial value: 0000 0004H

Bit Position	Bit Name	Function
31 to 3	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
2 to 0	BCLK2 to 0	Select the division ratio of the internal bus clock and BUSCLK pin (100 MHz). 000: Divided by 2 (Duty ratio: High 1, Low 1) 001: Divided by 3 (Duty ratio: High 1, Low 2) 010: Divided by 4 (Duty ratio: High 1, Low 1) 011: Divided by 5 (Duty ratio: High 2, Low 3) 100: Divided by 6 (Duty ratio: High 1, Low 1) (initial value) Other than above: Setting prohibited

15.2.4 Synchronous Burst Access Memory Controller Operation Mode Setting Register (SMC352MD)

This register is used to specify the operating mode of the synchronous burst access memory controller.

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

Cautions 1. This register is only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)".
No special sequence is required for reading the register.

2. When setting this register, only do so while the external memory area (1000 0000H to 1FFF FFFFH) is not accessed. Store programs in another area before running them.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMC352MD	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	MAGTD1	MAGTD0	SMCRDLTH	SMCWETH	SMCCLKTH
R/W	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	R/W	R/W	R/W	R/W	R/W

Address: BASE + 0124H
Initial value: 0000 0000H

Bit Position	Bit Name	Function
31 to 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4	MAGTD1	Fix the output from the MA16 to MA26 pins to low level. (Pins that function alternately as port pins output a low level only when used as port pins.) Note 1 0: Regular usage 1: Fix the output from the MA16 to MA26 pins to low level.
3	MAGTD0	Fix the output from the MA0 to MA15 pins to low level. (Pins that function alternately as port pins output a low level only when used as port pins.) Note 1 0: Regular usage 1: Fix the output from the MA0 to MA15 pins to low level.
2	SMCRDLTH	Select the SRAM read timing Note 2 0: SRAM data is latched at the rising edge of BUSCLK. 1: SRAM data is latched at the falling edge of BUSCLK.
1	SMCWETH	Select the SRAM WRZ0 to WRZ3 output mode. 0: SRAM WRZ0 to WRZ3 stays active during the period specified by the T_WP bit of the SET_CYCLES register. 1: After WRZ0 to WRZ3 is asserted, SRAM WRZ0 to WRZ3 stays active while the CSZ0 to CSZ3 signal is active.
0	SMCCLKTH	Select the SRAM BUSCLK output mode. 0: The internal clock signal of the synchronous burst access memory controller is output as is. 1: The clock signal is output only while the CSZ0 to CSZ3 signal is active. The timing examples in each mode are shown in 15.3.1(2), BUSCLK Masking.

Notes 1. This register becomes effective only when an ADMUXMODE terminal is high-level.

2. The setting of this register is valid only when synchronous access is being performed. When asynchronous access is being performed, the SRAM is read at the falling edge of the internal clock.

15.2.5 Synchronous Burst Access Memory Controller Direct Command Register (DIRECT_CMD)

This register is used to apply the values set in the synchronous burst access memory controller cycle setting register (SET_CYCLES) and synchronous burst access memory controller mode setting register (SET_OPMODE) to the synchronous burst access memory controller CSZn cycle register (SRAM_CYCLES0_n) and synchronous burst access memory controller CSZn mode register (OPMODE0_n) in each chip select area. By writing to this register, the values in these registers are applied to the corresponding registers in each chip select area.

- Access This register is only writable in 32-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECT_CMD	0	0	0	0	0	0	CHIP_NMBR			CMD_TYPE		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	W			W		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address 400A 8010H

Initial value —

Bit Position	Bit Name	Function
31 to 26, 20 to 0	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
25 to 23	CHIP_NMBR	Select the chip select area to which the register values are applied. 000: Apply values to the CSZ0 registers. 001: Apply values to the CSZ1 registers. 010: Apply values to the CSZ2 registers. 011: Apply values to the CSZ3 registers. 1xx: Setting prohibited
22, 21	CMD_TYPE	Specify the command type. 10: Register update Other than above: Setting prohibited

Remark: n = 0 to 3

15.2.6 Synchronous Burst Access Memory Controller Cycle Setting Register (SET_CYCLES)

This register is used to specify the clock cycles used for access to SRAM.

Specify values in this register and synchronous burst access memory controller mode setting register (SET_OPMODE), and then apply the values to each chip select area by using the synchronous burst access memory controller direct command register (DIRECT_CMD).

- Access This register is only writable in 32-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19 18 17	16 15 14	13 12 11	10 9 8	7 6 5 4	3 2 1 0	Address
SET_CYCLE	0	0	0	0	0	0	0	0	0	0	0	WE_TIME	T_TR	T_PC	T_WP	T_CEOE	T_WC	T_RC	400A 8014H Initial value —
R/W	0	0	0	0	0	0	0	0	0	0	0	W	W	W	W	W	W	W	

Bit Position	Bit Name	Function
31 to 21	—	Reserved. When writing to these bits, write 0.
20	WE_TIME	Specify when to assert the WRSTBZ signal. This setting is enabled when performing asynchronous access in multiplexed bus mode. 0: 2 cycles after the CSZ0 to CSZ3 signal is asserted. 1: The same time as the CSZ0 to CSZ3 signal is asserted.
19 to 17	T_TR	Specify the turnaround time inserted between SRAM access cycles. (tTR) 000: Setting prohibited 001: 1 clock cycle ... 111: 7 clock cycles The turnaround time is inserted when the following types of consecutive access are performed: - Read access -> Write access - Write access -> Read access - Read access -> Read access to another chip select area - The turnaround time is always inserted in multiplexed bus mode.
16 to 14	T_PC	Specify the page access time when reading a page. (tPC) Page access is enabled when performing asynchronous access in separate bus mode. 000: Setting prohibited 001: 1 clock cycle ... 111: 7 clock cycles
13 to 11	T_WP	Specify the time during which WRSTBZ is asserted. (tWP) 000: Setting prohibited 001: 1 clock cycle ... 111: 7 clock cycles If the SMCWETH bit of the SMC352MD register is 1, the WRSTBZ signal remains active while the CSZ0 to CSZ3 signal is active, regardless of the value set to the T_WP signal.

Bit Position	Bit Name	Function
10 to 8	T_CEOE	Specify the time from assertion of the CSZ0 to CSZ3 signal to assertion of the RDZ signal. (tCEOE[Note1]) 000: Setting prohibited 001: The RDZ signal is asserted 1 clock cycle after the CSZ0 to CSZ3 signal is asserted. ... 111: The RDZ signal is asserted 7 clock cycles after the CSZ0 to CSZ3 signal is asserted.
7 to 4	T_WC [Note 3]	Specify the time from assertion of the CSZ0 to CSZ3 signal to the start of writing. (tWC[Note 2]) 000x: Setting prohibited 0010: Writing starts 2 clock cycles after the CSZ0 to CSZ3 signal is asserted. ... 1111: Writing starts 15 clock cycles after the CSZ0 to CSZ3 signal is asserted. In single access, the value set in T_WC is the period where the CSZ0 to CSZ3 signal is asserted.
3 to 0	T_RC [Note 4]	Specify the time from assertion of the CSZ0 to CSZ3 signal to the start of reading. (tRC[Note 2]) 000x: Setting prohibited 0010: Reading starts 2 clock cycles after the CSZ0 to CSZ3 signal is asserted. ... 1111: Reading starts 15 clock cycles after the CSZ0 to CSZ3 signal is asserted. In single access, the value set in T_RC is the period where the CSZ0 to CSZ3 signal is asserted.

Notes 1. A setup in the following ranges is recommended for bus fight prevention at the time of multiplexer mode.

- Asynchronous access mode: Set up in the range from 011 to 111.

- Synchronous access mode: Set up in the range from 010 to 111.

2. Setting 2 clock cycles is prohibited in multiplexed bus mode.

Specify a setting from 0011 to 1111.

3. When a wait occurs, the write cycle is extended for a period during which the wait signal is asserted. For details, see Figure 15.23, Synchronous SRAM, Separate Bus Mode, Burst Write Access (4-beat), ADVZ Enabled.

4. When a wait occurs, the read cycle is extended for a period during which the wait signal is asserted. For details, see Figure 15.22, Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled.

15.2.7 Synchronous Burst Access Memory Controller Mode Setting Register (SET_OPMODE)

This register is used to specify the mode for access to SRAM.

Specify values in this register and synchronous burst access memory controller cycle setting register (SET_CYCLES), and then apply the values to each chip select area by using the synchronous burst access memory controller direct command register (DIRECT_CMD).

- Access This register is only writable in 32-bit units.

Address: 400A 8018H

Initial value: —

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 to 13	12	11	10	9 to 7	6	5 to 3	2	1 to 0
SET_OPMODE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BURST_ALIGN	BLS_TIME	ADV	0	WR_BL	WR_SYNC	RD_BL	RD_SYNC	MW
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W	W	W	0	W	W	W	W	W

Bit Position	Bit Name	Function
31 to 16	—	Reserved. When writing to these bits, write 0.
15 to 13	BURST_ALIGN	Specify the burst boundary. 000: No burst boundary 001: 32-data boundary 010: 64-data boundary 011: 128-data boundary 100: 256-data boundary Other than above: Setting prohibited
12	BLS_TIME	Specify when to assert the BENZ0 to BENZ3 signal. 0: The same time as the CSZ0 to CSZ3 signal is asserted. (Used as byte enable.) 1: The same time as the WRSTBZ signal is asserted. (Used as write byte enable.)
11	ADV	Specify whether to enable or disable the ADVZ pin. 0: Disabled (the ADVZ signal is fixed to high). 1: Enabled (the address becomes valid when the ADVZ signal is low level). The operation is as follows when the ADVZ pin is enabled: - The ADVZ signal remains active while the CSZ0 to CSZ3 signal is active during asynchronous access in separate bus mode. - Under any other conditions, the ADVZ signal remains active only for the first clock cycle.
10	—	Reserved. When writing to this bit, write 0.
9 to 7	WR_BL	Specify the burst length for write access. 000: Single access[Note] 001: Up to 4 data blocks 010: Up to 8 data blocks 011: Up to 16 data blocks Other than above: Setting prohibited

Note: Only single access can be specified when performing asynchronous access. Otherwise, setting is prohibited.

Bit Position	Bit Name	Function
6	WR_SYNC	Specify the access mode for write access. 0: Asynchronous access 1: Synchronous access The BUSCLK pin does not output a clock signal during asynchronous access.
5 to 3	RD_BL	Specify the burst length for read access. 000: Single access[Note] 001: Up to 4 data blocks 010: Up to 8 data blocks 011: Up to 16 data blocks Other than above: Setting prohibited
2	RD_SYNC	Specify the access mode for read access. 0: Asynchronous access 1: Synchronous access The BUSCLK pin does not output a clock signal during asynchronous access.
1, 0	MW	Specify the data bus width. When accessing the CSZ0 area, the BUS32EN pin determines the data bus width regardless of the setting in this field. 00: Setting prohibited 01: 16 bits 10: 32 bits 11: Setting prohibited

Note: Only single access can be specified when performing asynchronous access. Otherwise, setting is prohibited.

15.2.8 Synchronous Burst Access Memory Controller Refresh Setting Register (REF_PERIOD0)

This register is used to specify the number of times burst access can be executed consecutively.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
REF_PERIOD0	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	REF_PERIOD0				400A 8020H Initial value 0000 0000H
R/W	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	R/W				

Bit Position	Bit Name	Function
3 to 0	REF_PERIOD0	Specify the number of times burst access can be executed consecutively After burst transfer is performed the specified number of times, an idle cycle is inserted, and then the next burst transfer starts. The idle cycle specified by the T_TR bit of the SET_CYCLES register is inserted. 0000: No idle cycle is inserted. 0001: An idle cycle is inserted each time burst transfer is executed. 0010: An idle cycle is inserted after two consecutive burst transfers have been executed. ... 1111: An idle cycle is inserted after 15 consecutive burst transfers have been executed.

Caution: Set 0x0000_0001 in this register if the SMCWETH bit of the SMC352MD register is set to 1 enabling use of the address/data signal in separate bus mode.

15.2.9 Synchronous Burst Access Memory Controller CSZn Cycle Setting Registers (SRAM_CYCLES0_n)

These registers are used to reference the cycle settings specified for each chip select area.

The information set in the synchronous burst access memory controller cycle setting register (SET_CYCLES) can be read from each bit.

- Access This register is only readable in 32-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19–17	16–14	13–11	10–8	7–4	3–0	Address
SRAM_CYCLES0_n	0	0	0	0	0	0	0	0	0	0	0	WE_TIME	T_TR	T_PC	T_WP	T_CEOE	T_WC	T_RC	400A 8100H +20H×n Initial value 0002 B3CCH
R/W	0	0	0	0	0	0	0	0	0	0	0	R	R	R	R	R	R	R	

Remark: n = 0 to 3

15.2.10 Synchronous Burst Access Memory Controller CSZn Mode Registers (OPMODE0_0 to OPMODE0_3)

These registers are used to reference the operating mode settings specified for each chip select area.

The value set in the synchronous burst access memory controller mode setting register (SET_OPMODE) can be referenced by using the lower-order 16 bits of each register.

Access This register is only readable in 32-bit units.

Bits	31 to 24	23 to 16	15 to 13	12	11	10	9 to 7	6	5 to 3	2	1 to 0
OPMODE0_n	ADD_MATCH	ADD_MASK	BURST_ALIGN	BLS_TIME	ADV	0	WR_BL	WR_SYNC	RD_BL	RD_SYNC	MW
R/W	R	R	R	R	R	0	R	R	R	R	R

Address: 400A 8018H +20H×n

Initial value: —

Bit Position	Bit Name	Function
31 to 24	ADD_MATCH	The value specified as the base address of chip select areas can be read.
23 to 16	ADD_MASK	The value specified as the mask address of chip select areas can be read.
15 to 0	—	The value set in SET_OPMODE can be read.

Remark: n = 0 to 3

15.2.11 Register Setup Procedure

Be sure to set up the synchronous burst access memory controller setting registers during initialization by using the procedure shown below. These register settings cannot be changed dynamically during access to the external memory. Specify the register settings during initialization by using the program allocated to the internal instruction RAM.

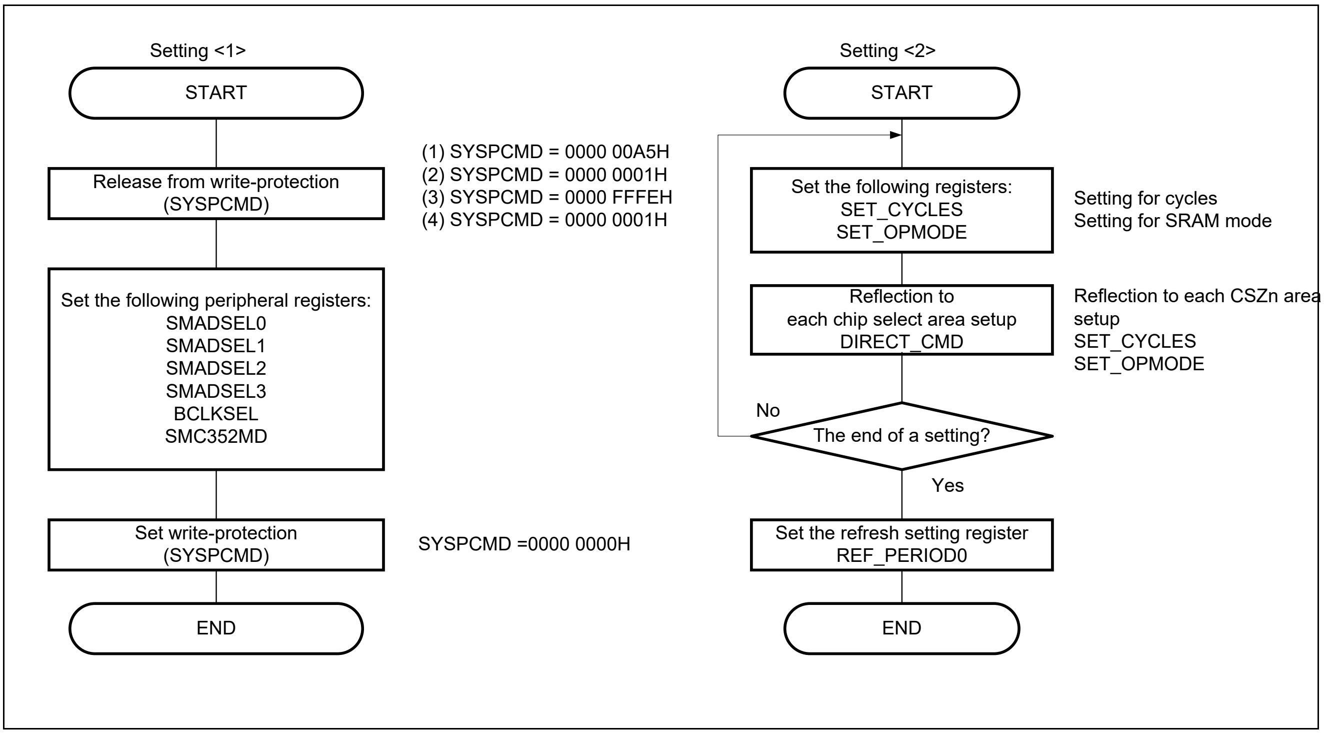


Figure 15.1 Register Setup Procedure

15.3 Function Details

15.3.1 Bus Clock Control Function

(1) BUSCLK Division

When using the synchronous burst access memory controller, the bus clock for the external memory interface (BUSCLK) can be used by dividing the system clock (100 MHz). By default, the system clock is divided by 6. A division factor of 2 to 6 can be selected. The bus clock is only output during synchronous SRAM access [Note].

- Division ratio: 1/2, 1/3, 1/4, 1/5, 1/6

Note: The bus clock is output for the CS active period + 1 cycle.

Remark: If the system clock is divided by 3, the duty ratio of the bus clock is 33.33% high. If the system clock is divided by 5, the duty ratio of the bus clock is 40% high. For other division factors, the duty ratio of the bus clock is 50%.

(2) BUSCLK Masking

The bus clock (BUSCLK) can be output for the period in which the CSZn signal is active, which is specified by the SMC352MD register.

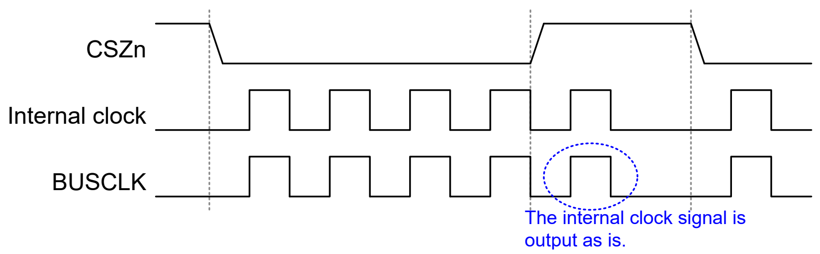


Figure 15.2 Clock Output Timing Example (SMC352MD.SMCCLKTH = 0)

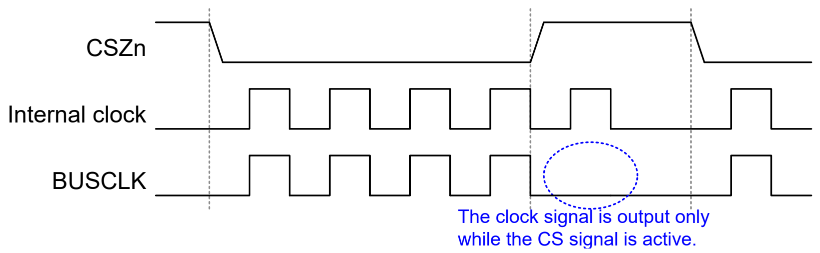


Figure 15.3 Clock Output Timing Example (SMC352MD.SMCCLKTH = 1)

Remark: n = 0 to 3

15.3.2 Address Output

The address signal output from the synchronous burst access memory controller to the external memory differs depending on the external bus width, however, the valid address signal is always output starting from the MA1 pin regardless of the bus width.

Bus Width	Address on Memory Map (256 MB Space)	Assignment of External Address Pins
32 bits	Address28 to Address2 bits	MA26 to MA0 pins
16 bits	Address27 to Address1 bits	MA26 to MA0 pins

15.3.3 Address/Data Multiplexing Feature

The address/data multiplexing feature enables address signals to be output from the data bus. By using this feature, the number of signal lines connected to the external memory can be reduced.
Use of the address/data multiplexing feature can be specified by using the ADMUXMODE pin.

External SRAM pins	In separate bus mode (ADMUXMODE = 0)		In multiplexed bus mode (ADMUXMODE = 1)		Remark
	16-bit bus mode (BUS32EN = 0)	32-bit bus mode (BUS32EN = 1)	16-bit bus mode (BUS32EN = 0)	32-bit bus mode (BUS32EN = 1)	
MA26 to MA0	Address27 to Address1	Address28 to Address2	Address27 to Address1	Address28 to Address2	The address signal is output regardless of the mode.
MD31 to MD16	—	Data31 to Data16	—	{5'h00, Address28 to Address2}, Data31 to Data0	For the address output timing in multiplexed bus mode, see "15.4, Memory Access Timing Example".Note
MD15 to MD0	Data15 to Data0	Data15 to Data0	Address16 to Address1, Data15 to Data0		

Note: Asynchronous access

Read: Figure 15.10, Asynchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Write: Figure 15.13, Asynchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled, WE_TIME = 0

Synchronous access

Read: Figure 15.16, Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Write: Figure 15.20, Synchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled

15.3.4 Write Enable Signal (WRZn) Cycle Extension

The write enable pin (WRZn) of the synchronous burst access memory controller is output only in the first cycle after the chip select signal (CSZn) is asserted when performing synchronous access. Some external peripheral devices cannot receive the write enable signal (WRZn) within one cycle. To solve this problem, the active period of the write enable signal (WRZn) can be extended while the chip select signal (CSZn) is active. To enable this feature, set the SMCWETH bit of the SMC352MD register to 1.

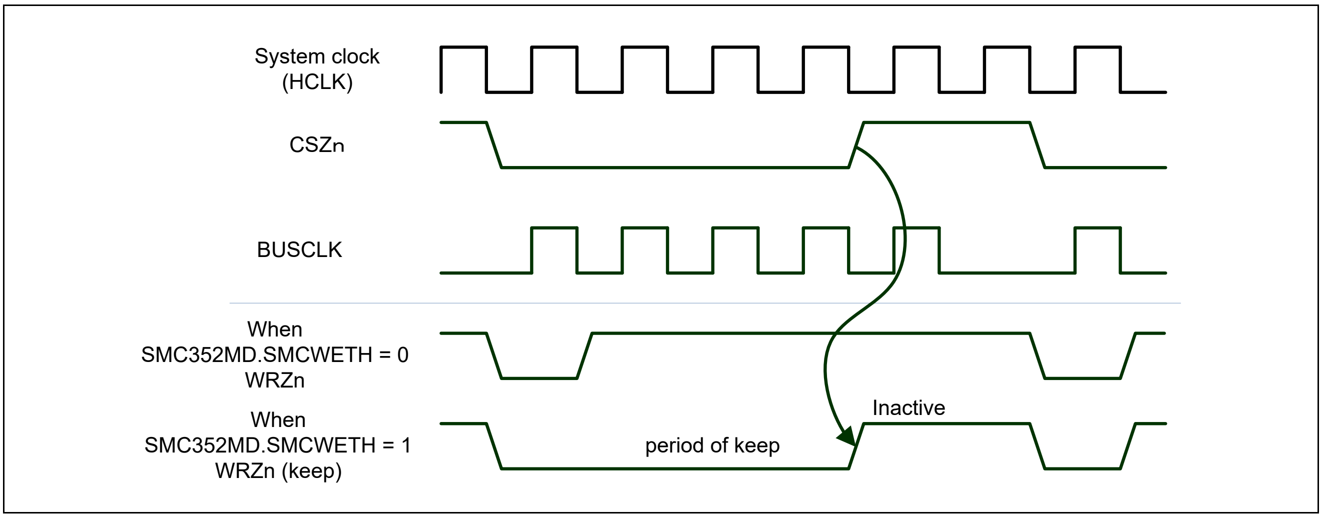


Figure 15.4 Write Enable Signal Operation

Remark: n = 0 to 3

15.3.5 Controlling the Data Read Timing

The timing at which to fetch read data during synchronous SRAM access can be adjusted. The rising or falling edge of BUSCLK output from an R-IN32M4-CL3 can be selected for this timing. If data is fetched at the rising edge of the clock, time for holding the data received from the external SRAM can be secured. If data is fetched at the falling edge of the clock, data setup time can be secured.

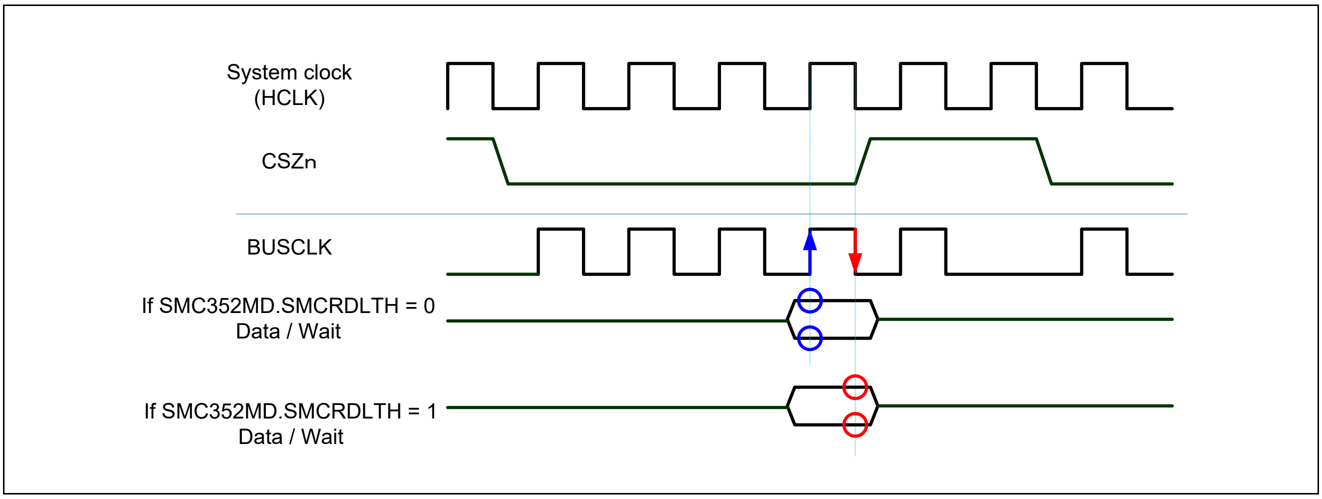


Figure 15.5 Read Data Timing Control

Remarks 1. n = 0 to 3

2. When operation is in asynchronous access mode, read data is always fetched at the falling edge of the system clock.

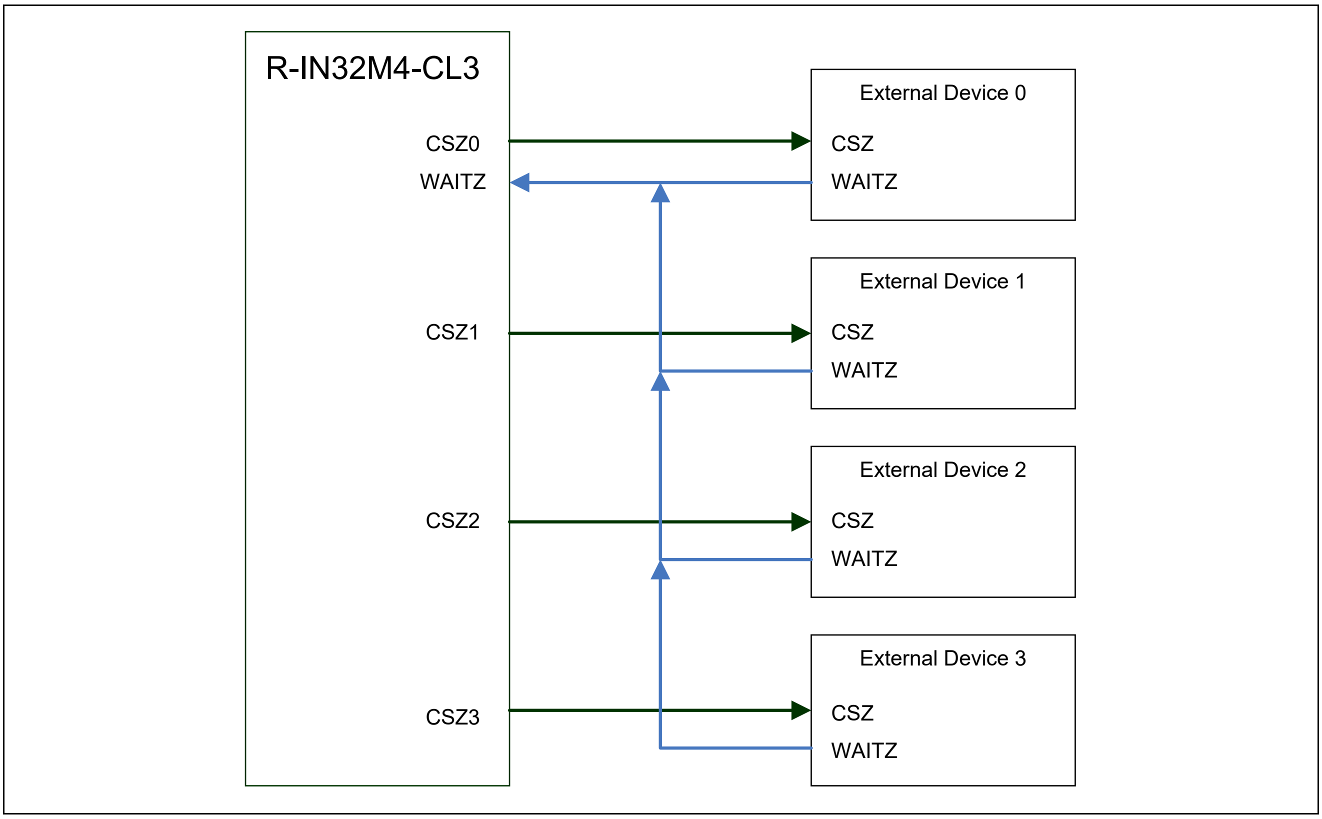
15.3.6 Wait Signal Control

The synchronous burst access memory controller can use up to four external wait input pins (WAITZ, WAITZ1 to WAITZ3) chip select areas. The WAITZSEL register is used to specify which external wait input pin is to be assigned to which chip select area. It is also possible to assign one wait pin to all four chip select areas.

For how to connect an R-IN32M4-CL3, the external devices, and external memory interface pins, refer to the R-IN32M4-CL3 User's Manual: Board Design.

(1) Connection example 1

Four external devices are connected. The wait signals are connected by using WAITZ via wired OR logic.



Remarks: The settings of the wait signals selection register are as follows.

WAITZSEL.WSEL0[3:0] = 1111B

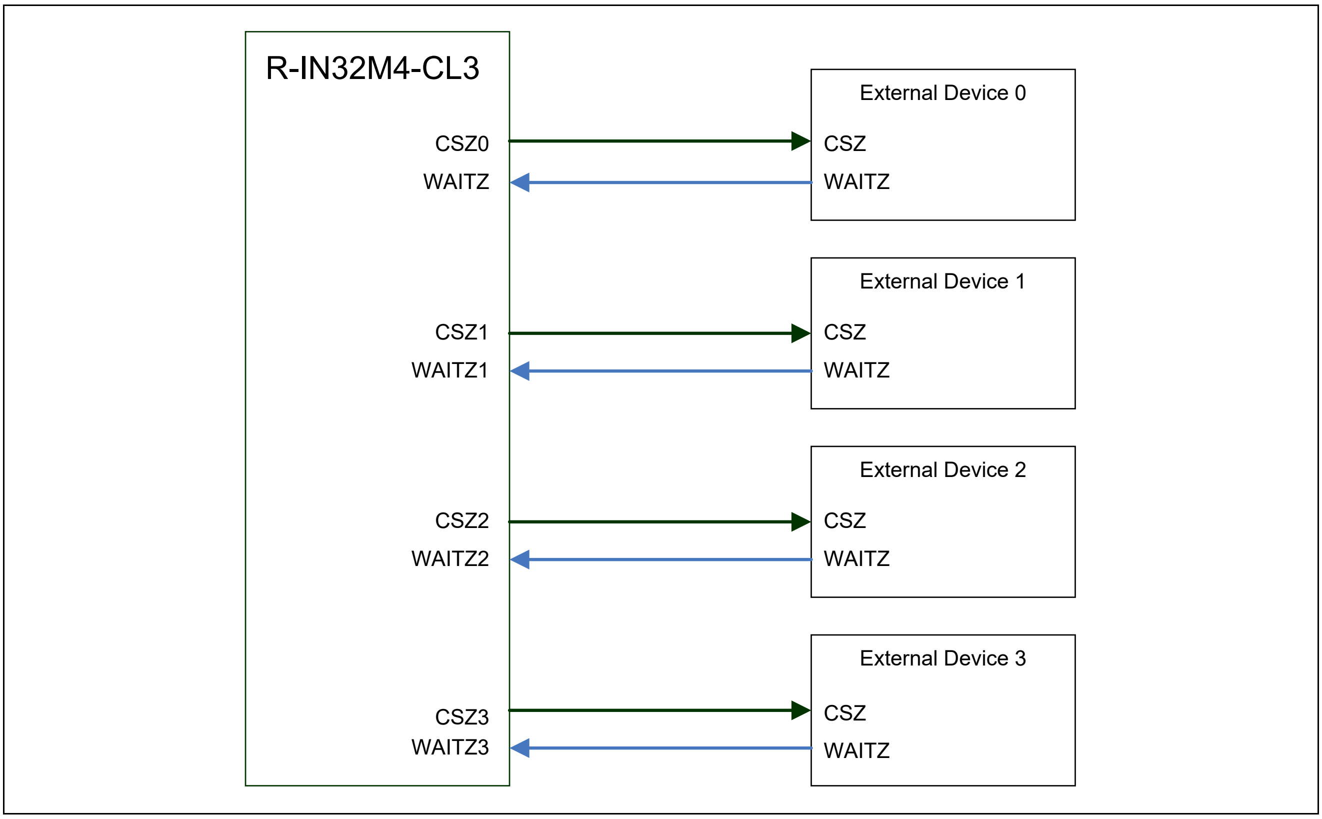
WAITZSEL.WSEL1[3:0] = 0000B

WAITZSEL.WSEL2[3:0] = 0000B

WAITZSEL.WSEL3[3:0] = 0000B

(2) Connection example 2

Four external devices are connected. The wait signals are connected individually.



Remarks: The settings of the wait signals selection register are as follows.

WAITZSEL.WSEL0[3:0] = 0001B

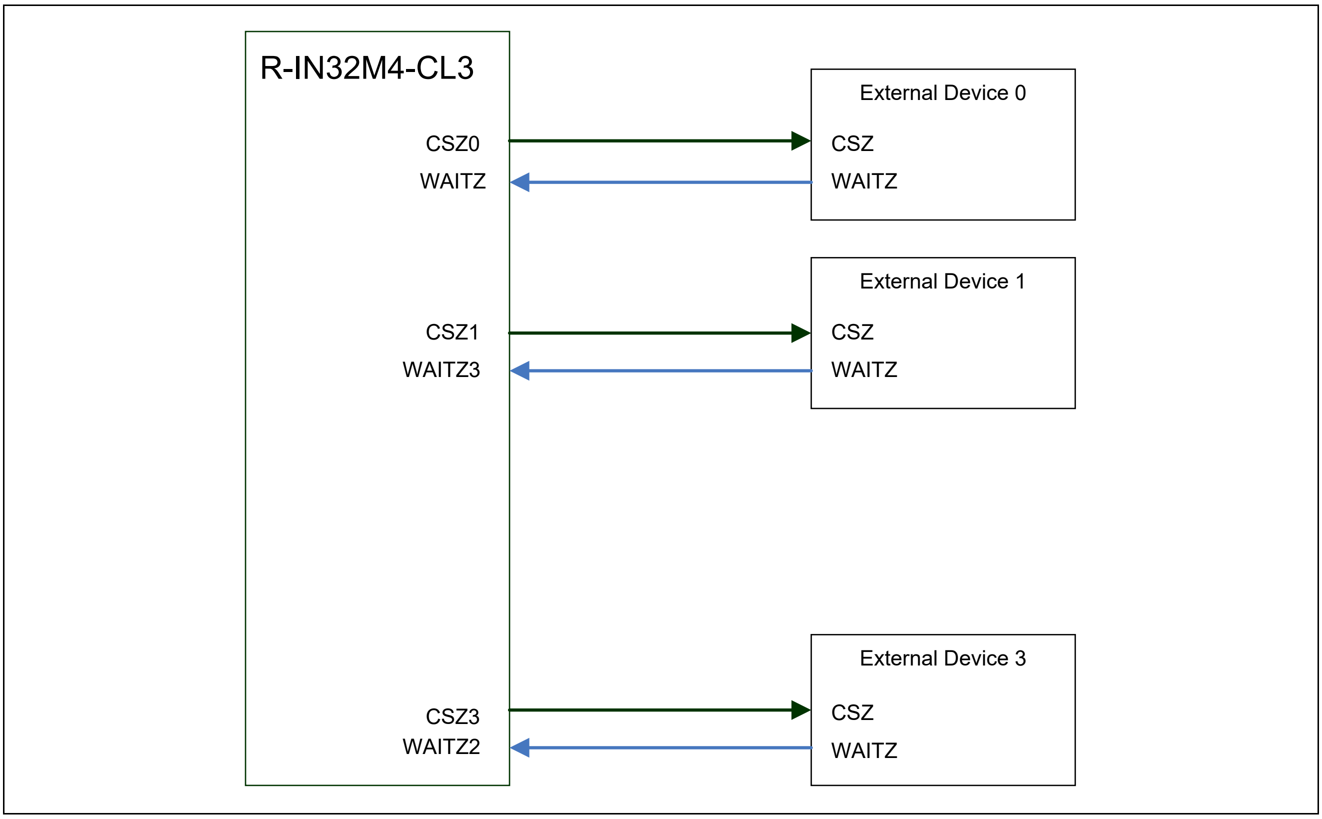
WAITZSEL.WSEL1[3:0] = 0010B

WAITZSEL.WSEL2[3:0] = 0100B

WAITZSEL.WSEL3[3:0] = 1000B

(3) Connection example 3

Three external devices are connected. The wait signals are connected individually. CSZ2 is not used. Assignment of the WAIT pins is changed.



Remark 1. The wait signals selection register (WAITZSEL) register can be used to select which interrupt corresponds to which chip select signal.

2. The settings of the wait signals selection register are as follows.

WAITZSEL.WSEL0[3:0] = 0001B

WAITZSEL.WSEL1[3:0] = 1000B

WAITZSEL.WSEL2[3:0] = 0000B

WAITZSEL.WSEL3[3:0] = 0100B

15.3.7 Specify the Operating Mode of the Synchronous Burst Access Memory Controller

Specify the operating mode for R-IN32M4-CL3 external pins MEMCSEL, ADMUXMODE, and BUS32EN.

Pin	Setting
MEMCSEL	Select whether to use the synchronous burst access memory controller or asynchronous SRAM memory controller. 0: Asynchronous SRAM memory controller 1: Synchronous burst access memory controller
ADMUXMODE	Select the bus mode for the address and data signals. 0: Separate bus mode 1: Multiplexed bus mode
BUS32EN	Specify the CSZ0 area bus width. 0: 16-bit bus 1: 32-bit bus

15.3.8 Switching External Memory Area Mapping

For the synchronous burst access memory controller, the address map and size of the chip select areas can be changed by using the SMADSEL0 to SMADSEL3 registers.

Cautions 1. The total size of all chip select areas is 256 MB.
2. The specifiable address space is from 1000 0000H to 1FFF FFFFH.
3. The chip select areas must not overlap. Specify base addresses and sizes such that the chip select areas do not overlap
4. When setting the registers, only do so while the external memory area (1000 0000H to 1FFF FFFFH) is not accessed. Store programs in another area before running them.

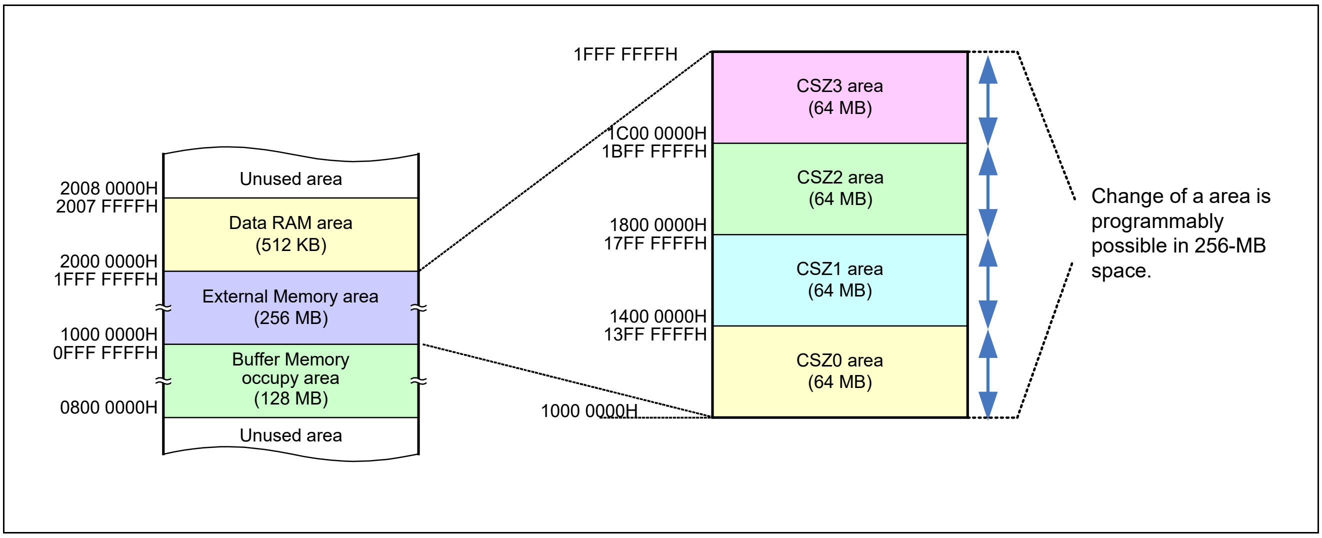


Figure 15.6 External Memory Space

15.4 Memory Access Timing Example

Memory access timing examples are shown below.

Table 15.2 Memory Access Timing Examples

Figure	Memory Type	Access Conditions	Page
Figure 15.7	Asynchronous SRAM	Read access, separate bus mode, ADVZ enabled	15-29
Figure 15.8	Asynchronous SRAM	Read access, separate bus mode, ADVZ disabled	15-30
Figure 15.9	Page ROM	Read access, separate bus mode, ADVZ enabled	15-31
Figure 15.10	Asynchronous SRAM	Read access, multiplexed bus mode, ADVZ enabled	15-32
Figure 15.11	Asynchronous SRAM	Write access, separate bus mode, ADVZ disabled	15-33
Figure 15.12	Asynchronous SRAM	Write access, separate bus mode, ADVZ enabled	15-34
Figure 15.13	Asynchronous SRAM	Write access, multiplexed bus mode, ADVZ enabled, WE_TIME = 0	15-35
Figure 15.14	Asynchronous SRAM	Write access, multiplexed bus mode, ADVZ enabled, WE_TIME = 1	15-36
Figure 15.15	Synchronous SRAM	Read access, separate bus mode, ADVZ enabled	15-37
Figure 15.16	Synchronous SRAM	Read access, multiplexed bus mode, ADVZ enabled	15-38
Figure 15.17	Synchronous SRAM	4-data burst read access, multiplexed bus mode, ADVZ enabled	15-39
Figure 15.18	Synchronous SRAM	Write access, separate bus mode, ADVZ enabled	15-40
Figure 15.19	Synchronous SRAM	8-data burst write access, separate bus mode, ADVZ enabled	15-41
Figure 15.20	Synchronous SRAM	Write access, multiplexed bus mode, ADVZ enabled	15-42
Figure 15.21	Synchronous SRAM	4-data burst write access, multiplexed bus mode, ADVZ enabled	15-43
Figure 15.22	Synchronous SRAM	Read, external wait timing	15-44
Figure 15.23	Synchronous SRAM	Write, external wait timing	15-45

15.4.1 Asynchronous Access Timing

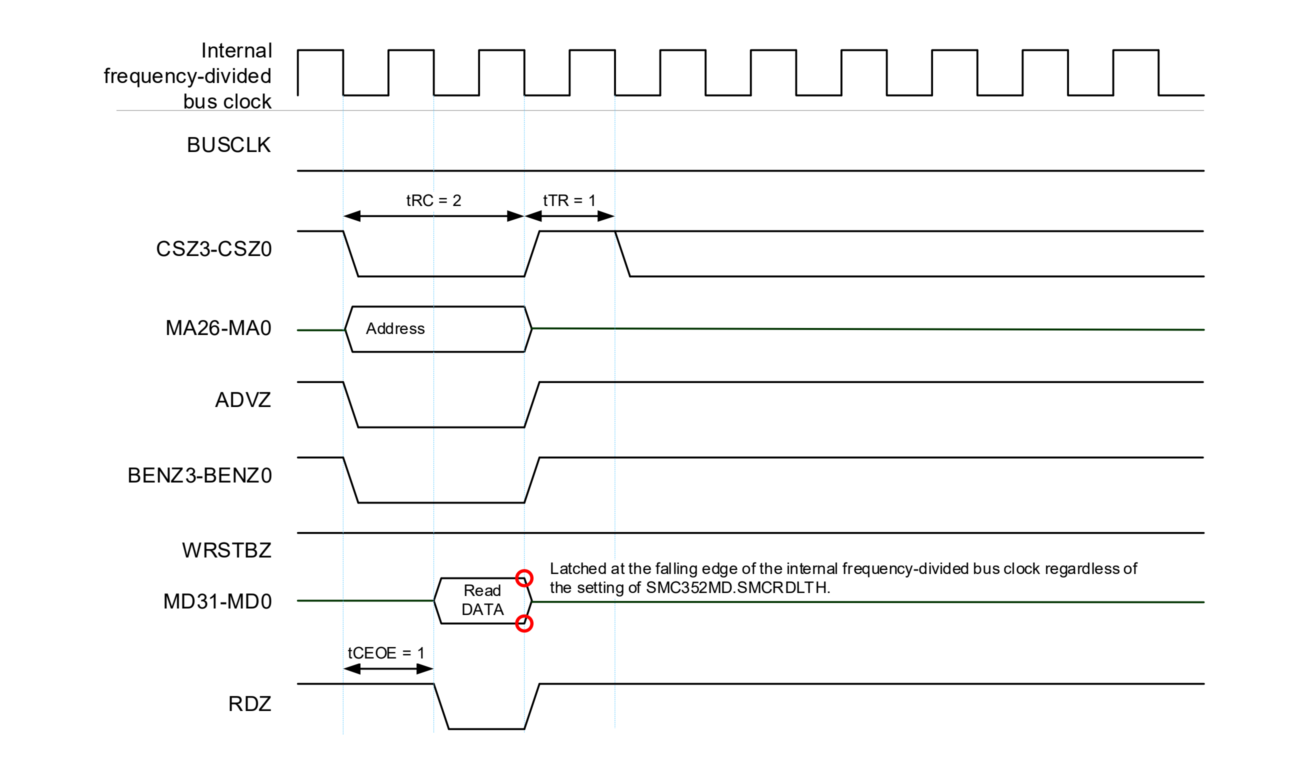


Figure 15.7 Asynchronous SRAM, Separate Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate mode)

SET_CYCLES.T_TR[2:0] = 001B (1 cycle)

T_CEOE[2:0] = 001B (1 cycle)

T_RC[3:0] = 0010B (2 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

RD_BL = 000B (single access)

RD_SYNC = 0B (asynchronous access)

MW[1:0] = 10B (bus width: 32 bits)

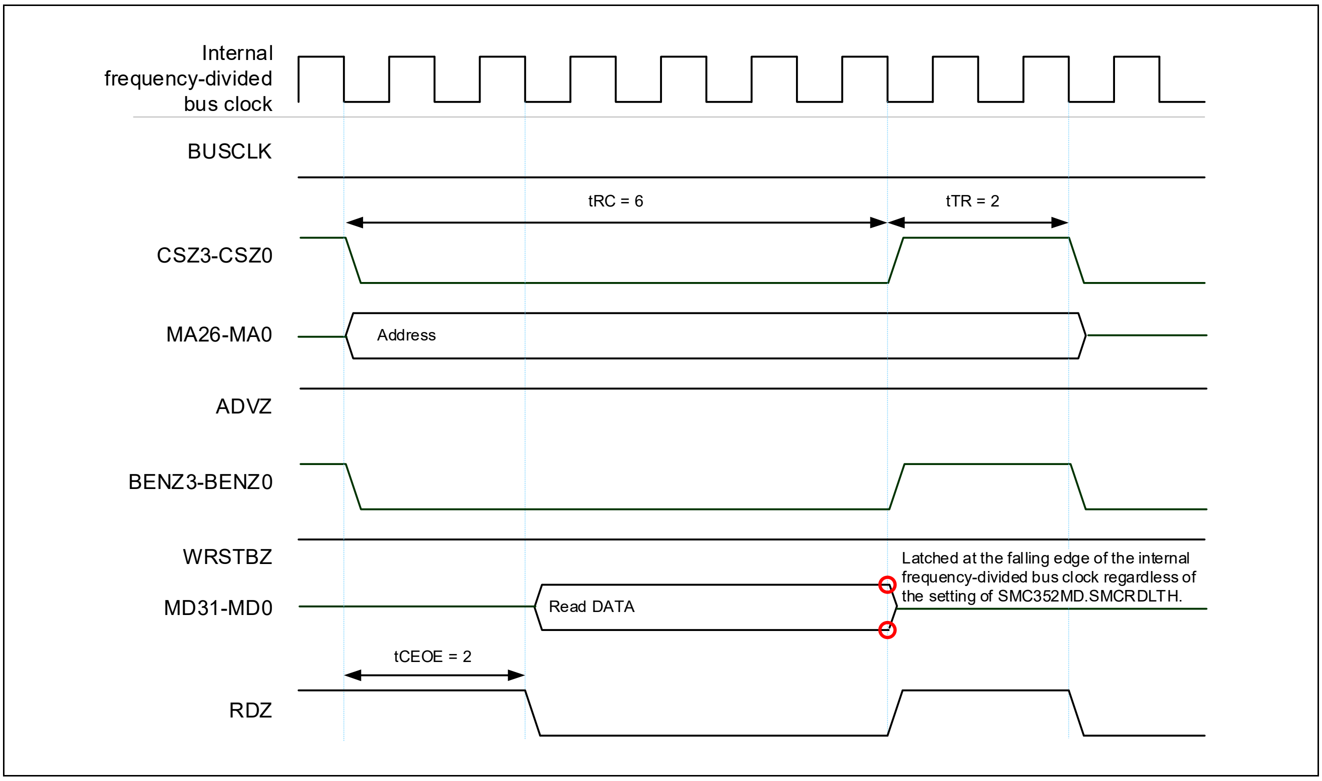


Figure 15.8 Asynchronous SRAM, Separate Bus Mode, Read Access, ADVZ Disabled

Remark: ADMUXMODE pin = Low level (separate mode)

SET_CYCLES.T_TR[2:0] = 010B (2 cycles)

T_CEOE[2:0] = 010B (2 cycles)

T_RC[3:0] = 0110B (6 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 0B (ADVZ disabled)

RD_BL = 000B (single access)

RD_SYNC = 0B (asynchronous access)

MW[1:0] = 10B (bus width: 32 bits)

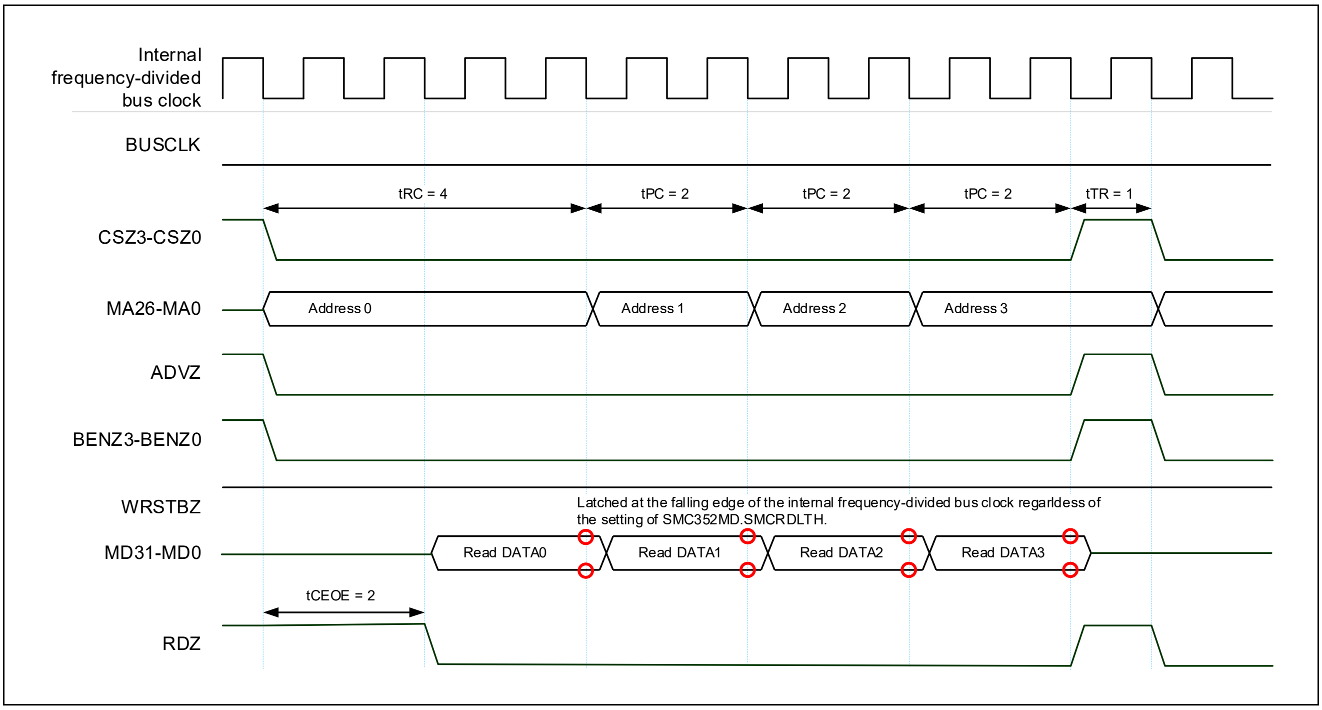


Figure 15.9 Asynchronous Page ROM, Separate Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate mode)

SET_CYCLES.T_TR[2:0] = 001B (1 cycle)

T_PC[2:0] = 010B (2 cycles)

T_CEOE[2:0] = 010B (2 cycles)

T_RC[3:0] = 0100B (4 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

RD_BL = 001B (up to 4 data blocks)

RD_SYNC = 0B (asynchronous access)

MW[1:0] = 10B (bus width: 32 bits)

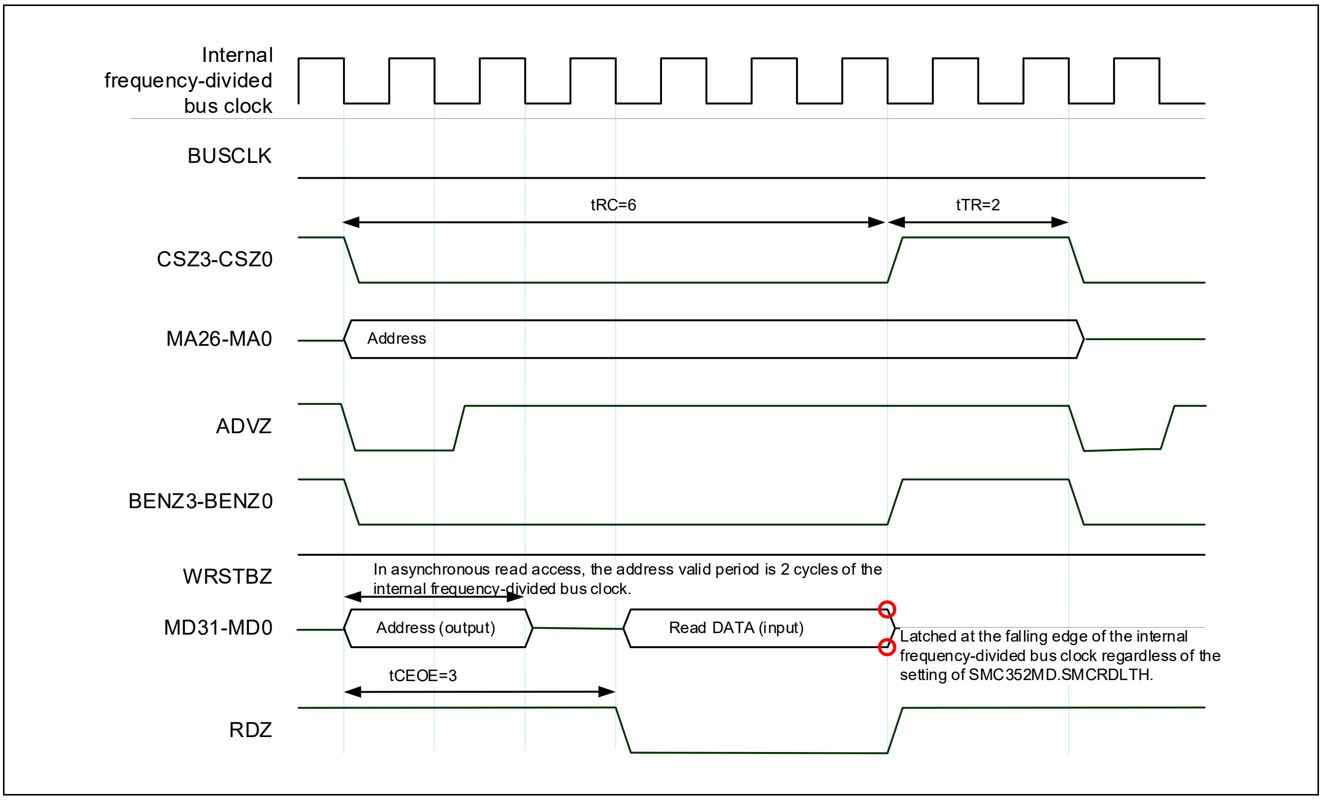


Figure 15.10 Asynchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed bus mode)

SET_CYCLES.T_TR[2:0] = 010B (2 cycles)

T_CEOE[2:0] = 011B (3 cycles)

T_RC[3:0] = 0110B (6 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

RD_BL = 000B (single access)

RD_SYNC = 0B (asynchronous access)

MW[1:0] = 10B (bus width: 32 bits)

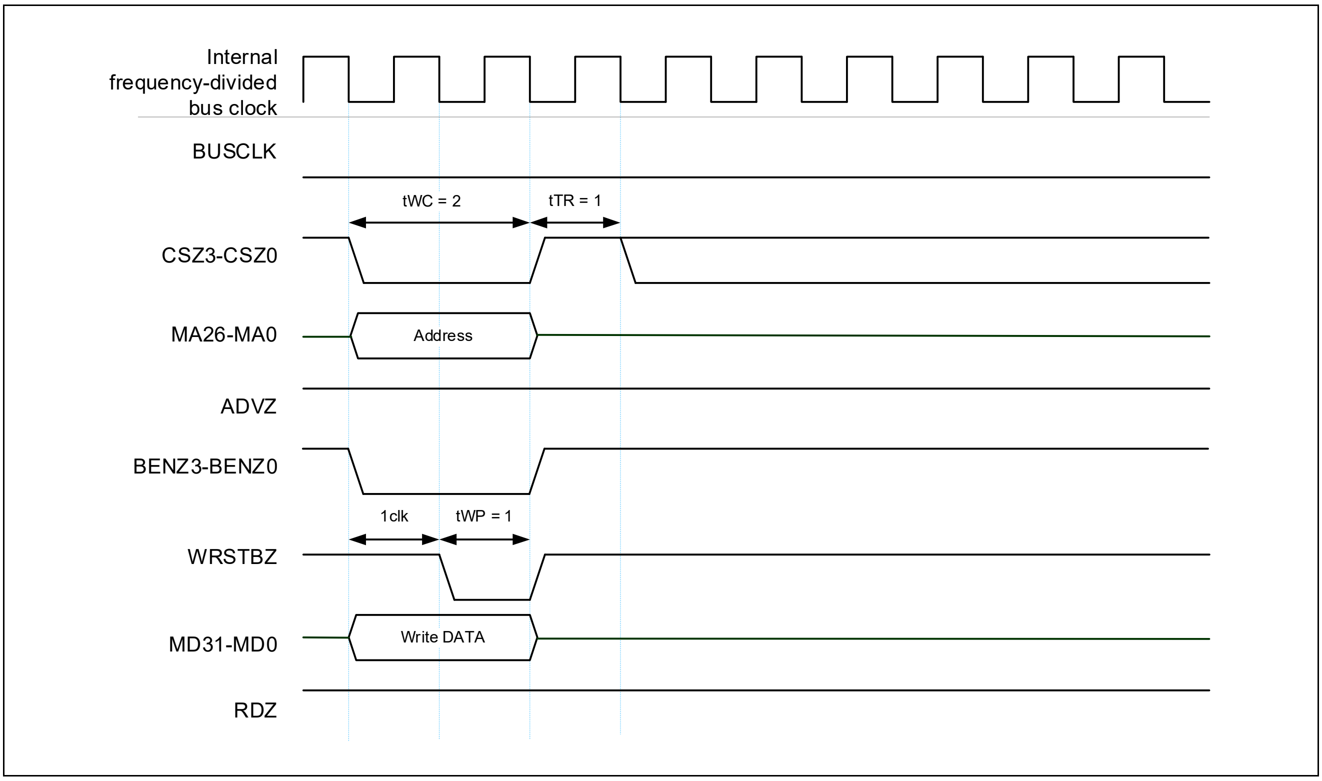


Figure 15.11 Asynchronous SRAM, Separate Bus Mode, Write Access, ADVZ Disabled

Remark: ADMUXMODE pin = Low level (separate mode)

SET_CYCLES.T_TR[2:0] = 001B (1 cycle)

T_WP[2:0] = 001B (1 cycle)

T_WC[3:0] = 0010B (2 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 0B (ADVZ disabled)

WR_BL = 000B (single access)

WR_SYNC = 0B (asynchronous access)

MW[1:0] = 10B (bus width: 32 bits)

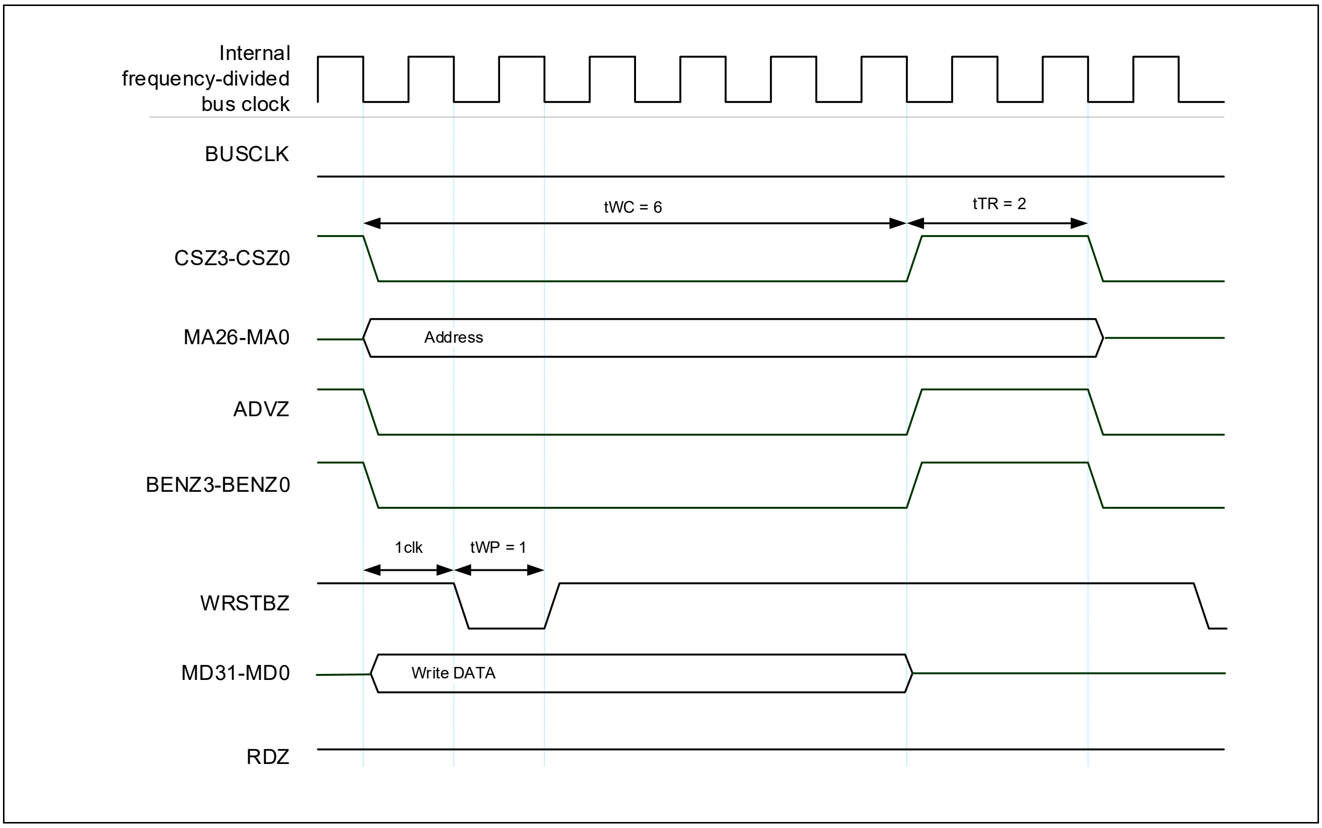


Figure 15.12 Asynchronous SRAM, Separate Bus Mode, Write Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate mode)

SET_CYCLES.T_TR[2:0] = 010B (2 cycles)

T_WP[2:0] = 001B (1 cycle)

T_WC[3:0] = 0110B (6 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

WR_BL = 000B (single access)

WR_SYNC = 0B (asynchronous access)

MW[1:0] = 10B (bus width: 32 bits)

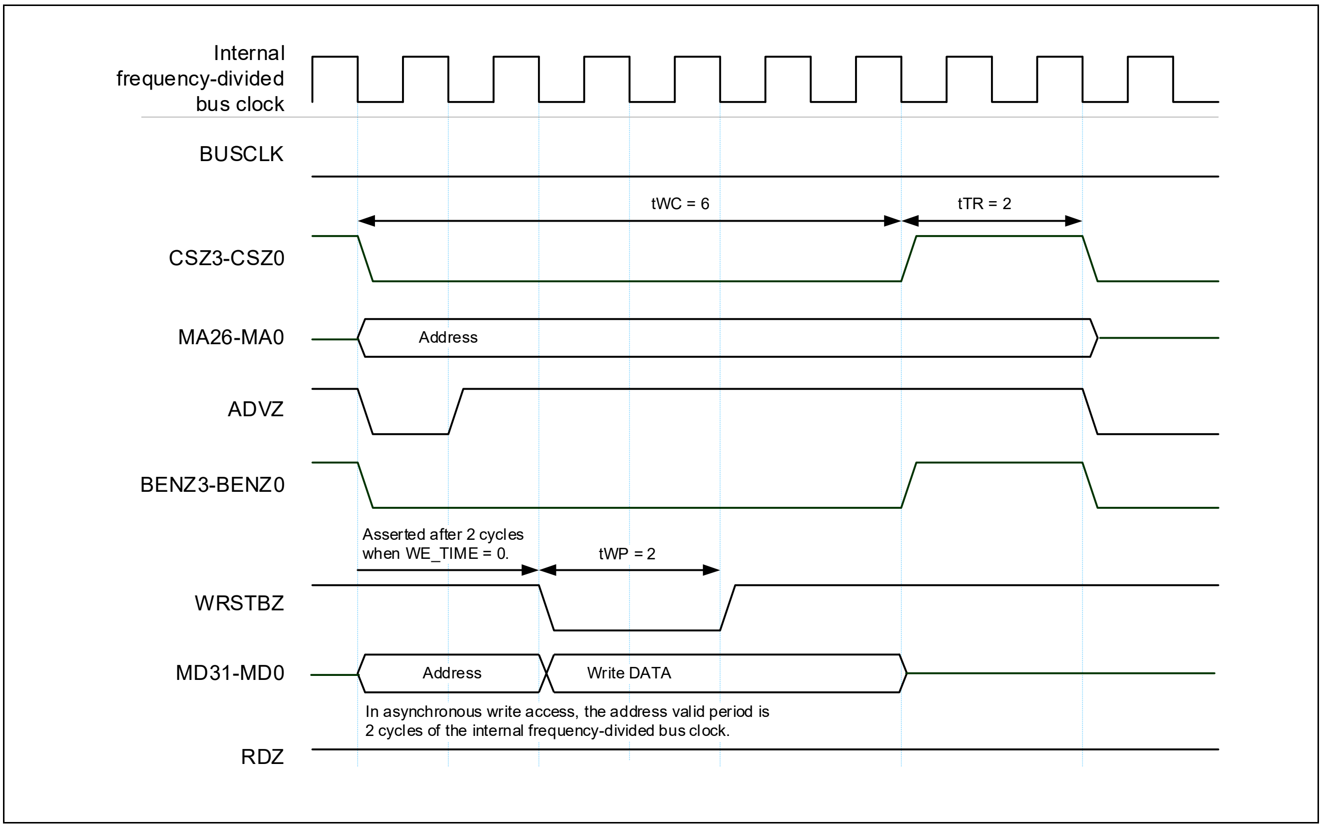


Figure 15.13 Asynchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled, WE_TIME = 0

Remark: ADMUXMODE pin = High level (multiplexed mode)

SET_CYCLES.WE_TIME = 0B (WRSTBZ is asserted 2 cycles after the CSZ is asserted)

T_TR[2:0] = 010B (2 cycles)

T_WP[2:0] = 010B (2 cycles)

T_WC[3:0] = 0110B (6 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

WR_BL = 000B (single access)

WR_SYNC = 0B (asynchronous access)

MW[1:0] = 10B (bus width: 32 bits)

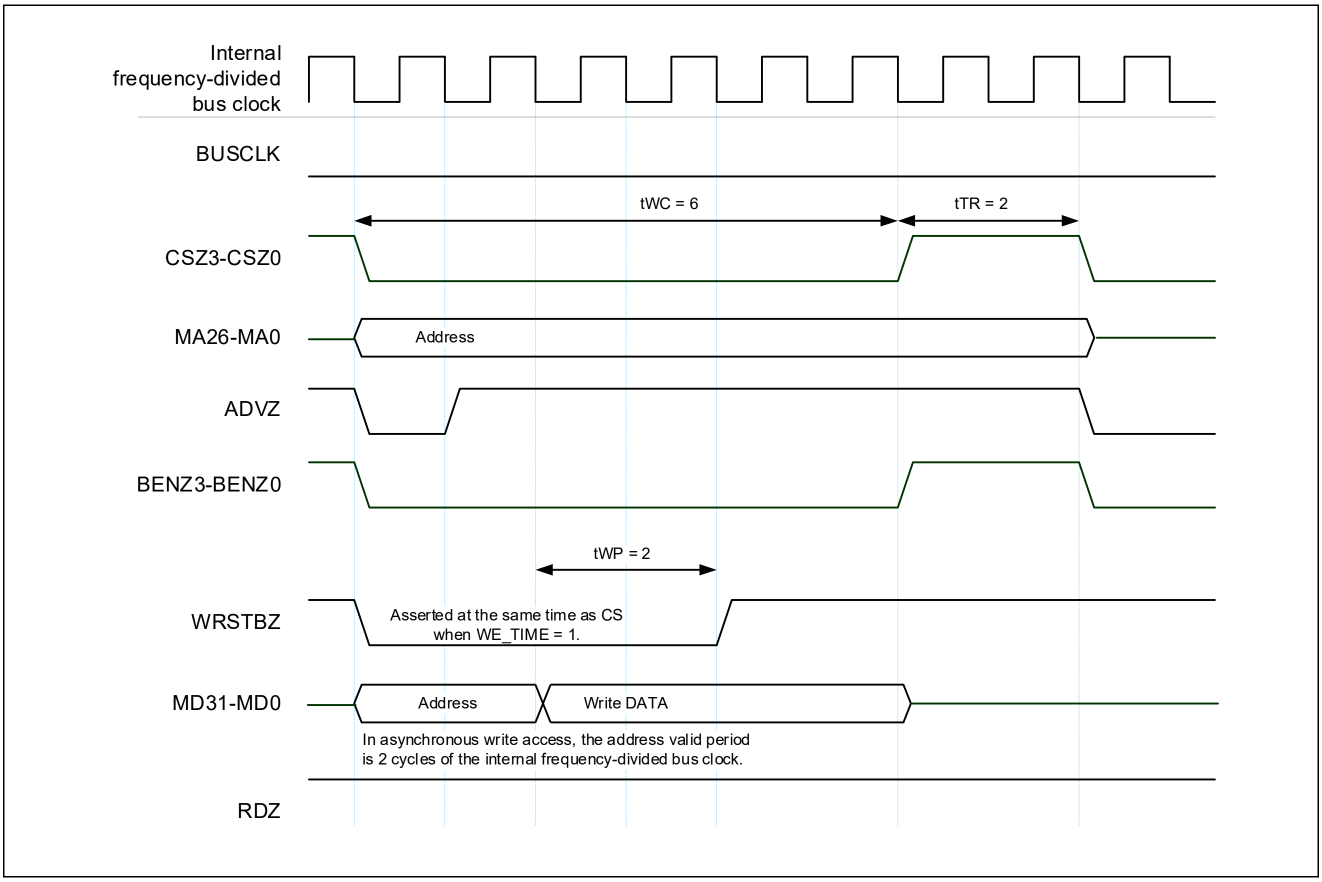


Figure 15.14 Asynchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled, WE_TIME = 1

Remark: ADMUXMODE pin = High level (multiplexed mode)

SET_CYCLES.WE_TIME = 1B (WRSTBZ is asserted at the same time as CSZ)

T_TR[2:0] = 010B (2 cycles)

T_WP[2:0] = 010B (2 cycles)

T_WC[3:0] = 0110B (6 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

WR_BL = 000B (single access)

WR_SYNC = 0B (asynchronous access)

MW[1:0] = 10B (bus width: 32 bits)

15.4.2 Synchronous Access Timing

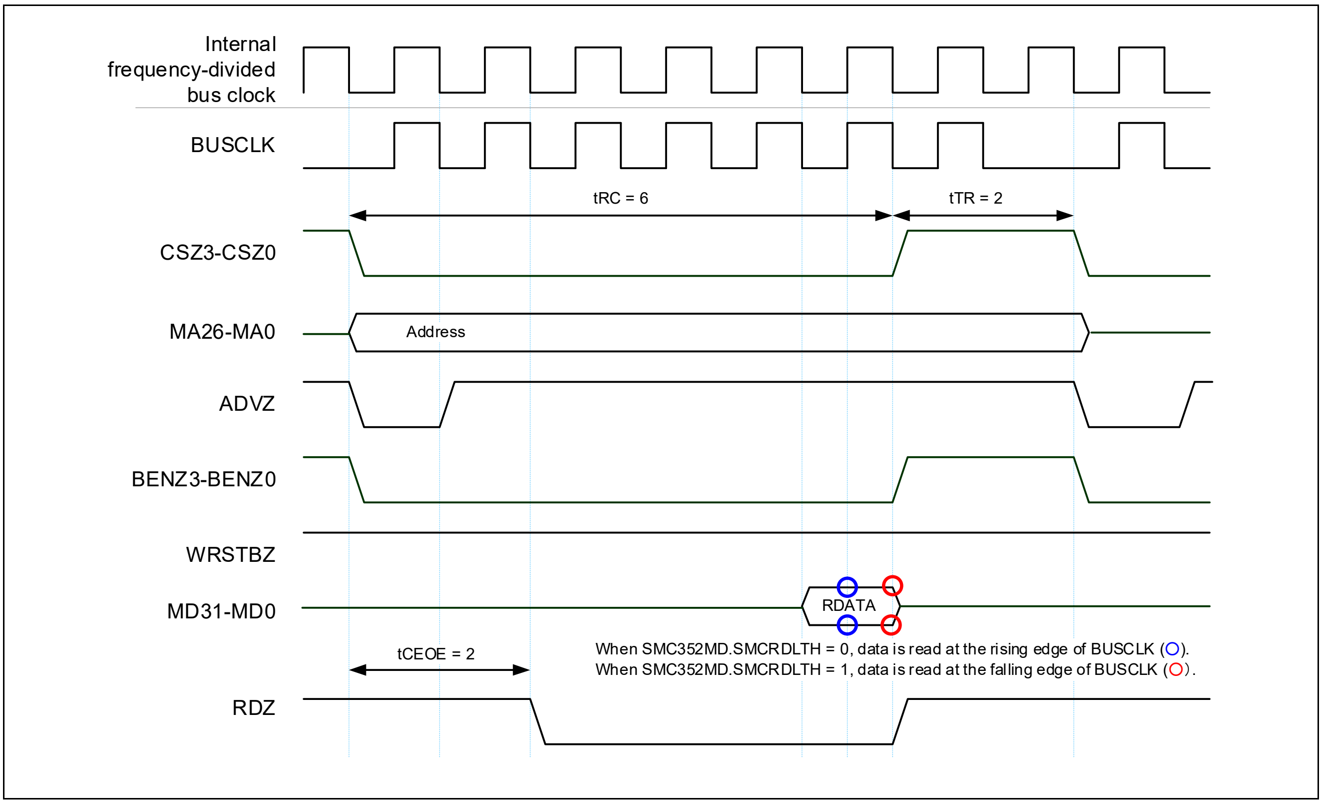


Figure 15.15 Synchronous SRAM, Separate Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate mode)

SET_CYCLES.T_TR[2:0] = 010B (2 cycles)

T_CEOE[2:0] = 010B (2 cycles)

T_RC[3:0] = 0110B (6 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

RD_BL = 000B (single access)

RD_SYNC = 1B (synchronous access)

MW[1:0] = 10B (bus width: 32 bits)

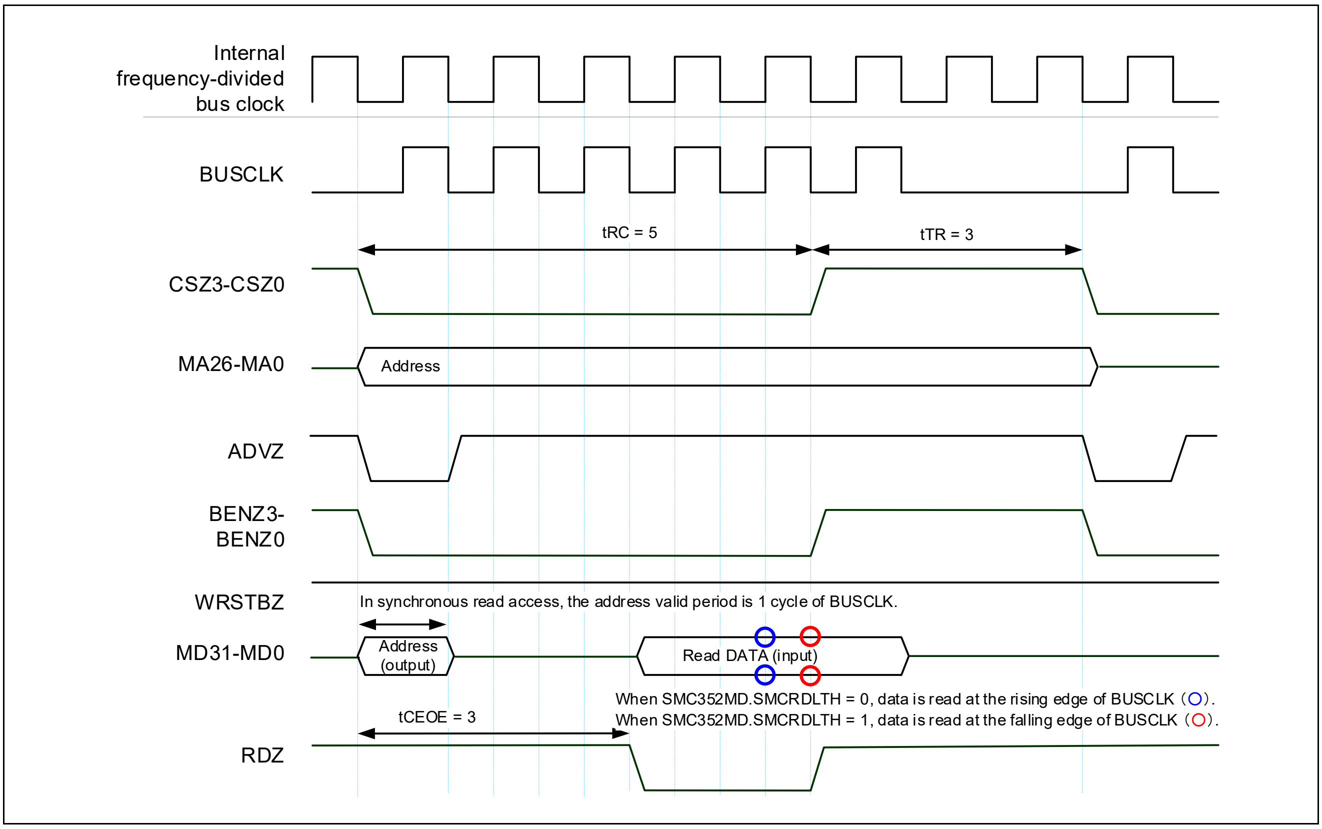


Figure 15.16 Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (multiplexed mode)

SET_CYCLES.T_TR[2:0] = 011B (3 cycles)

T_CEOE[2:0] = 011B (3 cycles)

T_RC[3:0] = 0101B (5 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

RD_BL = 000B (single access)

RD_SYNC = 1B (synchronous access)

MW[1:0] = 10B (bus width: 32 bits)

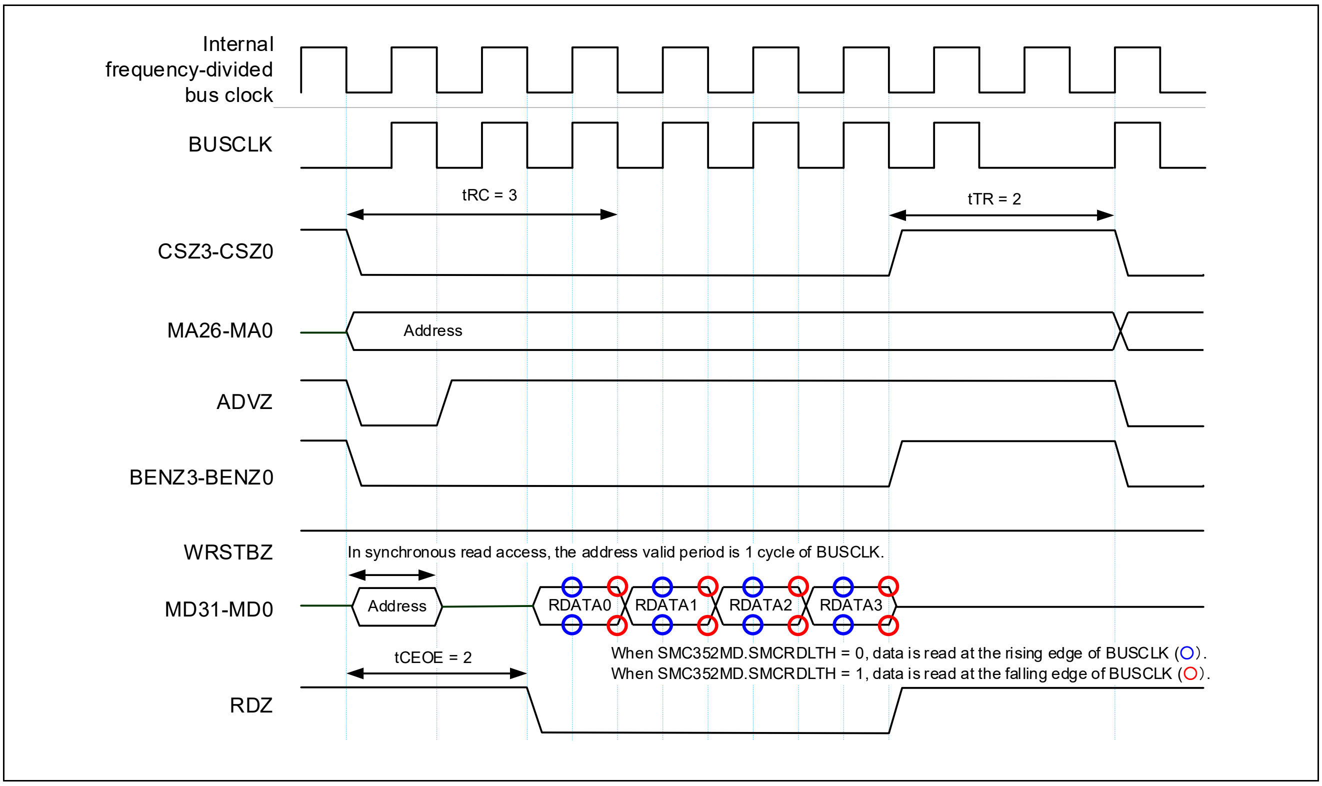


Figure 15.17 Synchronous SRAM, Multiplexed Bus Mode, Burst Read Access (4-Beat), ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed mode)

SET_CYCLES.T_TR[2:0] = 010B (2 cycles)

T_CEOE[2:0] = 010B (2 cycles)

T_RC[3:0] = 0011B (3 cycles)

SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)

BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)

ADV = 1B (ADVZ enabled)

RD_BL = 001B (up to 4 data blocks)

RD_SYNC = 1B (synchronous access)

MW[1:0] = 10B (bus width: 32 bits)

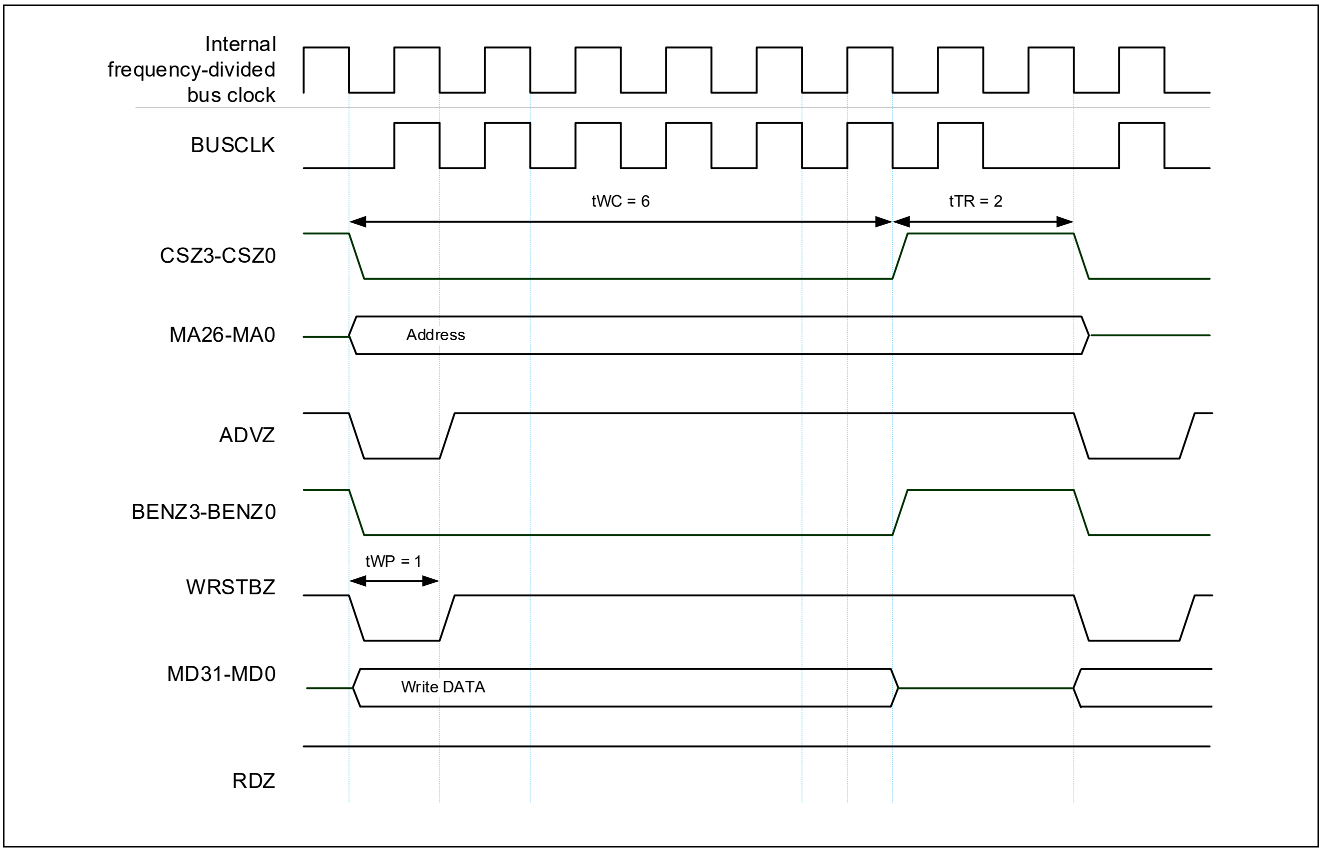


Figure 15.18 Synchronous SRAM, Separate Bus Mode, Write Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate mode)
SET_CYCLES.T_TR[2:0] = 010B (2 cycles)
T_WP[2:0] = 001B (1 cycle)
T_WC[3:0] = 0110B (6 cycles)
SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)
BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)
ADV = 1B (ADVZ enabled))
WR_BL = 000B (single access)
WR_SYNC = 1B (synchronous access)
MW[1:0] = 10B (bus width: 32 bits)

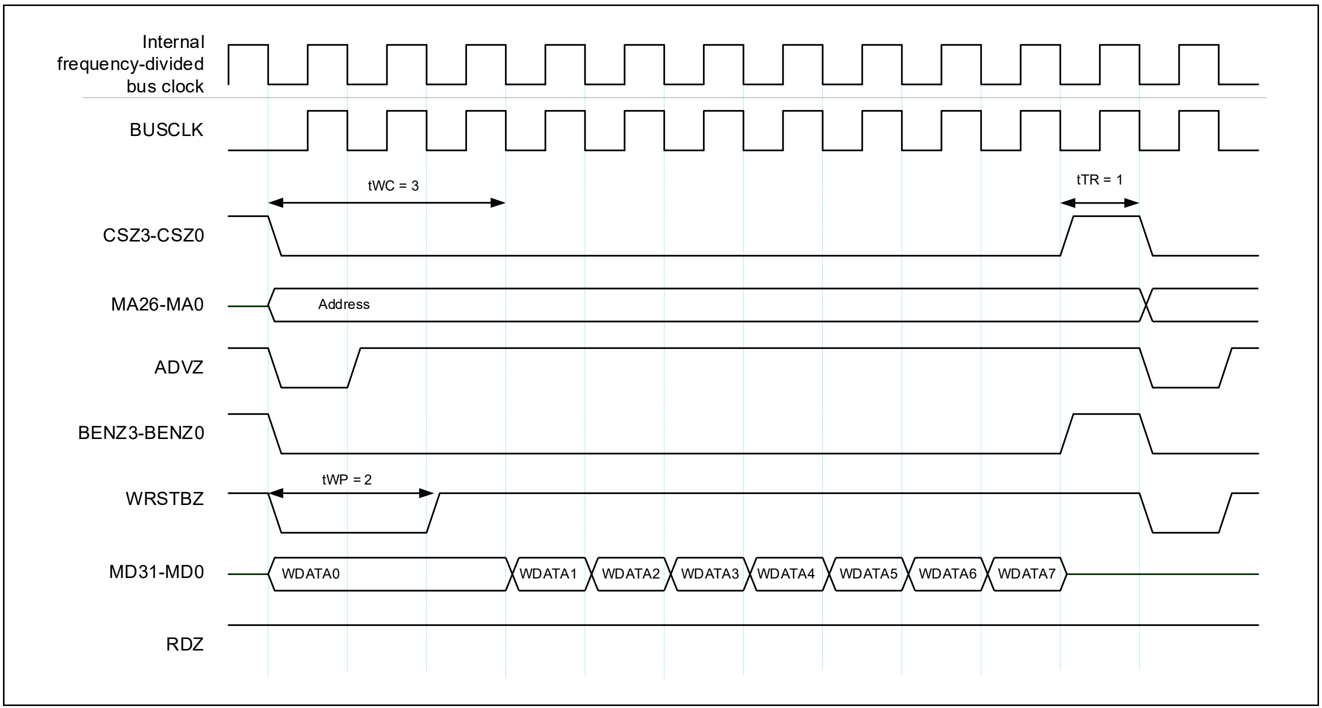


Figure 15.19 Synchronous SRAM, Separate Bus Mode, Burst Write Access (8-Beat), ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate mode)
SET_CYCLES.T_TR[2:0] = 001B (1 cycle)
T_WP[2:0] = 010B (2 cycles)
T_WC[3:0] = 0011B (3 cycles)
SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)
BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)
ADV = 1B (ADVZ enabled))
WR_BL = 010B (up to 8 data blocks)
WR_SYNC = 1B (synchronous access)
MW[1:0] = 10B (bus width: 32 bits)

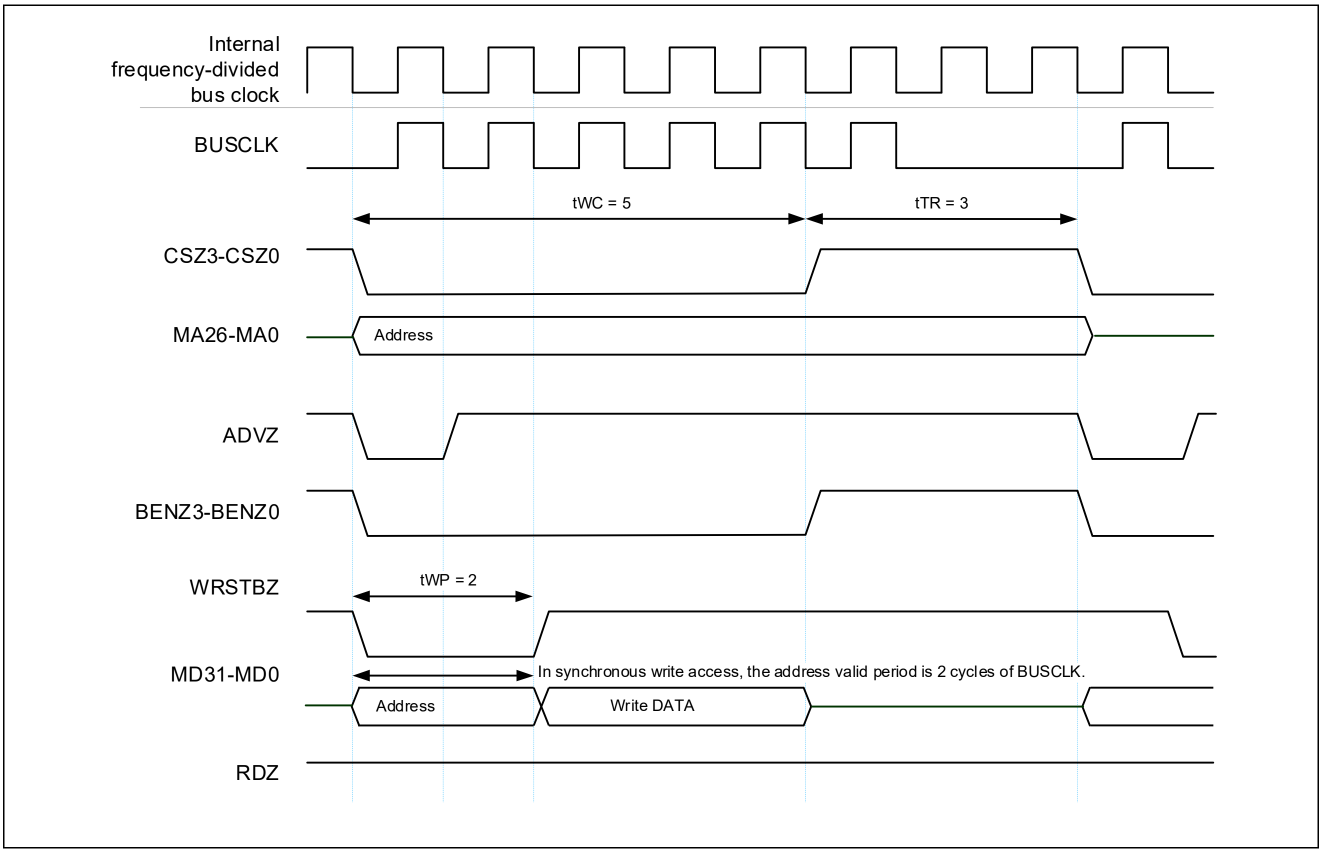


Figure 15.20 Synchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed mode)
SET_CYCLES.T_TR[2:0] = 011B (3 cycles)
T_WP[2:0] = 010B (2 cycles)
T_WC[3:0] = 0101B (5 cycles)
SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)
BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)
ADV = 1B (ADVZ enabled))
WR_BL = 000B (single access)
WR_SYNC = 1B (synchronous access)
MW[1:0] = 10B (bus width: 32 bits)

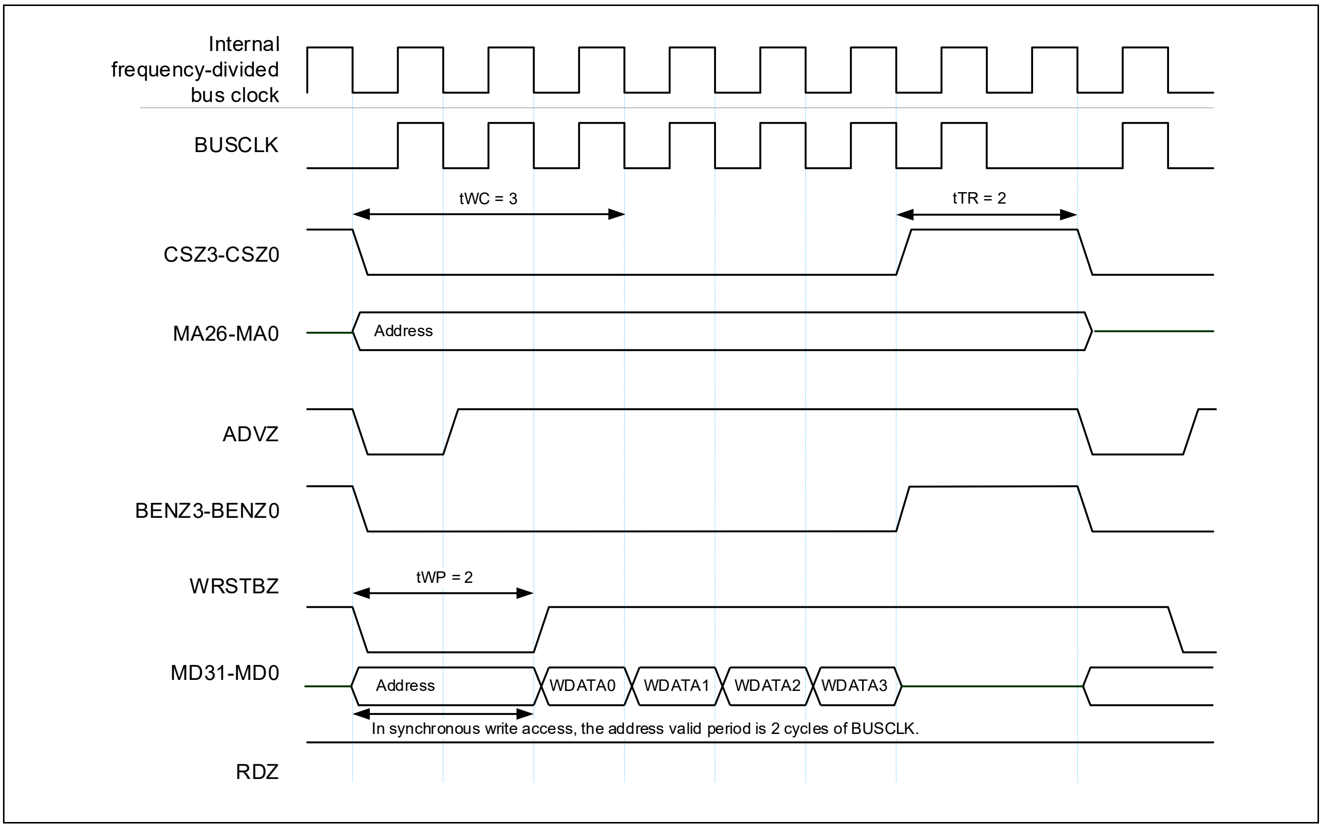


Figure 15.21 Synchronous SRAM, Multiplexed Bus Mode, Burst Write Access (4-Beat), ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed mode)
SET_CYCLES.T_TR[2:0] = 010B (2 cycles)
T_WP[2:0] = 010B (2 cycles)
T_WC[3:0] = 0011B (3 cycles)
SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)
BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)
ADV = 1B (ADVZ enabled))
WR_BL = 001B (up to 4 data blocks)
WR_SYNC = 1B (synchronous access)
MW[1:0] = 10B (bus width: 32 bits)

15.4.3 Wait Timing

Wait signals (WAITZ, WAIT1 to WAIT3) are only valid for synchronous access.

Caution: Wait signals (WAITZ, WAITZ1 to WAITZ3) are latched in synchronization with the internal clock, so the states of the wait signals are effective one cycle before the input is latched. When the setting of tRC and tWC is "N", the wait signals are effective after "N -1" cycles.

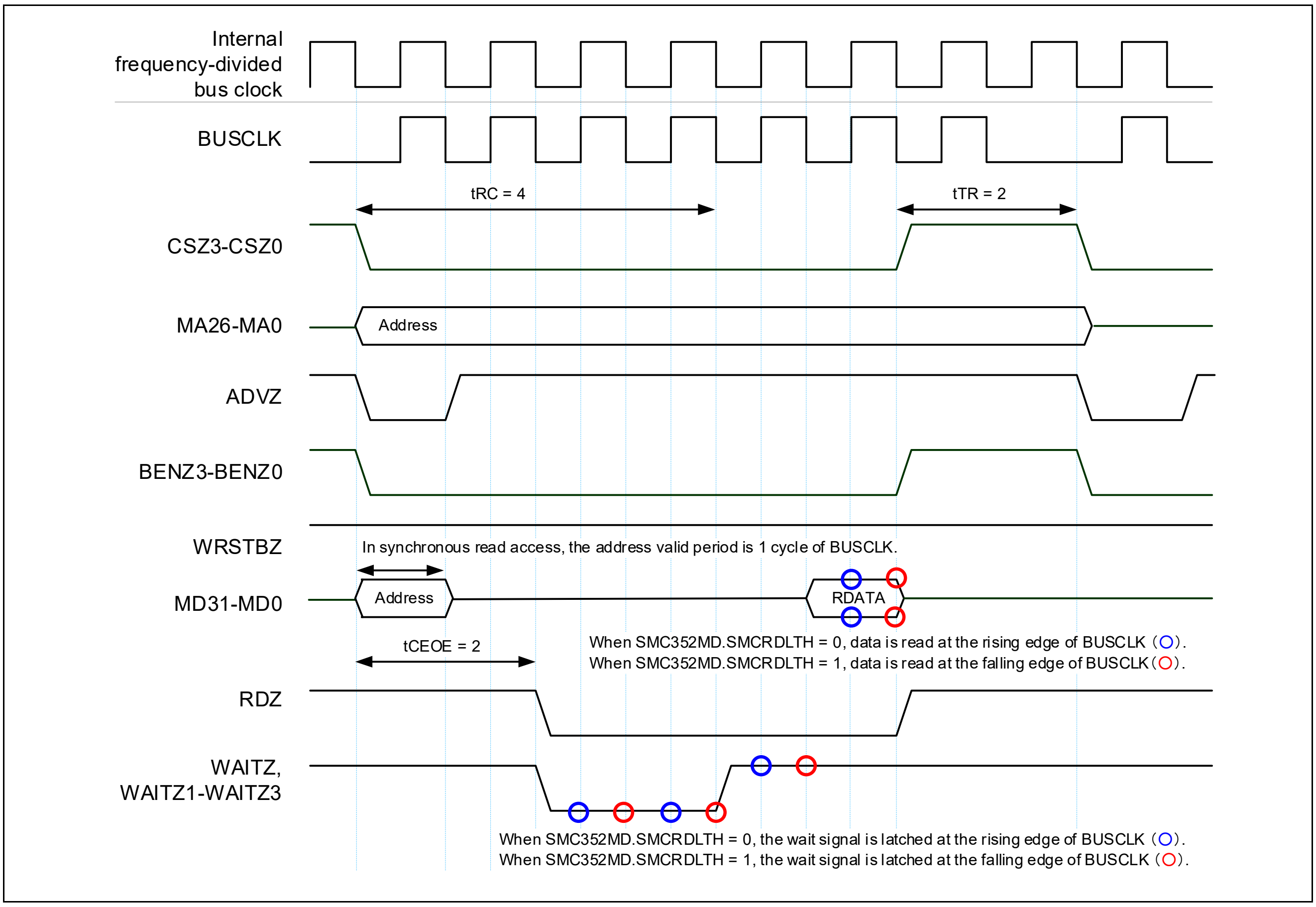


Figure 15.22 Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed mode)
SET_CYCLES.T_TR[2:0] = 010B (2 cycles)
T_CEOE[2:0] = 010B (2 cycles)
T_RC[3:0] = 0100B (4 cycles)
SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)
BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)
ADV = 1B (ADVZ enabled))
RD_BL = 000B (single access)
RD_SYNC = 1B (synchronous access)
MW[1:0] = 10B (bus width: 32 bits)

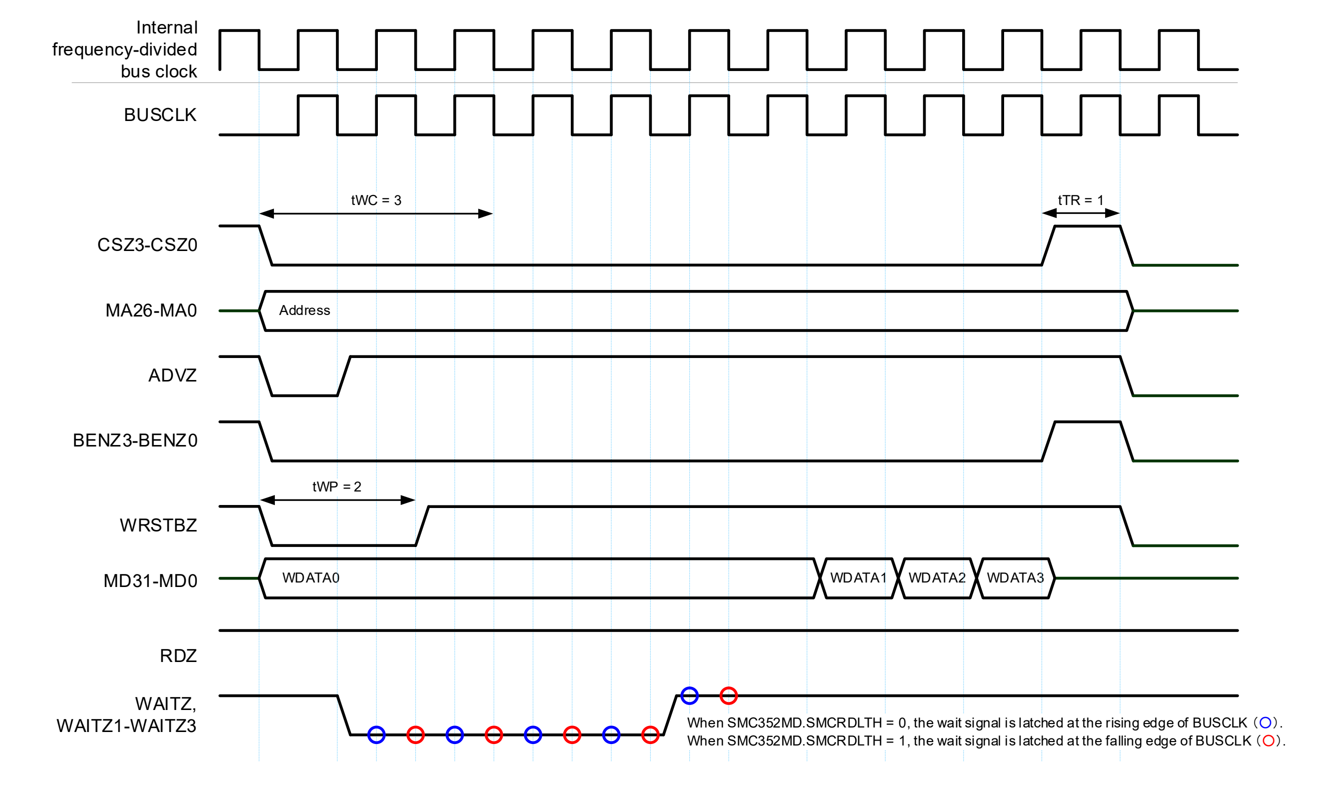


Figure 15.23 Synchronous SRAM, Separate Bus Mode, Burst Write Access (4-beat), ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate mode)
SET_CYCLES.T_TR[2:0] = 001B (1 cycle)
T_WP[2:0] = 0010B (2 cycles)
T_WC[3:0] = 0011B (3 cycles)
SET_OPMODE.BURST_ALIGN[2:0] = 000B (no burst boundary)
BLS_TIME = 0B (BENZ0 to BENZ3 pins used as byte enable)
ADV = 1B (ADVZ enabled))
WR_BL[2:0] = 001B (up to 4 data blocks)
WR_SYNC = 1B (synchronous access)
MW[1:0] = 10B (bus width: 32 bits)

Caution: Do not change the setting of the operating mode setting pins such as the MEMIFSEL and MEMCSEL pins during operation. Fix the setting before release from the reset state.

16. External MCU Interface

An external MCU interface is provided to allow use of the internal resources of R-IN32M4-CL3 by the external MCU. The external MCU interface is multiplexed with an external memory interface. When the MEMIFSEL pin is at the High level, the external MCU interface works. After power is initially supplied to the system, set the MEMIFSEL pin level by the time of de-assertion of whichever is the later of the PONRZ and RESETZ reset signals. Dynamic switching is not supported.
When the external MCU interface is used, booting of R-IN32M4-CL3 from the external MCU or serial flash memory is available. However, external memory access (to external ROM and SRAM) is not available.

The external MCU interface supports the asynchronous SRAM supported MCU connection mode and synchronous SRAM supported MCU connection mode. When the HIFSYNC pin is at the High level, it is the synchronous SRAM MCU interface. When the pin is at the Low level, it is the asynchronous SRAM MCU interface.
Furthermore, the external MCU interface supports the synchronous burst transfer supported MCU connection mode (clock synchronous), and therefore large volumes of data can be accessed at High speed. Placing both the MEMIFSEL pin and the MEMCSEL pin in the High level selects this mode.

Table 16.1 Mode of the External MCU Interface Selected by the Level on the Operating Mode Setting Pin

MEMIFSEL	MEMCSEL	HIFSYNC	ADMUXMODE	Function
Low	—	—	—	Not accessible from the external MCU (Operation is as the external memory interface.)
High	Low	Low	—	The chip is placed in asynchronous SRAM supported MCU connection mode. Connection of the bus clock signal to HBUSCLK is not required.
		High	—	The chip is placed in synchronous SRAM supported MCU connection mode. Connection of the bus clock signal to HBUSCLK is required.
	High	Low	Low	Setting prohibited
			High	Setting prohibited
		High	Low	The chip is placed in synchronous burst transfer supported MCU connection mode. (Separation of addresses and data)
			High	The chip is placed in synchronous burst transfer supported MCU connection mode. (Multiplexing of addresses and data)

Cautions 1. The BUS32EN, HWRZSEL, MEMCSEL, MEMIFSEL, and HIFSYNC pins do not support the dynamic switching. Determine the input values during a reset.

16.1 Memory MAP

A 2-Mbyte space is provided as the external MCU interface.

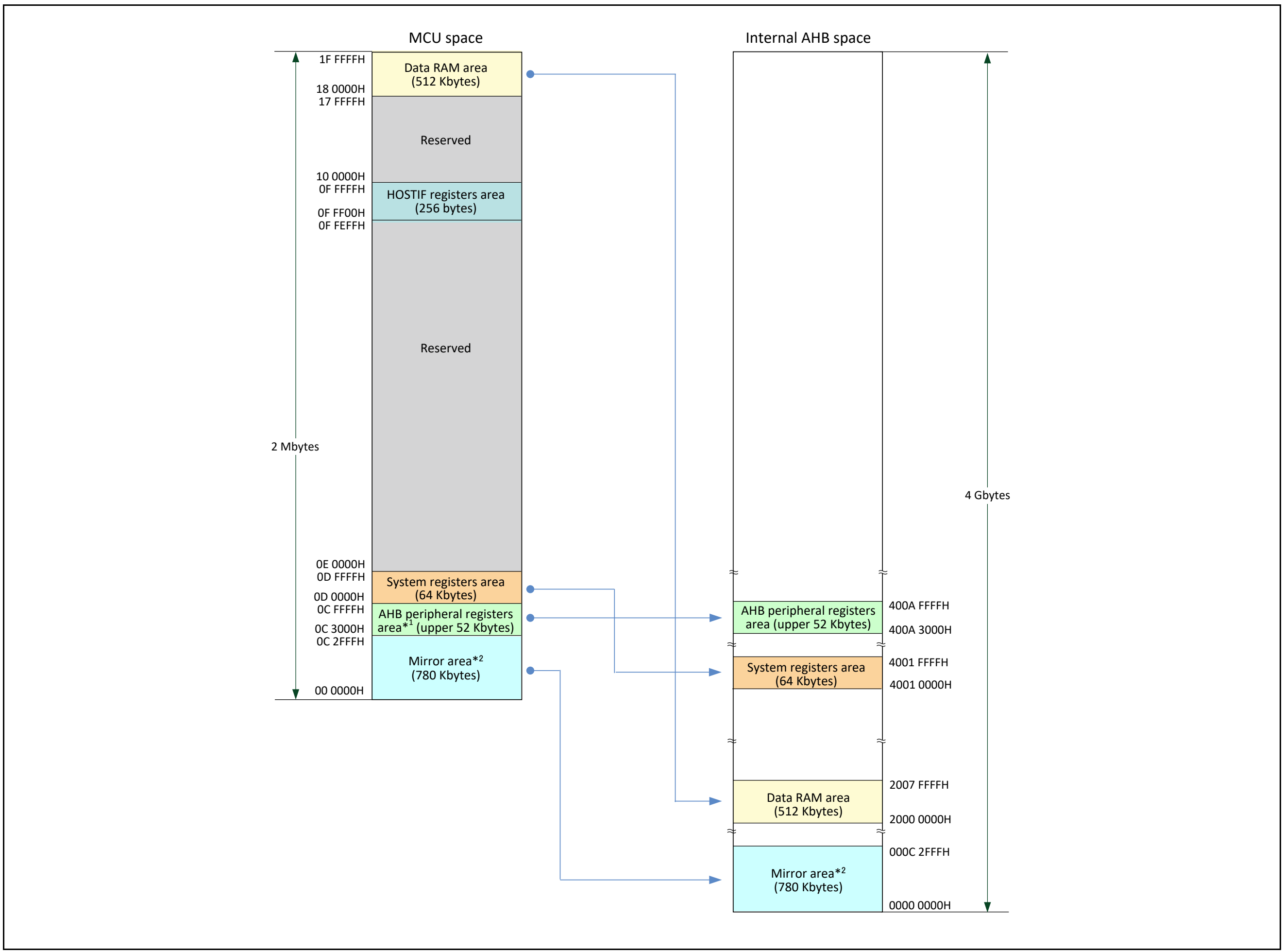


Figure 16.1 Memory Map (External MCU Interface Space)

Notes 1. The upper 52 Kbytes of the AHB peripheral registers area covers the range from the GPIO area to the synchronous burst memory controller control registers. For details, refer to Figure 3.2 "Memory Map (AHB Peripheral Registers Area)".

2. The mirror area is an area where one of the external memory area/serial flash memory area/instruction RAM area can be mirrored by setting the external pins (BOOT0, BOOT1). When access is from the external MCU interface, the following areas can be accessed.

• When {BOOT1, BOOT0} = 01 (External serial flash ROM boot):
Reserved area (access prohibited)

• When {BOOT1, BOOT0} = 10 (External MCU boot):
Instruction RAM area

For the list of slaves that can be accessed by the master, refer to Section 6 "Bus Architecture".
For details on the boot mode, refer to Section 7 "Booting Procedure".

16.2 Synchronous/Asynchronous SRAM Supported MCU Connection Modes

The external MCU interface supports the asynchronous SRAM supported MCU connection mode and synchronous SRAM supported MCU connection mode. When the HIFSYNC pin is at the high level, it is a synchronous SRAM MCU interface, and when HIFSYNC is at the low level, it is an asynchronous SRAM MCU interface.

Two modes are used when the MEMIFSEL pin is at the High level and the MEMCSEL pin is at the Low level.
After power is initially supplied, set the MEMIFSEL and MEMCSEL pin levels by the time of de-assertion of the reset signal. Dynamic switching is not supported.

16.2.1 Functional Overview

- Interface system
  - Asynchronous SRAM with wait control (for reading and writing)
  - Page ROM reading with wait control
- Synchronous relationship (set up with the HIFSYNC pin)
  - HBUSCLK synchronous mode (MAX: 50M Hz), asynchronous mode
- Bus width (set up with the BUS32EN pin)
  - 16 bits / 32 bits
- Transfer data size
  - 32 bits / 16 bits / 8 bits
- Write buffer: Two stages (when the synchronous mode is selected) or one stage (when the asynchronous mode is selected)
- Read buffer: Advance reading of up to 32 bytes is upported.
- Multiplexing of addresses and data
  - No multiplexing of addresses and data
- Checking of various states
  - Internal reset state (The state can be checked only while the asynchronous mode is set.)
  - States of the HIFSYNC and BUS32EN pins

16.2.2 Operation

(1) Supported transfer method

Table 16.2 Supported Transfer Method

Mode Setting			Supported Transfer Method	
MEMIFSEL	HIFSYNC	BUS32EN	Page Access Permitted Area	Page Access Prohibited Area
Low	—	—	Not accessible	Not accessible
High	Low (Asynchronous Mode)	Low (16-bit)	SRAM reading SRAM writing Page ROM reading	SRAM reading SRAM writing
		High (32-bit)		
	High (Synchronous Mode)	Low (16-bit)		
		High (32-bit)		

Remark: Page ROM reading can only be used for the area permitted by the HIFBCC and HIFPRC registers.

(2) Bus Sizing

The bus size for internal access is in accordance with the external bus width.

Table 16.3 Bus Sizing

HWRZSEL	BUS32EN	Access area in R-IN32M4-CL3: Area	Bus width	R/W	Targets for access through internal bus
L	L (16-bit)	Registers area	32-bit	R	All byte lanes (32-bit)
				W	*1
		AHB area to be buffered	32-bit	R	All byte lanes (32-bit)*2
				W	*1
		AHB area not to be buffered	32-bit	R	Specified byte lanes only
				W	*1
	H (32-bit)	Registers area	32-bit	R	All byte lanes (32-bit)
				W	*1
		AHB area to be buffered	32-bit	R	All byte lanes (32-bit)
				W	*1
		AHB area not to be buffered	32-bit	R	Specified byte lanes only
				W	*1
H	L (16-bit)	Registers area	32-bit	R	All byte lanes (32-bit)
				W	*1
		AHB area to be buffered	32-bit	R	All byte lanes (32-bit)*2
				W	*1
		AHB area not to be buffered	32-bit	R	Lower 16 bits or higher 16 bits*3
				W	*1
	H (32-bit)	Registers area	32-bit	R	All byte lanes (32-bit)
				W	*1
		AHB area to be buffered	32-bit	R	All byte lanes (32-bit)*2
				W	*1
		AHB area not to be buffered	32-bit	R	All byte lanes (32-bit)*3
				W	*1

Note 1: Only the specified byte lanes are to be written.
Note 2: The AHB area being buffered is always read in 32-bit units.
Note 3: When HWRZSEL = H, the AHB area not being buffered is read with the external bus width.

Caution 1. The respective access areas in the table represent the following.
• Registers area: External MCU interface registers area
• AHB area: Other than the above (Data RAM area or Network RAM area, etc.)

(3) Synchronous Mode and Asynchronous Mode

The synchronous SRAM supported MCU connection mode or asynchronous SRAM supported MCU connection mode is selectable by the setting of the HIFSYNC pin.
The synchronous relationship of each interface signal is described below.

Table 16.4 Synchronous Relationship of External MCU interface Signals

Signal Name	I/O	HIFSYNC (Selection of Synchronous Relationship)		
		H (Synchronous Mode)		L (Asynchronous Mode)
		WRITE	READ	
HCSZ	I	HBUSCLK synchronous	Asynchronous	Asynchronous
HPGCSZ	I	HBUSCLK synchronous	Asynchronous	Asynchronous
HA20–HA1	I	HBUSCLK synchronous	Asynchronous	Asynchronous
HRDZ	I	—	Asynchronous	Asynchronous
HWRSTBZ	I	HBUSCLK synchronous	—	Asynchronous
HWRZ3–HWRZ0, HBENZ3–HBENZ0	I	HBUSCLK synchronous	—	Asynchronous
HD31–HD0 (input)	I	HBUSCLK synchronous	—	Asynchronous
HD31–HD0 (output)	O	—	Asynchronous	Asynchronous
HWAITZ	O	HBUSCLK synchronous		Asynchronous
HERROUTZ	O	Asynchronous		

(4) Buffer reading

Buffer reading can be used for the data RAM area and others. Buffer reading is enabled when the RBUFONn bit of the HIFBCC register is set to 1. When buffer reading is enabled, up to 32 bytes are read in advance from each address read by the external MCU interface and stored in the buffer. The next time the address accessed by the external MCU interface matches the original address of the data stored in the advance-read buffer, the target data is read from the buffer. This improves throughput because data can be read from the advance-read buffer at high speed.

Remark: Data at addresses for advance reading are always read in the ascending order of addresses.

(5) Page ROM reading

In addition to buffer reading, page ROM reading can be used for the data RAM area and others.
To enable page ROM reading, set the RBUFONn bit of the HIFBCC register and the PAGEONn bit of the HIFPRC register to 1.
In reading from page ROM, the wait state is released (the HWAITZ signal output is at the High level) once all data in a page has been prepared for off-page reading. This improves throughput because on-page reading following off-page reading does not require wait processing.

Table 16.5 Page Size and On-Page Ratio

Page Size Setting	Address for Use in Page Judgment	On-Page Ratio (TYP)		Remark
HIFPRC.PAGESZ		BUS32EN	Ratio	
0B (8-byte)	HA[20:3]	0B (16-bit)	3/4	
		1B (32-bit)	1/2	
1B (16-byte)	HA[20:4]	0B (16-bit)	7/8	
		1B (32-bit)	3/4	

Cautions 1. Page ROM access to the area where page ROM reading is disabled is prohibited.
Attempted page ROM access to an area where page ROM reading is disabled may lead to a deadlock.

2. Areas where page ROM reading is enabled can be read in the same way as normal SRAM. However, since the chip is placed in the wait state until all data on a page have been provided, the latency increases in comparison with the case where page ROM reading is disabled.

3. In Page ROM reading, access across 16-byte boundaries is prohibited.
Start page ROM reading from the start of 16-byte boundaries such as xx00H and xx40H.

Remark: The page size is not affected by the bus width.
Large page size will increase the on-page ratio and transfer rate. Selecting 16 bytes as the page size is recommended unless this creates a problem for the system.

16.2.3 Basic Operation Timing of the External MCU Interface

(1) Access to the External MCU Interface Registers Area

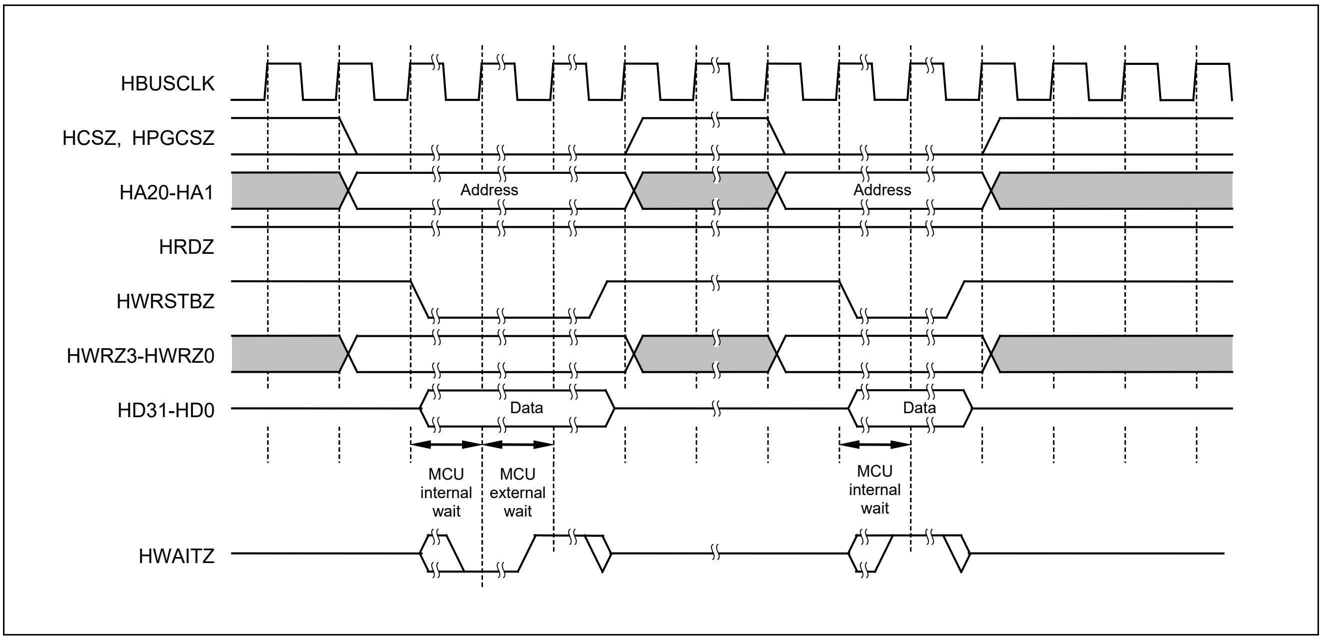


Figure 16.2 Writing to the External MCU Interface Registers Area (SRAM writing)

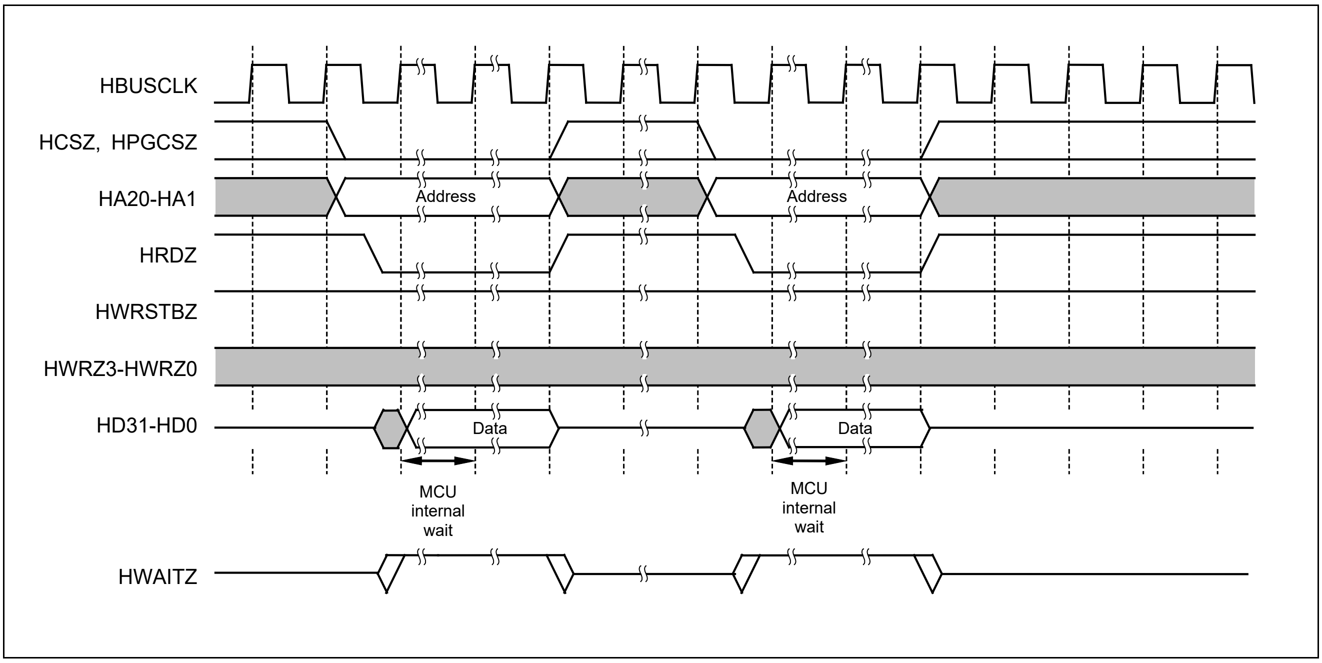


Figure 16.3 Reading from the External MCU Interface Registers Area (SRAM reading)

(2) Access to Other Areas

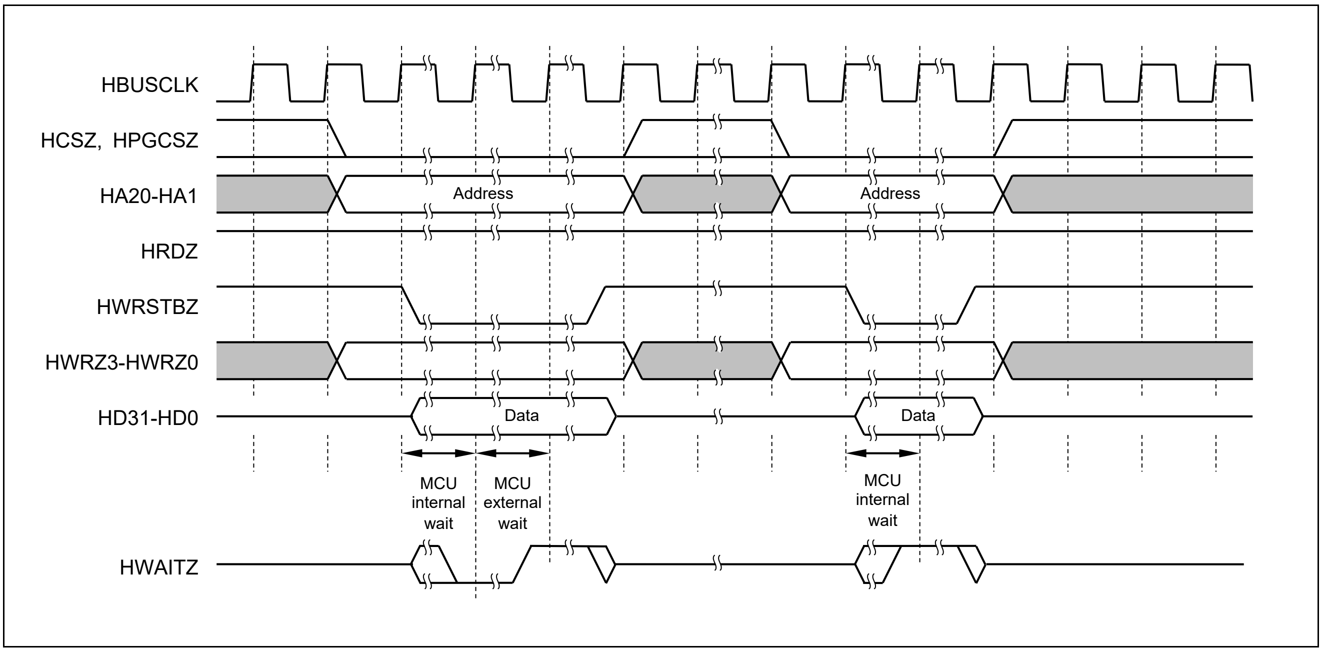


Figure 16.4 Writing to Other Areas (SRAM writing)

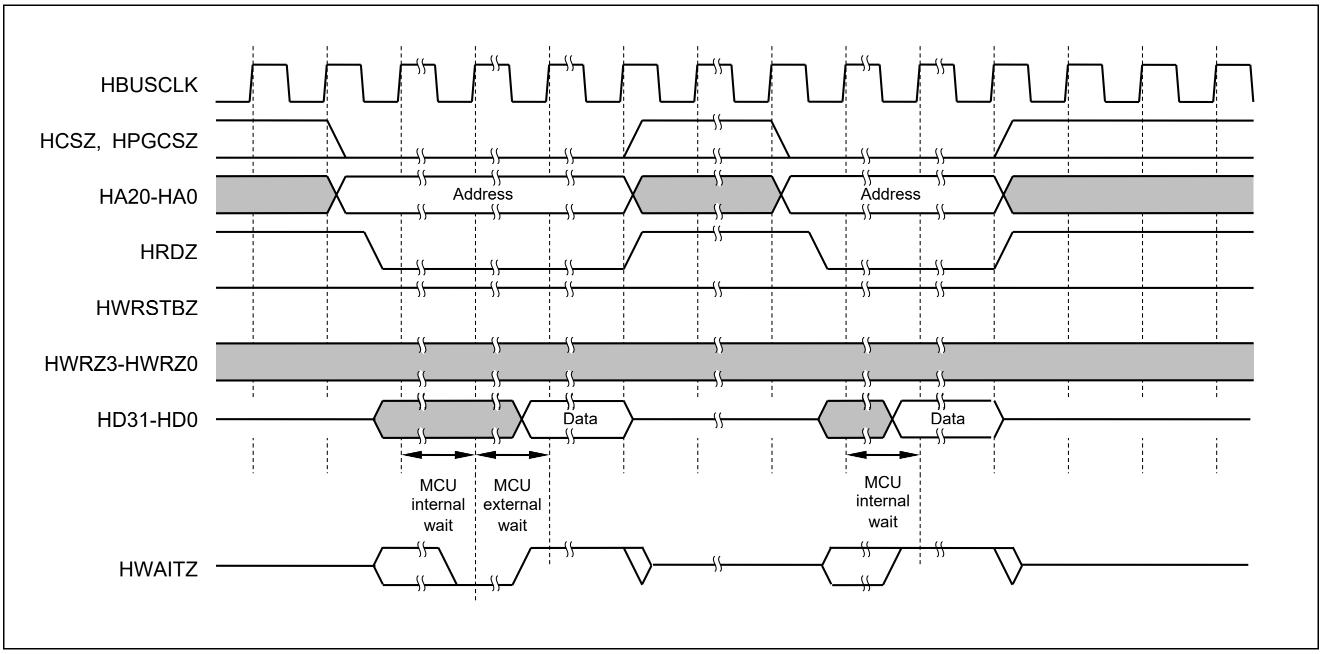


Figure 16.5 Reading from Other Areas (SRAM reading)

16.2.4 Asynchronous Connection Timing Adjustment of the External MCU Interface

(1) Outline of asynchronous connection timing adjustment

Adjustment of asynchronous connection timing is provided to adjust for variations in the relative timing between the external MCU interface and external MCU in asynchronous connection (HIFSYNC = L).

Table 16.6 Asynchronous Connection Timing Adjustment of the External MCU Interface

Transfer Method	Asynchronous Connection Timing Adjustment				Target Areas for Adjustment		
	Adjustment Timing		Setting Register		External MCU interface Registers Area	CC-Link IE	Other Areas
	Target Signal	Relativity Signal	Register Name	Bits Name			
SRAM writing	HCSZ HPGCSZ HA20–HA1 HWRZ HD31–HD0	HWRSTBZ	HIFBTC	WRSTD1–0	✓	—	✓
SRAM reading	HCSZ HPGCSZ HA20–HA1	HRDZ	HIFBTC	RDSTD1–0	—	—	✓
	HD31–HD0	HWAITZ	HIFBTC	RDDTS1–0	—	—	✓
Page ROM reading	HCSZ HPGCSZ HA20–HA1	HA20–HA1	HIFBTC	RDSTD1–0	—	—	✓
	HA20–HA1	HA20–HA1	HIFBTC	PASTD2–0	—	—	✓
	HD31–HD0	HWAITZ	HIFBTC	RDDTS1–0	—	—	✓

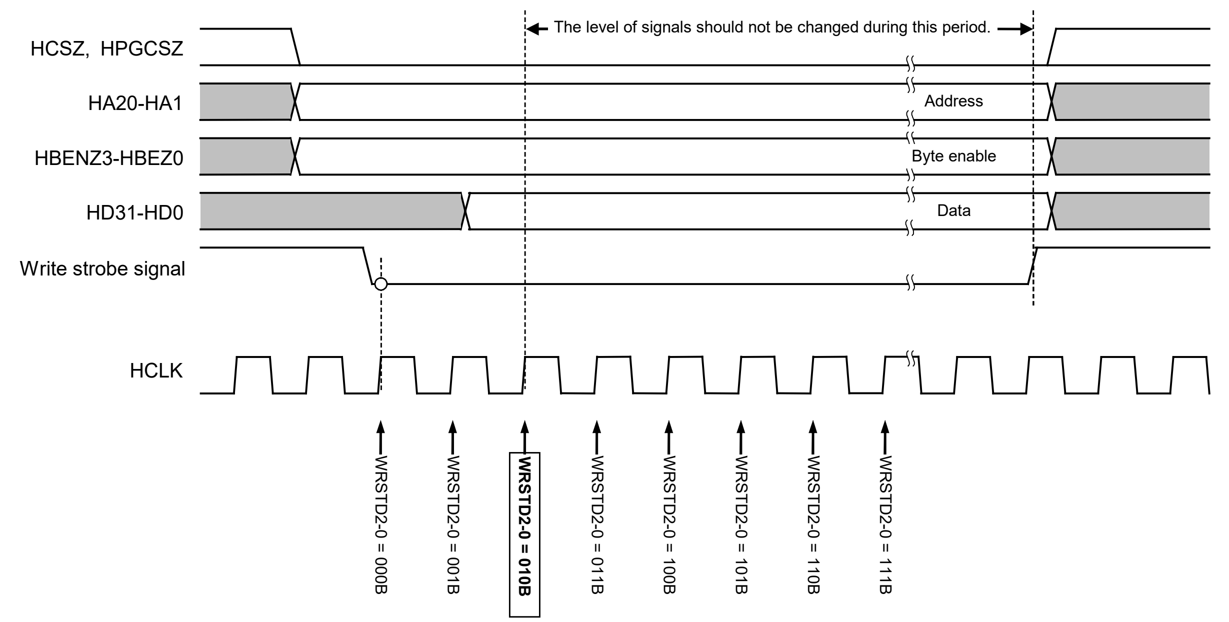
Cautions 1. Timing adjustment does not apply to the following accesses.
• Reading from the external MCU interface registers area
2. Timing adjustment affects access latency.

(2) Asynchronous Connection Timing Adjustment (SRAM writing)

Writing to internal resources starts in response to the detection of a falling edge of the write strobe signal (HWRSTBZ). The write strobe signal is selected as follows under the condition of input by the HWRZSEL pin or the BUS32EN pin. When externally writing to internal resources of R-IN32M4-CL3, stable addresses and data are required. R-IN32M4-CL3 has a function to adjust the timing for the sampling of addresses and data. Sampling timing adjustment can be handled by the HIFBTC register.

Table 16.7 Write Strobe Signal

Condition		Write Strobe Signal (Active Low)	Remark
HWRZSEL	BUS32EN		
Low	—	HWRSTBZ	
High	Low	HWRZ1 & HWRZ0	
	High	HWRZ3 & HWRZ2 & HWRZ1 & HWRZ0	

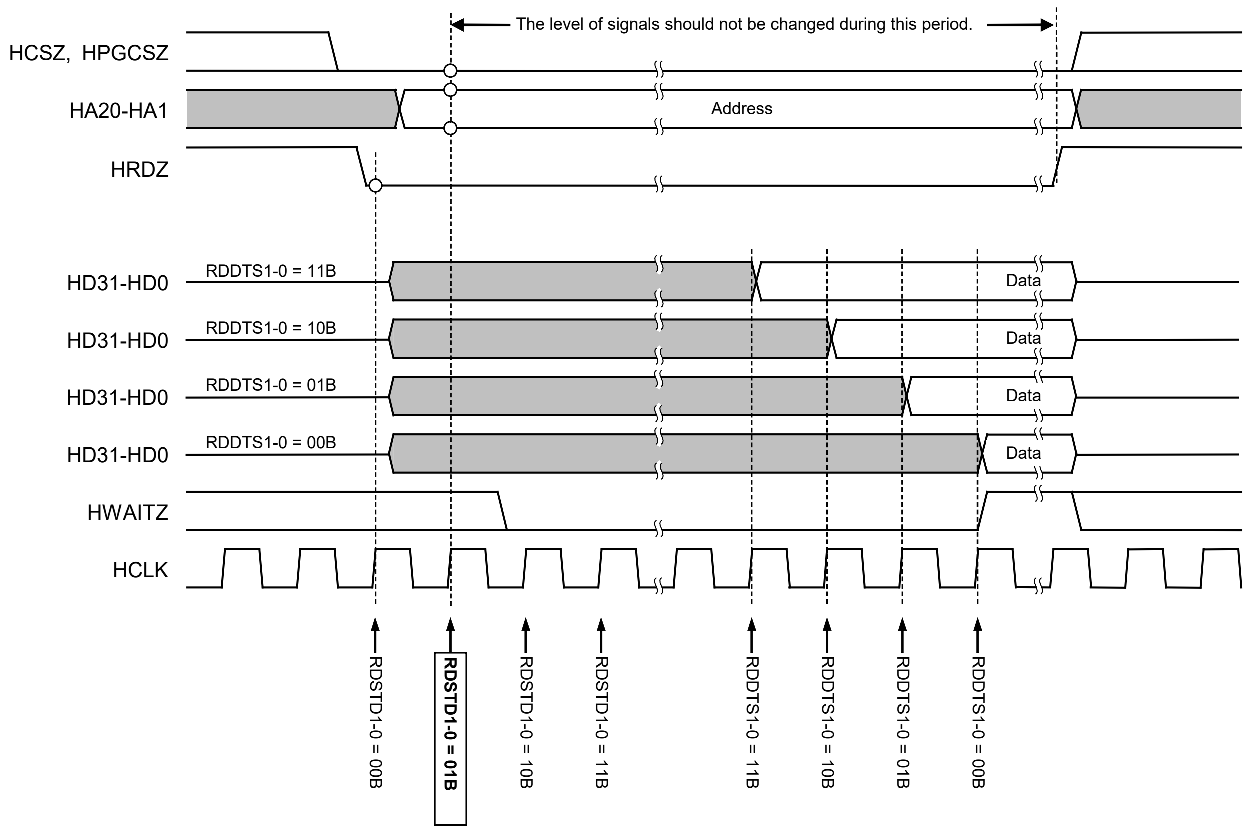


Remark: HCLK is an operation clock (100 MHz) for the internal system bus (AHB).

Figure 16.6 Timing Adjustment (SRAM writing)

(3) Asynchronous Connection Timing Adjustment (SRAM reading, page ROM reading)

Reading from internal resources starts in response to detection of a falling edge of the read strobe signal (HRDZ).
To ensure successful reading, the address and the HCSZ/HPGCSZ signal must be fixed before detection of the falling edge of the HRDZ signal. The timing for starting sampling can be adjusted by using the RDSTD1 and RDSTD0 bits of the HIFBTC register.
Furthermore, the time from fixing of the data (HD31 to HD0) to output of the High level as the HWAITZ signal can also be set. The time difference is set up by using the RDDTS1 and RDDTS0 bits of the HIFBTC register.



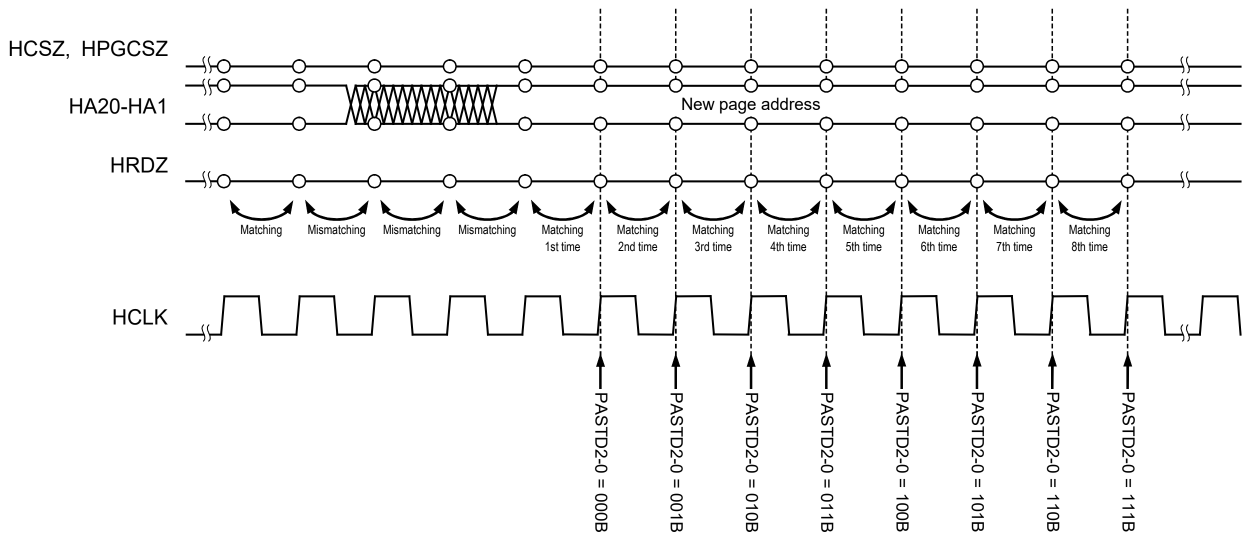
Remark: HCLK is an operation clock (100 MHz) for the internal system bus (AHB).

Figure 16.7 Timing Adjustment (SRAM reading, page ROM reading)

Caution: In reading SRAM, input a stable address signal during bus cycles after the start of sampling. Input of an unstable address signal may create a possibility of incorrect data being read and completion of the bus cycle not being possible without the HWAITZ signal de-asserted.

(4) Asynchronous Connection Timing Adjustment (page ROM reading)

Reading of a new page starts in response to detection of changes in the page address while reading the page ROM. To ensure successful read, stable address information is required. Timing adjustment is provided to sample stable addresses. The timing is adjusted by using the PASTD2 to PASTD0 bits of the HIFBTC register.



Remark: HCLK is an operating clock (100 MHz) of an internal system bus (AHB).

Figure 16.8 Timing Adjustment (page ROM reading)

16.2.5 Control Registers

(1) List of registers

The external MCU interface control registers are accessible by the MCU externally connected to R-IN32M4-CL3.

Address	Register Name	Symbol	R/W	Operation Unit [bit]			Initial value
				8	16	32	
0F FF00H	External MCU Interface Bus Control Register	HIFBCC	R/W	✓	✓	—	0001H
0F FF04H	External MCU Interface Timing Control Register	HIFBTC	R/W	✓	✓	—	3737H
0F FF08H	External MCU Interface Page ROM Control Register	HIFPRC	R/W	✓	✓	—	0000H
0F FF0CH	External MCU Interface Interrupt Request Control Register	HIFIRC	R/W	✓	✓	—	0000H
0F FF10H	External MCU Interface Error Source Register 0	HIFECR0	R	✓	✓	✓	0000 0000H
0F FF14H	External MCU Interface Error Source Register 1	HIFECR1	R	✓	✓	—	0000H
0F FF20H	External MCU Interface Monitor Register	HIFMON	R	✓	✓	—	Pin state*1
0F FF30H	External MCU Interface Specified Area Lower-limit Register	HIFXAL	R/W	✓	✓	—	0000H
0F FF34H	External MCU Interface Specified Area Upper-limit Register	HIFXAH	R/W	✓	✓	—	0000H

Remark 1. The initial value of the HIFMON register is determined in accordance with the state of the input pins (HIFSYNC and BUS32EN).

(2) External MCU Interface Bus Control Register (HIFBCC)

This register sets advance reading of the external MCU interface.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFBCC	0	0	WRPON	BSTON	0	0	0	RBUFONX	0	0	RBUFON5	0	0	0	RBUFON1	RBUFON0	0F FF00H	0001H
R/W	0	0	R/W	R/W	0	0	0	R/W	0	0	R/W	0	0	0	R/W	R/W		

Bit Position	Bit Name	Description
15–14	—	Reserved (Write 0 when writing. When reading, 0 is read.)
13	WRPON	Selects the type of burst transfer. 0: INCR4 1: WRAP4
12	BSTON	Specifies the method of AHB transfer for advance reading. 0: Single transfer is used. 1: Burst transfer is used.
11–9	—	Reserved (Write 0 when writing. When reading, 0 is read.)
8	RBUFONX	Enables or disables advance reading of the specified area. 0: Advance reading is disabled. 1: Advance reading is enabled.
7–6	—	Reserved (Write 0 when writing. When reading, 0 is read.)
5	RBUFON5	Enables or disables advance reading of the network RAM area.* 0: Advance reading is disabled. 1: Advance reading is enabled.
4–2	—	Reserved (Write 0 when writing. When reading, 0 is read.)
1	RBUFON1	Enables or disables advance reading of the mirror area. 0: Advance reading is disabled. 1: Advance reading is enabled.
0	RBUFON0	Enables or disables advance reading of the data RAM area. 0: Advance reading is disabled. 1: Advance reading is enabled.

Remarks 1. The register is only accessible by the external MCU.

2. Clearing of read buffers requires write access to any of the external MCU interface registers. To prevent erroneous writing to registers, write access to the HIFMON register (read-only register) is recommended.
Values written to the HIFMON register are ignored.

Table 16.8 Address Range for which Advance Reading and Page ROM Reading are Selectable

Target Macro	Address Range		Related Enable Bit	
	MCU Space	Internal AHB Space	Advance Reading	PageROM
(Specified Area)*1	{ XADRH [8:0], 12'hFFF } to { XADRL [8:0], 12'h000 }	Dependent on the specified area	HIFBCC. RBUFONX	HIFPRC. PAGEONX
Mirror Area	*2	*2	HIFBCC. RBUFON1	HIFPRC. PAGEON1
Data RAM	1F FFFFH to 18 0000H	2007 FFFFH to 2000 0000H	HIFBCC. RBUFON0	HIFPRC. PAGEON0

Notes 1. Some areas cannot be read in advance depending on the target macro even if advance reading is enabled.

2. The address range varies depending on the boot mode.
Address range for external MCU boot: 00 0000H to 0B FFFFH
Address range for external serial flash ROM boot: 00 0000H to 0C 2FFFH

(3) External MCU Interface Timing Control Register (HIFBTC)

This register sets timing adjustment of the external MCU interface.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFBTC	0	0	RDDTS1	RDDTS0	0	PASTD2	PASTD1	PASTD0	0	0	RDSTD1	RDSTD0	0	WRSTD2	WRSTD1	WRSTD0	0F FF04H	3737H
R/W	0	0	R/W	R/W	0	R/W	R/W	R/W	0	0	R/W	R/W	0	R/W	R/W	R/W		

Bit Position	Bit Name	Description
15–14	—	Reserved (Write 0 when writing. When reading, 0 is read.)
13–12	RDDTS1, RDDTS0	Sets the waiting time from fixing the HRD signal to de-assertion of the HWAITZ signal. 11: Wait for 3 clock cycles of HCLK. 10: Wait for 2 clock cycles of HCLK. 01: Wait for 1 clock cycle of HCLK. 00: No wait
11	—	Reserved (Write 0 when writing. When reading, 0 is read.)
10–8	PASTD2, PASTD1, PASTD0	Sets the stable waiting time of off-page detection. 111: Wait for 7 clock cycles of HCLK. 110: Wait for 6 clock cycles of HCLK. 101: Wait for 5 clock cycles of HCLK. 100: Wait for 4 clock cycles of HCLK. 011: Wait for 3 clock cycles of HCLK. 010: Wait for 2 clock cycles of HCLK. 001: Wait for 1 clock cycle of HCLK. 000: No wait
7–6	—	Reserved (Write 0 when writing. When reading, 0 is read.)
5–4	RDSTD1, RDSTD0	Sets the timing for detecting the start of read operation by the HRDZ signal. Adjusts the setup time of address input signals for falling edges of the HRDZ signal. 11: Delay by 3 HCLK clock cycles from the detection of the falling edge after synchronization 10: Delay by 2 HCLK clock cycles from the detection of the falling edge after synchronization 01: Delay by 1 HCLK clock cycle from the detection of the falling edge after synchronization 00: Simultaneous with the detection of the falling edge after synchronization
3	—	Reserved (Write 0 when writing. When reading, 0 is read.)
2–0	WRSTD2, WRSTD1, WRSTD0	Sets the timing for detecting the start of write operation by the HWRSTBZ signal. Adjusts the setup time of address input signals and write data input signals for falling edges of the HWRSTBZ signal. 111: Delay by 7 HCLK clock cycles from the detection of the falling edge after synchronization 110: Delay by 6 HCLK clock cycles from the detection of the falling edge after synchronization 101: Delay by 5 HCLK clock cycles from the detection of the falling edge after synchronization 100: Delay by 4 HCLK clock cycles from the detection of the falling edge after synchronization 011: Delay by 3 HCLK clock cycles from the detection of the falling edge after synchronization 010: Delay by 2 HCLK clock cycles from the detection of the falling edge after synchronization 001: Delay by 1 HCLK clock cycle from the detection of the falling edge after synchronization 000: Simultaneous with the detection of the falling edge after synchronization

Remark: The register is only accessible by the external MCU.

(4) External MCU Interface Page ROM Control Register (HIFPRC)

This register sets operation for access to the page ROM via the external MCU interface.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFPRC	0	0	0	PAGESZ	0	0	0	PAGEONX	0	0	PAGEON5	0	0	0	PAGEON1	PAGEON0	0F FF08H	0000H
R/W	0	0	0	R/W	0	0	0	R/W	0	0	R/W	0	0	0	R/W	R/W		

Bit Position	Bit Name	Description
15–13	—	Reserved (Write 0 when writing. When reading, 0 is read.)
12	PAGESZ	Sets the page size for page ROM reading.*1 0: 8-byte 1: 16-byte Note that access across 16-byte boundaries is prohibited.
11–9	—	Reserved (Write 0 when writing. When reading, 0 is read.)
8	PAGEONX	Sets page ROM reading of the specified area. 0: SRAM reading 1: Page ROM reading
7–6	—	Reserved (Write 0 when writing. When reading, 0 is read.)
5	PAGEON5	Sets page ROM reading of the network RAM area. 0: SRAM reading 1: Page ROM reading
4–2	—	Reserved (Write 0 when writing. When reading, 0 is read.)
1	PAGEON1	Sets page ROM reading of the mirror area. 0: SRAM reading 1: Page ROM reading
0	PAGEON0	Sets page ROM reading of the data RAM area. 0: SRAM reading 1: Page ROM reading

Note 1. The page size to be set in the PAGESZ bit must match the page size setting of the external MCU.

Remark: The register is only accessible by the external MCU.

(5) External MCU Interface Interrupt Request Control Register (HIFIRC)

This register sets error interrupt output to the external MCU.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFIRC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ERR RSP	0F FF0CH	0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W		

Bit Position	Bit Name	Description
15–1	—	Reserved (Write 0 when writing. When reading, 0 is read.)
0	ERRRSP	This bit is set to 1 on reception of an error response from an internal slave device. This bit is cleared to 0 by writing 0 to it. Writing 1 to this bit has no effect. While the setting of this bit is 1, the low level is output to the interrupt request signal HERROUTZ. 0: No error response 1: Error response

Remarks 1. The register is only accessible by the external MCU.

2. While the setting of the ERRRSP bit is 1, the external MCU I/F error source registers (HIFECR0, HIFECR1) are not updated even if a new error response was generated. The first error information is held in the external MCU I/F error source register.

(6) External MCU Interface Error Source Register 0 (HIFECR0)

When an error response is returned from an internal resource at the time of access by the external MCU, the address at which an error occurred is stored in the HIFECR0 register.
If a new error response is generated while the ERRRSP bit of the external MCU interface interrupt request control register (HIFIRC) is 1, that address formation is not stored.

HIFECR0_W	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	ERRAD31	ERRAD30	ERRAD29	ERRAD28	ERRAD27	ERRAD26	ERRAD25	ERRAD24	ERRAD23	ERRAD22	ERRAD21	ERRAD20	ERRAD19	ERRAD18	ERRAD17	ERRAD16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

HIFECR0_W	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	ERRAD15	ERRAD14	ERRAD13	ERRAD12	ERRAD11	ERRAD10	ERRAD9	ERRAD8	ERRAD7	ERRAD6	ERRAD5	ERRAD4	ERRAD3	ERRAD2	ERRAD1	ERRAD0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address 0F FF10H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31–0	ERRAD31–ERRAD0	These bits hold the address (32-bit address) at which an error occurred.

Cautions 1. When two or more errors occur, the address where the first error occurred is stored.

2. The setting of this register is not effective while the ERRRSP bit of the HIFIRC register is 0.

3. Clearing the ERRRSP bit of the HIFIRC register leads to updating of the setting of this register in response to the detection of a next error response. In interrupt processing, this register must be referenced before clearing the ERRRSP bit.

4. This register is only readable when the bus width of the external MCU interface is 32 bits.

Remark: The register is only accessible by the external MCU.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFECR0 _H0	ERR AD15	ERR AD14	ERR AD13	ERR AD12	ERR AD11	ERR AD10	ERR AD9	ERR AD8	ERR AD7	ERR AD6	ERR AD5	ERR AD4	ERR AD3	ERR AD2	ERR AD1	ERR AD0	0F FF10H	0000H
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		

Bit Position	Bit Name	Description
15–0	ERRAD15–ERRAD0	These bits hold the address (lower-order 16-bit address) at which an error occurred.

Cautions 1. When two or more errors occurred, the first access information is stored.

2. The setting of this register is not effective while the ERRRSP bit of the HIFIRC register is 0.

3. Clearing the ERRRSP bit of the HIFIRC register leads to updating of the setting of this register in response to the detection of a next error response. In interrupt processing, this register must be referenced before clearing the ERRRSP bit.

4. This register is only readable when the bus width of the external MCU interface is 16 bits.

Remark: The register is only accessible by the external MCU.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFECR0 _H1	ERR AD31	ERR AD30	ERR AD29	ERR AD28	ERR AD27	ERR AD26	ERR AD25	ERR AD24	ERR AD23	ERR AD22	ERR AD21	ERR AD20	ERR AD19	ERR AD18	ERR AD17	ERR AD16	0F FF12H	0000H
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		

Bit Position	Bit Name	Description
15–0	ERRAD31–ERRAD16	These bits hold the address (higher-order 16-bit address) at which an error occurred.

Cautions 1. When two or more errors occurred, the first access information is stored.

2. The setting of this register is not effective while the ERRRSP bit of the HIFIRC register is 0.

3. Clearing the ERRRSP bit of the HIFIRC register leads to updating of the setting of this register in response to the detection of a next error response. In interrupt processing, this register must be referenced before clearing the ERRRSP bit.

4. This register is only readable when the bus width of the external MCU interface is 16 bits.

Remark: The register is only accessible by the external MCU

(7) External MCU Interface Error Source Register 1 (HIFECR1)

When an error response is returned from an internal resource at the time of access by the external MCU, the information on whether this was caused by reading or writing and the access size when the error occurred is stored in the HIFECR1 register.
If a new error response is generated while the ERRRSP bit of the external MCU interface interrupt request control register (HIFIRC) is 1, that formation is not stored.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFECR1	0	0	0	0	0	0	0	0	0	0	0	0	ERRWR	ERRSZ2	ERRSZ1	ERRSZ0	0F FF14H	0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	R	R	R	R		

Bit Position	Bit Name	Description
15–4	—	Reserved (When reading, 0 is read.)
3	ERRWR	The information on whether an error was caused by reading or writing is stored. 0: Reading 1: Writing
2–0	ERRSZ2, ERRSZ1, ERRSZ0	The transfer size (access width) when the error occurred is stored. 000: 8-bit 001: 16-bit 010: 32-bit Other than the above: Prohibited access width

Cautions 1. When two or more errors occurred, the first access information is stored.

2. The setting of this register is not effective while the ERRRSP bit of the HIFIRC register is 0.

3. Clearing the ERRRSP bit of the HIFIRC register leads to updating of the setting of this register in response to the detection of a next error response. In interrupt processing, this register must be referenced before clearing the ERRRSP bit.

Remark: The register is only accessible by the external MCU

(8) External MCU Interface Monitor Register (HIFMON)

This register monitors input pins for the external MCU interface and their internal states.
This register can also be read during a reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFMON	0	0	0	0	0	0	0	1	0	0	0	MCMICONSET	HIFSYNC	BUS32EN	0	HIFRDY	0F FF20H	Pin states
R/W	0	0	0	0	0	0	0	1	0	0	0	R	R	R	0	R		

Bit Position	Bit Name	Description
15–9	—	Reserved (When reading, 0 is read.)
8	—	Reserved (When reading, 1 is read.)
7–5	—	Reserved (When reading, 0 is read.)
4	MCMICONSET	Indicate the state of the memory map of the external MCU interface. 0: Memory map space when CC-Link IE TSN is used 1: Memory map space when CC-Link IE Field is used
3	HIFSYNC	Indicate the state of the HIFSYNC pin. 0: Low level (asynchronous mode) 1: High level (synchronous mode)
2	BUS32EN	Indicate the state of the BUS32EN pin. 0: Low level (16 bits bus width) 1: High level (32 bits bus width)
1	—	Reserved (When reading, 0 is read.)
0	HIFRDY	Indicate the internal initialization state of the external MCU interface. 0: Under internal initialization 1: Completion of internal initialization

Note . During internal initialization processing (the HIFRDY bit is "0"), any access other than reading of the external MCU interface registers area is prohibited.
The HIFRDY bit is set to "0" at the start of a reset and changed to "1" when internal initialization processing is completed after release from the reset state, which allows access to internal resources of R-IN32M4-CL3.

Remark: The register is only accessible by the external MCU.

(9) External MCU Interface Specified Area Lower-limit Register (HIFXAL)

This register holds the lower-limit address of the specified area to be set in the MCU address space. The specified area is set by the combination of the settings of the external MCU interface specified area lower-limit register (HIFXAL) and the external MCU interface specified area upper-limit register (HIFXAH).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFXAL	0	0	0	0	0	0	0	XADRL8	XADRL7	XADRL6	XADRL5	XADRL4	XADRL3	XADRL2	XADRL1	XADRL0	0F FF30H	0000H
R/W	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Description
15–9	—	Reserved (Write 0 when writing. When reading, 0 is read.)
8–0	XADRL8–XADRL0	Sets the lower-limit address of the specified area to be set in the MCU address space. The lower-limit address to be set is HA[20:0] = { XADRL[8:0], 12’h000 }.

Caution: Before setting this register, set the HIFBCC.RBUFONX bit to 0.

Remark: This register can only be read or written by the external MCU.
The range from HA[20:0] = { XADRL[8:0], 12’h000 } to { XADRH[8:0], 12’hFFF } is the specified area.
If the values of XADRL[8:0] and XADRH[8:0] are equal, the specified 4-Kbyte area is selected.
If the value of XADRL[8:0] is greater than the value of XADRH[8:0], any existing specified area is lost and a new specified area is not secured.

(10) External MCU Interface Specified Area Upper-limit Register (HIFXAH)

This register holds the upper-limit address of the specified area to be set in the MCU address space. The specified area is set by the combination of the settings of the external MCU interface specified area lower-limit register (HIFXAL) and the external MCU interface specified area upper-limit register (HIFXAH).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFXAH	0	0	0	0	0	0	0	XADRH8	XADRH7	XADRH6	XADRH5	XADRH4	XADRH3	XADRH2	XADRH1	XADRH0	0F FF34H	0000H
R/W	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Description
15–9	—	Reserved (Write 0 when writing. When reading, 0 is read.)
8–0	XADRH8– XADRH0	Sets the upper-limit address of the specified area to be set in the MCU address space. The upper-limit address to be set is HA[20:0] = { XADRH[8:0], 12’hFFF }.

Caution: Before setting this register, set the HIFBCC.RBUFONX bit to 0.

Remark: This register can only be read or written by the external MCU.
The range from HA[20:0] = { XADRL[8:0], 12’h000 } to { XADRH[8:0], 12’hFFF } is the specified area.
If the values of XADRL[8:0] and XADRH[8:0] are equal, the specified 4-Kbyte area is selected.
If the value of XADRL[8:0] is greater than the value of XADRH[8:0], any existing specified area is lost and a new specified area is not secured.

16.2.6 Precautions

Precautions regarding the use of the external MCU interface in R-IN32M4-CL3 are described below.

Table 16.9 Register Settings for Each Area and Access Method

Area	Register setting		Access method		
	HIFPRC. PAGEONn	HIFBCC. RBUFONn	pageROM	SRAM	
			Read	Read	Write
Area not to be buffered	—	—	Setting prohibited*	OK	OK
Area to be buffered	0	0	Setting prohibited*	OK	OK
	0	1	Setting prohibited*	OK	OK
	1	0	Setting prohibited*	OK	OK
	1	1	OK	OK	OK

Note . Attempted access via the external MCU interface may lead to a deadlock.

16.3 Synchronous Burst Transfer supported MCU Connection Mode

High-level input to the MEMCSEL pin places the chip in synchronous burst transfer supported MCU connection mode. In this mode, clocked single transfer and burst transfer are supported.

16.3.1 Function overview

- Interface system
  - Single transfer (for reading and writing)
  - Burst transfer (for reading and writing)

- Synchronous relationship
  - HBUSCLK synchronous (50 MHz maximum)

- Bus width
  - 32 bits / 16 bits (Set with the BUS32EN pin.)

- Transfer data size
  - 32 bits / 16 bits / 8 bits

- Write buffer: 8 stages

- Read buffer: Advance reading of maximum 32 bytes is supported.

- Multiplexing of addresses and data
  - Multiplexing of addresses and data
  - Separation of addresses and data

- Checking of each state
  - States of the HIFSYNC and BUS32EN pins*1

Caution 1. In synchronous burst transfer supported MCU connection mode, the internal reset state cannot be checked by reading a register because the external MCU interface cannot be accessed until the internal reset signal is de-asserted.

16.3.2 Selecting Synchronous Burst Transfer supported MCU Connection Mode

In synchronous burst transfer supported MCU connection mode, the width of the external data bus is selected by the input of a signal to the BUS32EN pin, and multiplexing of addresses and data is selected by the level on the ADMUXMODE pin.

Table 16.10 Operating Mode Settings

Mode Setting Pins				Operating Mode
MEMCSEL	BUS32EN	HIFSYNC	ADMUXMODE	
H	L	L	—	Setting prohibited
		H	L	16-bit word synchronous SRAM, address/data separation
			H	16-bit word synchronous SRAM, address/data multiplexing
	H	L	—	Setting prohibited
		H	L	16-bit word synchronous SRAM, address/data separation
			H	16-bit word synchronous SRAM, address/data multiplexing

Caution: In synchronous burst transfer supported MCU connection mode, the asynchronous interface cannot be selected.

Table 16.11 Address Input in Synchronous Burst Transfer supported MCU Connection Mode

Setting Pins			Operating Mode	Byte Address [20:0] Acquisition Destination (MSB, …, LSB)
ADMUXMODE	BUS32EN	HIFSYNC		
L	L	—	AD separation 16-bit data bus Word address	{ HA[20:1], 1'b0 }
L	H	—	AD separation 32-bit data bus Word address	{ HA[19:1], 2'b00 }
H	L	L	Setting prohibited	—
H	L	H	AD multiplexing 16-bit data bus Word address	{ HA[20:17], HD[15:0], 1'b0 }
H	H	L	Setting prohibited	—
H	H	H	AD multiplexing 32-bit data bus Word address	{ HD[18:0], 2'b00 }

Remark: In this section and section 16.3.5, the HWDATA and HRDATA are aliases for the HD.
HWDATA: HD output signals of the external MCU
HRDATA: HD input signal of the external MCU

16.3.3 Write Status Mode and Write Strobe Mode

There are two types of write operations in synchronous SRAM type transfer mode: write status mode and write strobe mode. Either of these two operating modes is selected every bus cycle by the level on the HWRSTBZ pin to be sampled while the Low level is input on the HBCYSTZ pin.
When the HWRSTBZ pin is at the Low level while the Low level is input on the HBCYSTZ pin, write status mode is entered. In write status mode, the current bus cycle ends (the High level is sampled from the HCSZ signal) or the write bus cycle continues until the next bus cycle starts (the Low level is sampled from the HBCYSTZ pin).
When the HWRSTBZ pin is at the High level while the Low level is input on the HBCYSTZ pin and then the HWRSTBZ pin turns Low level, write strobe mode is entered. In write strobe mode, the write bus cycle continues until the current bus cycle ends (the High level is sampled from the HWRSTBZ pin).

16.3.4 Synchronous Burst Transfer Control Registers

(1) Register overview

The synchronous burst transfer control registers of the external MCU interface are accessible by the MCU externally connected to R-IN32M4-CL3.

Table 16.12 Synchronous Burst Transfer Control Registers of the External MCU Interface

Address	Register Name	Symbol	R/W	Access Size [bit]			Initial Value
				8	16	32	
0F FF80H	External MCU Interface Synchronous SRAM Control Register 0	HIFEXT0	R/W	✓	✓	—	0000H
0F FF84H	External MCU Interface Synchronous SRAM Control Register 1	HIFEXT1	R/W	✓	✓	—	0202H

Remark: The synchronous burst transfer control registers can only be accessed when the High-level is input on the MEMCSEL pin.

(2) External MCU Interface Synchronous Burst Transfer Control Register 0 (HIFEXT0)

This register sets operation for burst transfer to and from the external MCU.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFEXT0	MOD TRN	0	0	0	0	0	CND WEO	0	0	0	KES WTO	KES DTO	KES AVI	KES DTI	0	KES SBI	0F FF80H	0000H
R/W	R/W	0	0	0	0	0	R/W	0	0	0	R/W	R/W	R/W	R/W	0	R/W		

Bit Position	Bit Name	Description
15	MODTRN	Transfer mode selection 0: Single transfer only 1: Burst transfer is supported.
14–10	—	Reserved (Write 0 when writing. When reading, 0 is read.)
9	CNDWEO	WAIT release timing selection 0: Simultaneous with data 1: Precedes the data by one clock cycle.
8–6	—	Reserved (Write 0 when writing. When reading, 0 is read.)
5	KESWTO	Effective edge selection for HWAITZ output 0: Rising edge 1: Falling edge
4	KESDTO	Effective edge selection for data output 0: Rising edge 1: Falling edge
3	KESAVI	Effective edge selection for address input 0: Rising edge 1: Falling edge
2	KESDTI	Effective edge selection for data input 0: Rising edge 1: Falling edge
1	—	Reserved (Write 0 when writing. When reading, 0 is read.)
0	KESSBI	Effective edge selection for strobe signal input (HRDZ, HWRSTBZ) 0: Rising edge 1: Falling edge

Remarks 1. This register can be accessed only by the external MCU.

2. This register can only be accessed when the High-level is input on the MEMCSEL pin.

(3) External MCU Interface Synchronous Burst Transfer Control Register 1 (HIFEXT1)

This register sets operation for burst transfer to and from the external MCU.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
HIFEXT1	0	0	0	0	DLY RA3	DLY RA2	DLY RA1	DLY RA0	0	0	0	0	DLY WA3	DLY WA2	DLY WA1	DLY WA0	0F FF84H	0202H
R/W	0	0	0	0	R/W	R/W	R/W	R/W	0	0	0	0	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Description
15–12	—	Reserved (Write 0 when writing. When reading, 0 is read.)
11–8	DLYRA3– DLYRA0	Minimum time from the last input of the Low level on the BCYSTZ pin to the point where read data can be acquired. (trc) 0000: 3 1000: 11 0001: 4 1001: 12 0010: 5 1010: 13 0011: 6 1011: 14 0100: 7 1100: 15 0101: 8 1101: 16 0110: 9 1110: 17 0111: 10 1111: 18
7–4	—	Reserved (Write 0 when writing. When reading, 0 is read.)
3–0	DLYWA3– DLYWA0	Minimum time from the last input of the Low level on the BCYSTZ pin to the point where write data is received. (twc) 0000: 3* 1000: 9 0001: 3* 1001: 10 0010: 3 1010: 11 0011: 4 1011: 12 0100: 5 1100: 13 0101: 6 1101: 14 0110: 7 1110: 15 0111: 8 1111: 16

Remarks 1. This register can be accessed only by the external MCU.

2. This register can only be accessed when the High-level is input on the MEMCSEL pin.

16.3.5 Basic Operation Timing in Synchronous Burst Transfer Supported MCU Connection Mode

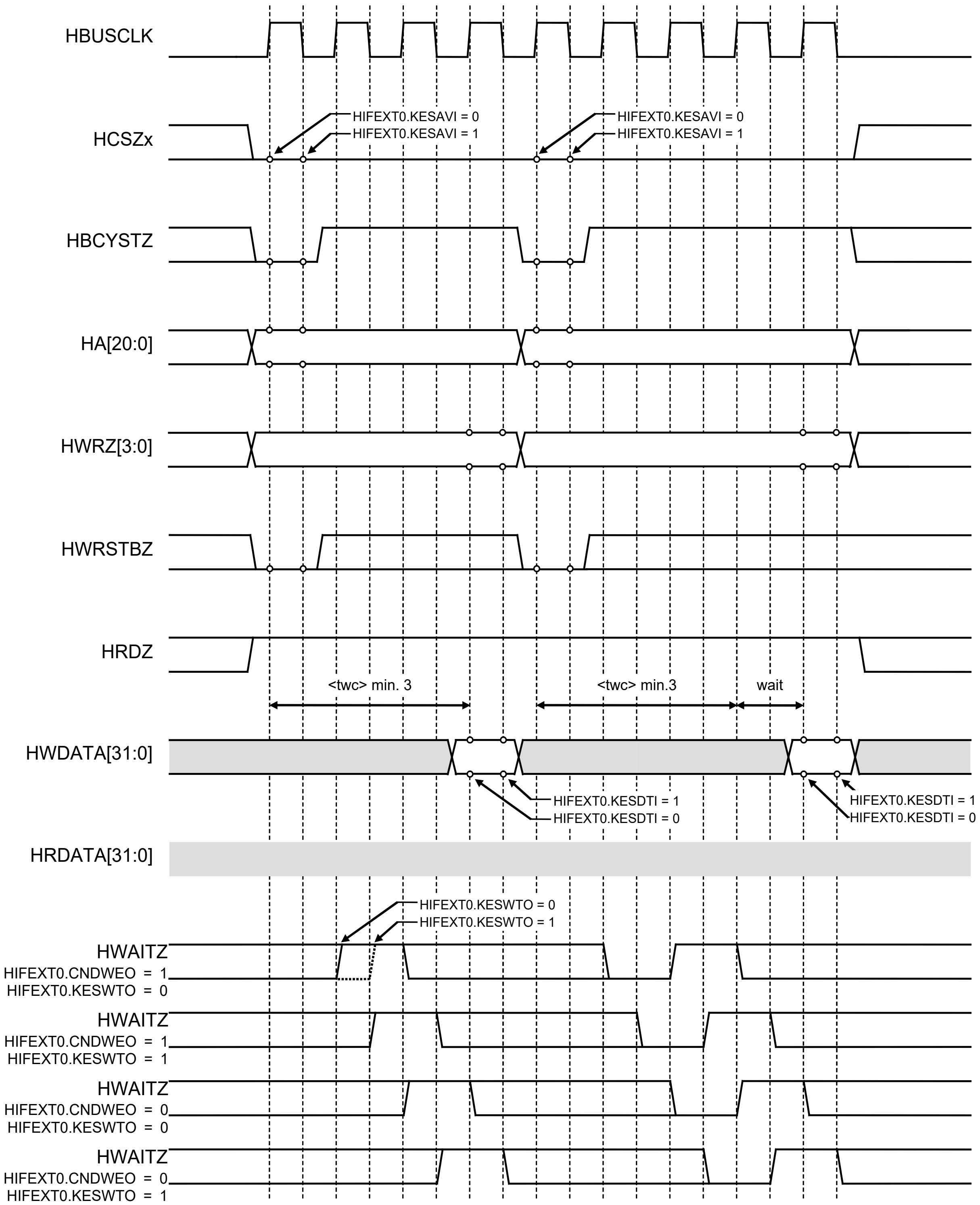


Figure 16.9 Writing by a Synchronous Burst Transfer Supporting MCU (Single Transfer, AD Separation, Write Status)

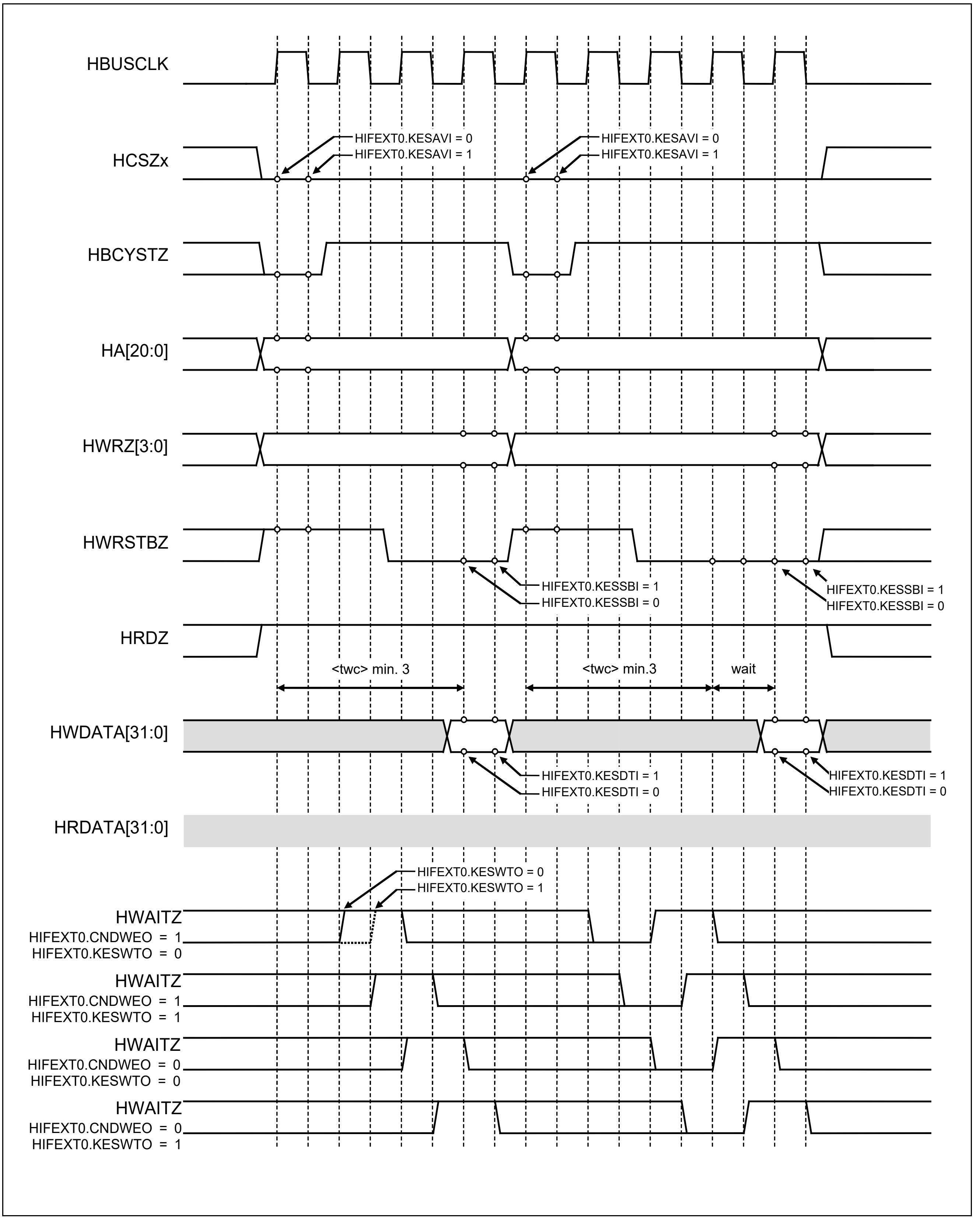


Figure 16.10 Writing by a Synchronous Burst Transfer Supporting MCU (Single Transfer, AD Separation, Write Strobe)

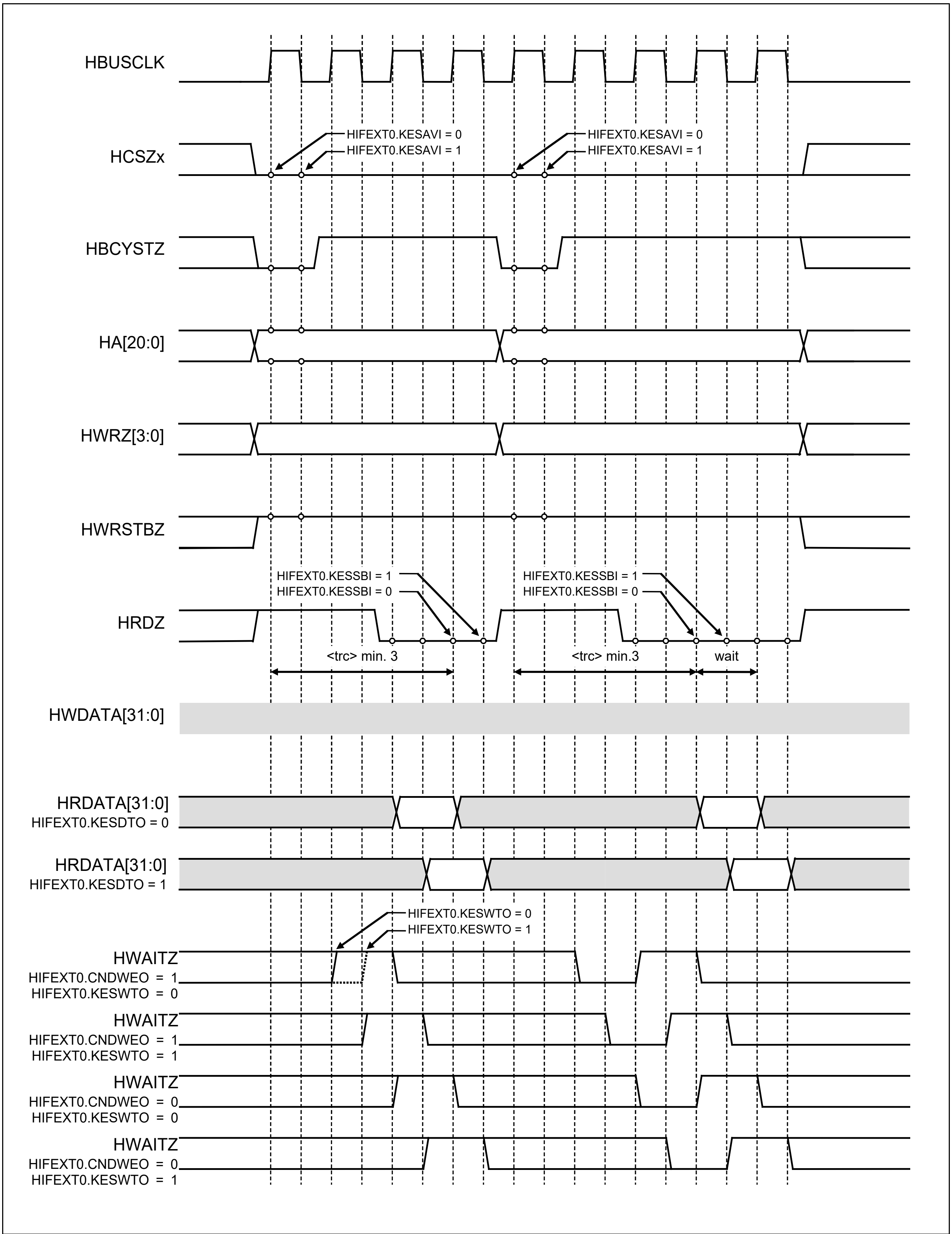


Figure 16.11 Reading by a Synchronous Burst Transfer Supporting MCU (Single Transfer, AD Separation)

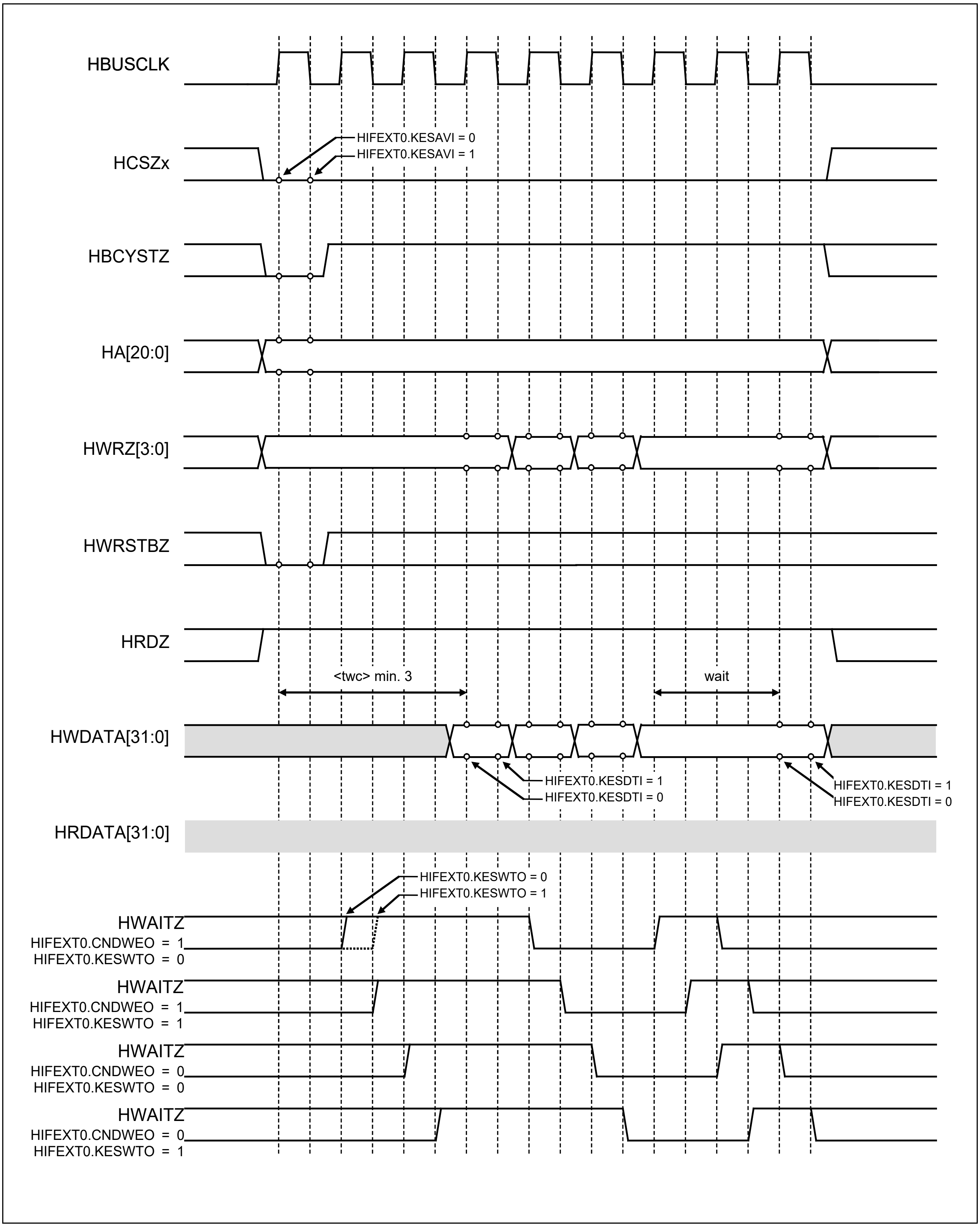


Figure 16.12 Writing by a Synchronous Burst Transfer Supporting MCU (Burst Transfer, AD Separation, Write Status)

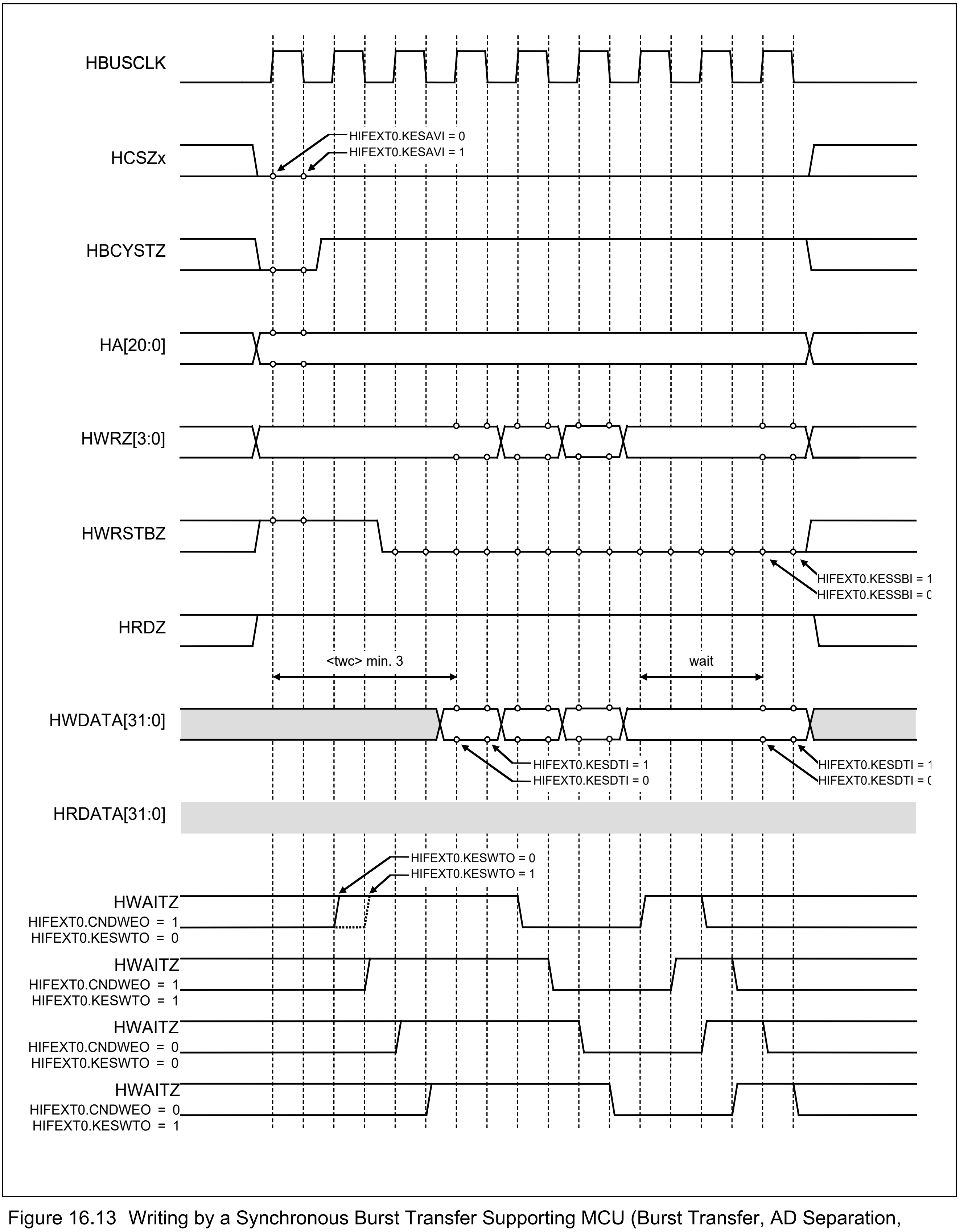


Figure 16.13 Writing by a Synchronous Burst Transfer Supporting MCU (Burst Transfer, AD Separation, Write Strobe)

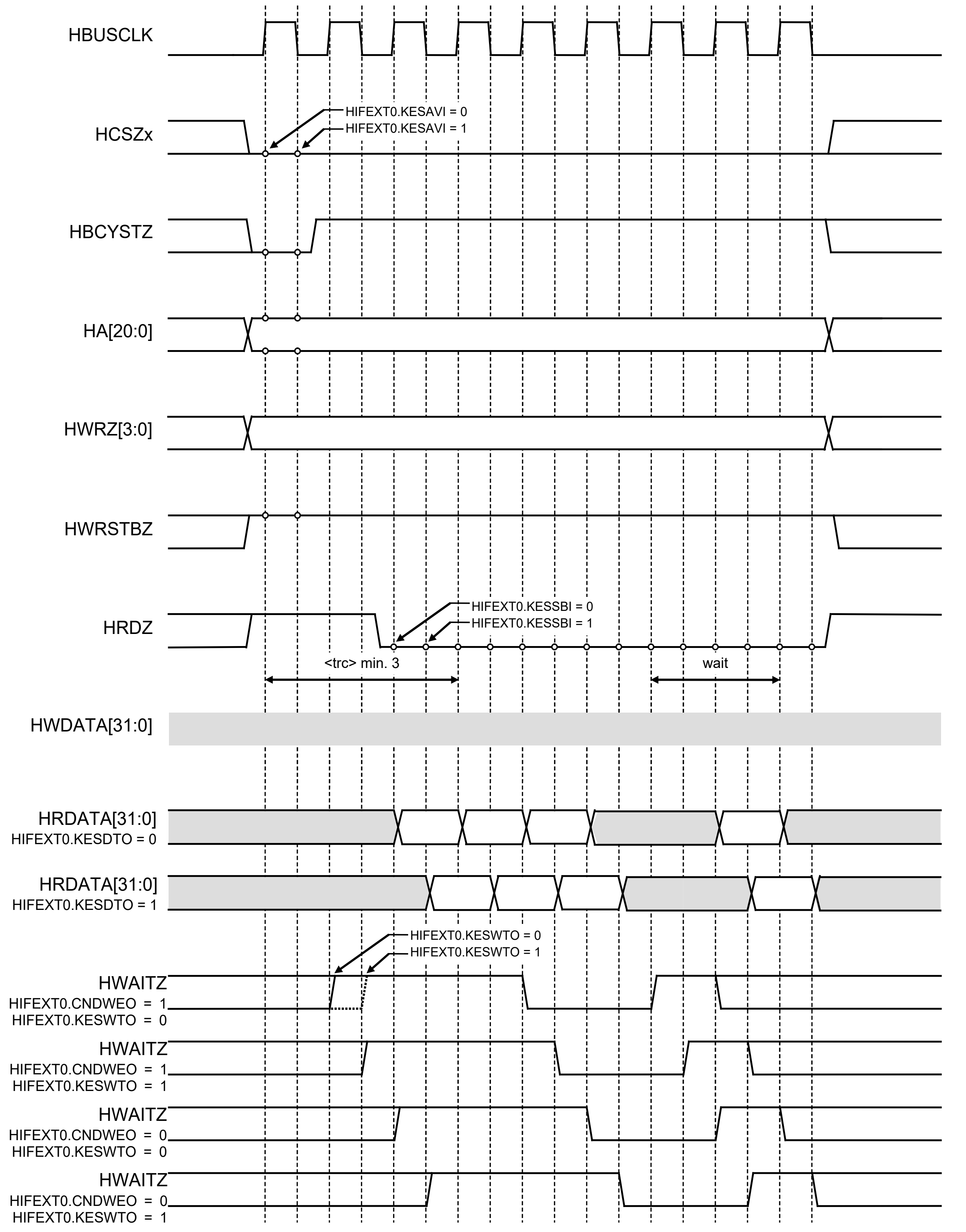


Figure 16.14 Reading by a Synchronous Burst Transfer Supporting MCU (Burst Transfer, AD Separation)

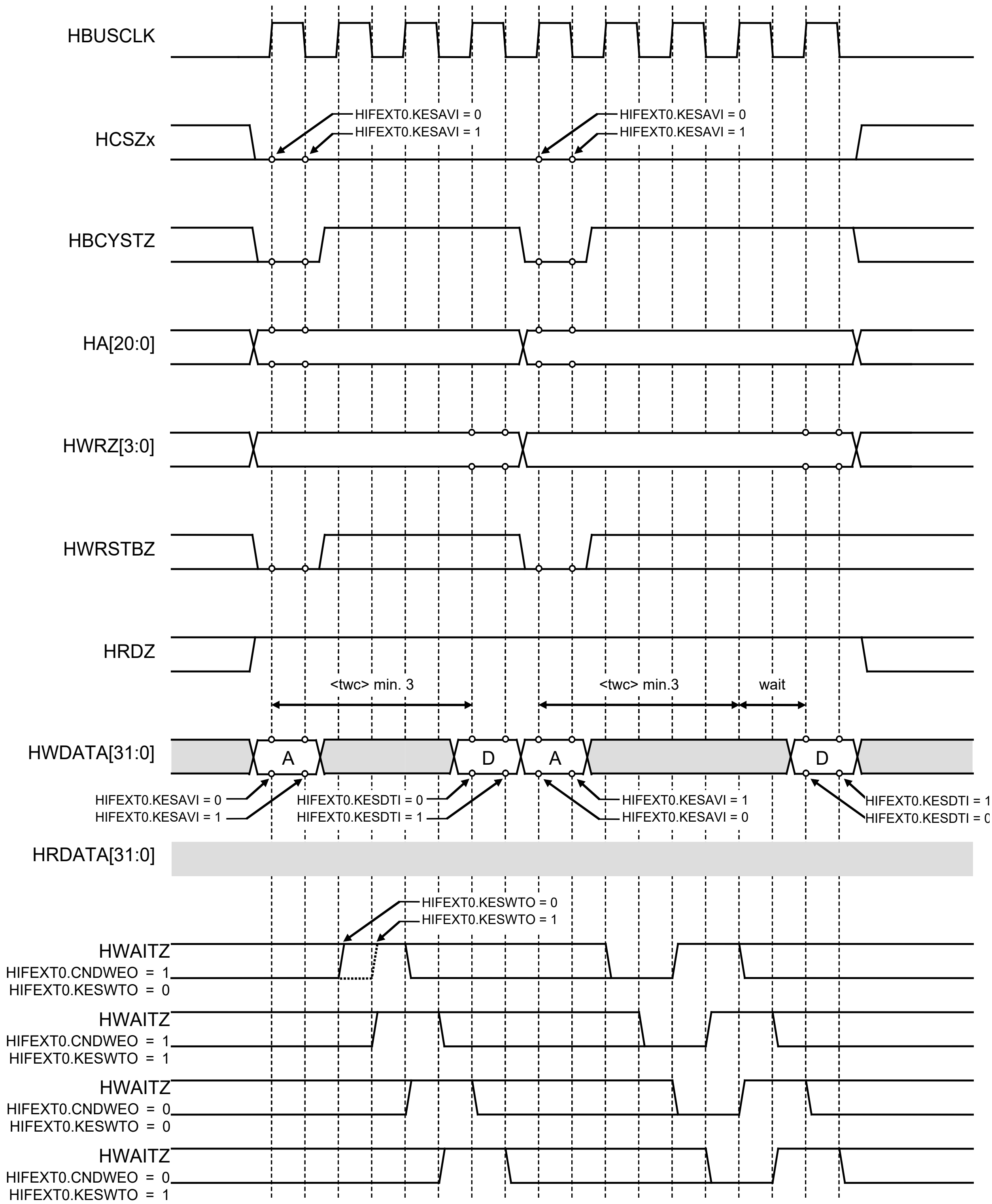


Figure 16.15 Writing by a Synchronous Burst Transfer Supporting MCU (Single Transfer, AD Multiplexing, Write Status)

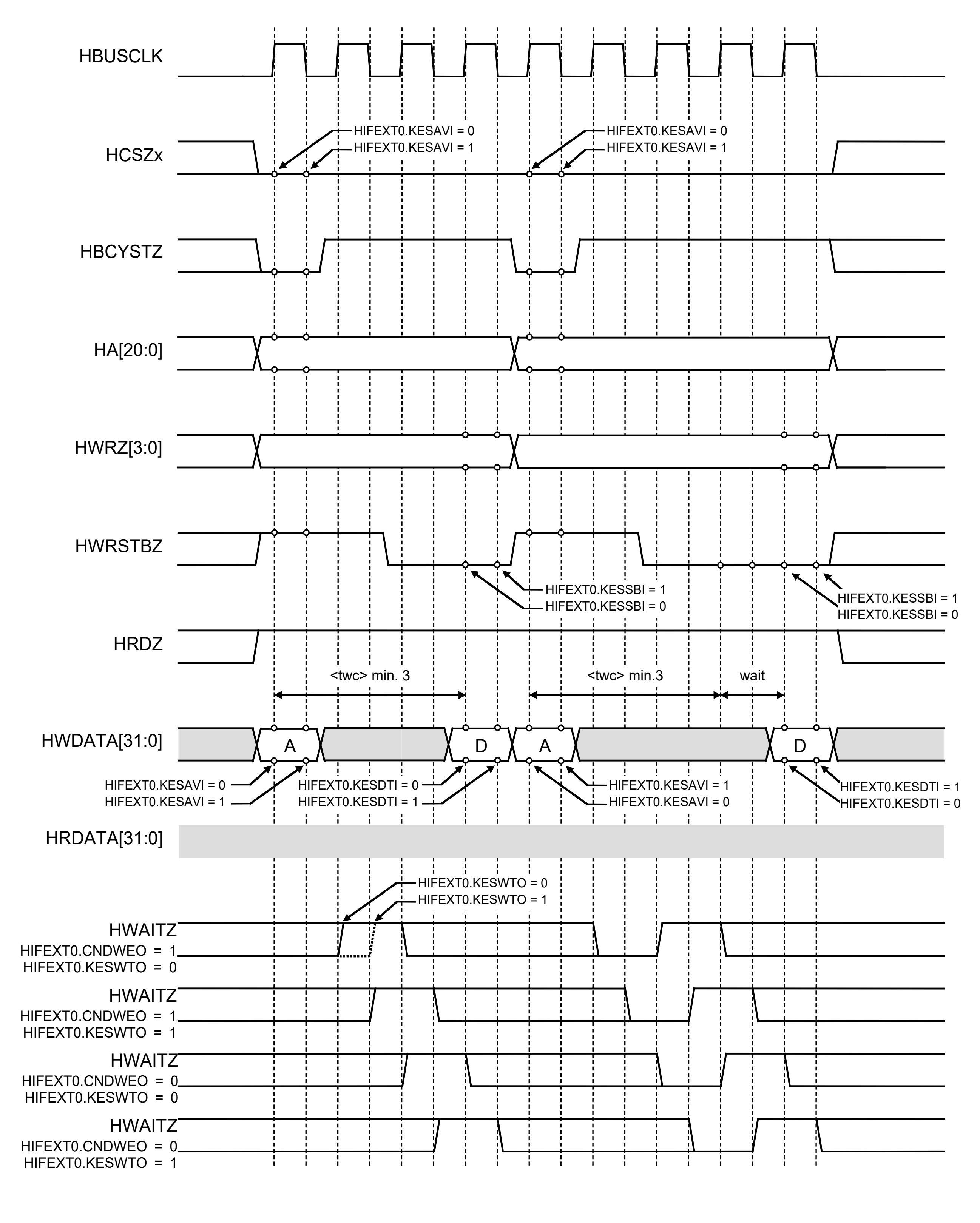


Figure 16.16 Writing by a Synchronous Burst Transfer Supporting MCU (Single Transfer, AD Multiplexing, Write Strobe)

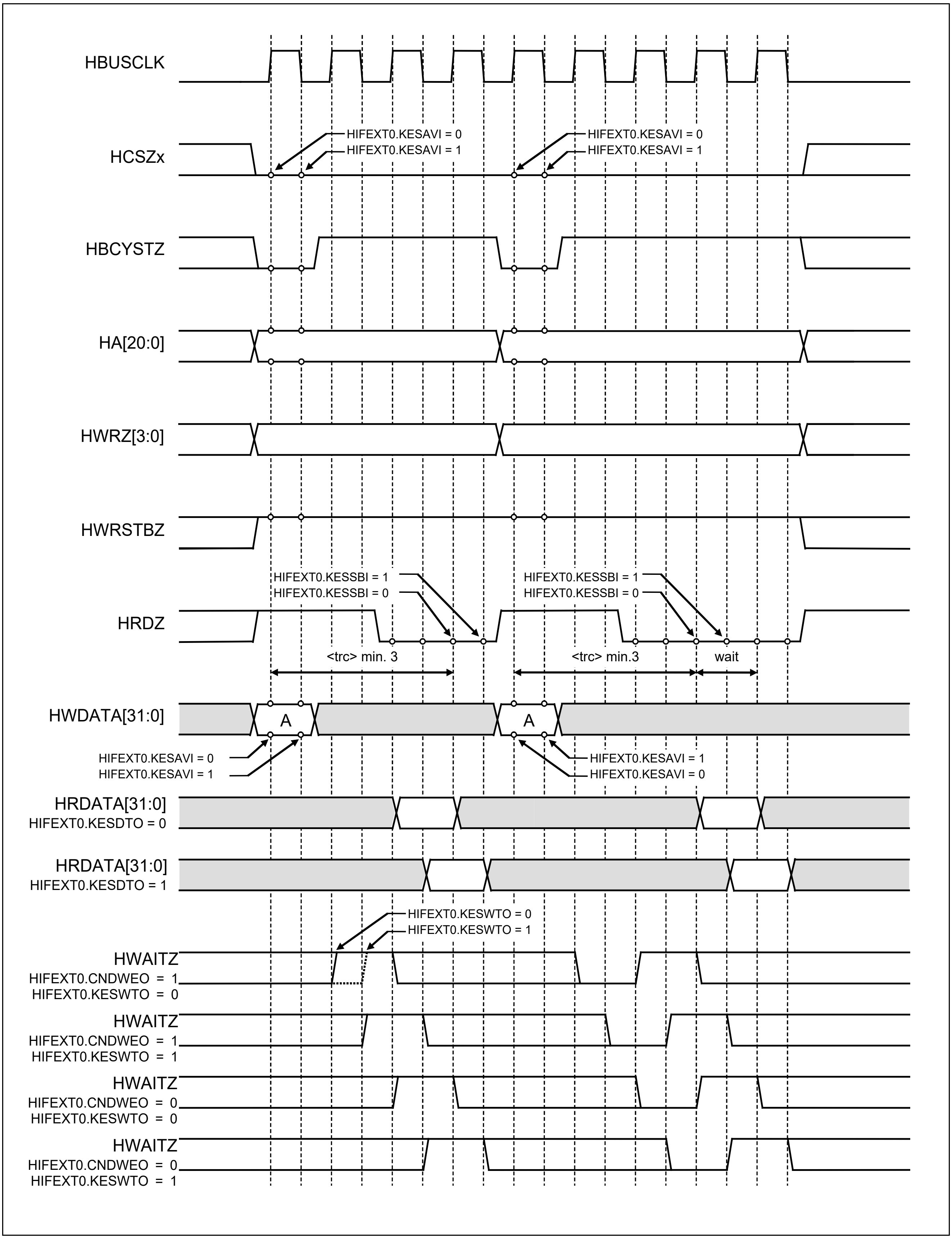


Figure 16.17 Reading by a Synchronous Burst Transfer Supporting MCU (Single Transfer, AD Multiplexing)

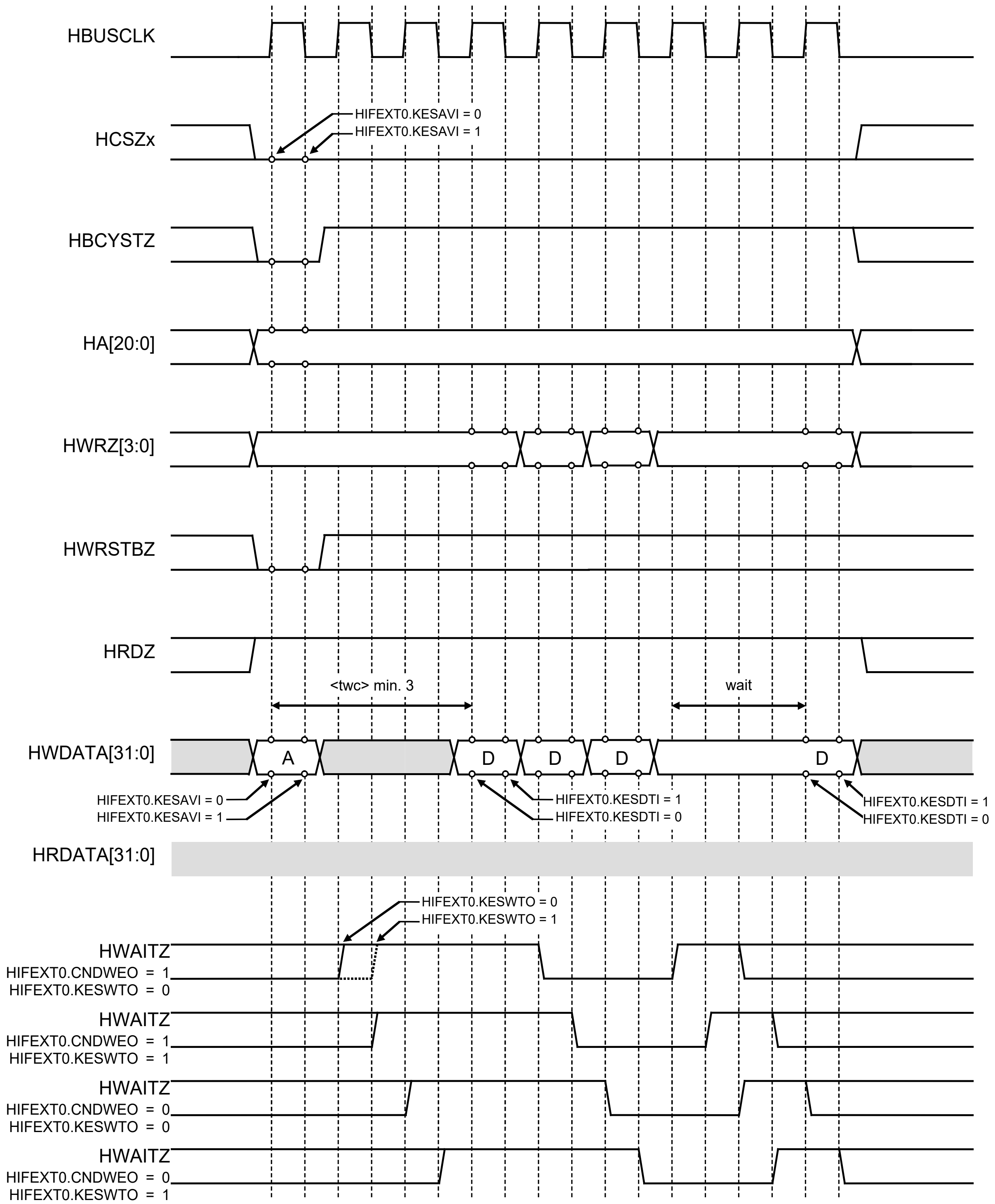


Figure 16.18 Writing by a Synchronous Burst Transfer Supporting MCU (Burst Transfer, AD Multiplexing, Write Status)

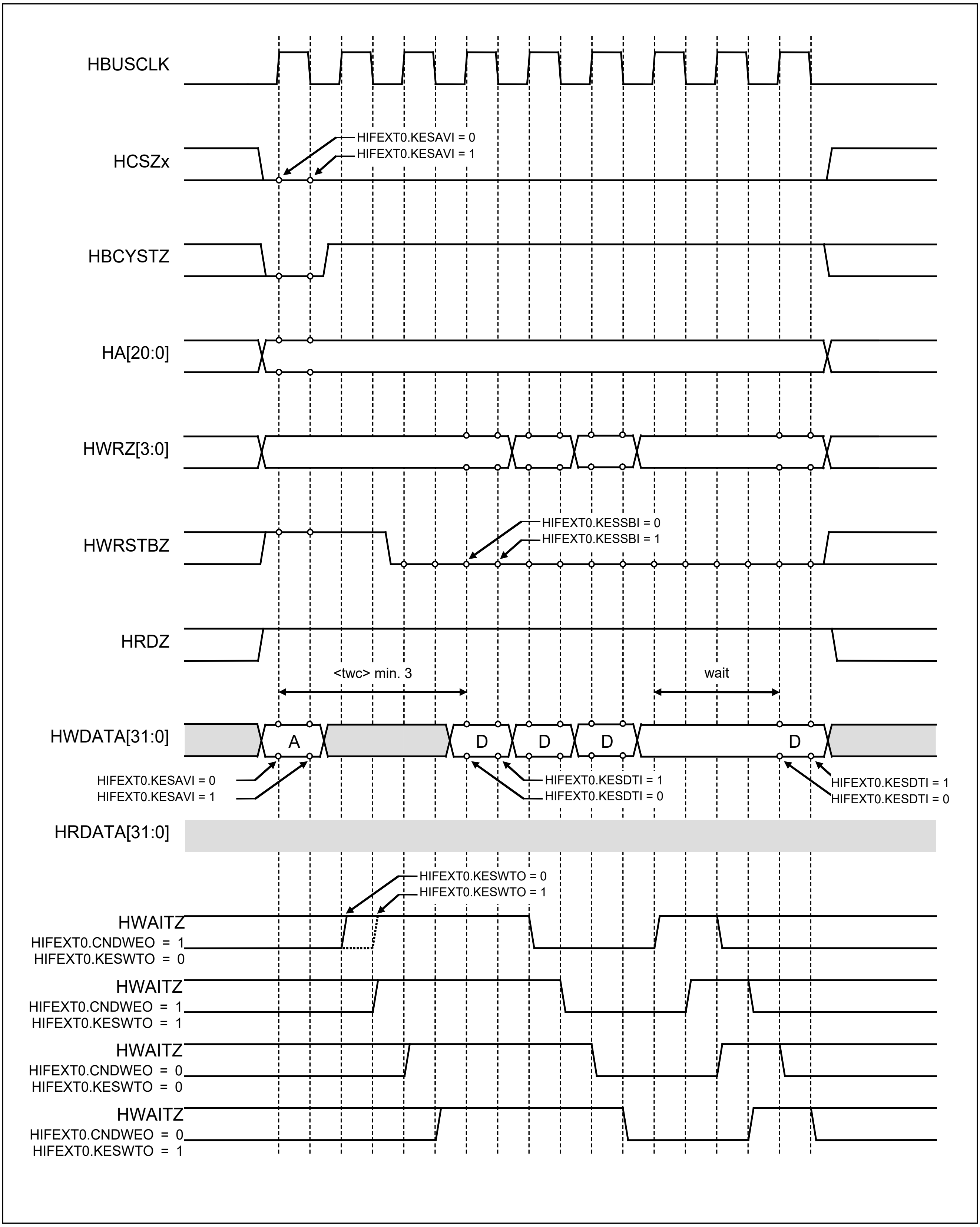


Figure 16.19 Writing by a Synchronous Burst Transfer Supporting MCU (Burst Transfer, AD Multiplexing, Write Strobe)

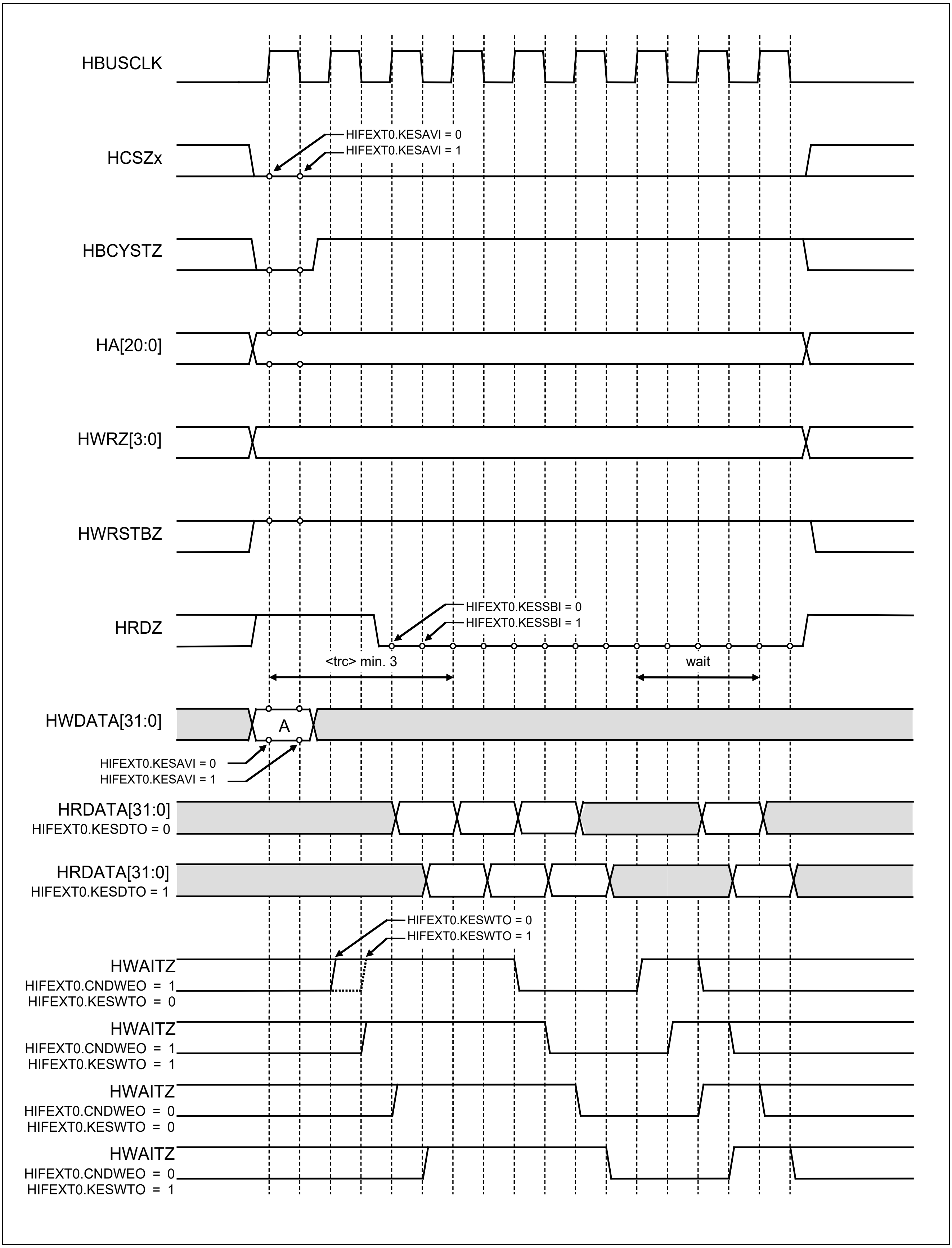


Figure 16.20 Reading by a Synchronous Burst Transfer Supporting MCU (Burst Transfer, AD Multiplexing)

16.3.6 Precautions

Precautions on the usage of the MCU connection mode of the synchronous burst transfer supporting are described below.

(1) Register settings for each area and available access methods

Table 16.13 Register Settings for Each Area Selected by the Level on the HPGCSZ Pin and Method of Access

Area	Register Setting			Access Method to HPGCSZ				Remark
	HIFEXT0. MODTRN	HIFPRC. PAGEONn	HIFBCC. RBUFONn	Burst Transfer W	Burst Transfer R	Single Transfer W	Single Transfer R	
Area not to be buffered	—	—	—	Prohibited	Prohibited	OK	OK	Single area
Area to be buffered	0	—	—	Prohibited	Prohibited	Prohibited	Prohibited	
	1	0	—	Prohibited	Prohibited	Prohibited	Prohibited	
	1	1	0	Prohibited	Prohibited	Prohibited	Prohibited	
	1	1	1	OK	OK	Prohibited	Prohibited	

Remark: For the burst transfer, be sure to set HIFEXT0.MODTRN = 1, HIFPRC.PAGEONn = 1, and HIFBCC.RUFONn = 1.

Table 16.14 Register Settings for Each Area Selected by the Level on the HCSZ Pin and Method of Access

Area	Register Setting			Access Method to HCSZ				Remark
	HIFEXT0. MODTRN	HIFPRC. PAGEONn	HIFBCC. RBUFONx	Burst Transfer W	Burst Transfer R	Single Transfer W	Single Transfer R	
All areas	—	—	—	Prohibited	Prohibited	OK	OK	

(2) Prohibition of burst transfer that spans boundaries of spaces and the advance reading area

Burst transfer that spans boundaries of the register space, SRAM space, or AHB space, or the boundaries of the area set for advance reading, is prohibited. Stop burst transfer before it crosses the boundaries of these spaces and the area.

(3) Timing for starting the internal bus cycle for writing

In bus cycles for writing in synchronous-burst transfer supported MCU connection mode, data for writing are sampled in synchronization with HBUSCLK and an access request is issued to the AHB control block on the next Rising edge of HBUSCLK.

Accordingly, if the connected external MCU only supplies the clock signal to the HBUSCLK pin during the bus cycle periods, actual writing to the target internal resource is put on hold until the next bus cycle for supplying the clock signal to the HBUSCLK pin starts.

If immediate completion of actual writing to the target internal resource is essential, add some bus cycles over which the clock signal is supplied to the HBUSCLK pin after the number of bus cycles that would otherwise be required for writing.

(4) Confirming reading from a control register

When reading from a given control register, the MCU does not wait for prior writing to the control register to be completed.
Accordingly, if a control register is read after writing to it but before its value is actually changed, the value before writing will be read.
To change the setting of a control register, confirm completion of the change to the setting by polling the register until the same value as was written is read.

(5) Access during the internal reset period

In synchronous burst transfer supported MCU connection mode, register values cannot be read during the internal reset period.

(6) De-asserting the reset signal

The internal reset signal output by the reset synchronization circuit is de-asserted in synchronization with HBUSCLK supplied by the external MCU. Accordingly, if the external MCU only supplies HBUSCLK during bus cycles of the R-IN32M4-CL3 and the internal reset signal is not de-asserted at the point at which the first bus cycle of the R-IN32M4-CL3 starts, so that bus cycle is not recognizable.

17. Serial Flash ROM Memory Controller

An R-IN32M4-CL3 device has an internal memory controller to connect a serial flash ROM for an SPI-compatible interface.

When the BOOT1 and BOOT0 bits for the corresponding pins are set to 0 and 1 respectively, booting is from the serial flash ROM.

17.1 Features

- SPI interface:
  - Three SPI protocols (extended SPI, dual-SPI, and quad-SPI) are supported.
  - SPI mode 0 and SPI mode 3 are supported (default: SPI mode 3).
  - The address width is 24 bits.
- Timing adjustment:
  A wide range of serial ROM products are available by setting the relevant register.
- ROM reading:
  - The bus cycles of the internal system bus for reading are automatically converted to SPI bus cycles.
  - Direct booting from the serial ROM
  - Instructions for reading, fast read, fast read dual output, fast read dual I/O, fast read quad output, and fast read quad I/O are supported.
  - Prefetching
  - Allows the use of polling
  - Prolongation of bus cycles for SPI access
- Direct communications:
  Instructions and functionality of various devices are supported under software control (erasure, programming, ID reading, power-down control, etc.)
- Maximum transfer clock rate: 50 MHz

17.2 Control Registers

To use the serial flash ROM memory controller, set the operating mode by using the control registers.

Table 17.1 Control Registers of the Serial Flash ROM Memory Controller

Register Name	Symbol	Address
Transfer mode control register	SFMSMD	400A 2400H
Chip selection control register	SFMSSC	400A 2404H
Clock control register	SFMSKC	400A 2408H
Status register	SFMSST	400A 240CH
Communications port register	SFMCOM	400A 2410H
Communications mode control register	SFMCMD	400A 2414H
Communications status register	SFMCST	400A 2418H
Instruction code register	SFMSIC	400A 2420H
Address mode control register	SFMSAC	400A 2424H
Dummy cycle control register	SFMSDC	400A 2428H
SPI protocol control register	SFMSPC	400A 2430H
Port control register	SFMPMD	400A 2434H
Data input timing control register	SFMDTC	400A 2438H
Version register	SFMVER	400A 244CH

Caution: The settings of the control registers of the serial flash ROM memory controller can be changed dynamically during the system operation.
Note, however, that when changing the settings of multiple control registers sequentially, an SPI bus cycle may start before the completion of changing the values of all the registers, so take care regarding the order of changing the register settings to ensure that the SPI bus timing specification is met in any stage of changing the register settings.

17.2.1 Transfer Mode Control Register (SFMSMD)

This register controls bus cycles for SPI access.

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 16, 14 to 12, and 2 to 0.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 400A 2400H Initial value 0000 0110H
SFMSMD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SFMCCE	0	0	0	SFMOSW	SFMOHW	SFMOEX	SFMMD3	SFMPAE	SFMPFE	SFMSE1	SFMSE0	0	SFMRM2	SFMRM1	SFMRM0	
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15	SFMCCE	Selects read instruction code. 0: Read instruction code that is set in the SFMSIC register (initial value) 1: Default read instruction code of each read format
14 to 12	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
11	SFMOSW	Selects adjustment of the setup time during serial data output. 0: Does not extend the high level period of SMSCK during serial data output (initial value). 1: Extends the high level period of SMSCK by one clock cycle during serial data output. This function takes effect only during serial data output.
10	SFMOHW	Selects adjustment of the hold time during serial data output. 0: Does not extend the low level period of SMSCK during serial data output (initial value). 1: Extends the low level period of SMSCK by one clock cycle during serial data output. This function takes effect only during serial data output.
9	SFMOEX	Extends the output enable signal for the serial interface I/O buffer. 0: Does not extend the output enable period of serial data (initial value). 1: Extends the output enable period of serial data by one SMSCK cycle. Only the output enable signal is extended; output data is not extended.
8	SFMMD3	Selects the SPI mode. 0: SPI mode 0 1: SPI mode 3 (initial value)
7	SFMPAE	Selects stopping of prefetching at locations other than byte boundaries. 0: Disables prefetching at locations other than byte boundaries (initial value). 1: Enables prefetching at locations other than byte boundaries.
6	SFMPFE	Selects prefetching. 0: Disables prefetching (initial value). 1: Enables prefetching.

(2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>5, 4</td><td>SFMSE1,
SFMSE0</td><td>Selects extension of the SMCSZ (chip select) signal after access to the SPI bus.
<table>
<tr><th>SFMSE1</th><th>SFMSE0</th><th>SMCSZ (chip select) signal extension mode</th></tr>
<tr><td>0</td><td>0</td><td>Does not extend the SMCSZ signal.</td></tr>
<tr><td>0</td><td>1</td><td>Extends the SMCSZ signal by up to 33 serial clock cycles (initial value).</td></tr>
<tr><td>1</td><td>0</td><td>Extends the SMCSZ signal by up to 129 serial clock cycles.</td></tr>
<tr><td>1</td><td>1</td><td>Extends the SMCSZ signal infinitely.</td></tr>
</table>
While the SMCSZ signal is at the high level, power consumption of the serial flash ROM is reduced.</td></tr>
<tr><td>3</td><td>—</td><td>Reserved. When writing to this bit, write 0. When read, 0 is returned.</td></tr>
<tr><td>2 to 0</td><td>SFMRM2–
SFMRM0</td><td>Selects the read mode of the serial flash ROM.
<table>
<tr><th>SFMRM2</th><th>SFMRM1</th><th>SFMRM0</th><th>Serial flash ROM read mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Reading (initial value)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Fast Read</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Fast Read Dual Output</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Fast Read Dual I/O</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Fast Read Quad Output</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Fast Read Quad I/O</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Setting prohibited (operation not guaranteed)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Setting prohibited (operation not guaranteed)</td></tr>
</table></td></tr>
</table>

17.2.2 Chip Selection Control Register (SFMSSC)

This register sets the timing of the chip select signal for the serial ROM.

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 6 to 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMSSC	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	SFMSLD	SFMSHD	SFMSW3	SFMSW2	SFMSW1	SFMSW0
R/W	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	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 2404H

Initial value 0000 0037H

Bit Position	Bit Name	Description
31 to 6	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
5	SFMSLD	Selects the output timing of the SMCSZ (chip select) signal. 0: Outputs SMSCK 0.5 clock cycles before the first rising edge of SMCLK. 1: Outputs SMSCK 1.5 clock cycles before the first rising edge of SMCLK (initial value).
4	SFMSHD	Selects the timing for de-asserting the SMCLK (chip select) signal. 0: De-asserts SMSCK 0.5 clock cycles after the last rising edge of SMCLK. 1: De-asserts SMSCK 1.5 clock cycles after the last rising edge of SMCLK (initial value).
3 to 0	SFMSW3– SFMSW0	Selects the minimum width at high level of the SMCSZ (chip select) signal.

SFMSW3	SFMSW2	SFMSW1	SFMSW0	Minimum width at high level of SMCSZ (chip select) signal
0	0	0	0	1 SMSCK cycle
0	0	0	1	2 SMSCK cycles
0	0	1	0	3 SMSCK cycles
0	0	1	1	4 SMSCK cycles
0	1	0	0	5 SMSCK cycles
0	1	0	1	6 SMSCK cycles
0	1	1	0	7 SMSCK cycles
0	1	1	1	8 SMSCK cycles (initial value)
1	0	0	0	9 SMSCK cycles
1	0	0	1	10 SMSCK cycles
1	0	1	0	11 SMSCK cycles
1	0	1	1	12 SMSCK cycles
1	1	0	0	13 SMSCK cycles
1	1	0	1	14 SMSCK cycles
1	1	1	0	15 SMSCK cycles
1	1	1	1	16 SMSCK cycles

17.2.3 Clock Control Register (SFMSKC)

This register specifies the operating speed of the SPI bus.

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 6 to 0.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMSKC	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	SFMDTY	SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0
R/W	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	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 2408H
Initial value 0000 0008H

Bit Position	Bit Name	Description
31 to 6	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
5	SFMDTY	Selects duty cycle correction for the SMSCK signal. 1: Delays the rising edge of the SMSCK signal by 0.5 cycles of HCLK. 0: The SMSCK signal is not adjusted.
4 to 0	SFMDV4–SFMDV0	Selects the serial clock (SMSCK) based on the internal system bus clock (HCLK).

SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection
0	0	0	0	0	HCLK/2
0	0	0	0	1	HCLK/3 Note
0	0	0	1	0	HCLK/4
0	0	0	1	1	HCLK/5 Note
0	0	1	0	0	HCLK/6
0	0	1	0	1	HCLK/7 Note
0	0	1	1	0	HCLK/8
0	0	1	1	1	HCLK/9 Note
0	1	0	0	0	HCLK/10 (initial value)
0	1	0	0	1	HCLK/11 Note
0	1	0	1	0	HCLK/12
0	1	0	1	1	HCLK/13 Note
0	1	1	0	0	HCLK/14
0	1	1	0	1	HCLK/15 Note
0	1	1	1	0	HCLK/16
0	1	1	1	1	HCLK/17 Note

Continued on next page

Note: When the clock frequency is divided by an odd number and duty-cycle correction is not in use, the width at high level of the SMSCK signal is 1 cycle of HCLK longer than the width at low level.

(2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>4 to 0</td><td>SFMDV4–
SFMDV0</td><td>Selects the serial clock (SMSCK) based on the internal system bus clock (HCLK).
<table>
<tr><th>SFMDV4</th><th>SFMDV3</th><th>SFMDV2</th><th>SFMDV1</th><th>SFMDV0</th><th>Serial clock selection</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>HCLK/18</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>HCLK/20</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>HCLK/22</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>HCLK/24</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>HCLK/26</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>HCLK/28</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>HCLK/30</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>HCLK/32</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>HCLK/34</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>HCLK/36</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>HCLK/38</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>HCLK/40</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>HCLK/42</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>HCLK/44</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>HCLK/46</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>HCLK/48</td></tr>
</table>
</td></tr>
</table>

Remark: HCLK: Internal system bus clock

17.2.4 Status Register (SFMSST)

This register is used to check the state of access to the serial flash ROM.

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

(1/2)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMSST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PFOFF	PFFUL	0	PFCNT4	PFCNT3	PFCNT2	PFCNT1	PFCNT0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	R	0	R	R	R	R	R

Address 400A 240CH
Initial value 0000 0080H

Bit Position	Bit Name	Description
31 to 8	—	Reserved. These bits are read as 0.
7	PFOFF	Indicates the state of prefetching.[Note] 0: Prefetching is active. 1: Prefetching is not enabled or inactive (initial value).
6	PFFUL	Indicates the state of the prefetch buffer. 0: The prefetch buffer has space available (initial value). 1: The prefetch buffer is full.
5	—	Reserved. This bit is read as 0.
4 to 0	PFCNT4–PFCNT0	Indicates the number of bytes of the prefetched data. (see table below)

PFCNT4–0					Number of bytes of prefetched data
0	0	0	0	0	No prefetched data (initial value)
0	0	0	0	1	1 byte of data prefetched
0	0	0	1	0	2 bytes of data prefetched
0	0	0	1	1	3 bytes of data prefetched
0	0	1	0	0	4 bytes of data prefetched
0	0	1	0	1	5 bytes of data prefetched
0	0	1	1	0	6 bytes of data prefetched
Other than the above					No other combination is available.

Continued on next page

Note: When the SFMPFE bit of the SFMSMD register is set to 1, prefetching is triggered by the first reading of the serial flash ROM and stops in response to writing to the SFMCMD register.
When prefetching is used for polling, if the PFOFF bit is set to 1, reading of the serial flash ROM data must be started, regardless of the value of the PFCNT4 to PFCNT0 bits.

(2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>4 to 0</td><td>PFCNT4–
PFCNT0</td><td>Indicates the number of bytes of the prefetched data.
<table>
<tr><th colspan="5">PFCNT4–0</th><th>Number of bytes of prefetched data</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>7 bytes of data prefetched</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>8 bytes of data prefetched</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>9 bytes of data prefetched</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>10 bytes of data prefetched</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>11 bytes of data prefetched</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>12 bytes of data prefetched</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>13 bytes of data prefetched</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>14 bytes of data prefetched</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>15 bytes of data prefetched</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>16 bytes of data prefetched</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>17 bytes of data prefetched</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>18 bytes of data prefetched</td></tr>
<tr><td colspan="5">Other than the above</td><td>No other combination is available.</td></tr>
</table>
</td></tr>
</table>

Note: When the SFMPFE bit of the SFMSMD register is set to 1, prefetching is triggered by the first reading of the serial flash ROM and stops in response to writing to the SFMCMD register.

When prefetching is used for polling, if the PFOFF bit is set to 1, reading of the serial flash ROM data must be started, regardless of the value of the PFCNT4 to PFCNT0 bits.

17.2.5 Communications Port Register (SFMCOM)

This is an I/O port used to output instruction codes, addresses, and write data to the serial flash ROM, as well as to retrieve read data and status information from the serial flash ROM.

When data is written to the SFMCOM port in direct communications mode (SFMCMD.DCOM = 1), the written data is transmitted to the serial ROM. When data is read from SFMCOM, one byte of data is received from the serial flash ROM and the received data is read.

When data is written to or read from SFMCOM, the chip select signal (SMCSZ) for the serial flash ROM becomes active. Even after the transmission or reception is completed, the chip select signal (SMCSZ) for the serial flash ROM remains active. The active chip select signal (SMCSZ) returns to the inactive state when desired data is written to the SFMCMD register described later.

Since serial flash ROM products from various vendors are not standardized in terms of commands and protocols, especially those related to programming and erasure, device-specific control is required. When the serial flash ROM is used with R-IN32M4-CL3 products, software control via SFMCOM is necessary for programming and erasure.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7 to 0
SFMCOM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SFMD[7:0]
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W

Address 400A 2410H

Initial value Undefined

Bit Position	Bit Name	Description
31 to 8	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
7 to 0	SFMD[7:0]	Communications port register to input and output data for direct transfer to and from the serial ROM. The read/write operations for this register are converted to SPI bus cycles. This register can only be accessed in direct communications mode (SFMCMD.DCOM = 1). Access to this this register will be ignored in ROM access mode (SFMCMD.DCOM = 0).

17.2.6 Communications Mode Control Register (SFMCMD)

This register is used to select the mode of communications between the system bus and SPI bus. There are two modes of communications between the system bus and SPI bus: ROM access mode and direct communications mode.

In ROM access mode, system bus cycles are automatically converted to SPI bus cycles, which allows the content of the serial ROM to be referenced as easily as the regular ROM.

In direct communications mode, data is repeatedly input to and output from the communication port register (SFMCOM) to establish an SPI bus cycle for transfer to and from the serial ROM via software control. This mode is used in programming and erasure of the serial ROM.

When the SFMCMD register is written, the chip select signal (SMCSZ) for the serial ROM, which has been active since the last access to SFMCOM, returns to the inactive state.

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 1 to 0.

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

Address 400A 2414H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	DCOM	Selects the mode of communications with the serial ROM. 0: ROM access mode (ROM access enabled) (initial value) 1: Direct communications mode (ROM access disabled)

Caution: Serial ROMs in general cannot respond to requests for reading, etc., except for certain processing such as status checking, over periods where programming, erasure, etc. is being handled within the device. Therefore, ensuring that operations such as programming and erasure, fetching of code, and other regular data-access operations are in an appropriate order is left to software. When designing software, create a flow of processing that is appropriate for the specifications of the serial ROM you are using.

17.2.7 Communications Status Register (SFMCST)

This register indicates the state of communications with the serial flash ROM.

If an attempt is made to access the space to which the serial flash ROM itself is allocated while SFMCMD.DCOM is set to 1 (direct communications mode), an error occurs and INTSFMC is generated.

INTSFMC is generated according to the level, which returns to the low level when the EROMR bit is cleared to 0.

Note that the EROMR bit cannot be set to 1.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMCST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EROMR	0	0	0	0	0	0	COMBSY
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	0	0	0	0	0	0	R/W

Address 400A 2418H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 8	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
7	EROMR	Indicates the state of detection of access to the ROM in direct communications mode. 0: No error (initial value) 1: Error (invalid access to the ROM was detected in direct communications mode)
6 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	COMBSY	Indicates the state of processing of SPI bus cycles in direct communications. 0: There are no SPI bus cycles being processed (initial value). 1: There are SPI bus cycles being processed.

Caution: Errors that are detectable by using the EROMR bit are limited to those related to the operation procedure of the direct communications register of an R-IN32M4-CL3. The specifications and restrictions on the individual serial ROM products from various vendors require software control.

17.2.8 Instruction Code Register (SFMSIC)

This register sets the timing of the serial ROM chip select signal

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 8 to 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 400A 2420H
SFMSIC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SFMCIC7	SFMCIC6	SFMCIC5	SFMCIC4	SFMCIC3	SFMCIC2	SFMCIC1	SFMCIC0	Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31 to 8	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
7 to 0	SFMCIC	Serial ROM read instruction code for replacement

17.2.9 Address Mode Control Register (SFMSAC)

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 5, 3, and 2 to 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 400A 2424H
SFMSAC	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	SFM4BC	0	0	SFMAS1	SFMAS0	Initial value 0000 0002H
R/W	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	R/W	0	0	R/W	R/W	

Bit Position	Bit Name	Description
31 to 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4	SFM4BC	Selects the default instruction code when a 4-byte address is selected. 0: Instruction code for 1-byte addresses is not used (initial value). 1: Instruction code for 4-byte addresses is used.
3, 2	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
1, 0	SFMAS1, SFMAS0	Selects the address width of the serial interface. 00: 1 byte 01: 2 bytes 10: 3 bytes (initial value) 11: 4 bytes

17.2.10 Dummy Cycle Control Register (SFMSDC)

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 16, 5, and 4 to 0.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMSDC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SFMXD7	SFMXD6	SFMXD5	SFMXD4	SFMXD3	SFMXD2	SFMXD1	SFMXD0	SFMXEN	SFMXST	0	0	SFMDN3	SFMDN2	SFMDN1	SFMDN0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	0	0	R/W	R/W	R/W	R/W

Address 400A 2428H

Initial value 0000 FF00H

Bit Position	Bit Name	Description
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 8	SFMXD7–0	Sets the value for selecting instruction-omission mode.
7	SFMXEN	Enables or disables instruction-omission mode. 0: Disables instruction-omission mode (initial value). 1: Enables instruction-omission mode.
6	SFMXST	Instruction omission status 0: Indicates that operation is in progress in normal (instruction not omitted) mode (initial value). 1: Indicates that operation is in progress in instruction-omission mode.
5, 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	SFMDN3–0	Selects the number of dummy cycles of fast read instructions.

SFMDN3	SFMDN2	SFMDN1	SFMDN0	Number of Dummy Cycles of Fast Read Instructions
0	0	0	0	Default number of cycles of each instruction format • Fast Read: 8 SMSCK cycles • Fast Read Dual Output: 8 SMSCK cycles • Fast Read Dual I/O: 4 SMSCK cycles • Fast Read Quad Output: 8 SMSCK cycles • Fast Read Quad I/O: 6 SMSCK cycles
0	0	0	1	3 SMSCK cycles Note
0	0	1	0	4 SMSCK cycles
0	0	1	1	5 SMSCK cycles
0	1	0	0	6 SMSCK cycles
0	1	0	1	7 SMSCK cycles
0	1	1	0	8 SMSCK cycles

Continued on next page

Note: To avoid contention in switching between input and output through the SMIO0 pin and the pin of the serial ROM to which it is connected, select at least 4 cycles of SMSCK as the number of dummy cycles for extending the signal output period by setting the SFMOEX bit in the SFMSMD register to 1.

(2/2)

Bit Position	Bit Name	Description
3 to 0	SFMDN3–0	Selects the number of dummy cycles of fast read instructions.

SFMDN3	SFMDN2	SFMDN1	SFMDN0	Number of Dummy Cycles of Fast Read Instructions
0	1	1	1	9 SMSCK cycles
1	0	0	0	10 SMSCK cycles
1	0	0	1	11 SMSCK cycles
1	0	1	0	12 SMSCK cycles
1	0	1	1	13 SMSCK cycles
1	1	0	0	14 SMSCK cycles
1	1	0	1	15 SMSCK cycles
1	1	1	0	16 SMSCK cycles
1	1	1	1	17 SMSCK cycles

17.2.11 SPI Protocol Control Register (SFMSPC)

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 5, 3, and 2 to 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMSPC	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	SFMSDE	0	0	SFMSPI1	SFMSPI0
R/W	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	R/W	0	0	R/W	R/W

Address 400A 2430H
Initial value 0000 0010H

Bit Position	Bit Name	Description
31 to 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4	SFMSDE	Selects the minimum time for input/output switching when the dual-SPI protocol or the quad-SPI protocol is selected. 1: Secures the minimum switching time equivalent to one cycle of SMSCK (initial value). 0: Does not secure the minimum switching time.
3, 2	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
1, 0	SFMSPI1, SFMSPI0	Selects the SPI protocol.[Note]

SFMSPI1	SFMSPI0	SPI Protocol
0	0	Extended SPI protocol (initial value)
0	1	Dual-SPI protocol
1	0	Quad-SPI protocol
1	1	Setting prohibited

Note: Switching of the protocols for the serial ROM requires software control.
Some products will require re-initialization following switching of the protocols between serial flash ROM memory control and the serial ROM control. Since the input of the reset signal cannot be used with the serial ROM, so the serial ROM must be initialized by cutting off and then resupplying power to it.

17.2.12 Port Mode Control Register (SFMPMD)

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 3, 1, and 0 to 0.

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

Address 400A 2434H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 3	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
2	SFMWPL	Specifies the level of the WP pin of the serial ROM device. 1: High level 0: Low level (initial value)
1, 0	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.

17.2.13 Data Input Timing Control Register (SFMDTC)

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 6, and 3 to 1 to 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMDTC	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	SFMDCL1	SFMDCL0	0	0	0	SFMDCD
R/W	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	R/W	R/W	0	0	0	R/W

Address 400A 2438H

Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 6	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
5, 4	SFMDCL1, SFMDCL0	Selects the latency in the reception of serial data.
3 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	SFMDCD	Selects the timing of the reception of serial data.[Note] 1: Delay by one cycle of HCLK from the falling edge of SMSCK 0: Same timing as the falling edge of SMSCK

Settings of SFMDCL1 and SFMDCL0 (bits 5, 4):

SFMDCL1	SFMDCL0	Reception Latency
0	0	Reception latency 1 (after 1 SMSCK cycle of serial ROM output) (initial value)
0	1	Reception latency 2 (after 2 SMSCK cycles of serial ROM output)
1	0	Reception latency 3 (after 3 SMSCK cycles of serial ROM output)
1	1	Setting prohibited

Note: Do not set this bit if the minimum delay of the serial data input by the serial flash ROM memory controller is shorter than one cycle of HCLK (internal bus clock) from the falling edge of SMSCK.

17.2.14 Version Register (SFMVER)

- Access This register can be read or written in 32-bit units.
  Be sure to set bits 31 to 16 to 0.

SFMVER — Address 400A 244CH, Initial value 0000 0300H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SFMVER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMVER	SFMVID15	SFMVID14	SFMVID13	SFMVID12	SFMVID11	SFMVID10	SFMVID9	SFMVID8	SFMVID7	SFMVID6	SFMVID5	SFMVID4	SFMVID3	SFMVID2	SFMVID1	SFMVID0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Description
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 0	SFMVID15–0	Indicates the version number.

17.3 Connection with Serial Flash ROM

An R-IN32M4-CL3 device is connected with the serial flash ROM as shown below.

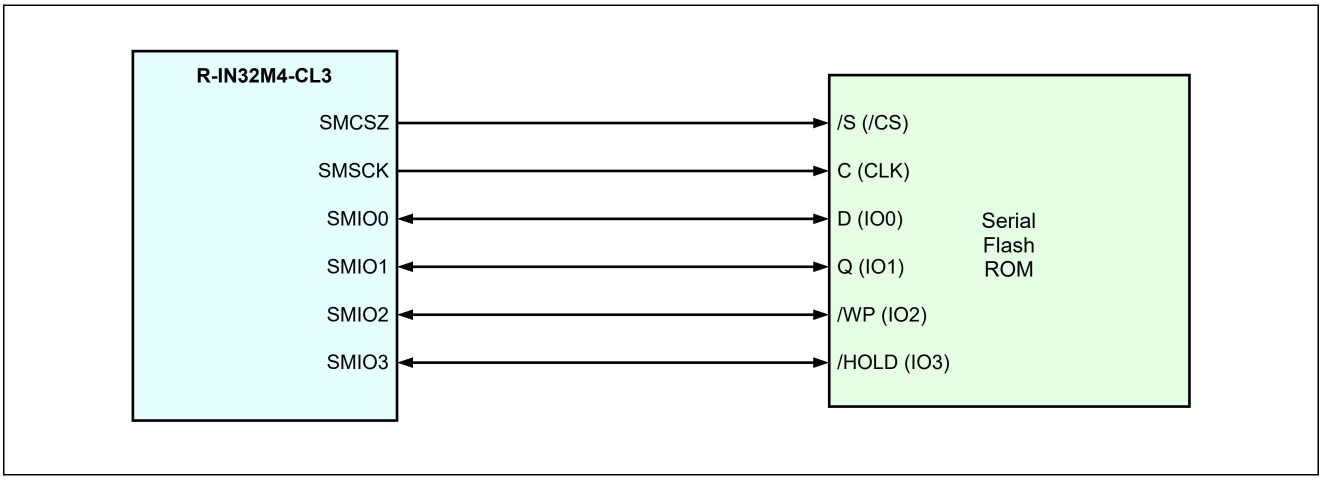


Figure 17.1 Connection with Serial Flash ROM

17.4 Operation

17.4.1 SPI Bus

(1) SPI Protocol

The extended SPI, dual-SPI, and quad-SPI protocols are supported as SPI protocols used for connection of the serial ROM. The initial state of SPI protocol is extended SPI, which can be changed by the SFMSPI bits in the SFMSPC register.

In the extended SPI protocol, instruction codes are always output from a single SMIO0 pin, and subsequent addresses and data are input and output by using one to four pins according to the instruction code format.

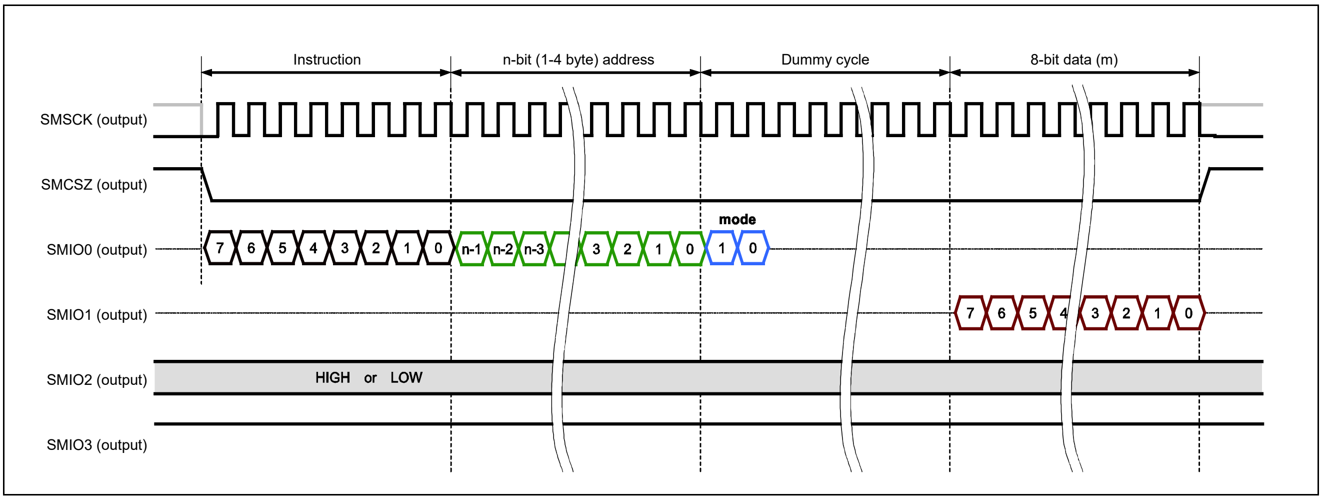


Figure 17.2 Example 1 of Extended SPI Protocol (Fast Read)

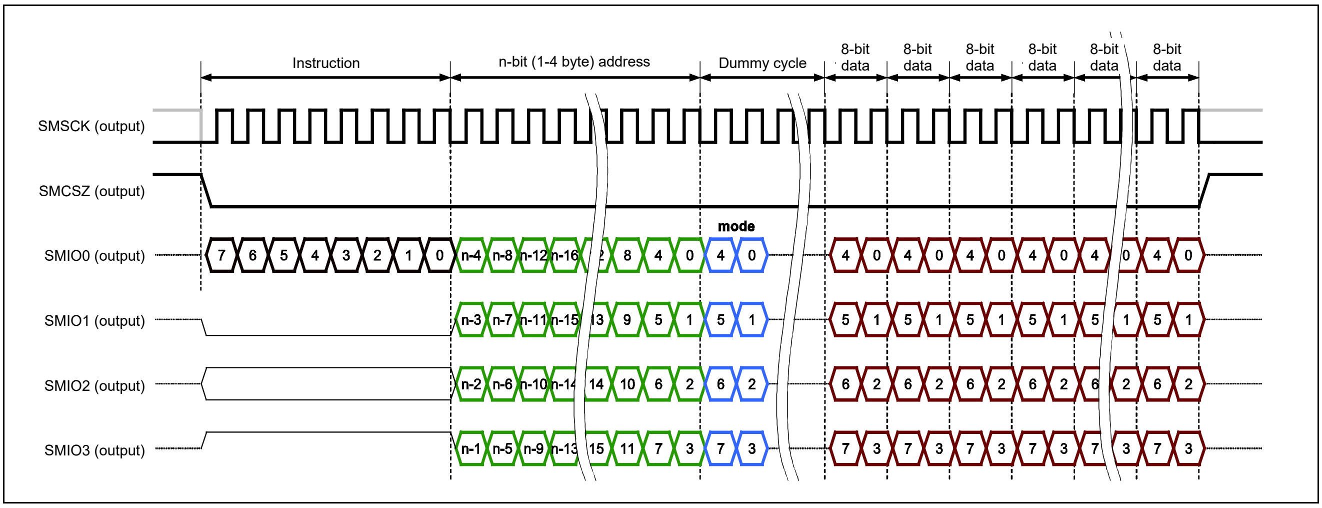


Figure 17.3 Example 2 of Extended SPI Protocol (Fast Read Quad I/O)

In the dual-SPI protocol, all signals (including instruction codes, addresses, and data) are input and output by using two pins SMIO0 and SMIO1.

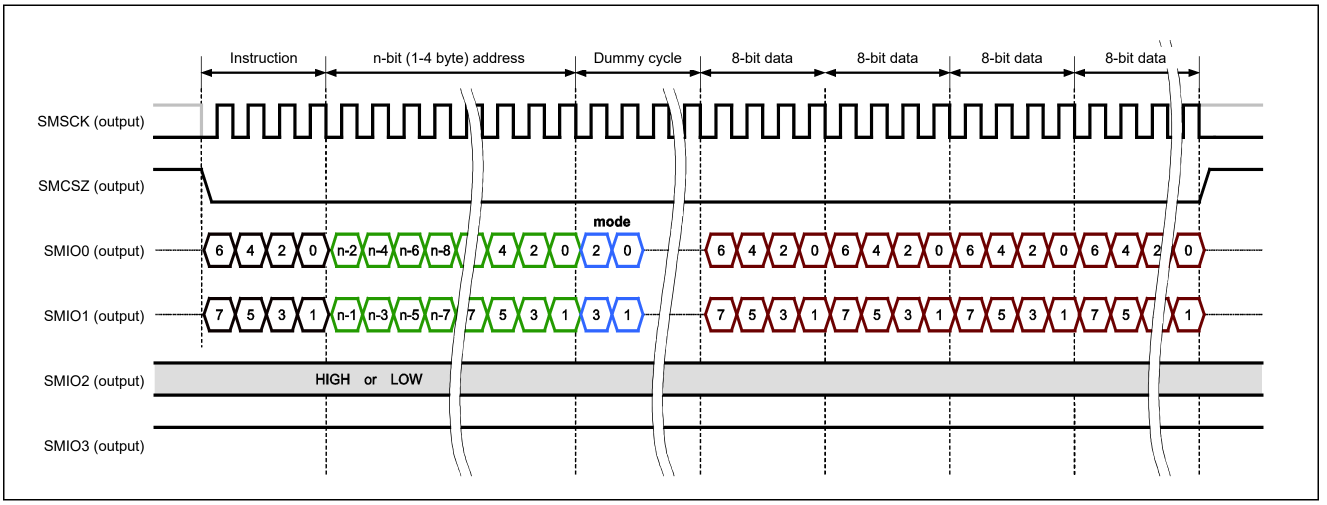


Figure 17.4 Example of Dual-SPI Protocol (Fast Read)

In the quad-SPI protocol, all signals (including instruction codes, addresses, and data) are input and output by using four pins SMIO0, SMIO1, SMIO2, and SMIO3.

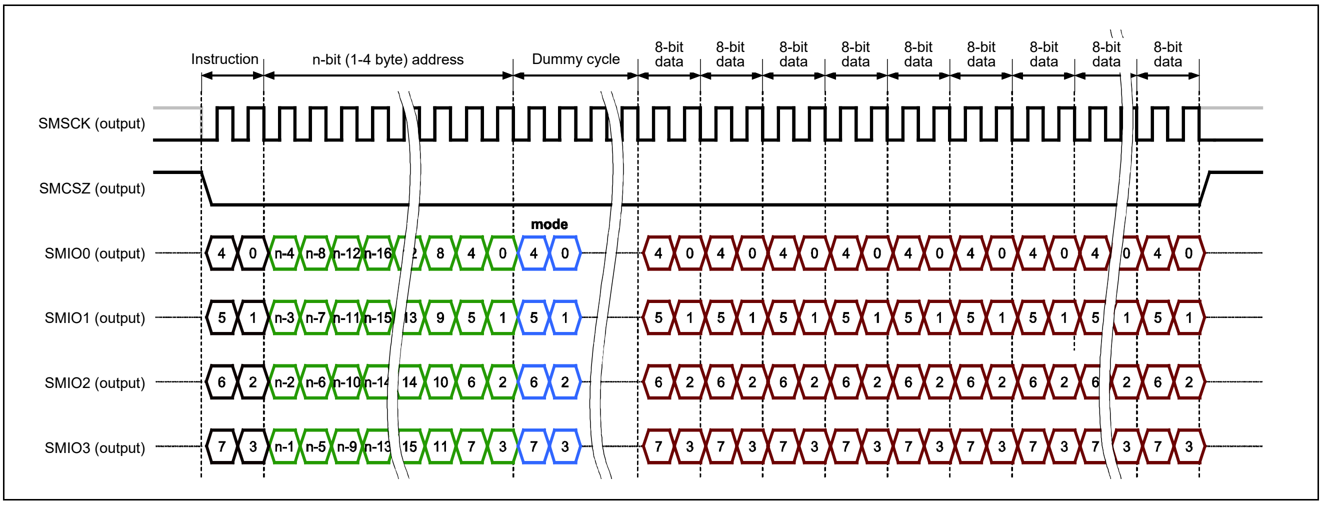


Figure 17.5 Quad SPI Protocol (Fast Read)

(2) SPI Mode

The serial flash ROM memory controller starts to operate in SPI mode 3 after release from the reset state. By changing the setting of the relevant register, it is possible to switch SPI mode 0 and SPI mode 3 during operation.

A difference between SPI mode 0 and SPI mode 3 is the level of the SMCLK signal on standby. In SPI mode 0, the standby level of the SMCLK signal is the low level. In SPI mode 3, the standby level of the SMCLK signal is the high level.

Serial data for output are output in synchronization with falling edges of the serial clock signal (SMSCK) and acquired in synchronization with rising edges of the serial clock signal (SMSCK).

Serial data for input are output in synchronization with falling edges of the serial clock signal (SMSCK) and acquired in synchronization with the following falling edges.

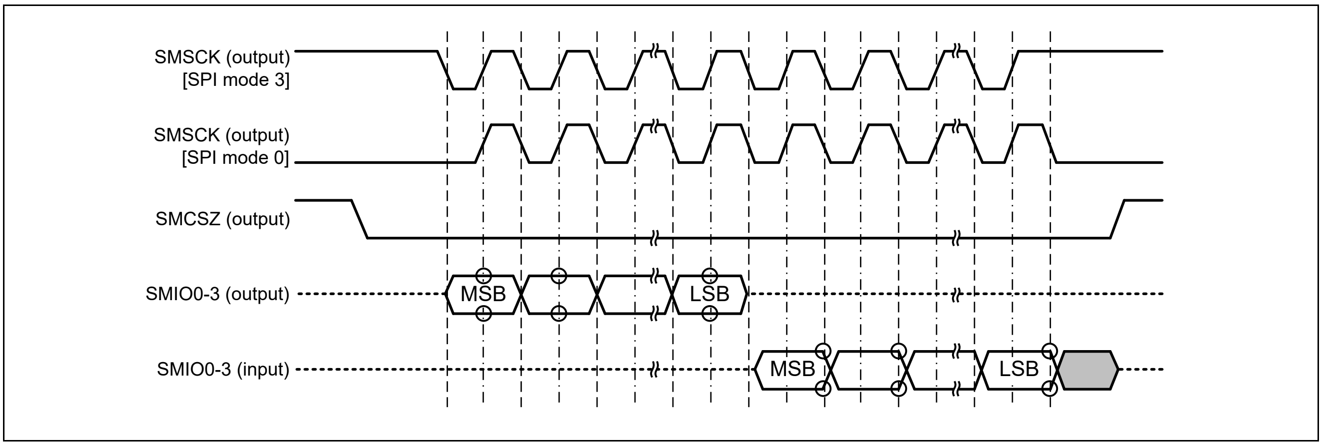


Figure 17.6 Basic Operation of SPI Bus

17.4.2 SPI Bus Timing Adjustment

The timing of SPI bus signals can be adjusted by setting the relevant registers.

The timing settings made here apply to all SPI bus accesses, regardless of access to the ROM or direct communications.

(1) Reference Cycle of the SPI Bus

The SPI bus operates according to the reference cycle that is an integral multiple of the AHCLK cycle.

This reference cycle can be selected from 2 to 48 times the AHCLK cycle by using SFMSKC.SFMDV4–SFMDV0.

(2) Duty Factor of the SMSCK Signal

When the reference cycle is an even multiple of the cycle of the internal system bus clock (HCLK), the width at high level of the SMSCK signal becomes equal to the width at low level. When the reference cycle is an odd multiple, the width at high level of the SMSCK signal becomes one HCLK clock cycle longer than the width at low level

To achieve a duty factor close to 50% for the SMSCK signal when the reference cycle is an odd multiple of the HCLK cycle, set SFMSKC.SFMDTY to 1. When SFMSKC.SFMDTY is set to 1, achieve a duty factor close to 50% by delaying the rising edge of the SMSCK output signal by half the HCLK cycle.

Note that, when the reference cycle is an even multiple of the HCLK cycle, the setting of SFMSKC.SFMDTY will be ignored.

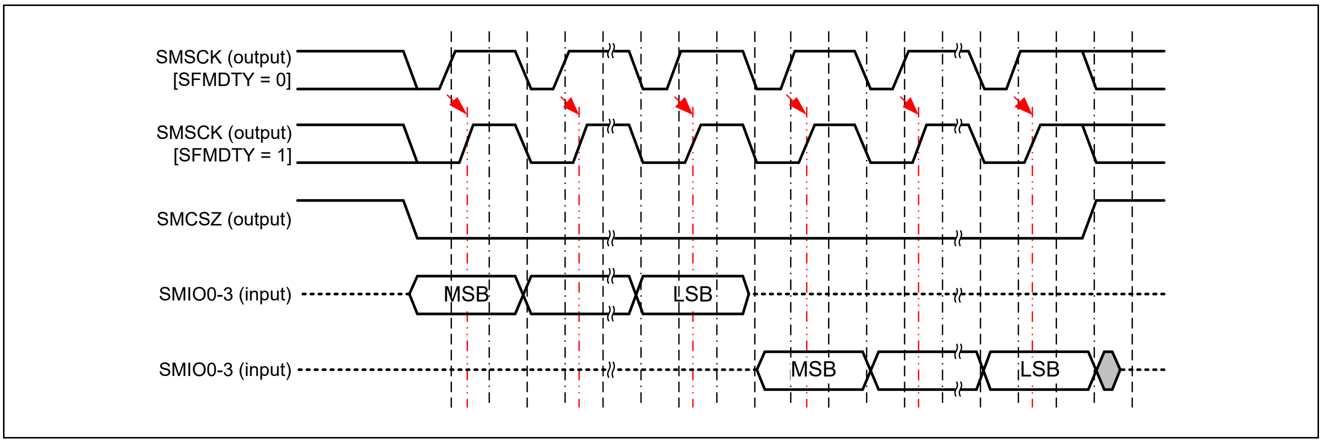


Figure 17.7 Correction of the SMSCK Signal Duty Factor by Using the SFMDTY Bit (Example of HCLK/3)

(3) Minimum Width at High Level of the SMCSZ Signal

The SMCSZ signal must remain at the high level (inactive) for a certain period of time between the adjacent SPI bus cycles in order to meet the non-selection time required by the device.

The minimum width at high level of the SMCSZ output signal can be selected from 1 to 16 times the reference cycle by using SFMSSC.SFMSW3–SFMSW0.

(4) Setup Time of the SMCSZ Signal

At the first rising edge of the SMSCK signal after the SMCSZ signal has been set to the low level, the setup time of the SMCSZ signal requested by the device must be met.

As the setup time of the SMCSZ signal, 0.5 cycles of SMSCK or 1.5 cycles of SMSCK can be selected by using SFMSSC.SFMSLD.

Note that the setting of SFMSSC.SFMSLD is also applied to securing of the setup time from the enable control of the output buffer for serial data output to the first rising edge of the SMSCK signal.

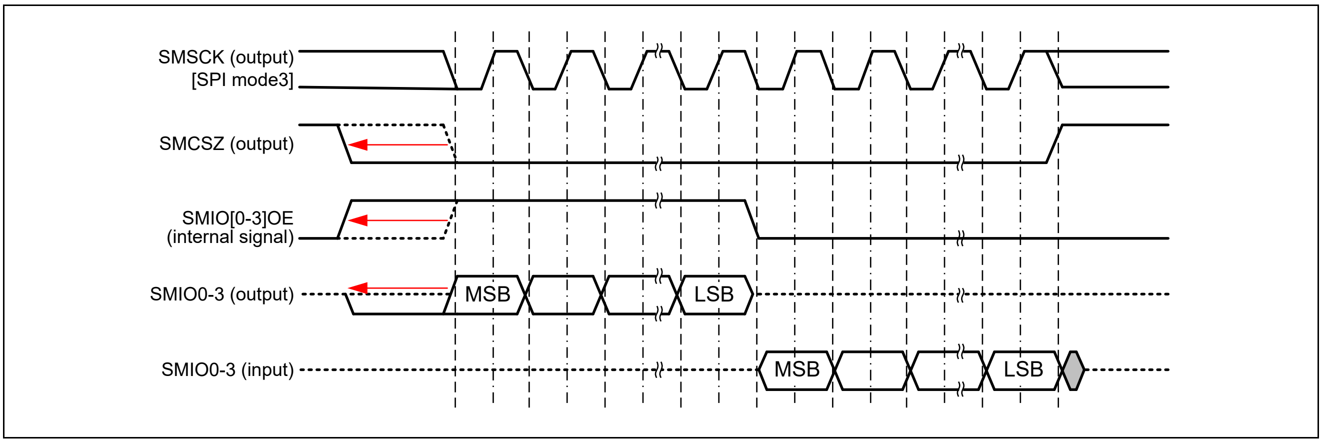


Figure 17.8 SMCSZ Signal Setup Time Adjustment by Using the SFMSLD Bit

(5) Hold Time of the SMCSZ Signal

When the SMCSZ signal is set to the high level from the last rising edge of the SMSCK signal, the SMCSZ hold time of the serial flash ROM must be met.

As the hold time of the SMCSZ signal, 0.5 cycles of SMSCK or 1.5 cycles of SMSCK can be selected by using SFMSSC.SFMSHD.

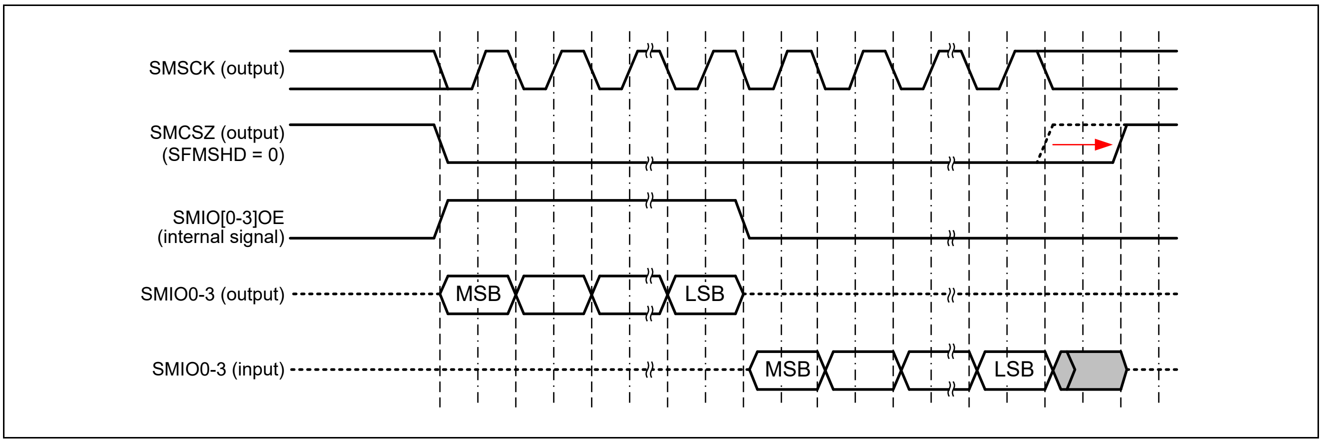


Figure 17.9 SMCSZ Signal Hold Time Adjustment by Using the SFMSHD Bit

(6) Output Enable Time of the Serial Data Output Buffer

The buffer output enable time of the SMIO0–3 pins can be extended by one SMSCK clock cycle by using SFMSMD.SFMOEX.

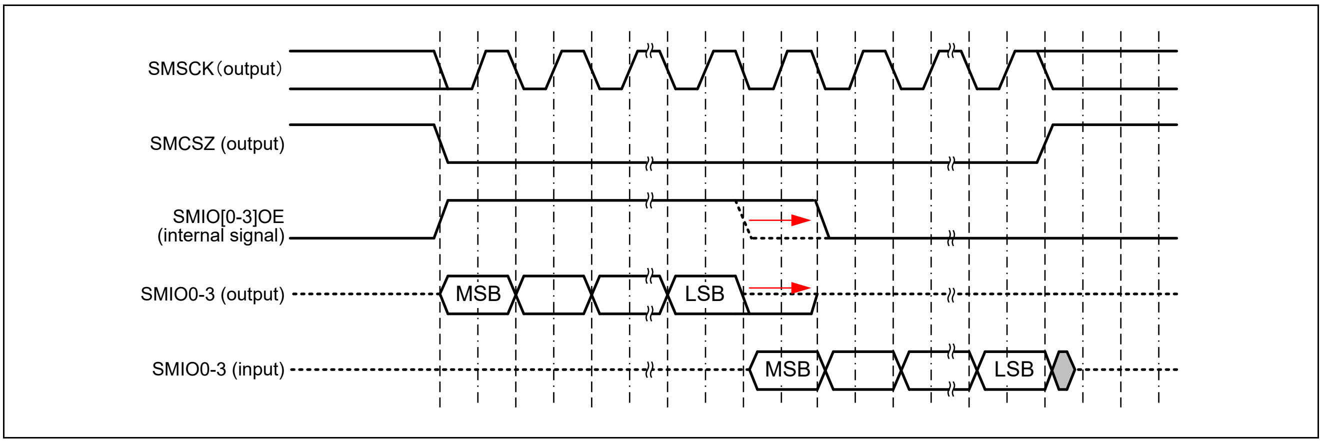


Figure 17.10 Output Enable Time Adjustment by Using the SFMOEX Bit

(7) Setup Time of Serial Data Output

Transmission of a command or address to the serial flash ROM must meet the setup time from serial data output to the rising edge of the SMSCK signal.

If this setup time is insufficient, SFMSMD.SFMOSW can be used to extend the setup time from serial data output to the rising edge of the SMSCK signal by one HCLK clock cycle.

If the SFMOSW bit is set to 1, the width at low level of SMSCK is extended by one HCLK clock cycle when serial data is transmitted during a data output cycle. This function does not take effect for serial data reception.

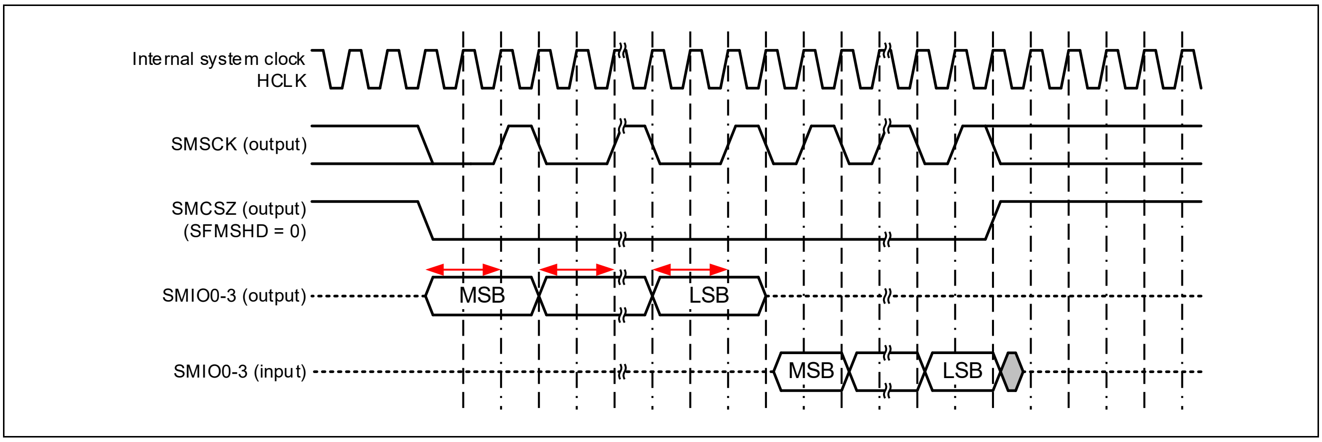


Figure 17.11 Serial Data Setup Time Adjustment by Using the SFMOSW Bit

(8) Hold Time of Serial Data Output

Transmission of a command or address to the serial flash ROM must meet the hold time from serial data output to the rising edge of the SMSCK signal.

If this hold time is insufficient, SFMSMD.SFMOHW can be used to extend the time from the rising edge of the SMSCK signal to the next change in the serial data by one HCLK clock cycle.

If the SFMOHW bit is set to 1, the width at high level of SMSCK is extended by one HCLK clock cycle when serial data is transmitted during a data output cycle. This function does not take effect for serial data reception.

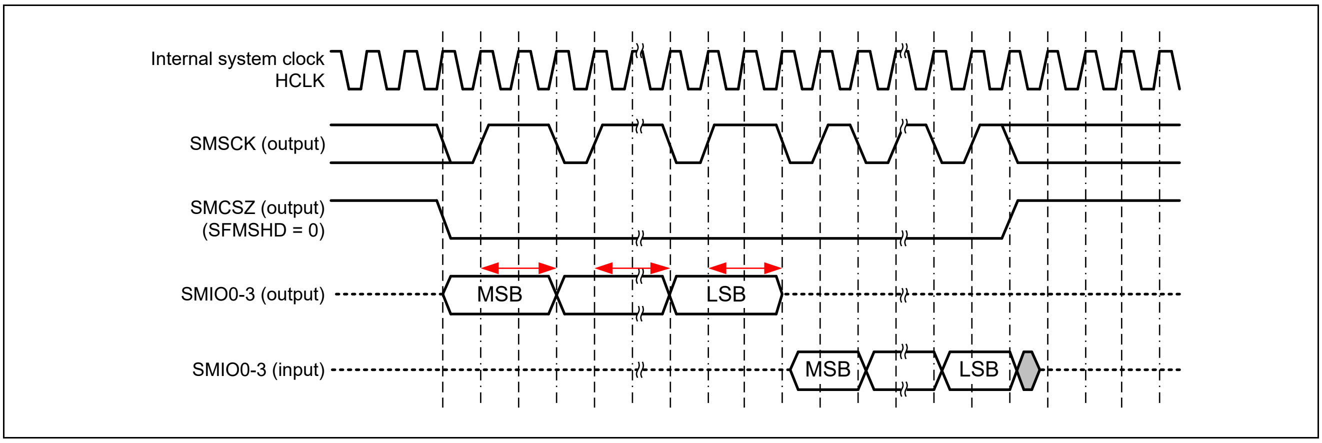


Figure 17.12 Serial Data Hold Time Adjustment by Using the SFMOHW Bit

(9) Latency in the Reception of Serial Data

The serial ROM outputs data in synchronization with falling edges of the SMSCK signal and receives it in synchronization with subsequent falling edges of the SMSCK signal. The delay from when the serial ROM starts to output data until it receives data is called the "reception latency" and the SFMDCL[1:0] bits of the SFMDTC register can be used to select a latency of from one to three cycles of SMSCK.

If a reception latency other than one SMSCK cycle is selected, additional clock cycles for adjusting the latency are inserted in the SPI bus cycle before the first cycle of data reception. From the viewpoint of the serial ROM, this operation appears as an increase in the number of clock cycles for data reception.

Additional cycles for adjusting the latency are not generated in SPI bus cycles where there is no data reception.

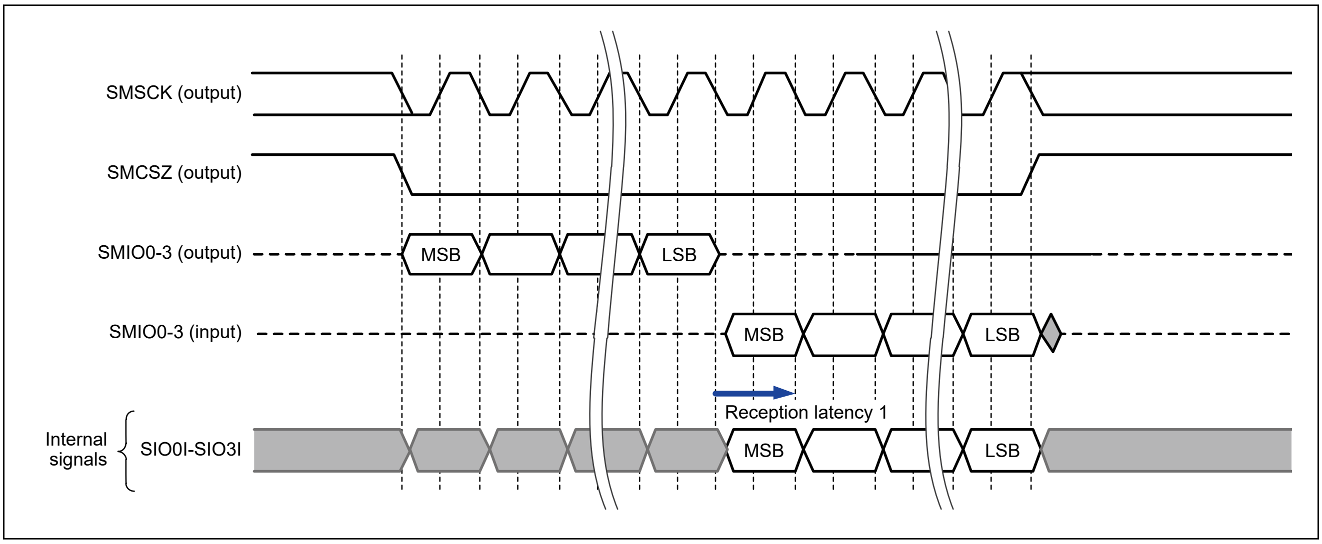


Figure 17.13 Reception Latency 1 (SFMDCL[1:0] = 00)

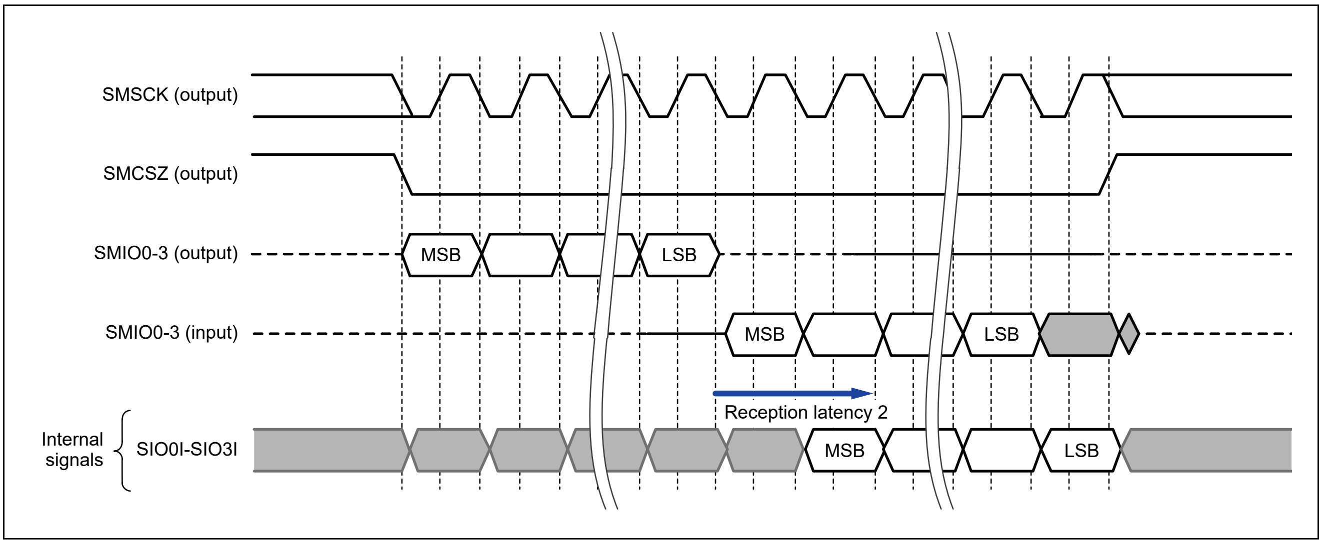


Figure 17.14 Reception Latency 2 (SFMDCL[1:0] = 01)

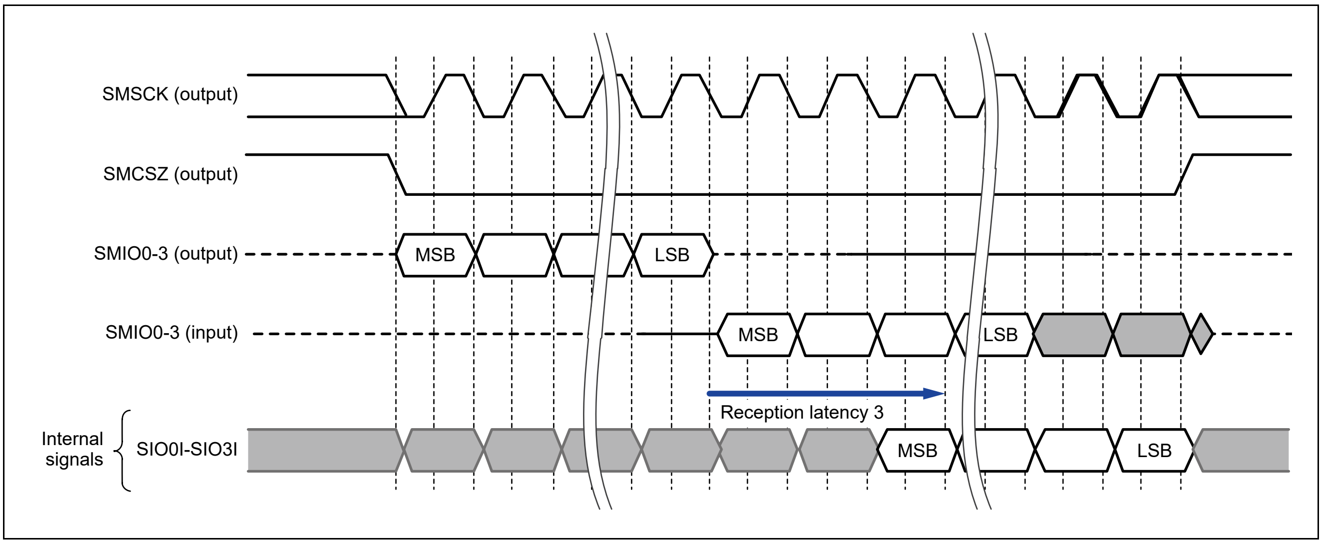


Figure 17.15 Reception Latency 3 (SFMDCL[1:0] = 10)

Caution: The timing diagrams in this document represent the timing of operation of reception with a latency of 1 unless otherwise specifically noted. If a reception latency other than 1 is selected, operation for reception may be suspended at locations other than byte boundaries within the serial ROM.

17.4.3 SPI Instruction Set for Use in Access to the Serial Flash ROM

(1) Types of SPI Instruction to Be Generated Automatically

When the serial flash ROM is accessed, an SPI bus cycle is automatically generated using the instructions listed below, according to the setting of the SFMSMD register.

When the reset signal is de-asserted, an instruction for release from the deep power-down state is automatically issued after a certain period of time.

Table 17.2 SPI Instruction Set to Be Generated Automatically

Instruction	Instruction Code	Number of Address Bytes	Number of Dummy Data Items	Number of Data Bytes	SFMRM bit Setting of SFMSMD Register
Standard reading	03H	3	—	1 to infinite	SFMRM[2:0] = 000B
Fast Read	0BH	3	1	1 to infinite	SFMRM[2:0] = 001B
Fast Read dual Output	3BH	3	1	1 to infinite	SFMRM[2:0] = 010B
Fast Read Dual I/O	BBH	3	1	1 to infinite	SFMRM[2:0] = 011B
Fast Read Quad Output	6BH	3	1	1 to infinite	SFMRM[2:0] = 100B
Fast Read Quad I/O	EBH	3	1	1 to infinite	SFMRM[2:0] = 101B
Release from deep power-down	ABH	—	—	—	—

(2) Instruction for Standard Reading

Standard reading is a common method of reading supported by a majority of serial flash ROMs.

When an SPI bus cycle starts, the SMCSZ signal becomes active and 03H is output as an instruction code. Next, a 24-bit address is transmitted, and then data is received.

In the initial state, this standard reading is selected.

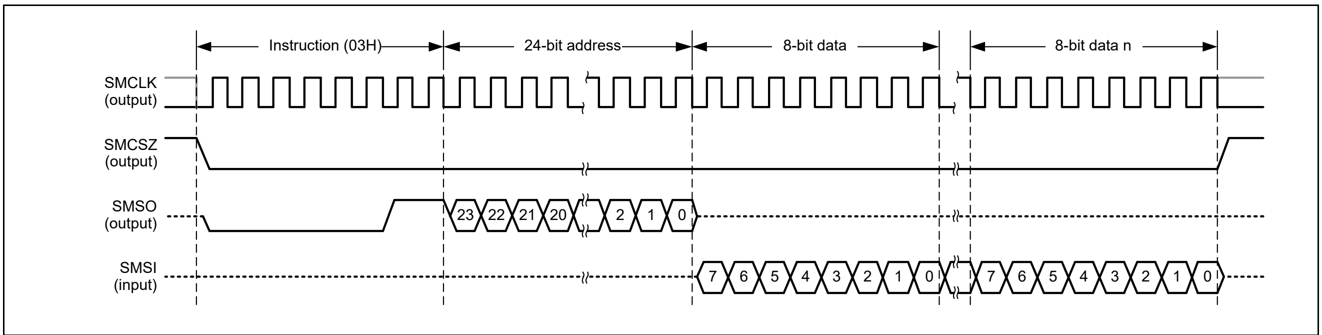


Figure 17.16 Bus Cycles for Standard Reading

(3) Instruction for Fast Read

Fast read is a method of reading that supports faster communications clock speeds than that for standard reading.

When an SPI bus cycle starts, the SMCSZ signal becomes active and 0BH is output as an instruction code. Next, a 24-bit address and 1-byte dummy data are transmitted, followed by reception of data.

The first two dummy cycles are used for selecting instruction-omission mode. When instruction-omission mode is selected, the same instruction as this one is also applied to the next SPI bus cycle and instruction code transmission in the next SPI bus cycle is omitted. For details of instruction-omission mode, see section 17.4.6, Instruction-Omission Mode Control.

For switching to fast read, use SFMSMD.SFMRM2, SFMSMD.SFMRM1, or SFMSMD.SFMRM0.

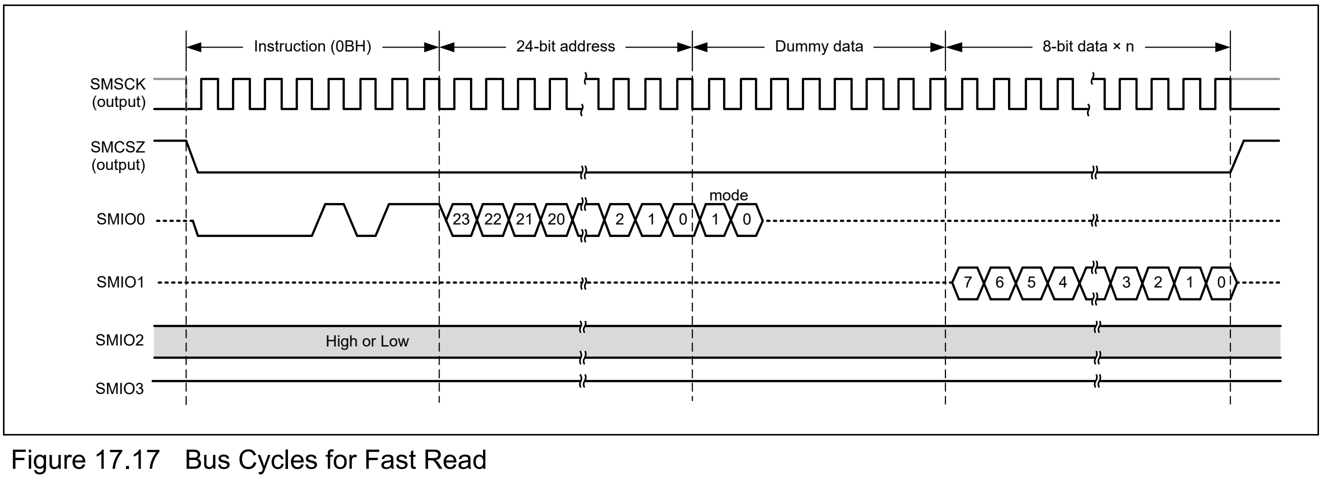


Figure 17.17 Bus Cycles for Fast Read

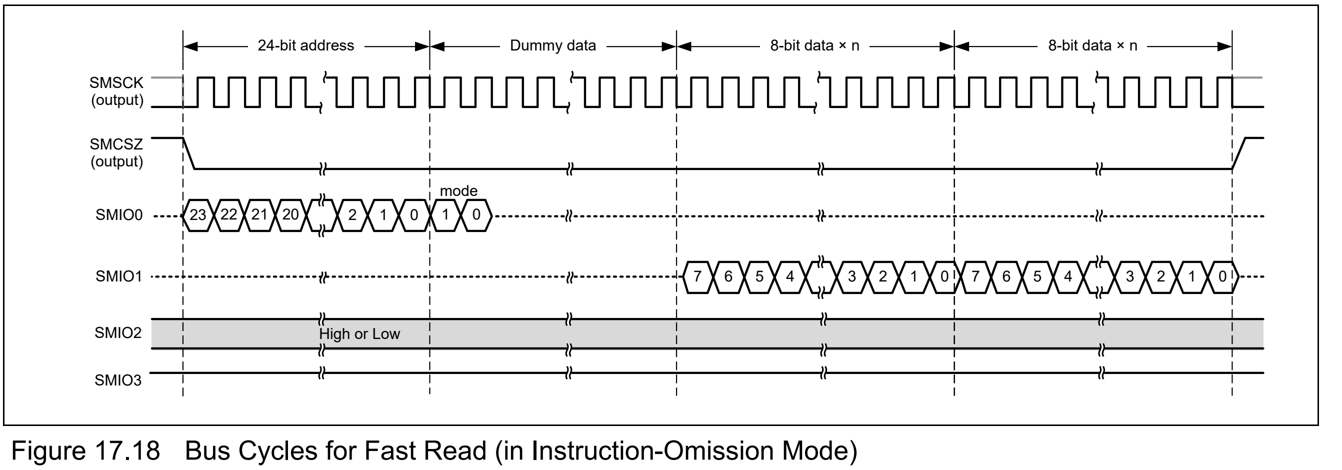


Figure 17.18 Bus Cycles for Fast Read (in Instruction-Omission Mode)

Caution: Use fast read only for serial flash ROMs that support fast read.

(4) Instruction for Fast Read Dual Output

Fast read dual output is a method of reading in which two signal lines are used for reception of data.

When an SPI bus cycle starts, the SMCSZ signal becomes active and 3BH is output as an instruction code. Next, a 24-bit address and 1-byte dummy data are transmitted, followed by reception of the data by using both the SMIO1 and SMIO0 pins.

Even-numbered bits of data are received via the SMIO0 pin, and odd-numbered bits are received via the SMIO1 pin.

The first two dummy cycles are used for selecting instruction-omission mode. When instruction-omission mode is selected, the same instruction as this one is also applied to the next SPI bus cycle and the transmission of instruction codes in the next SPI bus cycle is omitted. For details of instruction-omission mode, see section 17.4.6, Instruction-Omission Mode Control.

For switching to fast read dual output, use SFMSMD.SFMRM2, SFMSMD.SFMRM1, or SFMSMD.SFMRM0.

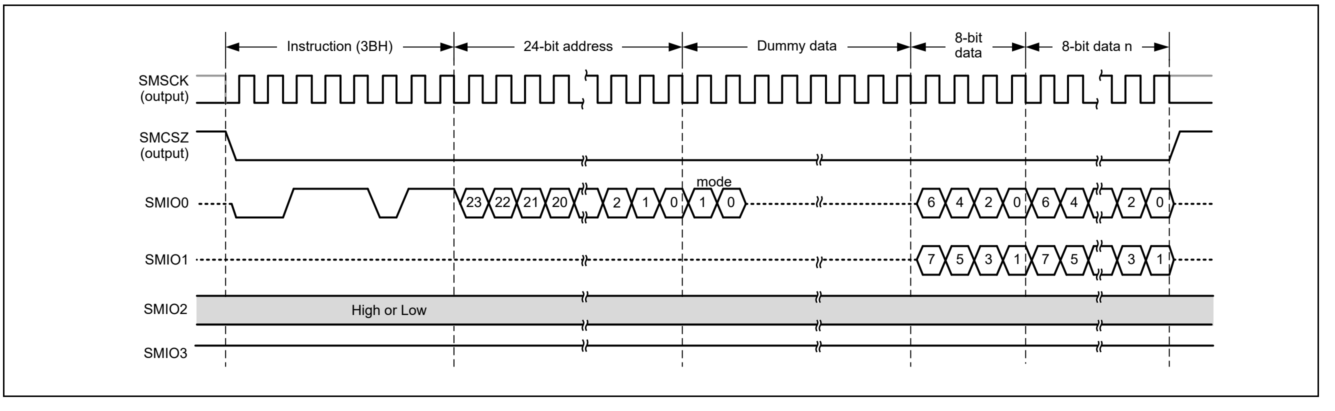


Figure 17.19 Bus Cycles for Fast Read Dual Output

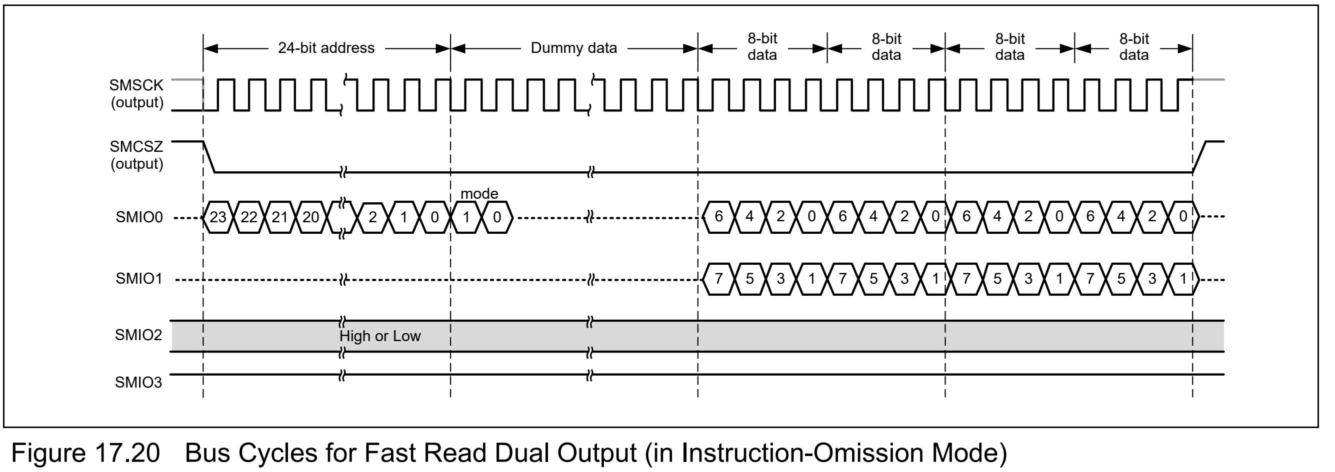


Figure 17.20 Bus Cycles for Fast Read Dual Output (in Instruction-Omission Mode)

Caution: Use fast read dual output for serial flash ROMs that support fast read dual output.

(5) Instruction for Fast Read Dual I/O

Fast read dual I/O is a method of reading in which two signal lines are used for transmission of addresses and reception of data.

When an SPI bus cycle starts, the SMCSZ signal becomes active and BBH is output as an instruction code. Next, a 24-bit address and 1-byte dummy data are transmitted by using the SMIO1 and SMIO0 pins, followed by reception of the data by using the SMIO1 and SMIO0 pins.

When an address and dummy data are transmitted and data is received, the SMIO0 pin is used for even-numbered bits and the SMIO1 pin is used for odd-numbered bits.

The first two dummy cycles are used for selecting instruction-omission mode. When instruction-omission mode is selected, the same instruction as this one is also applied to the next SPI bus cycle and the transmission of instruction codes in the next SPI bus cycle is omitted. For details of instruction-omission mode, see section 17.4.6, Instruction-Omission Mode Control.

For switching to fast read dual I/O, use SFMSMD.SFMRM2, SFMSMD.SFMRM1, or SFMSMD.SFMRM0.

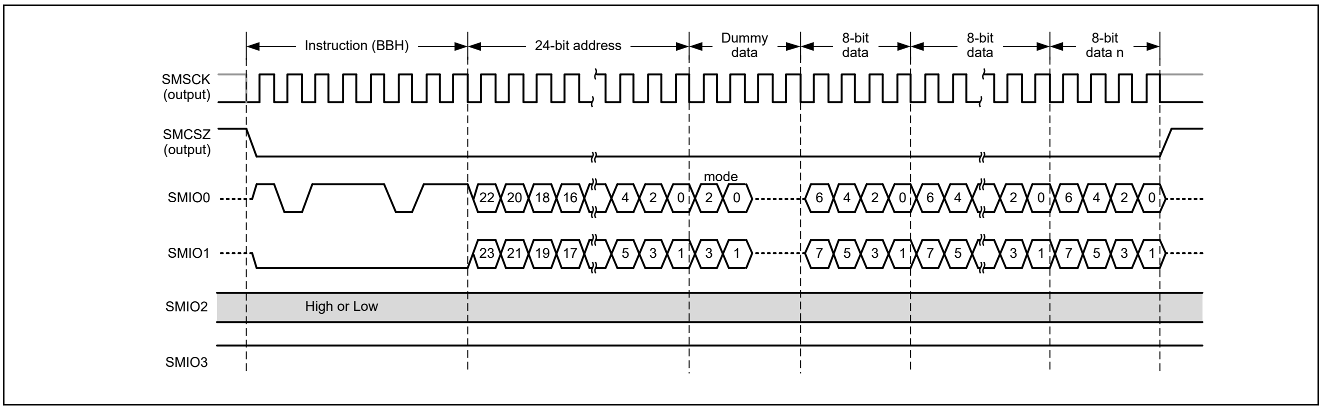


Figure 17.21 Bus Cycles for Fast Read Dual I/O

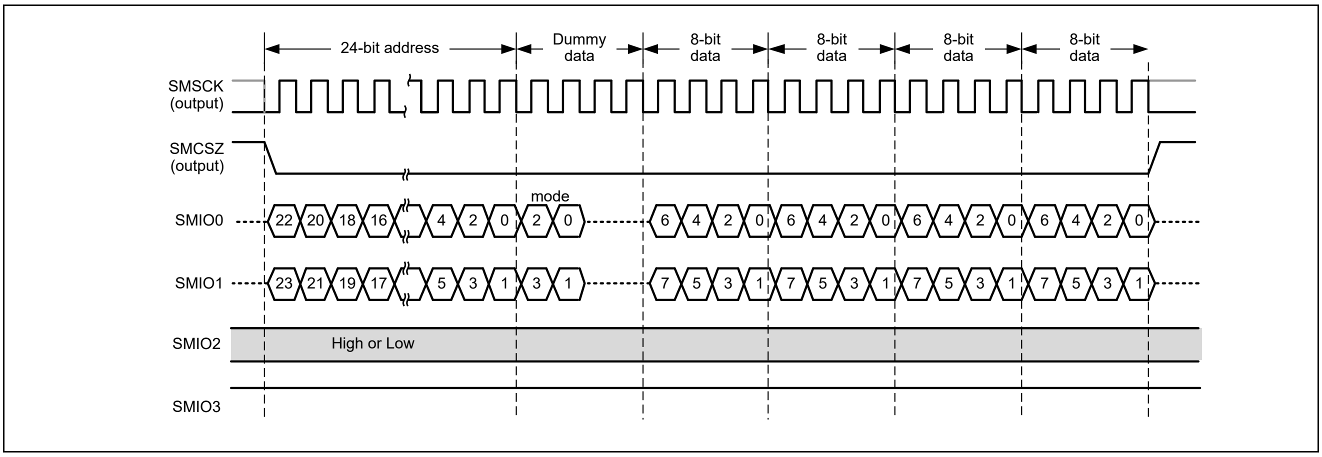


Figure 17.22 Bus Cycles for Fast Read Dual I/O (in Instruction-Omission Mode)

Caution: Use fast read dual I/O only for serial flash ROMs that support fast read dual I/O.

(6) Instruction for Fast Read Quad Output

Fast read quad output is a method of reading in which four signal lines are used for reception of data.

When an SPI bus cycle starts, the SMCSZ signal becomes active and 6BH is output as an instruction code. Next, a 24-bit address and 1-byte dummy data are transmitted by using the SMIO0 pin, followed by reception of the data by using the SMIO0, SMIO1, SMIO2, and SMIO3 pins.

The first two dummy cycles are used for selecting instruction-omission mode. When instruction-omission mode is selected, the same instruction as this one is also applied to the next SPI bus cycle and the transmission of instruction codes in the next SPI bus cycle is omitted. For details of instruction-omission mode, see section 17.4.6, Instruction-Omission Mode Control.

For switching to fast read quad output, use SFMSMD.SFMRM2, SFMSMD.SFMRM1, or SFMSMD.SFMRM0.

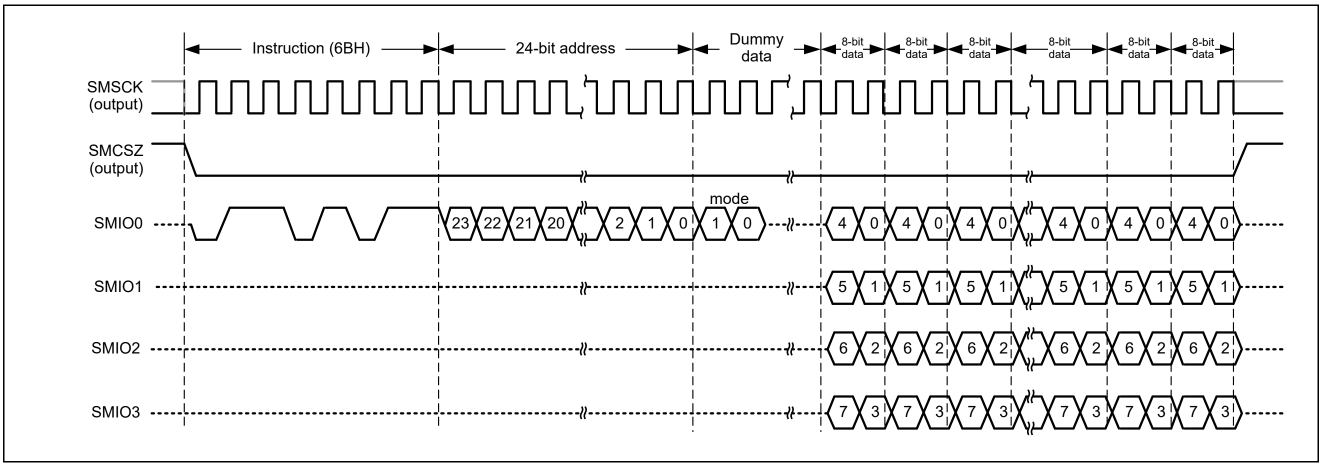


Figure 17.23 Bus Cycles for Fast Read Quad Output

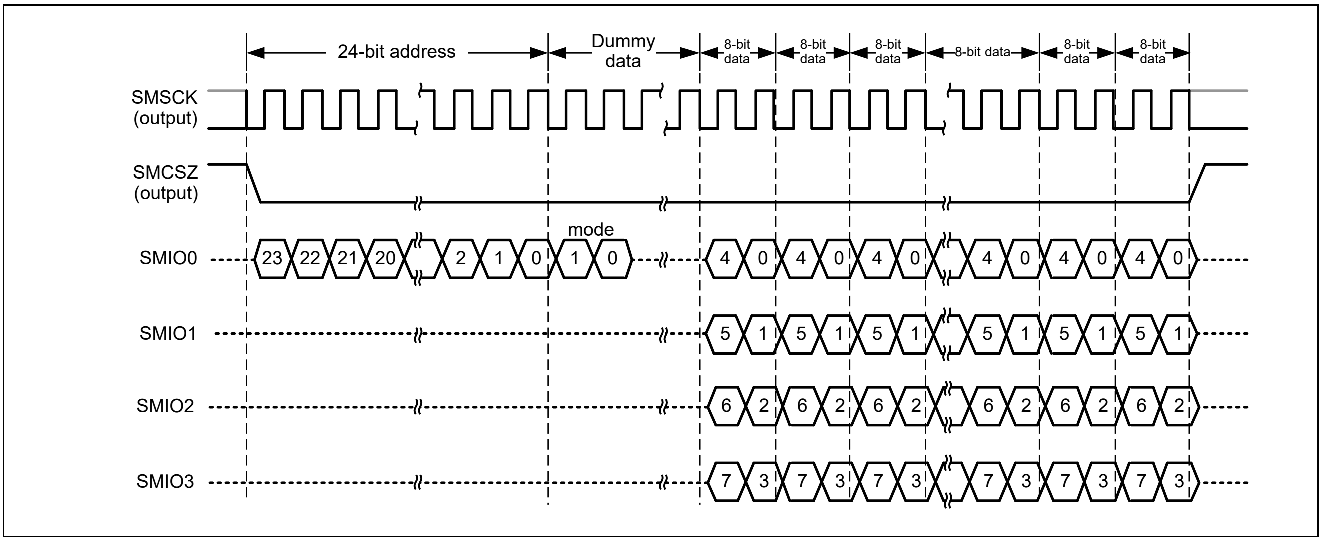


Figure 17.24 Bus Cycles for Fast Read Quad Output (in Instruction-Omission Mode)

Caution: Use fast read quad output only for serial flash ROMs that support fast read quad output.

(7) Instruction for Fast Read Quad I/O

Fast read quad I/O is a method of reading in which four signal lines are used for transmission of addresses and reception of data.

When an SPI bus cycle starts, the SMCSZ signal becomes active and EBH is output as an instruction code. Next, a 24-bit address and 1-byte dummy byte are transmitted by using the SMIO0, SMIO1, SMIO2, and SMIO3 pins, followed by reception of the data by using the SMIO0, SMIO1, SMIO2, and SMIO3 pins.

The first two dummy cycles are used for selecting instruction-omission mode. When instruction-omission mode is selected, the same instruction as this one is also applied to the next SPI bus cycle and instruction code transmission in the next SPI bus cycle is omitted. For details of instruction-omission mode, see section 17.4.6, Instruction-Omission Mode Control.

For switching to fast read quad I/O, use SFMSMD.SFMRM2, SFMSMD.SFMRM1, or SFMSMD.SFMRM0.

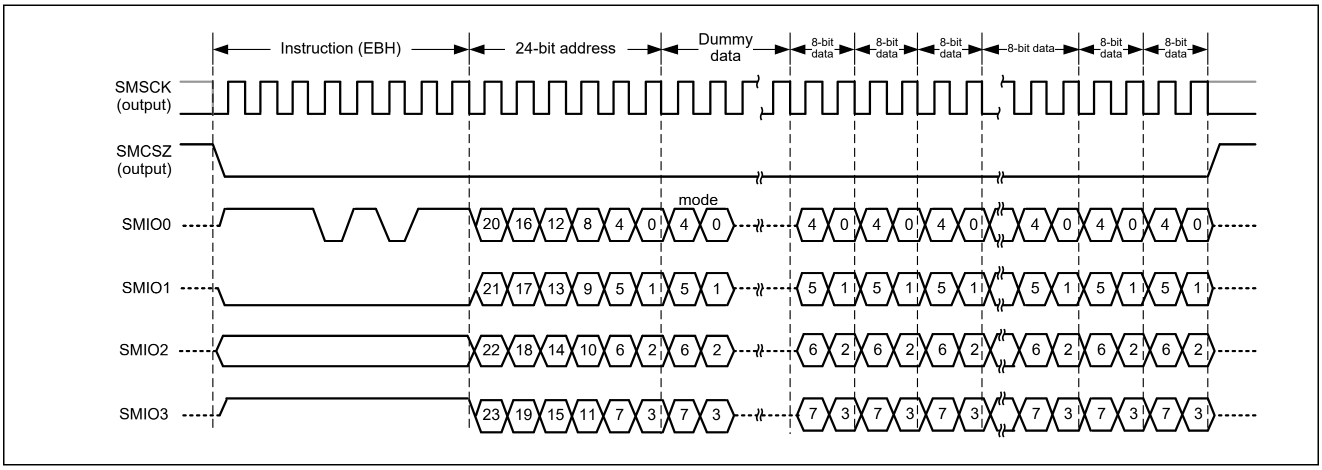


Figure 17.25 Bus Cycles for Fast Read Quad I/O

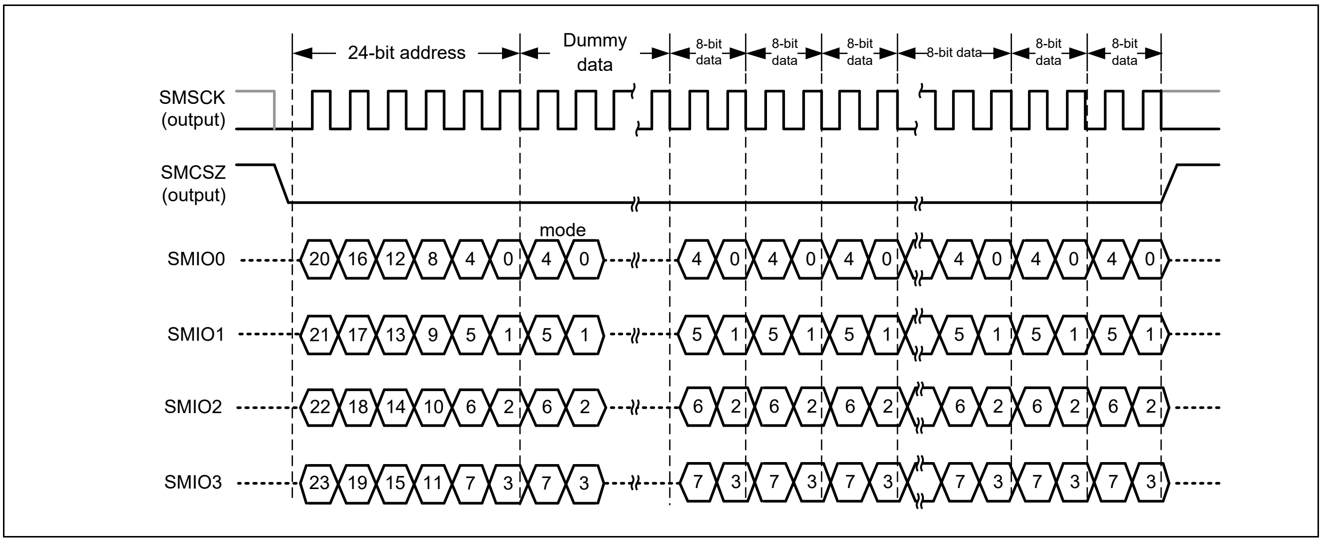


Figure 17.26 Bus Cycles for Fast Read Quad I/O (in Instruction-Omission Mode)

Caution: Use fast read quad I/O only for serial flash ROMs that support fast read quad I/O.

(8) Instruction for Release from Deep Power-Down

This is an instruction to return the serial flash ROM from the deep power-down state.

When an SPI bus cycle starts, the SMCSZ signal becomes active and ABH is output as an instruction code.

The instruction for release from deep power-down is automatically issued after release from the reset state.

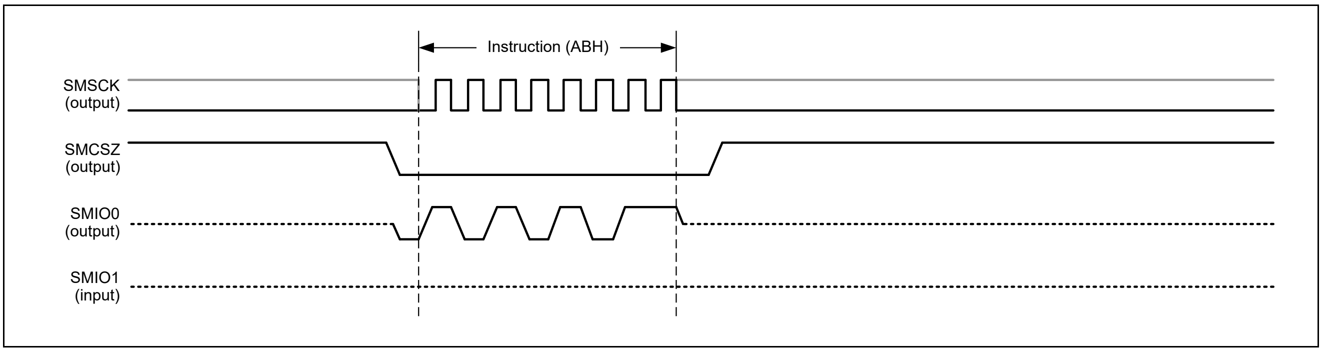


Figure 17.27 Bus Cycles for Release from Deep Power-Down

(9) Instruction for Entering 4-Byte Mode

This instruction is used to set the address width of the serial ROM to 4 bytes.

When an SPI bus cycle starts, the serial device select signal becomes active and the instruction code (B7H) is output.

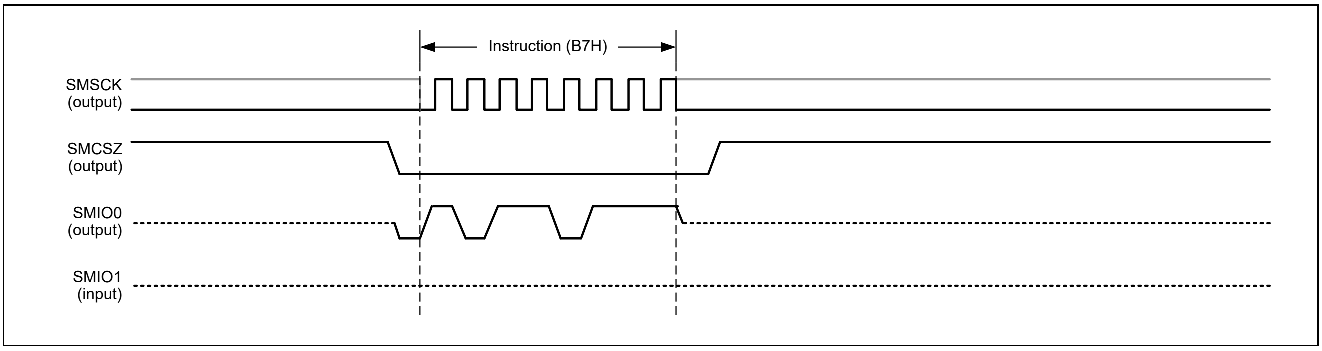


Figure 17.28 Bus Cycles for Entering 4-Byte Mode

(10) Instruction for Exiting 4-Byte Mode

This instruction is used to set the address width of the serial flash ROM to 3 bytes.

When an SPI bus cycle starts, the SMCSZ signal becomes active and an instruction code of E9H is output.

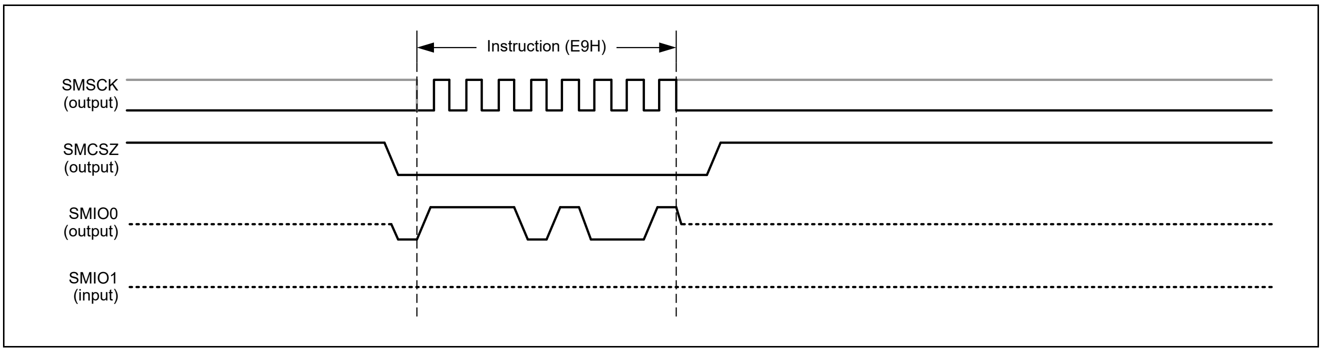


Figure 17.29 Bus Cycles for Exiting 4-Byte Mode

(11) Write Enable Instruction

This instruction is used to enable changing of the address width of the serial flash ROM.

When an SPI bus cycle starts, the SMCSZ signal becomes active and an instruction code of 06H is output.

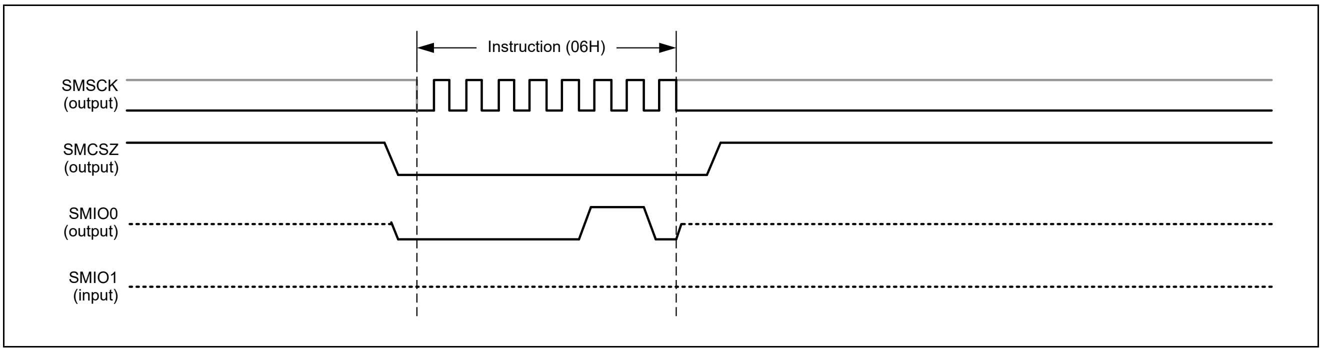


Figure 17.30 Bus Cycles for Enabling Writing

17.4.4 Modifying the SPI Bus Cycle

(1) ROM Reading by Individual Conversion

Internal system bus cycles for reading of the serial flash ROM are converted one by one to SPI bus cycles.

When a bus cycle for reading of the serial flash ROM is detected, the SMCSZ signal becomes active and an SPI bus cycle starts. When necessary data is received from the serial flash ROM, the SRMCSZ signal becomes inactive and the SPI bus cycle ends.

After that, when the next bus cycle for reading of the serial flash ROM is detected, the SMCSZ signal becomes active again after securing the minimum width at high level of the SMCSZ signal, and a new SPI bus cycle starts.

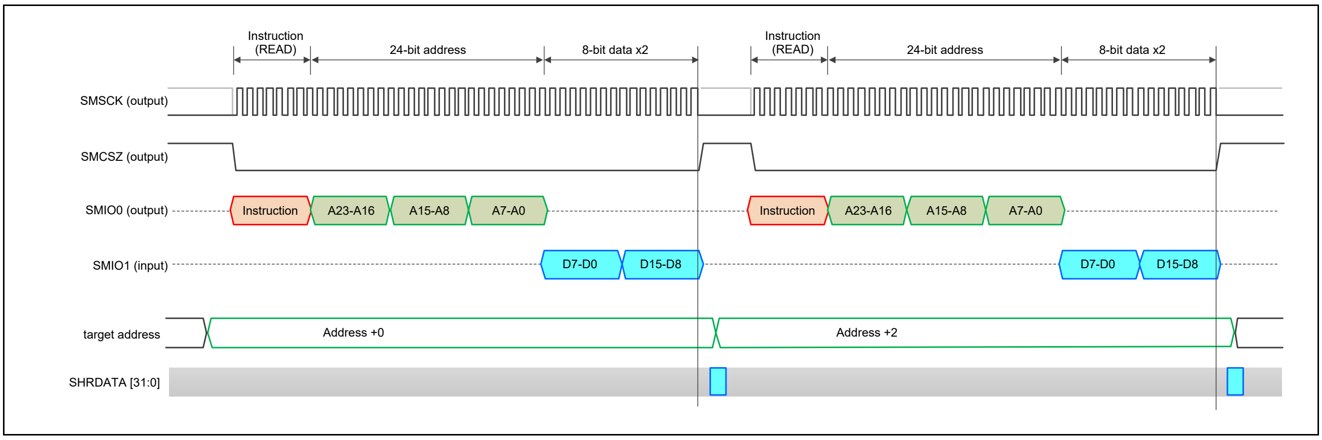


Figure 17.31 Continuous Data Reading by Individual Conversion

(2) ROM Reading by Using Prefetching

In cases such as when CPU instructions are executed or blocks of data are transferred, data is often read from consecutive addresses of the ROM.

Serial flash ROMs have functionality to repeat data reception without reissuing an instruction code or address. However, converting CPU-issued bus cycles individually results in separate SPI bus cycles, making it difficult to use the serial flash ROM efficiently.

The serial flash ROM memory controller of an R-IN32M4-CL3 features prefetching. Prefetching is enabled by setting SFMSMD.SFMPFE to 1.

When prefetching is enabled, each byte following the last byte to have been read from the ROM is continuously received and stored in a buffer without waiting for the next request to read from ROM. Next, when the CPU reads the ROM, the addresses are compared and, if the addresses match, the data in the buffer is transferred to the CPU. If the addresses do not match, the data in the buffer is discarded and a new SPI bus cycle is issued.

The size of the buffer for prefetching is 18 bytes. If this buffer becomes full, the SPI bus cycle ends once. After that, when the data in the buffer is read and the buffer has free space, a new SPI bus cycle automatically starts and prefetching continues.

Prefetching allows efficient data transfer in such cases as fetching of an instruction or block data transfer where data is read from consecutive addresses without any intervals.

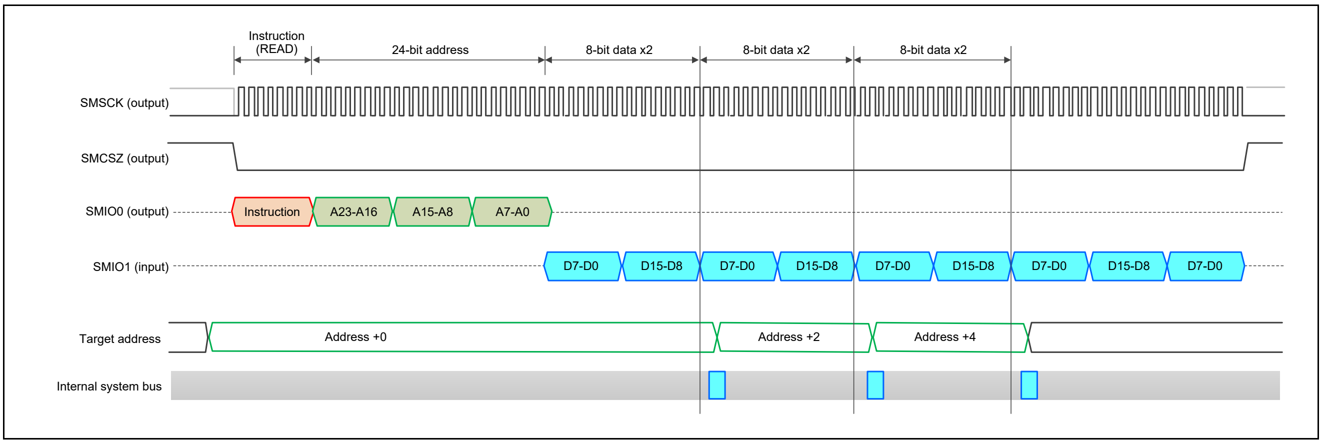


Figure 17.32 Continuous Data Reading by Using Prefetching

(3) Suspension of Prefetching

In the case of a bus cycle for reading from a given address in the ROM while serial transfer is already in progress for prefetching from another address, the current serial transfer that has become superfluous is suspended and a new SPI bus cycle starts.

Usually, such suspension of a serial transfer occurs at a byte boundary of received data. However, when SFMSMD.SFMPAE is set to 1, requests for suspension are accepted at locations other than byte boundaries. Note that, in the latter case, the serial flash ROM in use needs to support suspension of transfer at locations other than byte boundaries.

(4) ROM Reading by Using SPI Bus Cycle Extension

When a value other than 00B is set in SFMSMD.SFMSE1 or SFMSMD.SFMSE0, the supply of the SMSCK signal is stopped and the SMCSZ signal is kept at the low level, even after data has been acquired from the serial flash ROM, while the system waits for the next ROM read request with the SPI bus cycle on hold.

If the address of a next request to read from ROM directly follows that of the last request for reading, toggling of the SMSCK signal resumes and the subsequent data continues to be received. If the address of a next request to read from ROM does not directly follow that of the last request for reading, the SMCSZ signal is returned to the high level and the SPI bus cycle on hold is completed. After that, a new SPI bus cycle starts.

Use of this function reduces the overhead for instruction code and address transmission and allows efficient data transfer in such cases as when data is read intermittently from consecutive addresses.

The extension time of the SPI bus cycle is set by using SFMSMD.SFMSE1 and SFMSMD.SFMSE0. By default, the SMCSZ signal is extended by up to 33 serial clocks. After the specified extension time has elapsed, the SMCSZ signal is returned to the high level and the SPI bus cycle on hold is completed automatically.

Note that, if both SFMSMD.SFMSE1 and SFMSMD.SFMSE0 are set to 1, the SMCSZ signal is extended infinitely, in which case you need to take care on an increase in the power consumption of the serial flash ROM.

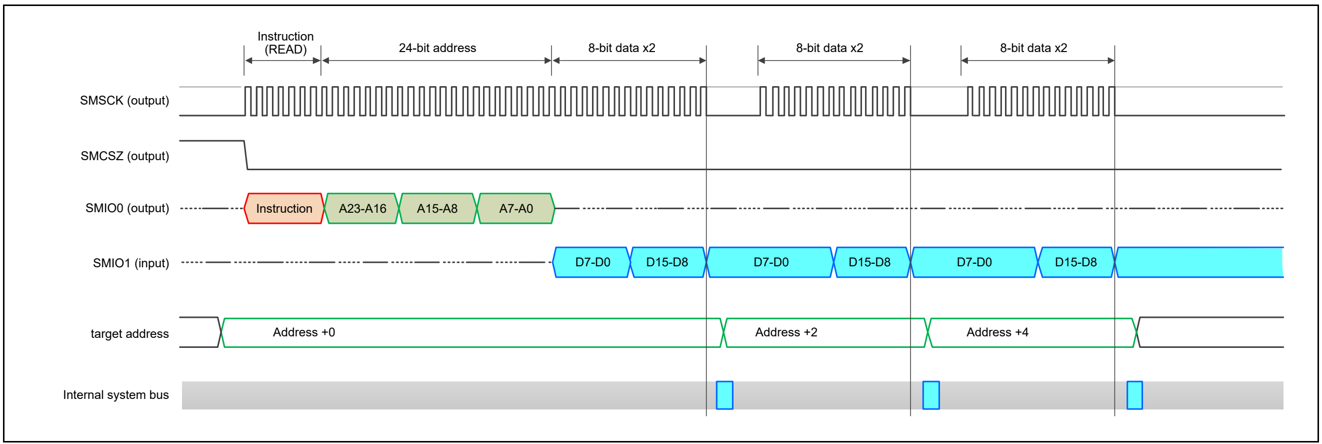


Figure 17.33 Continuous Data Reading by Using SPI Bus Cycle Extension

17.4.5 Automatic Release from the Deep Power-Down State

While in the deep power-down state, the serial flash ROM is unable to accept almost all kinds of instruction, including the read instruction, except for the release from deep power-down instruction.

On the other hand, since many serial flash ROMs perform power-on detection and internal logic initialization within the device in order to reduce the number of pins, their external pins do not include a reset input pin. Therefore, once the serial flash ROM enters the deep power-down state, it cannot be read until it is released from that state. This can lead to, for example, rebooting the system with a reset without switching it off causing a system malfunction.

To solve this problem, this serial flash ROM memory controller has an automatic deep power-down release function. After the release from deep power-down command is issued at the time the reset signal is de-asserted, the controller waits for 1025 cycles of the internal system bus clocks (SMSCK). When the system is configured to boot from the serial flash ROM, booting-up starts after this wait.

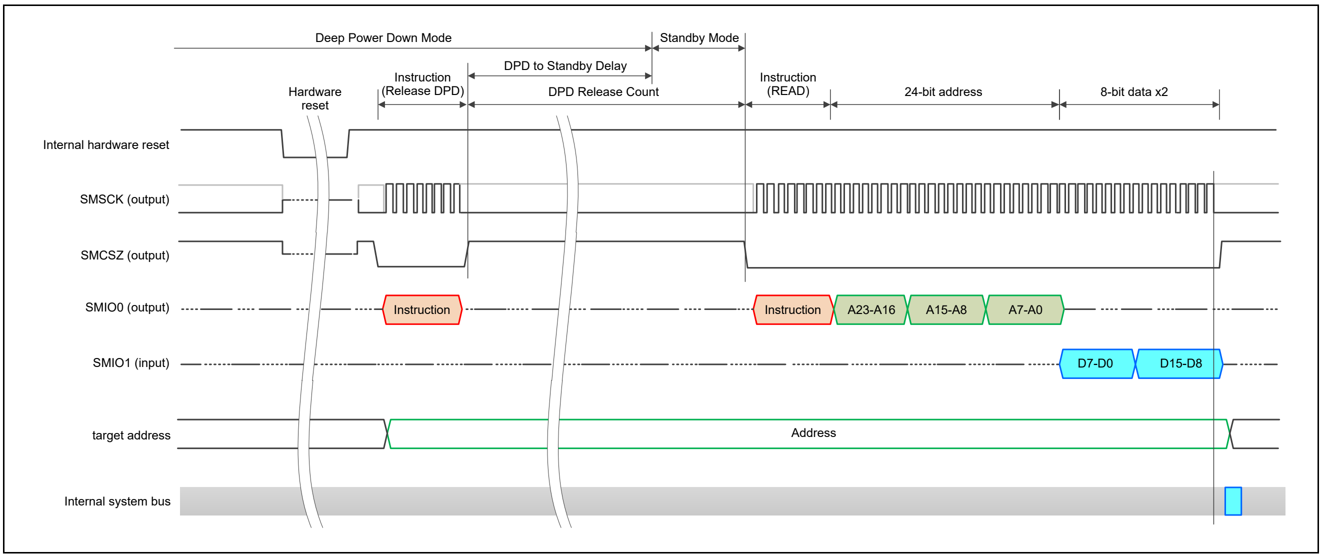


Figure 17.34 Operation for Automatic Release from the Deep Power-Down State

17.4.6 Instruction-Omission Mode Control

The latency of some commercially available serial ROM products can be reduced by omitting the reception of instruction codes for reading the ROM. The instruction code omission function for such serial ROM products is selected by the mode information received during a dummy cycle immediately before a cycle of the serial bus.

In dummy cycles of instructions for fast read, the serial flash ROM controller controls the instruction-omission mode of the serial ROM device by sending the values set in each half-byte of SFMSDC.SFMXD7–0 to the SMIO0, SMIO1, SMIO2, and SMIO3 pins during the first two cycles.

The control value for selecting the instruction-omission mode is unique to each serial ROM product, so the correct value must be set in SFMSDC.SFMXD7–0.

(1) Setting the Instruction-Omission Mode

When the value for selecting the instruction-omission mode specified for a serial ROM device is set in SFMSDC.SFMXD7–0 and SFMSDC.SFMXEN is set to 1, the value in SFMSDC.SFMXD7–0 is transferred to the serial ROM device during the first two cycles of the next Fast Read dummy read cycle, which places both the serial ROM controller and the serial ROM device in instruction-omission mode.

The actual completion of the procedure for the instruction-omission mode setting can be confirmed by reading 1 from SFMSDC.SFMXST.

Caution: Correctly set the value for selecting instruction-omission mode specified for the serial ROM device in SFMSDC.SFMXD7–0.
The serial ROM controller enters instruction-omission mode only from SFMSDC.SFMXEN regardless of the setting of SFMSDC.SFMXD7–0.

(2) Release from Instruction-Omission Mode

When the value for selecting release from instruction-omission mode specified for a serial ROM device is set in SFMSDC.SFMXD7–0 and SFMSDC.SFMXEN is set to 0, the value in SFMSDC.SFMXD7–0 is transferred to the serial ROM device during the first two cycles of the next Fast Read dummy read cycle, which releases both the serial ROM controller and the serial ROM device from instruction-omission mode.

The actual completion of the procedure for release from instruction-omission mode can be confirmed by reading 0 from SFMSDC.SFMXST.

Caution: Correctly set the value for selecting release from instruction-omission mode for the serial ROM device in SFMSDC.SFMXD7–0.
The serial ROM controller is released from instruction-omission mode only from SFMSDC.SFMXEN regardless of the setting of SFMSDC.SFMXD7–0.

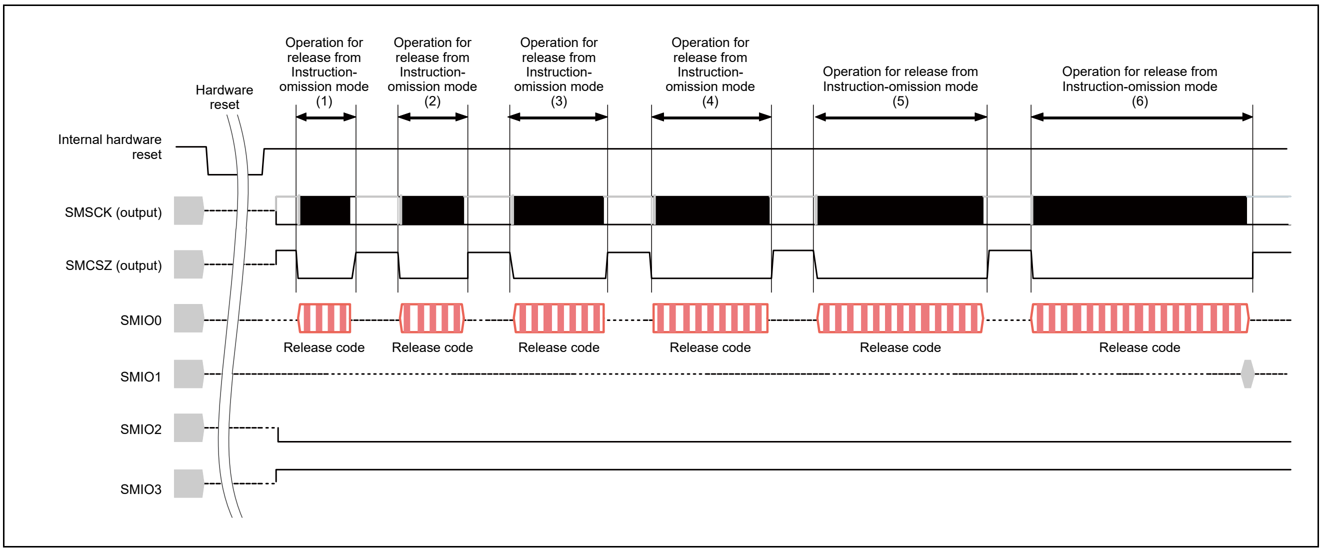


Figure 17.35 Automatic Release from Instruction-Omission Mode

(3) Automatic Release from Instruction-Omission Mode at the Time of a Reset

An R-IN32M4-CL3 handles automatic release from instruction-omission mode. Since the operating condition of the serial ROM device cannot be specified before the input of the reset signal, so from among the multiple serial ROM device states, it can be assumed to have been released from instruction-omission mode.

Table 17.3 Release Codes Used for Automatic Release from Instruction-Omission Mode

Releasing Operation	Release Code	Number of Bits	Target
Release from instruction omission (1)	00000011	8	24-bit address fast read quad IO
Release from instruction omission (2)	0000001111	10	32-bit address fast read quad IO
Release from instruction omission (3)	00000011111111	14	24-bit address fast read dual IO
Release from instruction omission (4)	000000111111111111	18	32-bit address fast read dual IO
Release from instruction omission (5)	00000011111111111111111111	26	24-bit address fast read 24-bit address fast read dual output 24-bit address fast read quad output
Release from instruction omission (6)	0000001111111111111111111111111111	34	32-bit address fast read 32-bit address fast read dual output 32-bit address fast read quad output

17.4.7 States of the SMIO2 and SMIO3 Pins

The states of the SMIO2 and SMIO3 pins change in conjunction with read mode of the serial interface selected by SFMSMD.SFMRM2–0.

Table 17.4 States of SMIO2 and SMIO3 Pins

SFMSMD.SFMRM2–0	SMIO2 Pin State Note1	SMIO3 Pin State Note2	Note
111	Setting prohibited		
110			
101	Input/output operation as a serial data line (Hi-Z in standby state)	Input/output operation as a serial data line (Hi-Z in standby state)	Fast Read Quad I/O
100			Fast Read Quad Output
011	SFMPMD.SFMWPL value output (Initial value: low level output)	High level output	Fast Read Dual I/O
010			Fast Read Dual Output
001			Fast Read
000			Reading (initial state)

Notes 1. The /WP pin function may be multiplexed with the SMIO2 pin of the serial ROM device.

2. The /HOLD or /RESET pin function may be multiplexed with the SMIO3 pin of the serial ROM device.

17.4.8 Direct Communications

(1) About Direct Communications

While the serial flash ROM memory controller of an R-IN32M4-CL3 is capable of converting ROM read bus cycles to SPI bus cycles automatically to read serial ROM data, the serial flash ROM has various functions such as reading, erasure, and programming of ID information, and reading of status information, in addition to reading of memory data. However, the instruction sets are not standardized among vendors and devices. As a means to perform these operations, therefore, transfer using arbitrary software-controlled SPI bus cycles is possible in addition to the usual three-wire serial interface. In an R-IN32M4-CL3, this is called direct communications.

(2) Direct Communications Mode

To handle direct communications with the serial flash ROM, set SFMCMD.DCOM to 1 to select direct communications mode.

In direct communications mode, normal reading of the ROM is prohibited. For the transition from direct communications mode to normal ROM access mode, clear SFMCMD.DCOM to 0.

Cautions 1. If the transfer is in progress after SFMCMD.DCOM has been rewritten, the mode will be changed after the completion of the transfer. If prefetching is in progress, the mode will be changed after the end of the transfer following the completion of the ongoing prefetching of one byte of data.

2. The program for switching direct communications mode and ROM access mode must be executed in a location other than the serial flash ROM. Also, when changing the mode, make sure that the cache fill operation is not performed and no access is made from a bus master such as the DMA controller.

(3) SPI Bus Cycle Generation in Direct Communications Mode

An SPI bus cycle for the serial flash ROM refers to a period of time during which SMCSZ is active. If the mode is changed to direct communications mode, SMCSZ becomes active (outputs the low level) in the first access to the communications port register (SFMCOM). After a series of I/O operations are performed via SFMCOM, SMCSZ becomes inactive when SFMCMD.DCOM is cleared to 0.

At this time, writing to the SFMCOM port is converted to transmission of one byte to the SPI bus. Similarly, reading from the SFMCOM port is converted to reception of one byte from the SPI bus.

Cautions 1. While direct communications mode is selected, writing to a register other than SFMCMD (such as SFMSMD, SFMSSC, SFMSKC, SFMSST, SFMCST, SFMSIC, SFMSAC, SFMSDC, SFMSPC, SFMPMD, and SFMDTC) is prohibited.

2. The completion of an SPI bus cycle by writing to a register other than SFMCMD is not guaranteed as official functionality.

17.5 Example of Configuration

This section describes the settings of the registers and serial flash ROM for standard reading, fast read dual I/O, and fast read quad I/O. The settings assume use of "TS-R-RIN32M4" from TESSERA TECHNOLOGY and serial flash ROM "MX25L6433F" in the starter kit from IAR Systems. Please make appropriate settings according to your usage environment. For details about the registers of the serial flash ROM memory controller, see section 17.2, Control Registers.

17.5.1 Standard Reading

(1) Operation

Standard reading is a common method of reading supported by most serial flash ROMs. For details, see section 17.4.3, SPI Instruction Set for Use in Access to the Serial Flash ROM.

(2) Settings of Registers

The following tables list examples of the settings of the registers for standard reading of the serial flash ROM.

(a) SFMSMD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMCCE	0			SFMOSW	SFMOHW	SFMOEX	SFMMD3	SFMPAE	SFMPFE	SFMSE[1:0]		0	SFMRD[2:0]		

Table 17.5 SFMSMD Register Settings for Standard Reading

Bit Name	Description
SFMCCE	Selects the read instruction code. 0: Read instruction code that is set in the SFMSIC register (initial value). 1: Default read instruction code of each read format.
SFMOSW	Selects adjustment of the setup time during serial data output. 0: Does not extend the high-level period of SMSCK during serial data output (initial value). 1: Extends the high-level period of SMSCK by one clock cycle during serial data output. This function takes effect only during serial data output.
SFMOHW	Selects adjustment of the hold time during serial data output. 0: Does not extend the low-level period of SMSCK during serial data output (initial value). 1: Extends the low-level period of SMSCK by one clock cycle during serial data output. This function takes effect only during serial data output.
SFMOEX	Extends the output enable signal for the serial interface I/O buffer. 0: Does not extend the output enable period of serial data (initial value). 1: Extends the output enable period of serial data by one SMSCK cycle. Only the output enable signal is extended; output data is not extended.
SFMMD3	Selects the SPI mode. 0: SPI mode 0 1: SPI mode 3 (initial value)
SFMPAE	Selects stopping of prefetching at locations other than byte boundaries. 0: Disables prefetching at locations other than byte boundaries (initial value). 1: Enables prefetching at locations other than byte boundaries.
SFMPFE	Selects prefetching. 0: Disables prefetching (initial value). 1: Enables prefetching.
SFMSE[1:0]	Selects extension of the SMCSZ signal after access to the SPI bus. 00: Does not extend the SMCSZ signal. 01: Extends the SMCSZ signal by up to 33 serial clock cycles (initial value). 10: Extends the SMCSZ signal by up to 129 serial clock cycles. 11: Extends the SMCSZ signal infinitely.
SFMRD[2:0]	Selects the read mode of the serial flash ROM. 000: Standard reading (initial value)

(b) SFMSSC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										SFMSLD	SFMSHD	SFMSW[3:0]			

Table 17.6 SFMSSC Register Settings for Standard Reading

Bit Name	Description
SFMSLD	Selects the output timing of the SMCSZ signal. 0: Outputs SMSCK 0.5 clock cycles before the first rising edge of SMCLK. 1: Outputs SMSCK 1.5 clock cycles before the first rising edge of SMCLK (initial value).
SFMSHD	Selects the timing for de-asserting the SMCSZ signal. 0: De-asserts SMSCK 0.5 clock cycles after the last rising edge of SMCLK. 1: De-asserts SMSCK 1.5 clock cycles after the last rising edge of SMCLK (initial value).
SFMSW[3:0]	Selects the minimum width at high level of the SMCSZ signal. <table><tr><td>SFMSW3</td><td>SFMSW2</td><td>SFMSW1</td><td>SFMSW0</td><td>Minimum width at high level of SMCSZ signal</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>8 SMSCK cycles (initial value)</td></tr></table>

(c) SFMSKC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										SFMDTY	SFMDV[4:0]				

Table 17.7 SFMSKC Register Settings for Standard Reading

Bit Name	Description
SFMDTY	Selects duty cycle correction for the SMSCK signal. 0: The SMSCK signal is not adjusted. 1: Delays the rising edge of the SMSCK signal by 0.5 cycles of HCLK.
SFMDV[4:0]	Selects SMSCK based on HCLK. SFMDV4 / SFMDV3 / SFMDV2 / SFMDV1 / SFMDV0 / Serial clock selection: 0 / 0 / 0 / 1 / 0 / HCLK/4

(3) Serial Flash ROM Setting

When using standard reading for the serial flash ROM, set the Quad Enable (QE) bit of the status register to 0. The flow for the setting is described below. Since QE of the status register is a non-volatile register, its setting is retained even if the device is powered down.

A) Issue the WREN command to check that the Write Enable Latch (WEL) bit of the status register is set to 1 and the serial flash ROM is writable.
B) Issue the WRSR command to set the QE bit of the status register to 0.
C) Issue the RDSR command to check that the Write In Progress (WIP) bit of the status register is set to 0 and writing to the serial flash ROM is completed
D) Issue the RDSR command to check that the QE bit of the status register is 0.

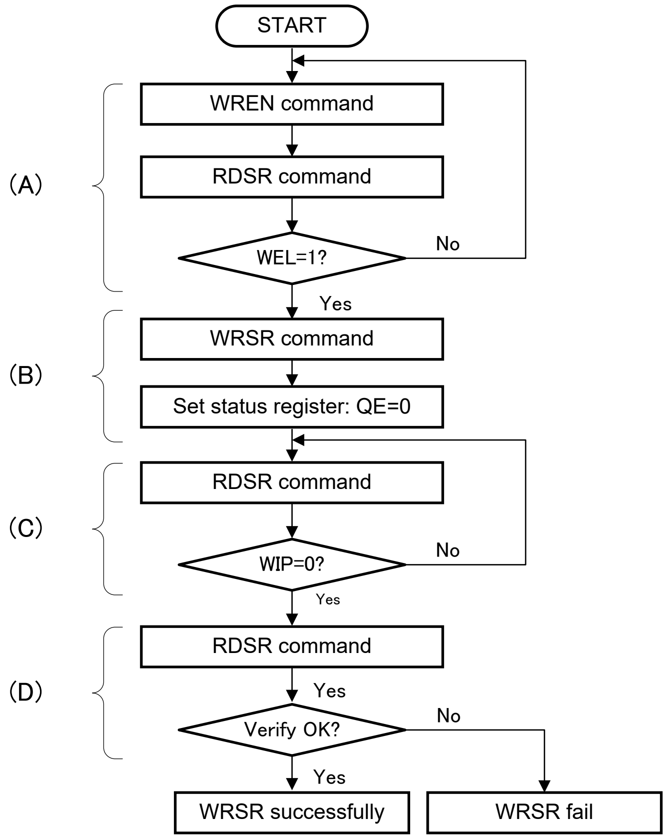


Figure 17.36 Serial Flash ROM Setting Flow (for Standard Reading)

17.5.2 Fast Read Dual I/O

(1) Operation

Fast read dual I/O is a method of reading in which two signal lines are used for transmission of addresses and reception of data. For details, see section 17.4.3, SPI Instruction Set for Use in Access to the Serial Flash ROM.

(2) Settings of Registers

The following tables list examples of the settings of the registers for fast read dual I/O of the serial flash ROM.

(a) SFMSMD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMCCE	0			SFMOSW	SFMOHW	SFMOEX	SFMMD3	SFMPAE	SFMPFE	SFMSE[1:0]		0	SFMRD[2:0]		

Table 17.8 SFMSMD Register Settings for Fast Read Dual I/O

Bit Name	Description
SFMCCE	Selects the read instruction code. 0: Read instruction code that is set in the SFMSIC register (initial value). 1: Default read instruction code of each read format.
SFMOSW	Selects adjustment of the setup time during serial data output. 0: Does not extend the high-level period of SMSCK during serial data output (initial value). 1: Extends the high-level period of SMSCK by one clock cycle during serial data output. This function takes effect only during serial data output.
SFMOHW	Selects adjustment of the hold time during serial data output. 0: Does not extend the low-level period of SMSCK during serial data output (initial value). 1: Extends the low level-period of SMSCK by one clock cycle during serial data output. This function takes effect only during serial data output.
SFMOEX	Extends the output enable signal for the serial interface I/O buffer. 0: Does not extend the output enable period of serial data (initial value). 1: Extends the output enable period of serial data by one SMSCK cycle. Only the output enable signal is extended; output data is not extended.
SFMMD3	Selects the SPI mode. 0: SPI mode 0 1: SPI mode 3 (initial value)
SFMPAE	Selects stopping of prefetching at locations other than byte boundaries. 0: Disables prefetching at locations other than byte boundaries (initial value). 1: Enables prefetching at locations other than byte boundaries.
SFMPFE	Selects prefetching. 0: Disables prefetching (initial value). 1: Enables prefetching.
SFMSE[1:0]	Selects extension of the SMCSZ (chip select) signal after access to the SPI bus. 00: Does not extend the SMCSZ signal. 01: Extends the SMCSZ signal by up to 33 serial clock cycles (initial value). 10: Extends the SMCSZ signal by up to 129 serial clock cycles. 11: Extends the SMCSZ signal infinitely.
SFMRD[2:0]	Selects the read mode of the serial flash ROM. 011: Fast Read Dual I/O

(b) SFMSSC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										SFMSLD	SFMSHD	SFMSW[3:0]			

Table 17.9 SFMSSC Register Settings for Fast Read Dual I/O

Bit Name	Description
SFMSLD	Selects the output timing of the SMCSZ signal. 0: Outputs SMSCK 0.5 clock cycles before the first rising edge of SMCLK. 1: Outputs SMSCK 1.5 clock cycles before the first rising edge of SMCLK (initial value).
SFMSHD	Selects the timing for de-asserting the SMCSZ signal. 0: De-asserts SMSCK 0.5 clock cycles after the last rising edge of SMCLK. 1: De-asserts SMSCK 1.5 clock cycles after the last rising edge of SMCLK (initial value).
SFMSW[3:0]	Selects the minimum width at high level of the SMCSZ signal.

SFMSW3	SFMSW2	SFMSW1	SFMSW0	Minimum width at high level of SMCSZ signal
0	1	1	1	8 SMSCK cycles (initial value)

(c) SFMSKC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										SFMDTY	SFMDV[4:0]				

Table 17.10 SFMSKC Register Settings for Fast Read Dual I/O

<table>
<tr><th>Bit Name</th><th>Description</th></tr>
<tr><td>SFMDTY</td><td>Selects duty cycle correction for the SMSCK signal.
0: The SMSCK signal is not adjusted.
1: Delays the rising edge of the SMSCK signal by 0.5 cycles of HCLK.</td></tr>
<tr><td>SFMDV[4:0]</td><td>Selects SMSCK based on HCLK.<table><tr><td>SFMDV4</td><td>SFMDV3</td><td>SFMDV2</td><td>SFMDV1</td><td>SFMDV0</td><td>Serial clock selection</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>HCLK/2</td></tr></table></td></tr>
</table>

(d) SFMSDC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMXD[7:0]								SFMX EN	SFMX ST	0		SFMDN[3:0]			

Table 17.11 SFMSDC Register Settings for Fast Read Dual I/O

<table>
<tr><th>Bit Name</th><th>Description</th></tr>
<tr><td>SFMXD[7:0]</td><td>Sets the value for selecting instruction-omission mode.</td></tr>
<tr><td>SFMXEN</td><td>Enables or disables instruction-omission mode.
0: Disables instruction-omission mode (initial value).</td></tr>
<tr><td>SFMXST</td><td>Instruction omission status
0: Indicates that operation is in progress in normal (instruction not omitted) mode (initial value).</td></tr>
<tr><td>SFMDN[3:0]</td><td>Selects the number of dummy cycles of fast read instructions.
<table>
<tr><th>SFMDN3</th><th>SFMDN2</th><th>SFMDN1</th><th>SFMDN0</th><th colspan="2">Number of Dummy Cycles of Fast Read Instructions</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="2">Default number of cycles of each instruction format</td></tr>
<tr><td>Fast Read Dual I/O</td><td>4 cycles of SMSCK</td></tr>
</table>
The number of dummy cycles depends on the device.
Please make a setting according to your usage environment.</td></tr>
</table>

(3) Serial Flash ROM Setting

When using fast read dual I/O for the serial flash ROM, set the QE bit of the status register and the Dummy Cycle (DC) bit of the configuration register to 0. The flow for the setting is described below. Since QE of the status register is a non-volatile register, its setting is retained even if the device is powered down.

A) Issue the WREN command to check that the WEL bit of the status register is set to 1 and the serial flash ROM is writable.
B) Issue the WRSR command to set the QE bit of the status register and the DC bit of the configuration register to 0.
C) Issue the RDSR command to check that the WIP bit of the status register is set to 0 and writing to the serial flash ROM is completed.
D) Issue the RDSR command to check that the QE bit of the status register is 0.
Issue the RDCR command to check that the DC bit of the configuration register is 0.

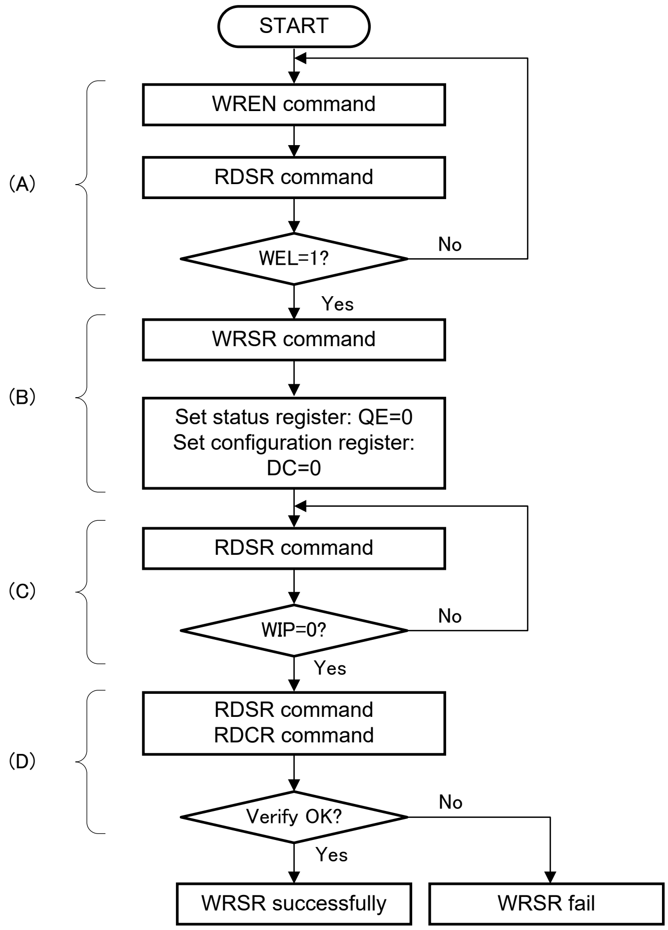


Figure 17.37 Serial Flash ROM Setting Flow (for Fast Read Dual I/O)

17.5.3 Fast Read Quad I/O

(1) Operation

Fast read quad I/O is a method of reading in which four signal lines are used for transmission of addresses and reception of data. For details, see section 17.4.3, SPI Instruction Set for Use in Access to the Serial Flash ROM.

(2) Settings of Registers

The following tables list examples of the settings of the registers for fast read quad I/O of the serial flash ROM.

(a) SFMSMD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMCCE	0			SFMOSW	SFMOHW	SFMOEX	SFMMD3	SFMPAE	SFMPFE	SFMSE[1:0]		0	SFMRD[2:0]		

Table 17.12 SFMSMD Register Settings for Fast Read Quad I/O

Bit Name	Description
SFMCCE	Selects the read instruction code. 0: Read instruction code that is set in the SFMSIC register (initial value). 1: Default read instruction code of each read format.
SFMOSW	Selects adjustment of the setup time during serial data output. 0: Does not extend the high-level period of SMSCK during serial data output (initial value). 1: Extends the high-level period of SMSCK by one clock cycle during serial data output. This function takes effect only during serial data output.
SFMOHW	Selects adjustment of the hold time during serial data output. 0: Does not extend the low-level period of SMSCK during serial data output (initial value). 1: Extends the low-level period of SMSCK by one clock cycle during serial data output. This function takes effect only during serial data output.
SFMOEX	Extends the output enable signal for the serial interface I/O buffer. 0: Does not extend the output enable period of serial data (initial value). 1: Extends the output enable period of serial data by one SMSCK cycle. Only the output enable signal is extended; output data is not extended.
SFMMD3	Selects the SPI mode. 0: SPI mode 0 1: SPI mode 3 (initial value)
SFMPAE	Selects stopping of prefetching at locations other than byte boundaries. 0: Disables prefetching at locations other than byte boundaries (initial value). 1: Enables prefetching at locations other than byte boundaries.
SFMPFE	Selects prefetching. 0: Disables prefetching (initial value). 1: Enables prefetching.
SFMSE[1:0]	Selects extension of the SMCSZ (chip select) signal after access to the SPI bus. 00: Does not extend the SMCSZ signal. 01: Extends the SMCSZ signal by up to 33 serial clock cycles (initial value). 10: Extends the SMCSZ signal by up to 129 serial clock cycles. 11: Extends the SMCSZ signal infinitely.
SFMRD[2:0]	Selects the read mode of the serial flash ROM. 101: Fast Read Quad I/O

(b) SFMSSC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										SFMSLD	SFMSHD	SFMSW[3:0]			

Table 17.13 SFMSSC Register Settings for Fast Read Quad I/O

<table>
<tr><th>Bit Name</th><th>Description</th></tr>
<tr><td>SFMSLD</td><td>Selects the output timing of the SMCSZ signal.
0: Outputs SMSCK 0.5 clock cycles before the first rising edge of SMCLK.
1: Outputs SMSCK 1.5 clock cycles before the first rising edge of SMCLK (initial value).</td></tr>
<tr><td>SFMSHD</td><td>Selects the timing for de-asserting the SMCSZ signal.
0: De-asserts SMSCK 0.5 clock cycles after the last rising edge of SMCLK.
1: De-asserts SMSCK 1.5 clock cycles after the last rising edge of SMCLK (initial value).</td></tr>
<tr><td>SFMSW[3:0]</td><td>Selects the minimum width at high level of the SMCSZ signal.
<table>
<tr><th>SFMSW3</th><th>SFMSW2</th><th>SFMSW1</th><th>SFMSW0</th><th>Minimum width at high level of SMCSZ signal</th></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>8 SMSCK cycles (initial value)</td></tr>
</table>
</td></tr>
</table>

(c) SFMSKC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										SFMDTY	SFMDV[4:0]				

Table 17.14 SFMSKC Register Settings for Fast Read Quad I/O

Bit Name	Description
SFMDTY	Selects duty cycle correction for the SMSCK signal. 0: The SMSCK signal is not adjusted. 1: Delays the rising edge of the SMSCK signal by 0.5 cycles of HCLK.
SFMDV[4:0]	Selects SMSCK based on HCLK. <table><tr><td>SFMDV4</td><td>SFMDV3</td><td>SFMDV2</td><td>SFMDV1</td><td>SFMDV0</td><td>Serial clock selection</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>HCLK/2</td></tr></table>

(d) SFMSDC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFMXD[7:0]								SFMX EN	SFMX ST	0		SFMDN[3:0]			

Table 17.15 SFMSDC Register Settings for Fast Read Quad I/O

<table>
<tr><th>Bit Name</th><th>Description</th></tr>
<tr><td>SFMXD[7:0]</td><td>Sets the value for selecting instruction-omission mode.</td></tr>
<tr><td>SFMXEN</td><td>Enables or disables instruction-omission mode.
0: Disables instruction-omission mode (initial value).</td></tr>
<tr><td>SFMXST</td><td>Instruction omission status
0: Indicates that operation is in progress in normal (instruction not omitted) mode (initial value).</td></tr>
<tr><td>SFMDN[3:0]</td><td>Selects the number of dummy cycles of fast read instructions.
<table>
<tr><th>SFMDN3</th><th>SFMDN2</th><th>SFMDN1</th><th>SFMDN0</th><th colspan="2">Number of Dummy Cycles of Fast Read Instructions</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="2">Default number of cycles of each instruction format</td></tr>
<tr><td>Fast Read Quad I/O</td><td>6 cycles of SMSCK</td></tr>
</table>
The number of dummy cycles depends on the device.
Please make a setting according to your usage environment.</td></tr>
</table>

(3) Serial Flash ROM Setting

When using fast read quad I/O for the serial flash ROM, set the QE bit of the status register to 1 and the DC bit of the configuration register to 0. The flow for the setting is described below. Since QE of the status register is a non-volatile register, its setting is retained even if the device is powered down.

A) Issue the WREN command to check that the WEL bit of the status register is set to 1 and the serial flash ROM is writable.
B) Issue the WRSR command to set the QE bit of the status register to 1 and the DC bit of the configuration register to 0.
C) Issue the RDSR command to check that the WIP bit of the status register is set to 0 and writing to the serial flash ROM is completed.
D) Issue the RDSR command to check that the QE bit of the status register is 1.
Issue the RDCR command to check that the DC bit of the configuration register is 0.

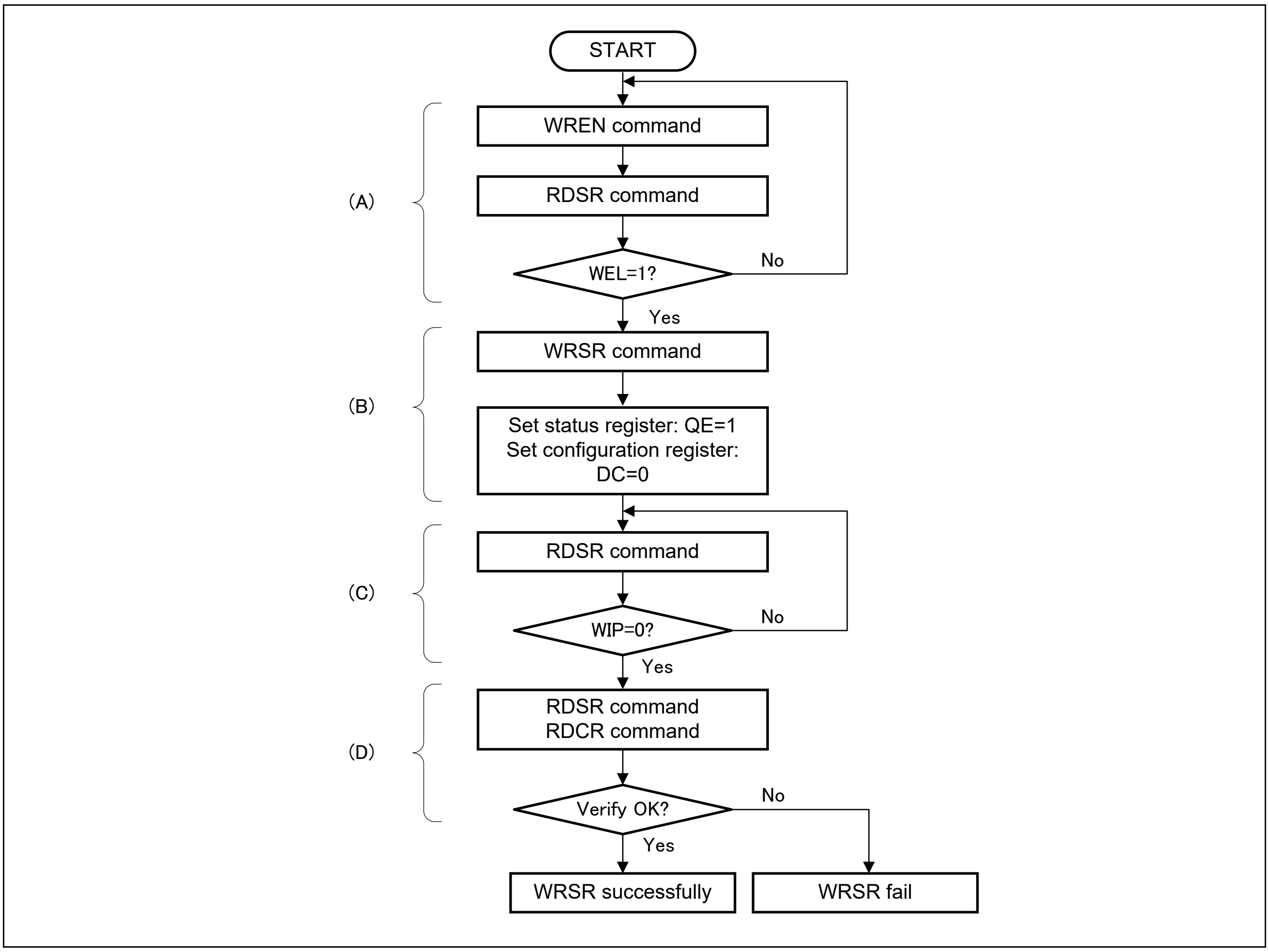


Figure 17.38 Serial Flash ROM Setting Flow (for Fast Read Quad I/O)

18. DMA Controllers

An R-IN32M4-CL3 incorporates two DMA controllers for the AHB bus, which consist of a total of five channels.

Table 18.1 R-IN32M4-CL3 DMA Controllers

Type of AHB DMA Controller	Number of Channels (expressed as "n" in the text)	Unit Number (expressed as "m" in the text)	External DMA Interface Pins	
General DMA controller	4	0	Channel 0	DMAREQZ0, DMAACKZ0, DMATCZ0
			Channel 1	DMAREQZ1, DMAACKZ1, DMATCZ1
			Channel 2	—
			Channel 3	—
DMA controller for real-time ports	1	1	RTDMAREQZ, RTDMAACKZ, RTDMATCZ	

- The meaning of n In this section, each channel of the DMA controllers is identified by "n".
- The meaning of m In this section, each unit of the DMA controllers is identified by "m".
  m = 0: General DMA controller
  m = 1: DMA controller for real-time ports

Remark: m = 0, 1
n = 0 to 3 when m = 0; n = 0 when m = 1

The DMA controllers serve as bus masters on the multi-layered AHB bus. Since they each have a dedicated layer, contention with other bus master is difficult to generate, enabling high throughput transfer. In an R-IN32M4-CL3, simultaneous transfer is possible since the general DMA controller and the DMA controller for real-time ports have individual AHB buses. For real-time ports in particular, a dedicated DMA controller is provided to prevent fluctuations in the time from a DMA transfer trigger until the actual transfer.

The DMA controllers control data transfer in response to requests for DMA in the form of the signals on the external DMAREQZ0, DMAREQZ1, and RTDMAREQZ pins, interrupt request signals, and the software trigger.

In addition, the widths at active level of the DMA acknowledge outputs (DMAACKZ0, DMAACKZ1, and RTDMAACKZ) and the mask widths for the DMA transfer request inputs (DMAREQZ0, DMAREQZ1, and RTDMAREQZ0) are selectable, easing the setting up of interfaces with external devices.

Caution: An R-IN32M4-CL3 employs a multi-layered internal bus architecture, so access by multiple masters can proceed at the same time unless contention between a bus master and slave arises.
In cases where different bus masters attempt access to the same slave, the result may not be as expected if this brings read-modify-write access and write access to the same address into contention. Take care with the flow of data so that such contention does not arise.

18.1 Features

18.1.1 Overview

- Number of channels: General DMA controller: 4 (Each channel is independent.)
  DMA controller for real-time ports: 1
- Number of buffer stages: General DMA controller: 16
  DMA controller for real-time ports: 4
- Transfer data size:
  - A size can be set for the source and destination independently.
  - Specifiable size: 8 to 512 bits
- Maximum number of transfer bytes: $2^{32}-1$ bytes (The DMA transfer volume is set in bytes.)
- Channel priority control
  - Fixed priority mode
  - Round robin mode (a channel that transferred data last is shifted to the lowest priority.)
- Methods of acquiring the transfer settings
  The data for use in DMA transfer is set in internal registers by using the following two modes.
  - Register mode
    DMA transfer is performed according to the control register in the DMA controller, which is set by the CPU. Conventional general DMA transfer is supported.
  - Link mode
    DMA transfer is performed according to a descriptor allocated in internal RAM or external memory.
    Various types of DMA transfer are possible. However, since a descriptor is accessed every DMA transfer, this mode is less responsive than register mode.
- Skipping
  A continuous access size and skip space size can be set respectively for the area for access in DMA transfer. After access to a set size for continuous access, the set skip space size can be skipped before access to the next address.
- Buffer data dumping
  Data in the buffer can be dumped when DMA transfer is forced to stop. After the data are dumped, DMA transfer is resumed.
- Suspension
  The ongoing DMA transaction can be suspended.
- DMA transfer interval setting
  The DMA transfer interval can be specified to adjust the bus occupancy ratio.
- Transfer mode
  - Single transfer mode
    When a DMA transfer request is generated, the DMAC acquires the right to use the bus and releases the bus each time it completes a transfer. After that, whenever a DMA transfer request is generated, this operation is repeated until the numbers of transfers specified in the control register are completed.
  - Block transfer mode
    When a DMA transfer request is generated, the DMAC acquires the right to use the bus and repeats data transfer until the numbers of transfers specified in the control register are completed. In this case, the bus is not occupied.

- Relationship for transfer targets
  In each DMA controller, the slaves with the "✓" marks below can be specified as the source/destination.

Table 18.2 Slaves as Targets for Transfer by the DMA Controller

Slaves as Targets for Transfer	General-Purpose DMAC	DMAC for Real-Time Ports
	Unit 0	Unit 1
Data RAM	✓	✓
Instruction RAM	✓	✓
Buffer RAM	✓	—
External Memory	✓	✓
Serial flash ROM	✓	—
Ethernet MAC Note 4	✓	✓
APB internal peripheral modules Note 1	✓	✓
Real-time ports	—	✓
General ports	✓	—
HWOS Note 2	—	—
DMA controller for real-time ports Note 3	—	—
General DMA controller Note 3	—	—

Remark: ✓: Specification with the source/destination is possible.
—: Specification with the source/destination is impossible.

Notes 1. The internal timer, serial interface, etc. are applied.
2. Hardware real-time OS
3. The register area of each DMA controller
4. The target module in an R-IN32M4-CL3 is CC-Link IE Field Network.

- Transfer request
  - Hardware request (A pin input or an interrupt request)
  - Software request
- Acknowledge output
  - Outputs an acknowledge signal to each channel.
- Terminal count output
  - Outputs a terminal count signal when the specified numbers of DMA transfers are completed.

Table 18.3 Relation between DMA Units/Channels and External DMA Interface Pins

The kind of AHB DMA Controller	DMA Unit/Channel	External DMA Interface Pins
General DMA controller	unit 0 / channel 0	DMAREQZ0, DMAACKZ0, DMATCZ0
	unit 0 / channel 1	DMAREQZ1, DMAACKZ1, DMATCZ1
	unit 0 / channel 2	None
	unit 0 / channel 3	None
DMA controller for real-time ports	unit 1	RTDMAREQZ, RTDMAACKZ, RTDMATCZ

Cautions 1. 0000 0000H–000B FFFFH secured as the instruction RAM area cannot be written directly. Writing to this area is via the instruction RAM mirror area (0400 0000H–040B FFFFH).

2. When writing to the instruction RAM area, observe the following conditions.

- **Write access must be in 32 bits (word) or 16 bits (half word).**
- **The number of bytes for transfer must be divisible by 32 bits (= 1 word = 4 bytes).**
- **The address of the 32-bit (= 1 word = 4 bytes) boundary should set as the start address.**
- **Write to the addresses continuously in the increment direction.**

18.2 Relation between DMA Units/Channels and DMA Triggers

The DMA trigger source registers (DTFR0 to DTFR3, and RTDTFR) are used to select the DMA transfer triggers from among interrupt requests in the form of input on the external interrupt pins or signals from internal peripheral modules, the software trigger, etc. The DMA transfer request, DMA acknowledge, and DMA terminal count signals of the external DMA interface are selected in the same way as trigger sources are allocated.

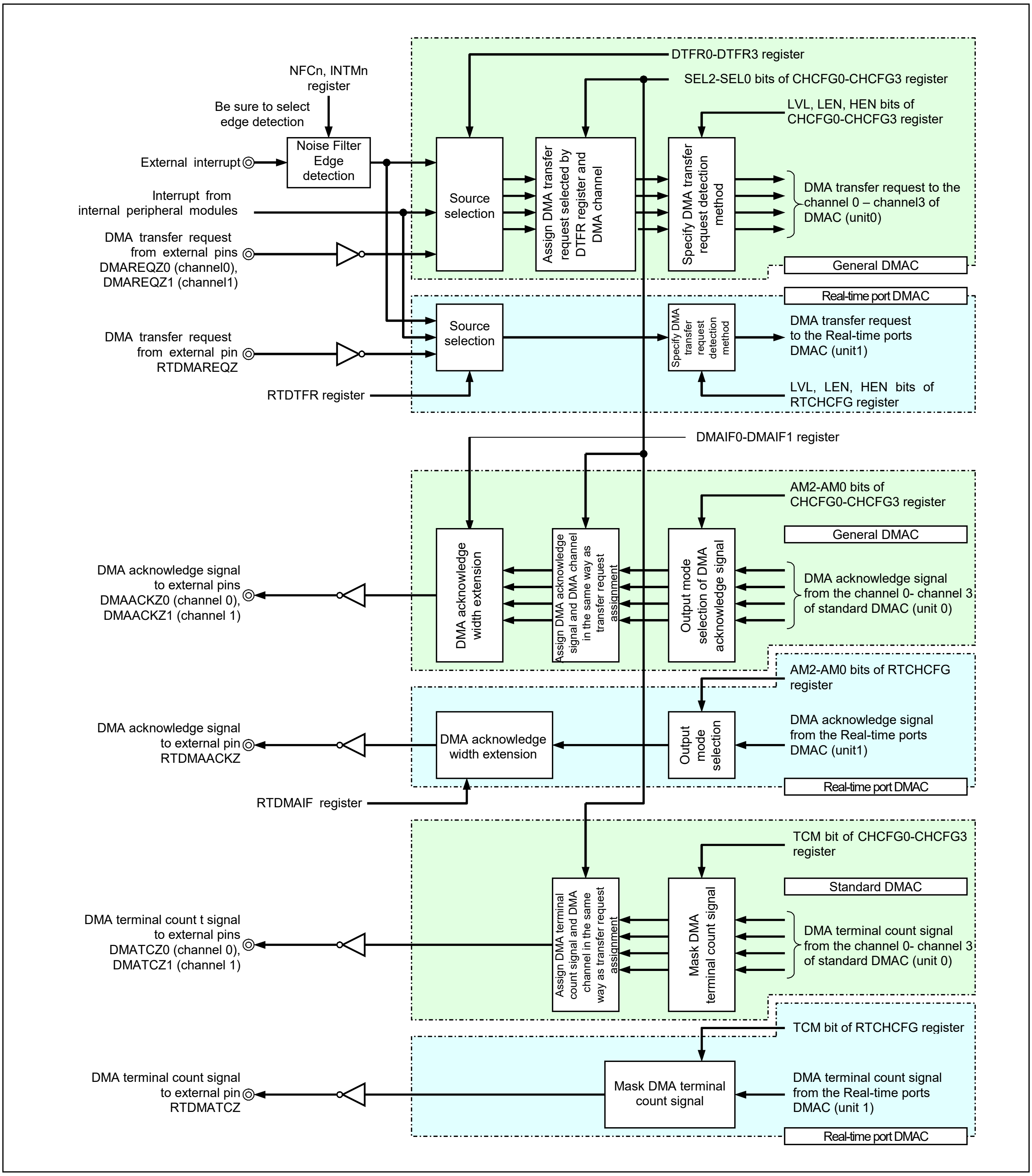


Figure 18.1 Relation between DMA Units/Channels and DMA Triggers

18.3 Terms and Definition

The terms used for the DMA controller are defined below.

Table 18.4 Definition of the Terms Used for the DMA Controller

Term	Definition
Burst	Means a single bus cycle.
DMA transfer	Refers to a single burst of read or write transfer executed by the DMAC.
DMA transaction	Refers to the fact that DMA transfer is completed for the total number of transfer bytes set in the DMAC, that is, the period of time it takes before the series of necessary DMA transfers is completed.
Descriptor	Means data describing DMA transfer settings that the DMAC loads in link mode.
Align	Refers to the state in which the address being transferred points to the beginning of the transfer size boundary. Specifically, the specified start address bit [($\log_2$ SIZE − 1): 0] is set to 0 (SIZE: transfer size [bytes]). Beat align: Refers to the state in which the transfer start address points to the beginning of the align boundary whose transfer size is set in SDS2–SDS0 (or DDS2–DSS0) of the CHCFGn register.
Unalign	Refers to the state in which the specified address does not point to the beginning of the align boundary of the transfer size. Specifically, the specified start address bit [($\log_2$ SIZE − 1): 0] is not set to 0 (SIZE: transfer size [bytes]). Beat unalign: Refers to the state in which the transfer start address does not point to the beginning of the align boundary whose transfer size is set in SDS2–SDS0 (or DDS2–DDS0) of the CHCFGn register.

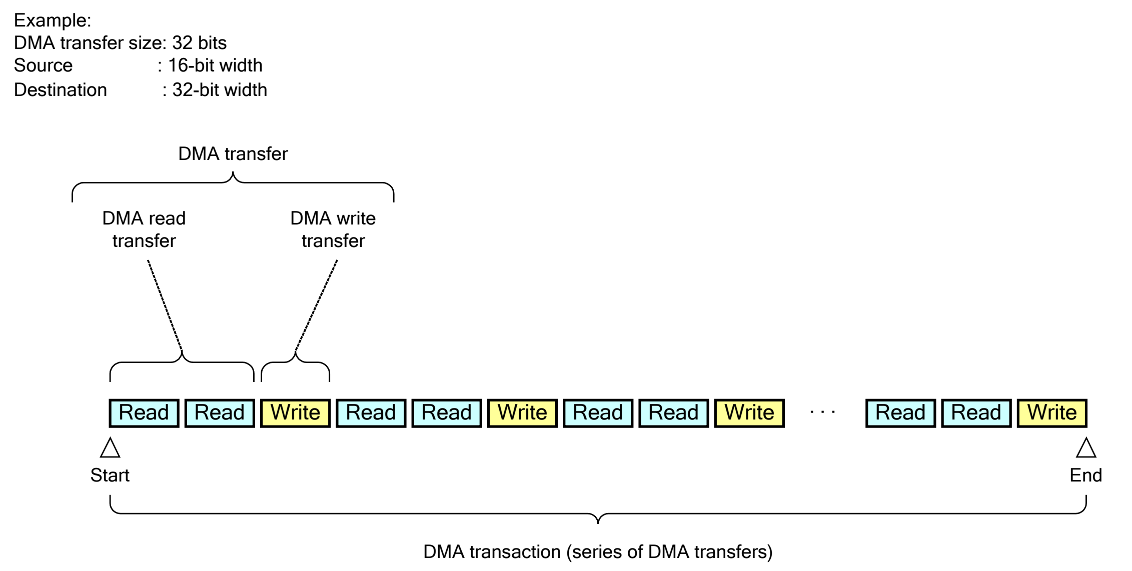


Figure 18.2 Name of Transfers

A single read or write transfer executed by the DMA controller is called a DMA transfer. To execute the series of set DMA transfers is called a DMA transaction.

18.4 DMA Controller Registers

18.4.1 Register Configuration

An R-IN32M4-CL3 has the general DMA controller and the DMA controller for real-time ports. The general DMA controller (unit 0) has four channels, and that for real-time ports (unit 1) has one channel. These controllers have the register sets listed below.

Table 18.5 DMA Controller Register Configuration

Register	Function
Next register set	This register set is used to set the source address, destination address, and the number of transfer bytes of the DMA transaction to be executed next. It consists of the Next 0 and Next 1 register sets. In register mode, set this register set using software. In link mode, the descriptor read data is automatically set in the Nex0 register set. The values of these register sets are loaded to Current register set and used for DMA transfer.
Current register set	This register set shows the source address, destination address, and the number of transfer bytes of the currently executed DMA transaction. The values are loaded from the Next 0/Next 1 register set (register mode) or the descriptor read data (link mode). They cannot be written directly using a program. The register set is automatically updated each time a DMA transaction is executed.
Channel register set	This register set is used to make DMA transfer settings. The settings made in this register set include the channel status indication, channel control, DMA transaction setting, and DMA transaction interval.
Link register set	This register set consists of the register for setting the address of the descriptor to be loaded next in link mode (Next link address register), the register for indicating the address of the currently executed descriptor (Current link address register), and the source/destination address register for the continuous space and skip space to be used when skipping is in use. The Current link address register is automatically updated at the time of descriptor read and cannot be written directly.
DMA control register	This register consists of the register for controlling the entire DMA unit and the register for indicating the status of each channel. It can be used to control the priority order of channels and check the status of each channel, such as enable, error, completion, terminal count, and suspend.
DMA interface register	This register consists of the DMA transfer interface signal control registers that set the timing of DMAREQZ and DMAACKZ signals programmable and the DMA trigger source registers that assign interrupt signals to the corresponding DMA channels.

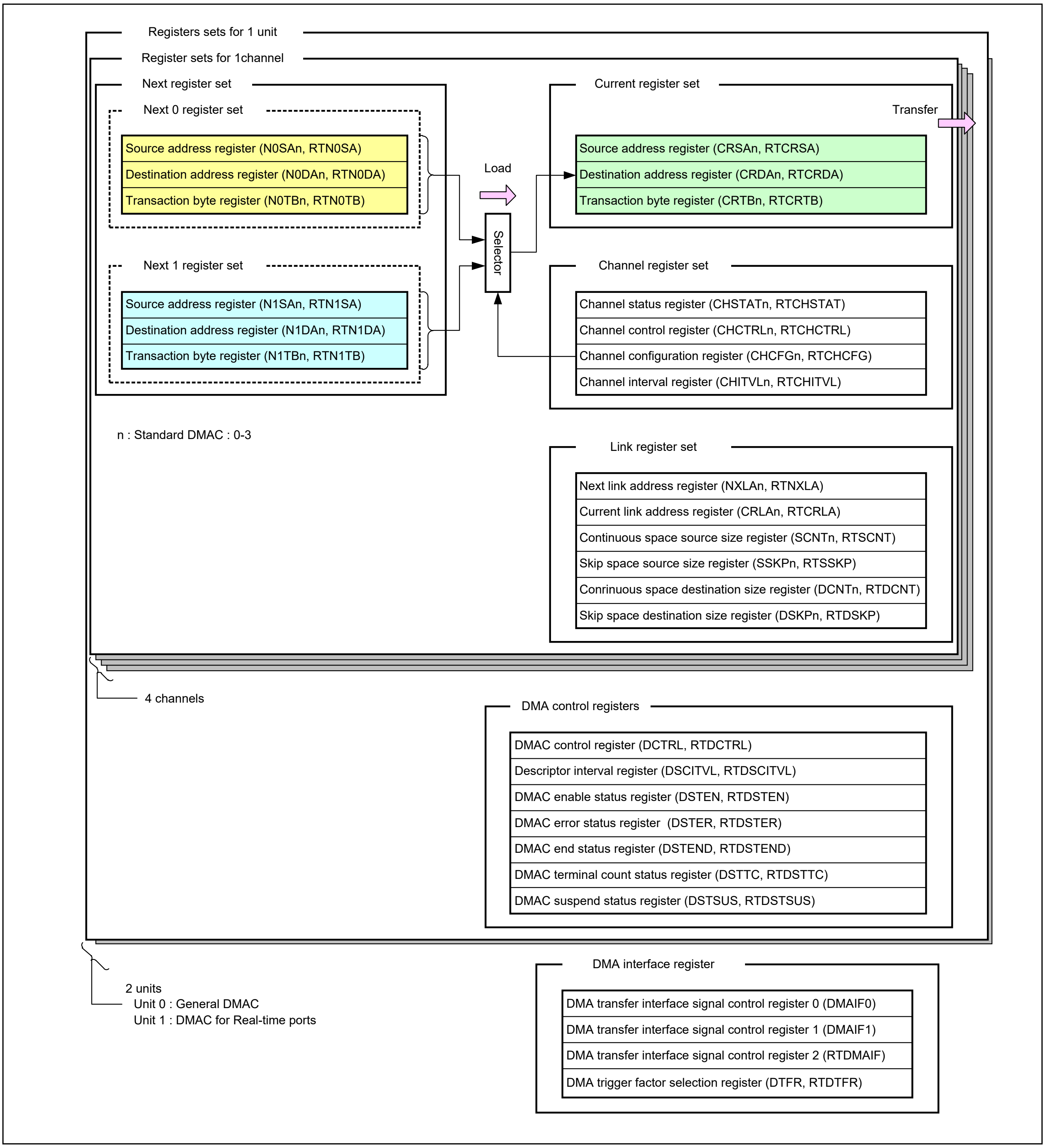


Figure 18.3 Register Block Diagram of DMA

Remark: n: General-purpose DMAC: 0 to 3

18.4.2 Control Register Outline

Table 18.6 DMA Controller Control Registers

(1/4)

Register Name	Symbol	Address
Next 0 source address register 0	N0SA0	400A 2800H
Next 0 destination address register 0	N0DA0	400A 2804H
Next 0 transaction byte register 0	N0TB0	400A 2808H
Next 1 source address register 0	N1SA0	400A 280CH
Next 1 destination address register 0	N1DA0	400A 2810H
Next 1 transaction byte register 0	N1TB0	400A 2814H
Current source address register 0	CRSA0	400A 2818H
Current destination address register 0	CRDA0	400A 281CH
Current transaction byte register 0	CRTB0	400A 2820H
Channel status register 0	CHSTAT0	400A 2824H
Channel control register 0	CHCTRL0	400A 2828H
Channel configuration register 0	CHCFG0	400A 282CH
Channel interval register 0	CHITVL0	400A 2830H
Next link address register 0	NXLA0	400A 2838H
Current link address register 0	CRLA0	400A 283CH
Next 0 source address register 1	N0SA1	400A 2840H
Next 0 destination address register 1	N0DA1	400A 2844H
Next 0 transaction byte register 1	N0TB1	400A 2848H
Next 1 source address register 1	N1SA1	400A 284CH
Next 1 destination address register 1	N1DA1	400A 2850H
Next 1 transaction byte register 1	N1TB1	400A 2854H
Current source address register 1	CRSA1	400A 2858H
Current destination address register 1	CRDA1	400A 285CH
Current transaction byte register 1	CRTB1	400A 2860H
Channel status register 1	CHSTAT1	400A 2864H
Channel control register 1	CHCTRL1	400A 2868H
Channel configuration register 1	CHCFG1	400A 286CH
Channel interval register 1	CHITVL1	400A 2870H
Next link address register 1	NXLA1	400A 2878H
Current link address register 1	CRLA1	400A 287CH

(2/4)

Register Name	Symbol	Address
Next 0 source address register 2	N0SA2	400A 2880H
Next 0 destination address register 2	N0DA2	400A 2884H
Next 0 transaction byte register 2	N0TB2	400A 2888H
Next 1 source address register 2	N1SA2	400A 288CH
Next 1 destination address register 2	N1DA2	400A 2890H
Next 1 transaction byte register 2	N1TB2	400A 2894H
Current source address register 2	CRSA2	400A 2898H
Current destination address register 2	CRDA2	400A 289CH
Current transaction byte register 2	CRTB2	400A 28A0H
Channel status register 2	CHSTAT2	400A 28A4H
Channel control register 2	CHCTRL2	400A 28A8H
Channel configuration register 2	CHCFG2	400A 28ACH
Channel interval register 2	CHITVL2	400A 28B0H
Next link address register 2	NXLA2	400A 28B8H
Current link address register 2	CRLA2	400A 28BCH
Next 0 source address register 3	N0SA3	400A 28C0H
Next 0 destination address register 3	N0DA3	400A 28C4H
Next 0 transaction byte register 3	N0TB3	400A 28C8H
Next 1 source address register 3	N1SA3	400A 28CCH
Next 1 destination address register 3	N1DA3	400A 28D0H
Next 1 transaction byte register 3	N1TB3	400A 28D4H
Current source address register 3	CRSA3	400A 28D8H
Current destination address register 3	CRDA3	400A 28DCH
Current transaction byte register 3	CRTB3	400A 28E0H
Channel status register 3	CHSTAT3	400A 28E4H
Channel control register 3	CHCTRL3	400A 28E8H
Channel configuration register 3	CHCFG3	400A 28ECH
Channel interval register 3	CHITVL3	400A 28F0H
Next link address register 3	NXLA3	400A 28F8H
Current link address register 3	CRLA3	400A 28FCH

(3/4)

Register Name	Symbol	Address
Continuous space source size register 0	SCNT0	400A 2A00H
Skip space source size register 0	SSKP0	400A 2A04H
Continuous space destination size register 0	DCNT0	400A 2A08H
Skip space destination size register 0	DSKP0	400A 2A0CH
Continuous space source size register 1	SCNT1	400A 2A20H
Skip space source size register 1	SSKP1	400A 2A24H
Continuous space destination size register 1	DCNT1	400A 2A28H
Skip space destination size register 1	DSKP1	400A 2A2CH
Continuous space source size register 2	SCNT2	400A 2A40H
Skip space source size register 2	SSKP2	400A 2A44H
Continuous space destination size register 2	DCNT2	400A 2A48H
Skip space destination size register 2	DSKP2	400A 2A4CH
Continuous space source size register 3	SCNT3	400A 2A60H
Skip space source size register 3	SSKP3	400A 2A64H
Continuous space destination size register 3	DCNT3	400A 2A68H
Skip space destination size register 3	DSKP3	400A 2A6CH
DMAC control register	DCTRL	400A 2B00H
DMAC descriptor interval register	DSCITVL	400A 2B04H
DMAC enable status register	DSTEN	400A 2B10H
DMAC error status register	DSTER	400A 2B14H
DMAC end status register	DSTEND	400A 2B18H
DMAC terminal count status register	DSTTC	400A 2B1CH
DMAC suspend status register	DSTSUS	400A 2B20H
RTDMAC Next 0 source address register	RTN0SA	400A 2C00H
RTDMAC Next 0 destination address register	RTN0DA	400A 2C04H
RTDMAC Next 0 transaction byte register	RTN0TB	400A 2C08H
RTDMAC Next 1 source address register	RTN1SA	400A 2C0CH
RTDMAC Next 1 destination address register	RTN1DA	400A 2C10H
RTDMAC Next 1 transaction byte register	RTN1TB	400A 2C14H
RTDMAC Current source address register	RTCRSA	400A 2C18H
RTDMAC Current destination address register	RTCRDA	400A 2C1CH
RTDMAC Current transaction byte register	RTCRTB	400A 2C20H
RTDMAC Channel status register	RTCHSTAT	400A 2C24H
RTDMAC Channel control register	RTCHCTRL	400A 2C28H
RTDMAC Channel configuration register	RTCHCFG	400A 2C2CH
RTDMAC Channel interval register	RTCHITVL	400A 2C30H
RTDMAC Next link address register	RTNXLA	400A 2C38H
RTDMAC Current link address register	RTCRLA	400A 2C3CH

(4/4)

Register Name	Symbol	Address
RTDMAC Continuous space source size register	RTSCNT	400A 2E00H
RTDMAC Skip space source size register	RTSSKP	400A 2E04H
RTDMAC Continuous space destination size register	RTDCNT	400A 2E08H
RTDMAC Skip space destination size register	RTDSKP	400A 2E0CH
RTDMAC control register	RTDCTRL	400A 2F00H
RTDMAC descriptor interval register	RTDSCITVL	400A 2F04H
RTDMAC enable status register	RTDSTEN	400A 2F10H
RTDMAC error status register	RTDSTER	400A 2F14H
RTDMAC end status register	RTDSTEND	400A 2F18H
RTDMAC terminal count status register	RTDSTTC	400A 2F1CH
RTDMAC suspend status register	RTDSTSUS	400A 2F20H
DMA transfer interface signal control register 0	DMAIFC0	4001 0720H
DMA transfer interface signal control register 1	DMAIFC1	4001 0724H
DMA transfer interface signal control register 2	RTDMAIFC	4001 0728H
DMA trigger source register 0	DTFR0	4001 0730H
DMA trigger source register 1	DTFR1	4001 0734H
DMA trigger source register 2	DTFR2	4001 0738H
DMA trigger source register 3	DTFR3	4001 073CH
DMA trigger source register 4	RTDTFR	4001 0740H

18.4.3 General DMA Controller Register Set

18.4.3.1 Next Register Set

The Next register set is loaded to the Current register set.

(1) Next Source Address Registers (N0SAn, N1SAn)

These registers set the DMA source address of General DMA controller (unit 0)/channel n.

N0SAn is for the Next 0 register set, and N1SAn is for the Next 1 register set.

In write-only mode in which write operations are performed continuously with the same value (CHCFGn.WONLY = 1), the register is used to set data to be written continuously (see section 18.7.4, Write-Only Mode).

- Access These registers can be read or written in units of 32 bits.

Register	Bits 31–0	Address	Initial value
N0SAn (R/W)	Normal mode: Source address Write-only mode: Write data (R/W)	400A 2800H + 40H × n	0000 0000H
N1SAn (R/W)	Normal mode: Source address Write-only mode: Write data (R/W)	400A 280CH + 40H × n	0000 0000H

Bit Position	Bit Name	Description
31 to 0	SA31–SA0	Source address in normal mode Sets the start address of the DMA transfer source.
	WD31–WD0	Write data in write-only mode Sets data to be written continuously in write-only mode in which write operations are performed continuously with the same value.

Caution: In a link mode transfer, the N0SAn register is overwritten by the descriptor read data.

Remark: n = 0 to 3

(2) Next Destination Address Registers (N0DAn, N1DAn)

These registers set the DMA destination address of general DMA controller (unit 0)/channel n.

N0DAn is for the Next 0 register set, and N1DAn is for the Next 1 register set.

- Access These registers can be read or written in units of 32 bits.

Register	Bits 31 to 0	Address	Initial value
N0DAn	Destination address	400A 2804H + 40H × n	0000 0000H
R/W	R/W		
N1DAn	Destination address	400A 2810H + 40H × n	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	DA31–DA0	Destination address. Sets the start address of the DMA destination.

Caution: In a link mode transfer, the N0DAn register is overwritten by the descriptor read data.

Remark: n = 0 to 3

(3) Next Transaction Byte Registers (N0TBn, N1TBn)

These registers set the total number of transfer bytes (DMA transaction) of the general DMA controller (unit 0)/channel n.

N0TBn is for the Next 0 register set, and N1TBn is for the Next 1 register set.

- Access These registers can be read or written in units of 32 bits.

Register	Bits 31 to 0	Address	Initial value
N0TBn	Transaction byte	400A 2808H + 40H × n	0000 0000H
R/W	R/W		
N1TBn	Transaction byte	400A 2814H + 40H × n	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	TB31–TB0	Number of transaction bytes. Sets the total number of bytes of a DMA transaction.

Cautions 1. Set the number of transfers in the total number of bytes.

2. Setting '0' as the number of transaction bytes is prohibited.

3. In a link mode transfer, the N0TBn register is overwritten by the descriptor read data.

Remark: n = 0 to 3

18.4.3.2 Current Register Set

The Current register set is a set of read-only registers that indicate the DMA transfer source address, destination address, and total number of transfer bytes.

The set values are loaded from the Next 0/Next 1 register set when in register mode and from the descriptor read data when in link mode. Values cannot be written using software.

(1) Current Source Address Register (CRSAn)

This register indicates the DMA source address of the general DMA controller (unit 0)/channel n.

- Access The register can only be read in units of 32 bits.

	31 … 0	Address	Initial value
CRSAn	Source address	400A 2818H + 40H × n	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRSA31–CRSA0	Current source address registers. Indicates the read address of the next DMA transaction. During the DMA transaction, the register is automatically updated (fixed when CHCFGn.SAD = 1 or undefined when CHCFGn.WONLY = 1). The initial value is loaded from one of the following registers. In register mode: Loads the source address from N0SAn / N1SAn. In link mode: Loads the source address from the descriptor. (The descriptor read data is assigned to the N0SAn register and, at the time of transfer, assigned to the CRSAn register.) This register is updated when the read operation for the DMA transfer is completed. The register should be read when DMA is not in progress (when CHSTATn.TACT = 0). The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.

Remark: n = 0 to 3

(2) Current Destination Address Register (CRDAn)

This register indicates the DMA destination address of the general DMA controller (unit 0)/channel n.

- Access The register can only be read in units of 32 bits.

	Bits 31 to 0	Address	Initial value
CRDAn	Destination address	400A 281CH + 40H × n	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRDA31–CRDA0	Current destination address registers. Indicates the write address of the next DMA transaction. During the DMA transaction, the register is automatically updated (fixed when CHCFGn.SAD = 1 or undefined when CHCFGn.WONLY = 1). The initial value is loaded from one of the following registers. In register mode: Loads the destination address from N0DAn / N1DAn. In link mode: Loads the destination address from the descriptor. (The descriptor read data is assigned to the N0DAn register and, at the time of transfer, assigned to the CRDAn register.) This register is updated when the write operation for the DMA transfer is completed. The register should be read when DMA is not in progress (when CHSTATn.TACT = 0). The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.

Remark: n = 0 to 3

(3) Current Transaction Byte Register (CRTBn)

This register indicates the total number of transfer bytes of the general DMA controller (unit 0)/channel n. Its value becomes 0000 0000H at the end of the DMA transaction (the series of DMA transfers).

- Access The register can only be read in units of 32 bits.

	31 – 0	Address	Initial value
CRTBn	Transaction byte data	400A 2820H + 40H × n	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRTB31–CRTB0	Current transaction byte registers. Indicates the number of bytes remaining to be transferred during the currently executed DMA transaction (the series of DMA transfers). During the DMA transaction, the register value is automatically decremented. The initial value is loaded from one of the following registers. In register mode: Loads the number of transfer bytes from N0TBAn/N1TBn. In link mode: Loads the number of transfer bytes from the descriptor. (The descriptor read data is assigned to the N0TBn register and, at the time of transfer, assigned to the CRTBn register.) This register is updated when the write operation for the DMA transfer is completed. The register should be read when DMA is not in progress (when CHSTATn.TACT = 0). The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.

Remark: n = 0 to 3

(4) Channel Register Set

The channel register set is a set of registers used to set the DMA transfer operation and DMA transfer mode, as well as to read the status information.

(a) Channel status register (CHSTATn)

This register reads the status of the general DMA controller (unit 0)/channel n.

- Access The register can only be read in units of 32 bits.

(1/6)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
CHSTAT n	DNUM								0	0	0	0	0	SWPRQ	DMARQM	INTM	0	0	0	0	MODE	DER	DW	DL	SR	TC	END	ER	SUS	TACT	RQST	EN	400A 2824H + 40H × n Initial value 0000 0000H
R/W	R	R	R	R	R	R	R	R	0	0	0	0	0	R	R	R	0	0	0	0	R	R	R	R	R	R	R	R	R	R	R	R	

Bit Position	Bit Name	Description
31 to 24	DNUM	Indicates the number of valid bytes in the buffer. Data in the buffer refers to data that has been read from the source but not yet written to the destination. Increment condition: • The DMA read transfer is completed. Decrement condition: • The DMA write transfer is completed. Condition for clearing this bit to 0: • CHSTATn.EN bit clearing condition. • The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)
23 to 19	—	Reserved. These bits return 0 when read.
18	SWPRQ	Indicates the status of the forced dump request. The status of the dump request initiated by the CHCTRLn.SETSSWPRQ bit is indicated. 0: Forced dump request not asserted. 1: Forced dump request asserted. Condition for setting this bit to 1: • The CHCTRLn.SETSSWPRQ bit is set to 1. Condition for clearing this bit to 0: • Forced dumping clears all data from the buffer. • CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)

Remark: n = 0 to 3

(2/6)

Bit Position	Bit Name	Description	
17	DMARQM	Indicates the temporary mask status of the DMA transfer request input. 0: Not masked. 1: Temporarily masked.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• The CHCTRLn.SETDMARQM bit is set to 1.	• The CHCTRLn. CLRDMARQM bit is set to 1. • The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)
16	INTM	Indicates the temporary mask status of the INTDMAn interrupt output. 0: Temporarily mask released. 1: Temporarily mask applied.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• The CHCTRLn.SETINTM bit is set to 1.	• The CHCTRLn.CLRINTM bit is set to 1. • The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)
15 to 12	—	Reserved. These bits return 0 when read.	
11	MODE	Indicates the DMA mode. This reflects the value of the DMS bit of the CHCFGn register. 0: Register mode 1: Link mode	
10	DER	Descriptor error bit. This bit is set to 1 when the LV bit (descriptor enable/disable bit) of the header of the read descriptor is set to 0 (the descriptor is disabled) in link mode. It is not dependent on the value of the CHCFGn.DIM bit. 0: There is no descriptor error. 1: There is a descriptor error.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• The LV bit of the descriptor header is set to 0 (the descriptor is disabled) when CHCFGn.DRRP is set to 0 in link mode (the descriptor continues to be read until the descriptor is enabled (LV = 1)).	• CHCTRLn.CLRDER bit is set to 1. (The DER bit, i.e., this bit is cleared.) • CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)

Remark: n = 0 to 3

(3/6)

<table>
<tr><th>Bit Position</th><th>Bit Position</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">9</td><td rowspan="3">DW</td><td colspan="2">This bit is set to 1 during a writeback to the descriptor in link mode.
If a bus error [Note] is received during the writeback to the descriptor, the bit remains set and not cleared to 0.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for setting this bit to 1</td></tr>
<tr><td>• The writeback of the header starts in link mode.</td><td>• The writeback of the header is completed in link mode.
• The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)</td></tr>
<tr><td rowspan="3">8</td><td rowspan="3">DL</td><td colspan="2">This bit is set to 1 during while loading the descriptor in link mode.
If a bus error [Note] is received while loading the descriptor, the bit remains set and not cleared to 0.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The descriptor is being loaded in link mode.</td><td>• Loading of the descriptor is completed in link mode.
• The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)</td></tr>
<tr><td rowspan="3">7</td><td rowspan="3">SR</td><td colspan="2">Indicates the register set selected in register mode.
0: Next 0 register set.
1: Next 1 register set</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• CHCFGn.RSEL is set to 1. (when Next 1 register set is selected)</td><td>• CHCFGn.RSEL is set to 0. (when Next 1 register set is selected)</td></tr>
<tr><td rowspan="3">6</td><td rowspan="3">TC</td><td colspan="2">This bit is set to 1 when the DMA transaction (the series of DMA transfers) is completed.
It is set to 1 only when CHCFGn.TCM is set to 0 (DMATCZp: terminal count output enable).</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The total numbers of transfer bytes set in the CRTBn register have been transferred in register mode.
• The total number of transfer bytes set in the CRTBn register have been transferred when WBD is set to 1 for the descriptor header (the writeback of the LV bit of the header is disabled) in link mode.
• The descriptor writeback is completed when WBD is set to 0 for the descriptor header in link mode.</td><td>• The CHCTRLn.CLRTC bit is set to 1. (The TCn bit, i.e., this bit is cleared.)
• The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)</td></tr>
</table>

Note: If a reserved area in the memory map is specified as the destination for access, the internal bus (AHB) generates a bus error (address decode error).
This bit can be cleared to 0 by setting the CHCTRLn.SWRST bit to 1.

Remark: n = 0 to 3; p = 0, 1

(4/6)

Bit Position	Bit Name	Description	
5	END	This bit is set to 1 when the DMA transaction (the series of DMA transfers) is completed and INTDMAn occurs.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0

		• The condition for setting the TC bit to 1 and the following condition are met: CHCFGn.DEM = 0 (INTDMAn) DMA transfer completion interrupt output is enabled)
		• The following condition are all met in link mode:
		  • The LV bit of the descriptor header is set to 0 (descriptor disabled).
		  • CHCFGn.DRRP is set to 0. (When the LV bit of the descriptor header is set to 0, the DER bit is set to 1, causing a descriptor error and stopping the DMA transfer.)
		  • CHCFGn.DIM is set to 0. (When the LV bit of the descriptor header is set to 0, the descriptor error interrupt (INTDMAn) is enabled.)
			• The CHCTRLn.CLREND bit is set to 1. (The ENDn bit, i.e., this bit is cleared.)
			• The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)

4	ER Note 1	This bit is set to 1 when a transfer error Note 2 occurs during DMA transfer and the INTDMAERR0 interrupt occurs.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• A DMA transfer error occurs Note 2	• The CHCTRLn.SWRSTn bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)
3	SUS	Indicates the suspended state of DMA channel n. 0: DMA channel n is not suspended. 1: DMA channel n is suspended.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• During the DMA transaction (the series of DMA transfers) for DMA channel n, the CHCTRLn.SETSUS bit is set to 1 and the DMA transaction for DMA channel n is suspended.	• The CHCTRLn.CLRSUS bit is set to 1. (Release from the suspended state) • The CHCTRLn.CLREN bit is set to 1. • The condition for clearing CHSTATn.EN bit is met.
2	TACT	Indicates whether DMA channel n is active. This bit is used to check that DMA channel n is completely inactive. 0: DMA is inactive on DMA channel n. 1: DMA is active on DMA channel n.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• The CHCTRLn.SETEN bit is set to 1. (The system waits for the start of descriptor read or DMA trigger.)	• The CHSTATn.EN is set to 0 and the entire DMA transaction (the series of DMA transfers) is completed.

Notes 1. If transfer proceeds while the ER bit is set to 1, use processing to handle the series of associated DMA transfers as invalid.

2. A bus error occurs during access to an undefined area, etc.

Remark: n = 0 to 3

(5/6)

Bit Position	Bit Name	Description	
1	RQST	Indicates whether a transfer request has been received. 0: A DMA transfer request has not been received. 1: A DMA transfer request has been received.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• The CHCTRLn.STG bit is set to 1. (When DMA is started by software.) • A DMA transfer request is received in response to the DMA transfer trigger selected by the SELn bit of the CHCFGn register.	• The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.) • The CHCTRLn.CLRRQ bit is set to 1. (The RQST bit, i.e., this bit is cleared.) • The DMA transfer ends in single transfer mode (CHCFGn.TM = 0). (By using the CHCFGn.REQD bit, the DMAACKZp output timing can be selected as either when it is read or when it is written. The condition for clearing this bit to 0 is when the read or write • The entire DMA transaction (the series of DMA transfers) is completed in register mode. (When CHCFGn.REN is set to 0 (the next DMA transfer is not performed by using the Next register set specified by the CHCFGn.RSEL bit after the DMA transaction (the series of DMA transfers) is completed).) • The DMA transfer for the last descriptor is completed in link mode. (When the LE bit of the descriptor header is set to 1 (link end).) • The DMA transfer is stopped during descriptor read in link mode (when LV is set to 0 and DRRP is set to 0 in the header). (LV = 0: Descriptor disabled) (CHCFGn.DRRP = 0: When the LV bit of the descriptor header is set to 0, the DERn bit is set to 1, causing a descriptor error and stopping the DMA transfer.) • CHCFGn.DEM is set to 0 (when the DMA transfer completion interrupt (INTDMAn) output is enabled and the DMA transaction (the series of DMA transfers) is completed). • A bus error [Note] occurs.

Note: A bus error occurs during access to an undefined area, etc.

Remark: n = 0 to 3; p = 0, 1

(6/6)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">0</td><td rowspan="3">EN</td><td colspan="2">Indicates whether the operation of DMA channel n is enabled or disabled.
0: Operation disabled.
1: Operation enabled</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• CHCTRLn.SETEN is set to 1.</td><td>• The CHCTRLn.SWRST bit is set to 1.
(The channel status register (CHSTATn, i.e., this register) is cleared.)
• The CHCTRLn.CLREN bit is set to 1.
(The EN bit, i.e., this bit is cleared.)
• The entire DMA transaction (the series of DMA transfers) is completed in register mode.
(When CHCFGn.REN is set to 0 (the next DMA transfer is not performed by using the Next register set specified by the CHCFGn.RSEL bit after the DMA transaction (the series of DMA transfers) is completed).)
• The DMA transfer for the last descriptor is completed in link mode.
(When the LE bit of the descriptor header is set to 1 (link end).) (When the WBD bit of the descriptor header is set to 0, this bit is cleared upon the completion of writeback.)
• A bus error Note occurs.</td></tr>
</table>

Note: A bus error occurs during access to an undefined area, etc.

Remark: n = 0 to 3

Cautions 1. If transfer proceeds while the ER bit is set to 1, use processing to handle the series of associated DMA transfers as invalid.

2. To stop the DMA transaction (the series of DMA transfers), mask or clear the transfer request or clear the EN bit (follow the procedure described in section 18.8.13, Suspending Transfer).

3. If the same DMA channel is requested to perform a transfer by using both the DMA transfer request signal and a software-initiated transfer request (i.e., by setting the CHCRTLn.STG bit to 1), the source of the request cannot be identified. Only one of the two transfer requests should be used at a time.

4. When making a software-initiated transfer request, check the Current register or other data to ensure that the last requested DMA transfer has been completed, before manipulating the CHCTRLn.STG bit.

(b) Channel control register (CHCTRLn)

This register controls the DMA transfer operation of the general DMA controller (unit 0)/channel n.

- Access — The register can only be written in units of 32 bits. Any bit of the register does not affect the operation if 0 is written to it. A read operation results in 0 being read from all the bits.

(1/3)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCTRLn	0	0	0	0	0	0	0	0	0	0	0	0	CLRDMARQM	SETDMARQM	CLRINTM	SETINTM	0	SETSSWPRQ	0	SETREN	0	0	CLRSUS	SETSUS	CLRDER	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
R/W	0	0	0	0	0	0	0	0	0	0	0	0	W	W	W	W	0	W	0	W	0	0	W	W	W	W	W	W	W	W	W	W

Address: 400A 2828H + 40H × n

Initial value: 0000 0000H

Bit Position	Bit Name	Description
31 to 20	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
19	CLRDMARQM	Clears the temporary mask status for DMA transfer request input. When this bit is set to 1, the temporary mask status for hardware DMA transfer requests is cleared. This clears the CHSTATn.DMARQM (temporary mask status for DMA transfer requests) bit to 0. 0: Does not affect the operation. 1: Releases the temporary mask status for hardware DMA transfer requests enabled by setting SETDMARQM to 1.
18	SETDMARQM	Sets the mask status for DMA transfer request input. When this bit is set to 1, the temporary mask status is set for hardware DMA transfer requests. This sets the CHSTATn.DMARQM (temporary mask status for DMA transfer requests) bit to 1. 0: Does not affect the operation. 1: Masks hardware DMA transfer requests temporarily.
17	CLRINTM	Clears the mask status for INTDMAn output. When this bit is set to 1, the mask status INTDMAn output is released. This clears the CHSTATn.INTM (temporary mask status for INTDMAn interrupt output) bit to 0. If the mask is released when the DMA transfer has been completed, INTDMAn is not output. 0: Does not affect the operation 1: Releases the mask status for INTDMAn output enabled by setting SETINTM to 1.

Remark: n = 0 to 3

(2/3)

Bit Position	Bit Name	Description
16	SETINTM	Sets the mask status for INTDMAn output. When this bit is set to 1, the temporary mask status is set for INTDMAn output. This sets the CHSTATn.INTM (temporary mask status for INTDMAn output) bit to 1. 0: Does not affect the operation. 1: Masks INTDMAn output.
15	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.
14	SETSSWPRQ	Forces the buffer to dump data. When this bit is set to 1, the buffer is forced to dump the data stored in it (see section 18.8.7, Forced Dumping). Note that, when CHCFGn.REQD is set to 1 and DMAACKZp is asserted at the time of writing, forced dumping cannot be used. 0: Does not affect the operation. 1: Forces the buffer data not yet written to the destination to be written (dumped) to the destination.
13	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.
12	SETREN	Set this bit to 1 to proceed to the next DMA transfer using the Next register set specified by the CHCFGn.RSEL bit after a DMA transaction (the series of DMA transfers) is completed in register mode. This sets the CHCFGn.REN bit to 1. For details, see the description of the REN bit of the channel configuration register (CHCFGn). 0: Does not affect the operation. 1: Sets CHCFGn.REN to 1.
11, 10	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
9	CLRSUS	Releases the suspended state of the active DMA channel n. If this bit is set to 1 when CHSTATn.SUS is set to 1, DMA channel n is released from the suspended state. 0: Does not affect the operation. 1: Releases suspension of the currently executed DMA transfer.
8	SETSUS	Sets the suspended state of the active DMA channel n. If this bit is set to 1 when CHSTATn.EN is set to 1 (the operation of DMA channel n is enabled), the active DMA channel n is placed in the suspended state. 0: Does not affect the operation. 1: Suspends the currently executed DMA transfer.
7	CLRDER	Clears the descriptor error in link mode. When this bit is set to 1, the CHSTATn.DER (descriptor error) bit is cleared to 0. 0: Does not affect the operation. 1: CHSTATn.DER (descriptor error) bit to 0.

Remark: n = 0 to 3; p = 0, 1

(3/3)

Bit Position	Bit Name	Description
6	CLRTC	Clears the terminal count (DMA transaction (the series of DMA transfers) completion) status. When this bit is set to 1, the CHSTATn.TC (terminal count) bit is cleared to 0. 0: Does not affect the operation. 1: Clears the CHSTATn.TC (terminal count) bit to 0.
5	CLREND	Clears the CHSTATn.END bit, which is set at the same time a DMA transaction (the series of DMA transfers) is completed and INTDMAn occurs. When this bit is set to 1, the CHSTATn.END bit is cleared to 0. 0: Does not affect the operation. 1: Clears the CHSTATn.END bit to 0.
4	CLRRQ	Clears the DMA transfer request. When this bit is set to 1, the CHSTATn.RQST (DMA transfer request) bit is cleared to 0. 0: Does not affect the operation. 1: Clears the CHSTATn.RQST (DMA transfer request) bit to 0.
3	SWRST	Executes software reset for DMA channel n. When this bit is set to 1, software reset is executed and each bit of the channel status register (CHSTATn) for which this operation is defined as the clearing condition is cleared to 0. Set this bit to 1 when the transfer on DMA channel n is completely stopped. To see whether the DMA channel transfer is completely stopped, check that both CHSTATn.EN and CHSTATn.TACT are set to 0. 0: Does not affect the operation. 1: Clear each bit of the CHSTATn register to 0 for which SWRST is defined as the clearing condition
2	STG	Serves as a software trigger for starting a DMA transfer by software. When this bit is set to 1, an internal transfer request is set (software trigger). If this bit is set to 1 at the same time as the SWRST bit, setting of the SWRST bit (software reset) is given priority. 0: Does not affect the operation. 1: Sets a transfer request by software (sets the CHSTATn.RQST bit to 1).
1	CLREN	Stops the operation of DMA channel n. When this bit is set to 1, the CHSTATn.EN bit is cleared to 0 and the operation of DMA channel n is stopped (for details, see section 18.8.13, Suspending Transfer). 0: Does not affect the operation. 1: Stops the operation of DMA channel n (clears the CHSTATn.EN bit to 0).
0	SETEN	Enables the operation of DMA channel n. When this bit is set to 1, the CHSTATn.EN bit is set to 1 and the operation of DMA channel n is enabled. If this bit is set to 1 at the same time as the SWRST bit, setting of the SWRST bit (software reset) is given priority. 0: Does not affect the operation. 1: Enables the operation of DMA channel n (sets the CHSTATn.EN bit to 1).

Remark: n = 0 to 3

(c) Channel configuration register (CHCFGn)

This register sets the DMA operation mode of the general DMA controller (unit 0)/channel n.

- Access The register can be read or written in units of 32 bits.

(1/7)

CHCFGn — Address: 400A 282CH + 40H × n — Initial value: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19–16	15–12	11	10–8	7	6	5	4	3	2–0
CHCFGn	DMS	REN	RSW	RSEL	SBE	DIM	TCM	DEM	WONLY	TM	DAD	SAD	DDS3–DDS0	SDS3–SDS0	DRRP Note 2	AM2–AM0	0	LVL	HEN	LEN	REQD	SEL2–SEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W R/W R/W R/W	R/W R/W R/W R/W	R/W	R/W R/W R/W	0	R/W	R/W	R/W	R/W	R/W R/W R/W

Bit Position	Bit Name	Description	
31	DMS	Selects the DMA operation mode. 0: Register mode (initial value) 1: Link mode	
30	REN	Selects whether to proceed to the next DMA transfer following the completion of the DMA transaction (the series of DMA transfers). When proceeding to the next DMA transfer, the Next register set selected by the RSEL bit is used to perform the DMA transfer. This setting is valid only in register mode. When this bit is set to 1 during the DMA transaction, we recommend using the SETERN bit of the CHCTRLn register. 0: Does not proceed to the next transfer. 1: Proceed to the next transfer (the Next register set selected by the RSEL bit is used).	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• This bit is set to 1. • The CHCTRLn.SETREN bit is set to 1.	• This bit is cleared to 0. • When REN is set to 1, a DMA transaction (the series of DMA transfers) is completed.
29	RSW	Selects whether to invert the RSEL (Next register set selection) bit when a DMA transaction (the series of DMA transfers) is completed. This setting is valid only in register mode. 0: Does not invert RSEL after a DMA transaction (the series of DMA transfers) is completed (initial value). 1: Inverts RSEL after a DMA transaction (the series of DMA transfers) is completed.	

Remark: n = 0 to 3

(2/7)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="2">Description</th></tr>
<tr><td>28</td><td>RSEL</td><td colspan="2">Selects the Next register set to be used for the next DMA transfer.
This setting is valid only in register mode.
When RSW is set to 1, the bit is automatically inverted upon the completion of a DMA transaction (the series of DMA transfers).
0: Uses the Next 0 register set (initial value).
1: Uses the Next 1 register set</td></tr>
<tr><td>27</td><td>SBE</td><td colspan="2">Selects how to handle the data already read into the buffer, if the operation of DMA channel n is stopped by setting CHCTRLn.CLREN to 1 during a DMA transaction (the series of DMA transfers). Note that, if REQD is set to 1 and the mode is selected in which DMAACKZp is output at the time of writing, this bit cannot be set to 1.
0: Stops the transfer without dumping (writing) the buffer data (initial value).
1: Stops the transfer after dumping (writing) the buffer data.</td></tr>
<tr><td>26</td><td>DIM</td><td colspan="2">Selects how the descriptor error interrupt (INTDMAERR0) behaves if the LV bit of the descriptor header is set to 0 in link mode.
0: Does not mask INTDMAERR0 (initial value).
1: Masks INTDMAERR0.</td></tr>
<tr><td rowspan="3">25</td><td rowspan="3">TCM</td><td colspan="2">Masks terminal count output (DMATCZp).
If this bit is set to 1 when the terminal count is output, DMATCZp is not output. CHSTATn.TC is not set to 1, either. In this case, the bit is automatically cleared to 0 in register mode or not cleared to 0 in link mode.
Use this bit when controlling DMA transfers by software.
0: Does not mask (enables terminal count output (DMATCZp); initial value).
1: Masks (disables terminal count output (DMATCZp).</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• This bit is set to 1.</td><td>• This bit is cleared to 0.
• The DMA transaction (the series of DMA transfers) is completed when this bit is set to 1 in register mode.</td></tr>
<tr><td rowspan="3">24</td><td rowspan="3">DEM</td><td colspan="2">Selects how INTDMAn behaves when a DMA transaction (the series of DMA transfers) is completed.
If this bit is set to 1 when INTDMAn occurs, INTDMAn is not output. CHSTATn.END is not set to 1, either. In this case, the bit is automatically cleared to 0 in register mode or not cleared to 0 in link mode.
0: Does not mask (enable INTDMAn output, initial value).
1: Masks (disables INTDMAn output).</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• This bit is set to 1.</td><td>• This bit is cleared to 0.
• The DMA transaction (the series of DMA transfers) is completed when this bit is set to 1 in register mode.</td></tr>
</table>

Remark: n = 0 to 3; p = 0, 1

(3/7)

Bit Position	Bit Name	Description
23	WONLY	Selects normal mode or write-only mode. In write-only mode, the data set in the Next source address register (N0SAn or N1SAn) is written to the address indicated by the Next destination address register (N0DAn or N1DAn). Use the write-only mode to perform write operations continuously with the same value. 0: Normal mode (initial value) 1: Write-only mode.
22	TM	Selects the DMA transfer mode. 0: Single transfer mode (performs a single transfer for each DMA transfer request; initial value). 1: Block transfer mode (transfers the number of bytes set in the transaction byte register for a DMA transfer request).
21	DAD	Sets the counting direction of the destination address of DMA channel n. 0: Increment (initial value) 1: Fixed **Caution: Do not select 1 (fixed) in DAD when the destination is using skip mode or the beats are not aligned on the destination side.**
20	SAD	Sets the counting direction of the source address of DMA channel n. 0: Increment (initial value) 1: Fixed **Caution: Do not select 1 (fixed) in SAD when the source is using skip mode or the beats are not aligned on the source side.**
19	DDS3	Selects normal mode or skip mode for DMA destination addressing. 0: Normal mode (initial value) 1: Skip mode

Remark: n = 0 to 3

(4/7)

Bit Position	Bit Name	Description
18 to 16	DDS2–DDS0	Sets the transfer size of the DMA destination.

DDS2	DDS1	DDS0	DMA Destination Transfer Size
0	0	0	8 bits (initial value)
0	0	1	16 bits
0	1	0	32 bits
0	1	1	Setting prohibited
1	0	0	128 bits
1	0	1	256 bits
1	1	0	512 bits [Note]
1	1	1	Setting prohibited

Note: These bits can be set only when addresses are aligned in units of the transfer size.

Bit Position	Bit Name	Description
15	SDS3	Selects normal mode or skip mode for DMA source addressing. 0: Normal mode (initial value) 1: Skip mode
14 to 12	SDS2–SDS0	Sets the transfer size of the DMA source.

SDS2	SDS1	SDS0	DMA Source Transfer Size
0	0	0	8 bits (initial value)
0	0	1	16 bits
0	1	0	32 bits
0	1	1	Setting prohibited
1	0	0	128 bits
1	0	1	256 bits
1	1	0	512 bits [Note]
1	1	1	Setting prohibited

Note: These bits can be set only when addresses are aligned in units of the transfer size.

Remark: n = 0 to 3

(5/7)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>11</td><td>DRRP</td><td>Selects the operation if the descriptor header is disabled (LV = 0) in link mode.
0: Sets the CHSTATn.DER (descriptor error) bit to 1 and stops the operation (initial value).
1: Continues to read the same descriptor until LV becomes 1. When LV becomes 1, a DMA transfer is started by using that descriptor. To set the interval at which the descriptor is to be read, use the descriptor interval register (DSCITVL).</td></tr>
<tr><td>10 to 8</td><td>AM2–
AM0</td><td>Selects the output mode of the DMA acknowledge signal.
<table>
<tr><th>AM2</th><th>AM1</th><th>AM0</th><th>DMA Acknowledge Signal (DMAACKZp) Output Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Pulse mode Note 1 (initial value)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Level mode.
The active level is maintained until the DMA transfer request (DMAREQZp) becomes inactive.</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>Bus cycle mode Note 2
The active level is maintained during a DMA transfer bus cycle.</td></tr>
<tr><td>1</td><td>X</td><td>X</td><td>DMA acknowledge signal (DMAACKZp) output disabled</td></tr>
</table>
Notes 1. A pulse of one BUSCLK cycle is output as the DMAACKZp signal.
2. In bus cycle mode, the DMA acknowledge signal is output following the point at which acquisition of bus mastership is requested. For this reason, the DMA acknowledge signal is output earlier than the actual DMA bus cycle, and a bus cycle of an internal master which has previously acquired mastership of the same bus may proceed at this time.

Cautions 1. The settings of AM2 to AM0 do not affect the actual operation while the interrupt request signal from on-chip peripheral modules and external interrupt input are selected.
2. The settings of AM2 to AM0 may duplicate those of the DMAIFCp register. In general, however, when the DMAACKZp signal is set to the level mode by using AM2 to AM0, the DMAIFCn register should be left at its initial value. Conversely, when the DMAIFCn register is used to extend the DMAACKZp pulse width or for the DMAREQZp mask function, set AM2 to AM0 to select the pulse mode.

Remark: X: Don't Care</td></tr>
<tr><td>7</td><td>—</td><td>Reserved. When writing to this bit, write 0. When read, 0 is returned.</td></tr>
</table>

Remark: n = 0 to 3; p = 0, 1

(6/7)

Bit Position	Bit Name	Description
6	LVL	Selects the method of detection of the DMA transfer request signal.
5	HEN	A DMA transfer request is chosen by the DMA trigger source register n (DTFRn). The method of detection of the DMA transfer request signal differs with the selected DMA transfer request.
4	LEN	**[When the DMA transfer request signal is a DMA request signal of an external pin]** An internal DMA interface is positive logic. A DMA interface terminal (DMAREQZp, DMAACKZp, and DMATCZp) is negative logic. Since the signals of the DMA interface pins are inverted at the connection to the system bus DMAC signals, the opposite logic to that selected by the settings of the HENn and LENn bits is chosen.

LVLn	HENn	LENn	Detection Method of DMA Transfer Request Signal (DMAREQZp)		
				Internal Signal	External Pin
0	0	0	Edge detection	Detection disabled	
0	0	1		Falling edge detection	Rising edge detection
0	1	0		Rising edge detection	Falling edge detection
0	1	1		Rising/falling edge detection (Does not recommend)	
1	0	0	Level detection	Detection disabled	
1	0	1		Low level detection	High level detection
1	1	0		High level detection	Low level detection
1	1	1		Setting prohibited	

[When DMA transfer request signal is an interrupt signal (The signal which starts with INT).]

LVLn	HENn	LENn	Detection Procedure of the DMA Transfer Request Signal by an Interrupt Signal	
0	0	0	Edge detection	Detection disabled
0	0	1		Low level detection
0	1	0		High level detection
0	1	1		Setting prohibited
1	×	×	Level detection	Setting prohibited

Bit Position	Bit Name	Description
3	REQD	Selects when DMAACKZp is to become active. Usually, set this bit so that DMAACKZp is output to the side on which DMAREQZp is asserted. 0: DMAACKZp is active when reading (DMAREQZp is the source). 1: DMAACKZp is active when writing (DMAREQZp is the destination).

Remark: n = 0 to 3; p = 0, 1

(7/7)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>2 to 0</td><td>SEL2–
SEL0</td><td>Selects the DMA interface signal for each channel.
Usually, set the same value as the channel number.
Only if the priority needs to be replaced within the channel of an external DMA transfer request, change a DMA trigger by using the SEL1 or SEL0 bit.
<table>
<tr><th>SEL2</th><th>SEL1</th><th>SEL0</th><th>DMA Trigger Selection</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>The DMA transfer source selected by DTFR0 is chosen.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>The DMA transfer source selected by DTFR1 is chosen.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>The DMA transfer source selected by DTFR2 is chosen.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>The DMA transfer source selected by DTFR3 is chosen.</td></tr>
<tr><td colspan="3">Other than the above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Remark: p = 0, 1

(d) Channel interval register (CHITVLn)

This register sets the DMA transfer interval of the general DMA controller (unit 0)/channel n.

The specifiable interval values are the internal system bus clock (HCLK) cycle × the value of ITVL15–ITVL0.

- Access The register can be read or written in units of 32 bits.

For details, see section 18.8.9, Interval Counting.

	31 to 16	15 to 0	Address	Initial value
CHITVLn	0	ITVL15–ITVL0	400A 2830H + 40H × n	0000 0000H
R/W	0	R/W		

Bit Position	Bit Name	Description
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 0	ITVL15–ITVL0	Set the DMA transfer interval of DMA channel n.

Remark: n = 0 to 3

(5) Link Register Set

This is a register set that indicates the link addresses in link mode.

When DMA is started by setting a descriptor address in the NXLAn register, the hardware loads the value of the NXLAn register to the CRLAn register and the descriptor is read. The DMAC starts a DMA transaction based on that descriptor value. The NXLAn register is automatically updated based on the link address value in the read descriptor, and its value is used as the descriptor address for the next DMA transaction.

Remark: n = 0 to 3

(a) Next link address register (NXLAn)

This register sets the link address of the general DMA controller (unit 0)/channel n.

Set the address to which the descriptor in link mode is allocated.

- Access The register can be read or written in units of 32 bits.

For information about link mode, see section 18.7.3, Link Mode.

	31–2	1	0	Address	Initial value
NXLAn	NXLA31–NXLA2	0	0	400A 2838H + 40H × n	0000 0000H
R/W	R/W	0	0		

Initial Value	Bit Name	Description
31 to 0	NXLA31–NXLA2	Sets the link address in link mode. Only an address aligned by word (32 bits) can be set. The lower-order two bits are fixed at 0.

Remark: n = 0 to 3

(b) Current link address register (CRLAn)

This register indicates the address of the descriptor currently executed in link mode.

- Access The register can only be read in units of 32 bits.

	31–0	Address	Initial value
CRLAn	CRLA31–CRLA0	400A 283CH + 40H × n	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRLA31–CRLA0	Indicates the address of the descriptor currently executed in link mode.

Remark: n = 0 to 3

(c) Continuous space source size register (SCNTn)

This register sets the size of the continuous access space for access to the source by the general DMA controller (unit 0)/channel n in bytes. The register is used in combination with the skip space source size register (SSKPn).

To use skip mode for the source address, set the SDS3 bit of the channel configuration register (CHCFGn) to 1.

Do not set the SAD bit of the channel configuration register (CHCFGn) to 1 (fixed at the source address).

Moreover, please do not set “0000 0000H” to this register in skip mode.

- Access The register can be read or written in units of 32 bits.

	31 … 0	Address	Initial value
SCNTn	SCNT31–SCNT0	400A 2A00H + 20H × n	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	SCNT31–SCNT0	Specifies the size of the continuous access space for the source address in bytes in skip mode.

Remark: n = 0 to 3

(d) Skip space source size register (SSKPn)

This register sets the size of the skip space for access to the source by the general DMA controller (unit 0)/channel n in bytes. The register is used in combination with continuous space source size register n (SCNTn).

To use skip mode for the source address, set the SDS3 bit of the channel configuration register (CHCFGn) to 1.

Do not set the SAD bit of the channel configuration register (CHCFGn) to 1 (fixed at the source address).

- Access The register can be read or written in units of 32 bits.

	31 … 0	Address	Address
SSKPn	SSKP31–SSKP0	400A 2A04H + 20H × n	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	SSKP31–SSKP0	Specifies the size of the skip space for the source address in bytes in skip mode.

Remark: n = 0 to 3

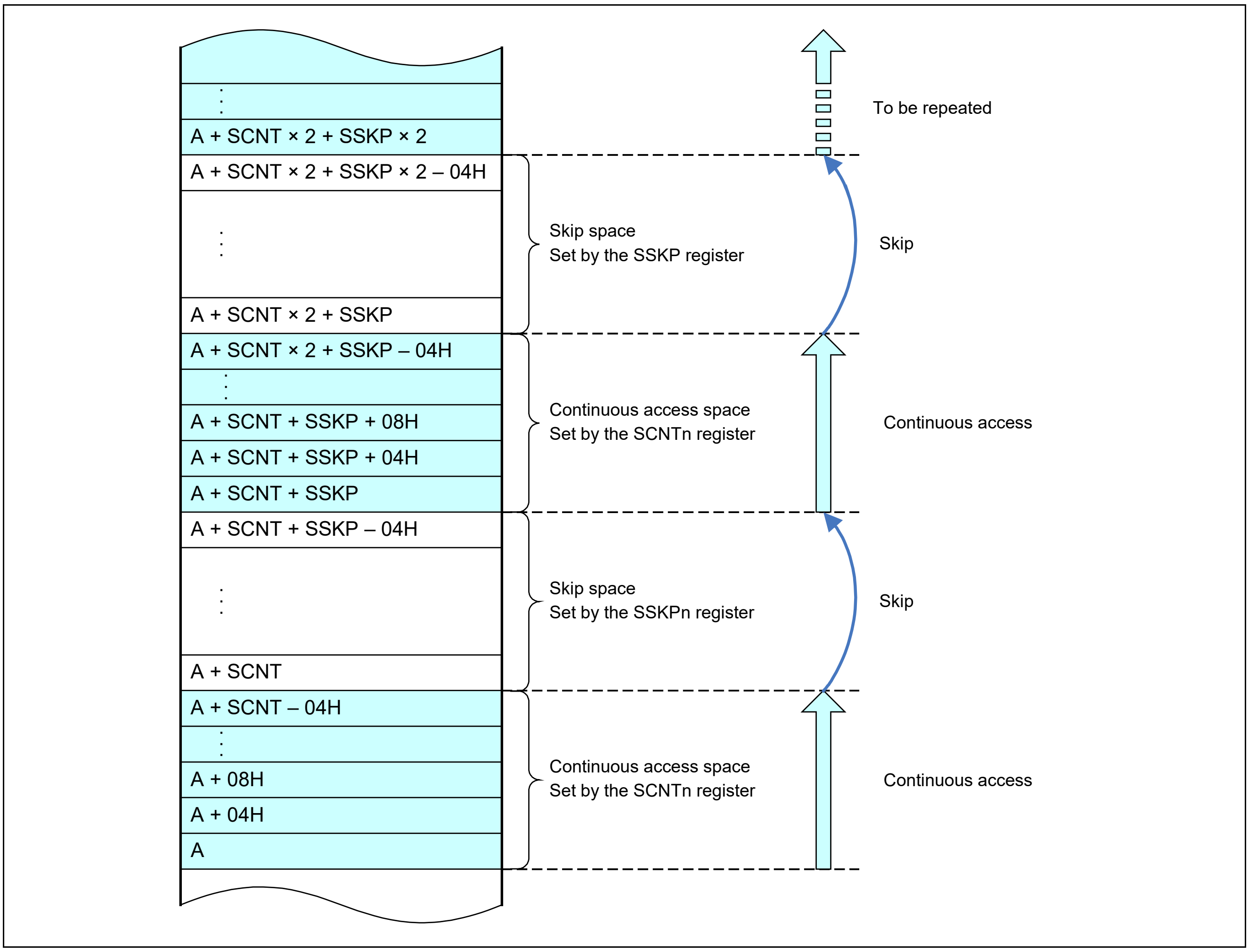


Figure 18.4 Relationship between the SSKPn and SCNTn Registers in Skip Mode

Remark: The values of SCNTn and SSKPn can be set independently of the source address and the value of the SDS2–SDS0 bit (source data size) of the channel configuration register (CHCFGn).
The DMA controller is accessed in the size set by SDS2 to SDS0, and only valid data is retrieved into the buffer.

(e) Continuous space destination size register (DCNTn)

This register sets the size of the continuous access space for access to the destination by the general DMA controller (unit 0)/channel n in bytes. The register is used in combination with the skip space destination size register (DSKPn).

To use skip mode for the destination address, set the DDS3 bit of the channel configuration register (CHCFGn) to 1.

Do not set the DAD bit of the channel configuration register (CHCFGn) to 1 (fixed at the destination address).

Also, do not set 0000 0000H in this register in skip mode.

- Access The register can be read or written in units of 32 bits.

	31 … 0	Address	Initial value
DCNTn	DCNT31–DCNT0	400A 2A08H + 20H × n	0000 0000H
R/W	R/W		

Bit Position	Bit Position	Description
31 to 0	DCNT31–DCNT0	Specifies the size of the continuous access space for the destination address in bytes in skip mode.

Remark: n = 0 to 3

(6) Skip Space Destination Size Register (DSKPn)

This register sets the size of the skip space for access to the destination by the general DMA controller (unit 0)/channel n in bytes.
The register is used in combination with the continuous space destination size register (DCNTn).

To use skip mode for the destination address, set the DDS3 bit of the channel configuration register (CHCFGn) to 1.

Do not set the DAD bit of the channel configuration register (CHCFGn) to 1 (fixed at the destination address).

- Access The register can be read or written in units of 32 bits.

	31 … 0	Address	Initial value
DSKPn	DSKP31–DSKP0	400A 2A0CH + 20H × n	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	DSKP31–DSKP0	Specifies the size of the skip space for the destination address in bytes in skip mode.

Remark: n = 0 to 3

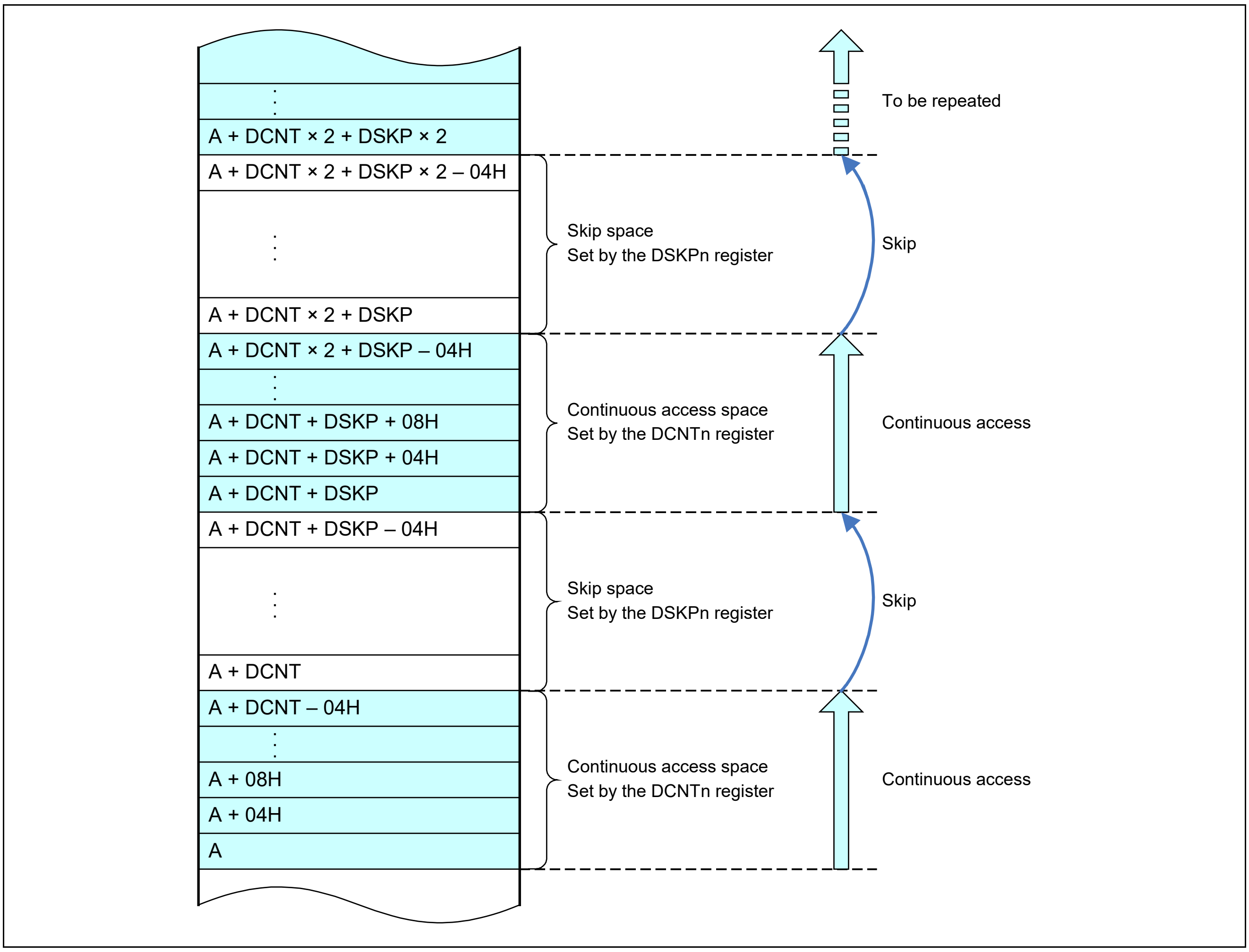


Figure 18.5 Relationship between the DSKPn and DCNTn Registers in Skip Mode

Remark: The values of DCNTn and DSKPn can be set independently of the destination address and the value of the DDS2–DDS0 bit (destination data size) of the channel configuration register (CHCFGn). The DMA controller only writes to the specified space in combinations of sizes equal to or smaller than that set by DDS2 to DDS0.

(7) DMA Control Registers

The DMA control register is for control that applies in common to all channels of the general DMA controller (unit 0).

(a) DMAC control register (DCTRL)

This register selects the transfer priority control mode.

Be sure to set 0 to bits 31 to 1.

- Access The register can be read or written in units of 32 bits.

	31–1	0	Address	Initial value
DCTRL	0	PR	400A 2B00H	0000 0000H
R/W	0	R/W		

Bit Position	Bit Name	Description
31 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	PR	Selects the transfer priority control mode (see section 18.8.2, DMA Unit Priority Control). 0: Fixed priority mode 1: Round robin mode

(b) Descriptor interval register (DSCITVL)

If the descriptor header is read in link mode when the DRRP bit of the channel configuration register (CHCFGn) set to 1, and if the LV bit is set to 0 (descriptor disabled), the descriptor continues to be read until LV becomes 1.

This register sets the interval at which the descriptor is to be read in such a case. It can be set in units of the internal system bus clock (HCLK) cycle × 256.

- Access The register can be read or written in units of 32 bits.

	31–16	15–8	7–0	Address	Initial value
DSCITVL	0	DITVL15–DITVL8	0	400A 2B04H	0000 0000H
R/W	0	R/W	0		

Bit Position	Bit Name	Description
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 8	DITVL15–DITVL8	Sets the interval at which the descriptor header continues to be read until the LV bit becomes 1. The descriptor is read in the (DITVL15–DITVL8 value) × 256 × internal system bus clock (HCLK) cycles.
7 to 0	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.

Remark: n = 0 to 3

(c) DMAC enable status register (DSTEN)

This register indicates the state of the EN (enable) bit of all the DMA channels.

- Access The register can only be read in units of 32 bits.
  Writing to the register does not change the value of any of its bits.

To set EN to 1 (enable DMA channel n), set the SETEN bit of the channel control register (CHCTRLn) to 1.
To set EN to 0 (disable DMA channel n), set the CLREN bit of the channel control register (CHCTRLn) to 1.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTEN	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	EN3	EN2	EN1	EN0
R/W	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	R	R	R	R

Address: 400A 2B10H
Initial value: 0000 0000H

Bit Position	Bit Name	Description
31 to 4	—	Reserved. These bits return 0 when read.
3	EN3	Indicates the EN (DMA enable) bit of DMA channel 3.
2	EN2	Indicates the EN (DMA enable) bit of DMA channel 2.
1	EN1	Indicates the EN (DMA enable) bit of DMA channel 1.
0	EN0	Indicates the EN (DMA enable) bit of DMA channel 0

Remarks 1. The EN bit of each DMA Channel is the 0th bit of the channel status register (CHSTATn).
2. n = 0 to 3

(d) DMAC error status register (DSTER)

This register indicates the state of the ER (error) bit of all the DMA channels.

- Access The register can only be read in units of 32 bits.
  Writing to this register does not change the value of the bits.

If an error occurs in a DMA transfer bus cycle, 1 is set. To clear the bit to 0, the SWRST bit of the channel control register (CHCTRLn) needs to be set to 1. If any ER bit is set to 1, the series of the associated DMA transfers should be handled as an invalid transaction.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTER	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	ER3	ER2	ER1	ER0
R/W	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	R	R	R	R

Address 400A 2B14H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 4	—	Reserved. These bits return 0 when read.
3	ER3	Indicates the ER (DMA transfer error) bit of DMA channel 3.
2	ER2	Indicates the ER (DMA transfer error) bit of DMA channel 2.
1	ER1	Indicates the ER (DMA transfer error) bit of DMA channel 1.
0	ER0	Indicates the ER (DMA transfer error) bit of DMA channel 0.

Remarks 1. The ER bit of each DMA channel is the 4th bit of the channel status register (CHSTATn).
2. n = 0 to 3

(e) DMAC end status register (DSTEND)

This register indicates the state of the END bit (indicating generation of INTDMAn on completion of the DMA transaction (the series of DMA transfers)) of all the DMA channels.

- Access The register can only be read in units of 32 bits.
  Writing to the register does not change the value of any of its bits.

For information about the setting and clearing conditions, see the description of the END bit of the channel status register (CHSTATn).

DSTEND — Address 400A 2B18H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTEND	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	END3	END2	END1	END0
R/W	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	R	R	R	R

Bit Position	Bit Name	Description
31 to 4	—	Reserved. These bits return 0 when read.
3	END3	Indicates the END (DMA transaction completion) bit of DMA channel 3.
2	END2	Indicates the END (DMA transaction completion) bit of DMA channel 2.
1	END1	Indicates the END (DMA transaction completion) bit of DMA channel 1.
0	END0	Indicates the END (DMA transaction completion) bit of DMA channel 0.

Remarks 1. The END bit of each DMA channel is the 5th bit of the channel status register (CHSTATn).
2. n = 0 to 3

Table 18.7 Correspondence between DMA End Status Registers and Interrupt Signals

Register Name	Bits Name	Corresponding Transfer Completion Interrupt Signal
DSTEND	END0	INTDMA0
	END1	INTDMA1
	END2	INTDMA2
	END3	INTDMA3

(f) DMAC terminal count status register (DSTTC)

This register indicates the state of the TC bit (indicating the completion of the DMA transaction (the series of DMA transfers)) of all the DMA channels.

- Access The register can only be read in units of 32 bits.
  Writing to the register does not change the value of any of its bits.

For information about the setting and clearing conditions, see the description of the TC bit of the channel status register (CHSTATn).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 400A 2B1CH
DSTTC	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	TC3	TC2	TC1	TC0	Initial value 0000 0000H
R/W	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	R	R	R	R	

Bit Position	Bit Name	Description
31 to 4	—	Reserved. These bits return 0 when read.
3	TC3	Indicates the TC (DMA transaction completion) bit of DMA channel 3.
2	TC2	Indicates the TC (DMA transaction completion) bit of DMA channel 2.
1	TC1	Indicates the TC (DMA transaction completion) bit of DMA channel 1.
0	TC0	Indicates the TC (DMA transaction completion) bit of DMA channel 0.

Remarks 1. The TC bit of each DMA channel is the 6th bit of the channel status register (CHSTATn).

2. n = 0 to 3

(g) DMAC suspend status register (DSTSUS)

This register indicates the state of the SUS (suspended state) bit of all the DMA channels.

- Access The register can only be read in units of 32 bits.
  Writing to the register does not change the value of any of its bits.

To set SUS to 1 (set the suspended state), set the SETSUS bit of the channel control register (CHCTRLn) to 1.
To set SUS to 0 (release from the suspended state), set the CLRSUS bit of the channel control register (CHCTRLn) to 1.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 400A 2B20H
DSTSUS	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	SUS3	SUS2	SUS1	SUS0	Initial value 0000 0000H
R/W	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	R	R	R	R	

Bit Position	Bit Name	Description
31 to 4	—	Reserved. These bits return 0 when read.
3	SUS3	Indicates the SUS (suspended state) bit of DMA channel 3.
2	SUS2	Indicates the SUS (suspended state) bit of DMA channel 2.
1	SUS1	Indicates the SUS (suspended state) bit of DMA channel 1.
0	SUS0	Indicates the SUS (suspended state) bit of DMA channel 0.

Remarks 1. The SUS bit of each DMA channel is in the 3rd bit of the channel status register (CHSTATn).

2. n = 0 to 3

18.4.4 Register Set of DMA Controller for Real-Time Ports

18.4.4.1 Next Register Set

The Next register set is loaded to the Current register set.

(1) Next Source Address Registers (RTN0SA, RTN1SA)

These registers set the DMA source address of the DMA controller for real-time ports.

RTN0SA is for the Next 0 register set, and RTN1SA is for the Next 1 register set.

In the write-only mode in which write operations are performed continuously with the same value (CRTHCFG.WONLY = 1), the register is used to set data to be written continuously (see section 18.7.4, Write-Only Mode).

- Access These registers can be read or written in units of 32 bits.

Register	Bits 31–0	Address	Initial value
RTN0SA (R/W)	Normal mode: Source address Write-only mode: Write data (R/W)	400A 2C00H	0000 0000H
RTN1SA (R/W)	Normal mode: Source address Write-only mode: Write data (R/W)	400A 2C0CH	0000 0000H

Bit Position	Bit Name	Description
31 to 0	SA31–SA0	Source address in normal mode Sets the start address of the DMA transfer source.
	WD31–WD0	Write data in write-only mode Sets data to be written continuously in write-only mode in which write operations are performed continuously with the same value.

Caution: In a link mode transfer, the RTN0SA register is overwritten by the descriptor read data.

(2) Next Destination Address Registers (RTN0DA, RTN1DA)

These registers set the DMA destination address of the DMA controller for real-time ports.

RTN0DAn is for the Next 0 register set, and RTN1DAn is for the Next 1 register set.

- Access These registers can be read or written in units of 32 bits.

Register	Bits 31–0	Address	Initial value
RTN0DA	Destination address	400A 2C04H	0000 0000H
R/W	R/W		
RTN1DA	Destination address	400A 2C10H	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	DA31–DA0	Destination address. Sets the start address of the DMA destination.

Caution: In a link mode transfer, the RTN0DA register is overwritten by the descriptor read data.

(3) Next Transaction Byte Registers (RTN0TB, RTN1TB)

These registers set the total number of transfer bytes (DMA transaction) of the DMA controller for real-time ports.

RTN0TB is for the Next 0 register set, and RTN1TB is for the Next 1 register set.

- Access These registers can be read or written in units of 32 bits.

Register	Bits 31–0	Address	Initial value
RTN0TB	Transaction byte	400A 2C08H	0000 0000H
R/W	R/W		
RTN1TB	Transaction byte	400A 2C14H	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	TB31–TB0	Number of transaction bytes. Sets the total number of bytes of a DMA transaction.

Cautions 1. Set the number of transfers by using the total number of bytes.

2. Setting '0' as the number of transaction bytes is prohibited.

3. In a link mode transfer, the RTN0TB register is overwritten by the descriptor read data.

18.4.4.2 Current Register Set

The Current register set is a set of read-only registers that indicate DMA source address, destination address, and total number of transfer bytes.

The set values are loaded from the Next 0/Next 1 register set when in register mode and from the descriptor read data when in link mode. Values cannot be written using software

(1) Current Source Address Register (RTCRSA)

This register indicates the DMA source address of the DMA controller for real-time ports.

- Access The register can only be read in units of 32 bits.

	31 … 0	Address	Initial value
RTCRSA	Source address	400A 2C18H	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRSA31–CRSA0	Current source address register. Indicates the read address of the next DMA transaction. During the DMA transaction, the register is automatically updated (fixed when RTCHCFG.SAD = 1 or undefined when RTCHCFG.WONLY = 1). The initial value is loaded from one of the following registers. In register mode: Loads the source address from RTN0SA / RTN1SA. In link mode: Loads the source address from the descriptor. (The descriptor read data is assigned to the RTN0SA register and, at the time of transfer, assigned to the RTCRSA register.) This register is updated when the read operation for the DMA transfer is completed. The register should be read when DMA is not in progress (when RTCHSTAT.TACT = 0). The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.

(2) Current Destination Address Register (RTCRDA)

This register indicates the DMA destination address of the DMA controller for real-time ports.

- Access The register can only be read in units of 32 bits.

	31 … 0	Address	Initial value
RTCRDA	Destination address	400A 2C1CH	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRDA31–CRDA0	Current destination address register. Indicates the write address of the next DMA transaction. During the DMA transaction, the register is automatically updated (fixed when RTCHCFGn.SAD = 1 or undefined when RTCHCFGn.WONLY = 1). The initial value is loaded from one of the following registers. In register mode: Loads the destination address from RTN0DA/RTN1DA. In link mode: Loads the destination address from the descriptor. (The descriptor read data is assigned to the RTN0DA register and, at the time of transfer, assigned to the RTCRDA register.) This register is updated when the write operation for the DMA transfer is completed. The register should be read when DMA is not in progress (when RTCHSTAT.TACT = 0). The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.

(3) Current Transaction Byte Register (RTCRTB)

This register indicates the total number of transfer bytes of the DMA controller for real-time ports.
Its value becomes 0000 0000H at the end of the DMA transaction (the series of DMA transfers).

- Access The register can only be read in units of 32 bits.

	31 … 0	Address	Initial value
RTCRTB	Transaction byte data	400A 2C20H	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRTB31–CRTB0	Current transaction byte register. Indicates the number of remaining transfer bytes during the DMA transaction (the series of DMA transfers) currently executed. During the DMA transaction, the register value is automatically decremented. The initial value is loaded from one of the following registers. In register mode: Loads the number of transfer bytes from RTN0TBA/RTN1TB. In link mode: Loads the number of transfer bytes from the descriptor. (The descriptor read data is assigned to the RTN0TB register and, at the time of transfer, assigned to the RTCRTB register.) This register is updated when the write operation for the DMA transfer is completed. The register should be read when DMA is not in progress (when RTCHSTAT.TACT = 0). The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.

(4) Channel Register Set

The channel register set is a set of registers used to set the DMA transfer operation and DMA transfer mode and read the status information.

(a) Channel status register (RTCHSTAT)

This register reads the status of the DMA controller for real-time ports.

- Access The register can only be read in units of 32 bits.

(1/6)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCHSTAT	DNUM								0	0	0	0	0	SWPRQ	DMARQM	INTM	0	0	0	0	MODE	DER	DW	DL	SR	TC	END	ER	SUS	TACT	RQST	EN
R/W	R	R	R	R	R	R	R	R	0	0	0	0	0	R	R	R	0	0	0	0	R	R	R	R	R	R	R	R	R	R	R	R

Address 400A 2C24H

Initial value 0000 0000H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="6">Description</th></tr>
<tr><td rowspan="3">31 to 24</td><td rowspan="3">DNUM</td><td colspan="6">Indicates the number of valid bytes in the buffer.
Data in the buffer refers to data that has been read from the source but not yet written to the destination.</td></tr>
<tr><td colspan="2">Increment condition</td><td colspan="2">Decrement condition</td><td colspan="2">Condition for clearing this bit to 0</td></tr>
<tr><td colspan="2">• The DMA read transfer is completed.</td><td colspan="2">• The DMA write transfer is completed.</td><td colspan="2">• The RTCHSTAT.EN bit clearing condition.
• The RTCHCTRL.SWRST bit is set to 1. (Channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
<tr><td>23 to 19</td><td>—</td><td colspan="6">Reserved. These bits return 0 when read.</td></tr>
<tr><td rowspan="4">18</td><td rowspan="4">SWPRQ</td><td colspan="6">Indicates the status of the forced dump request.
The status of the dump request initiated by the RTCHCTRL.SETSSWPRQ bit is indicated.
0: Forced dump request not asserted.
1: Forced dump request asserted.</td></tr>
<tr><td colspan="3">Condition for setting this bit to 1</td><td colspan="3">Condition for clearing this bit to 0</td></tr>
<tr><td colspan="3" rowspan="2">• The CHCTRLn.SETSSWPRQ bit is set to 1.</td><td colspan="3">• Forced dumping clears all data from the buffer.</td></tr>
<tr><td colspan="3">• RTCHCTRL.SWRST bit is set to 1.
(The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
</table>

(2/6)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">17</td><td rowspan="3">DMARQM</td><td colspan="2">Indicates the temporary mask status of the DMA transfer request input.
0: Not masked.
1: Temporarily masked</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The RTCHCTRL.SETDMARQM bit is set to 1.</td><td>• The RTCHCTRL.CLRDMARQM bit is set to 1.
• The RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
<tr><td rowspan="3">16</td><td rowspan="3">INTM</td><td colspan="2">Indicates the temporary mask status of the INTRTDMA interrupt output.
0: Temporarily mask released.
1: Temporarily mask applied</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The RTCHCTRL.SETINTM bit is set to 1.</td><td>• The RTCHCTRL.CLRINTM bit is set to 1.
• The RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
<tr><td>15 to 12</td><td>—</td><td colspan="2">Reserved. These bits return 0 when read.</td></tr>
<tr><td>11</td><td>MODE</td><td colspan="2">Indicates the DMA mode. This reflects the value of the DMS bit of the RTCHCFG register.
0: Register mode
1: Link mode</td></tr>
<tr><td rowspan="3">10</td><td rowspan="3">DER</td><td colspan="2">Descriptor error bit.
This bit is set to 1 when the LV bit (descriptor enable/disable bit) of the header of the read descriptor is set to 0 (the descriptor is disabled) in link mode. It is not dependent on the value of the RTCHCFG.DIM bit.
0: There is no descriptor error.
1: There is a descriptor error.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The LV bit of the descriptor header is set to 0 (the descriptor is disabled) when RTCHCFG.DRRP is set to 0 in link mode (the descriptor continues to be read until the descriptor is enabled (LV = 1)).</td><td>• RTCHCTRL.CLRDER bit is set to 1. (The DER bit, i.e., this bit is cleared.)
• RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
</table>

(3/6)

<table>
<tr><th>Bit Position</th><th>Bit Position</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">9</td><td rowspan="3">DW</td><td colspan="2">This bit is set to 1 during a writeback to the descriptor in link mode.
If a bus error [Note] is received during the writeback to the descriptor, the bit remains set and not cleared to 0.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for setting this bit to 1</td></tr>
<tr><td>• The writeback of the header starts in link mode.</td><td>• The writeback of the header is completed in link mode.
• The RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
<tr><td rowspan="3">8</td><td rowspan="3">DL</td><td colspan="2">This bit is set to 1 while loading the descriptor in link mode.
If a bus error [Note] is received while loading the descriptor, the bit remains set and not cleared to 0.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The descriptor is being loaded in link mode.</td><td>• Loading of the descriptor is completed in link mode.
• The RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
<tr><td rowspan="3">7</td><td rowspan="3">SR</td><td colspan="2">Indicates the register set selected in register mode.
0: Next 0 register set.
1: Next 1 register set</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• RTCHCFG.RSEL is set to 1. (when Next 1 register set is selected)</td><td>• RTCHCFG.RSEL is set to 0. (when Next 1 register set is selected)</td></tr>
<tr><td rowspan="3">6</td><td rowspan="3">TC</td><td colspan="2">This bit is set to 1 when the DMA transaction (the series of DMA transfers) is completed.
It is set to 1 only when RTCHCFG.TCM is set to 0 (RTDMATCZ: terminal count output enable).</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The total number of transfer bytes set in the RTCRTB register have been transferred in register mode.
• The total number of transfer bytes set in the CRTB4 register have been transferred when WBD is set to 1 for the descriptor header (the writeback of the LV bit of the header is disabled) in link mode.
• The descriptor writeback is completed when WBD is set to 0 for the descriptor header in link mode.</td><td>• The RTCHCTRL.CLRTC bit is set to 1. (The TCn bit, i.e., this bit is cleared.)
• The RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
</table>

Note: A bus error occurs during access to an undefined area, etc.
This bit can be cleared to 0 by setting the RTCHCTRL.SWRST bit to 1.

(4/6)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">5</td><td rowspan="3">END</td><td colspan="2">This bit is set to 1 when the DMA transaction (the series of DMA transfers) is completed and INTRTDMA occurs.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The condition for setting the TC bit to 1 and the following condition are met: RTCHCFG.DEM = 0 (INTRTDMA) DMA transfer completion interrupt output is enabled)
• The following condition are all met in link mode:
  • The LV bit of the descriptor header is set to 0 (descriptor disabled).
  • RTCHCFG.DRRP is set to 0.
  (When the LV bit of the descriptor header is set to 0, the DER bit is set to 1, causing a descriptor error and stopping the DMA transfer.)
  • RTCHCFG.DIM is set to 0. (When the LV bit of the descriptor header is set to 0, the descriptor error interrupt (INTRTDMA) is enabled.)</td><td>• The RTCHCTRL.CLREND bit is set to 1.
(The END bit, i.e., this bit is cleared.)
• The RTCHCTRL.SWRST bit is set to 1.
(The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
<tr><td rowspan="3">4</td><td rowspan="3">ER Note 1</td><td colspan="2">This bit is set to 1 when a transfer error Note 2 occurs during DMA transfer and the INTRTDMAERR interrupt occurs.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• A DMA transfer error occurs Note 2</td><td>• The RTCHCTRL.SWRSTn bit is set to 1.
(The channel status register (RTCHSTAT, i.e., this register) is cleared.)</td></tr>
<tr><td rowspan="3">3</td><td rowspan="3">SUS</td><td colspan="2">Indicates the suspend state of DMA channel n.
0: DMA channel n is not suspended.
1: DMA channel n is suspended.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• During the DMA transaction (the series of DMA transfers) for DMA channel n, the RTCHCTRL.SETSUS bit is set to 1 and the DMA transaction for DMA channel n is suspended.</td><td>• The RTCHCTRL.CLRSUS bit is set to 1.
(Release from the suspended state)
• The RTCHCTRL.CLREN bit is set to 1.
• The condition for clearing RTCHSTATn.EN bit is met.</td></tr>
<tr><td rowspan="3">2</td><td rowspan="3">TACT</td><td colspan="2">Indicates whether DMA channel n is active.
This bit is used to check that DMA channel n is completely inactive.
0: DMA is inactive.
1: DMA is active.</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• The RTCHCTRL.SETEN bit is set to 1.
(The system waits for the start of descriptor read or DMA trigger.)</td><td>• The RTCHSTAT.EN is set to 0 and the entire DMA transaction (the series of DMA transfers) is completed.</td></tr>
</table>

Notes 1. If transfer proceeds while the ER bit is set to 1, use processing to handle the series of associated DMA transfers as invalid.

2. A bus error occurs during access to an undefined area, etc.

(5/6)

Bit Position	Bit Name	Description	
1	RQST	Indicates whether a transfer request has been received. 0: A DMA transfer request has not been received. 1: A DMA transfer request has been received.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		• The RTCHCTRL.STG bit is set to 1 (when DMA is started by software). • A DMA transfer request is received in response to the DMA transfer trigger selected by the SELn bit of the RTCHCFG register.	• The RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.) • The RTCHCTRL.CLRRQ bit is set to 1. (The RQST bit, i.e., this bit is cleared.) • The DMA transfer ends in single transfer mode (RTCHCFG.TM = 0). (By using the RTCHCFG.REQD bit, the RTDMAACKZ output timing can be selected as either when it is read or when it is written. The condition for clearing this bit to 0 is when the read or write • The entire DMA transaction (the series of DMA transfers) is completed in register mode. (When RTCHCFG.REN is set to 0 (the next DMA transfer is not performed by using the Next register set specified by the RTCHCFG.RSEL bit after the DMA transaction (the series of DMA transfers) is completed).) • The DMA transfer for the last descriptor is completed in link mode. (When the LE bit of the descriptor header is set to 1 (link end).) • The DMA transfer is stopped during descriptor read in link mode (when LV is set to 0 and DRRP is set to 0 in the header). (LV = 0: Descriptor disabled) (RTCHCFG.DRRP = 0: When the LV bit of the descriptor header is set to 0, the DERn bit is set to 1, causing a descriptor error and stopping the DMA transfer.) • RTCHCFG.DEM is set to 0 (when the DMA transfer completion interrupt (INTRTDMAn) output is enabled and the DMA transaction (the series of DMA transfers) is completed). • A bus error [Note] occurs

Note: A bus error occurs during access to an undefined area, etc.

(6/6)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">0</td><td rowspan="3">EN</td><td colspan="2">Indicates whether the operation of the DMA controller is enabled or disabled.
0: Operation disabled
(The DMA transfer request which occurred during the stop of operation is suspended.)
1: Operation enabled</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• RTCHCTRL.SETEN is set to 1.</td><td>• The RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)
• The RTCHCTRL.CLREN bit is set to 1. (The EN bit, i.e., this bit is cleared.)
• The entire DMA transaction (the series of DMA transfers) is completed in register mode.
(When RTCHCFG.REN is set to 0 (the next DMA transfer is not performed by using the Next register set specified by the RTCHCFG.RSEL bit after the DMA transaction (the series of DMA transfers) is completed).)
• The DMA transfer for the last descriptor is completed in link mode.
(When the LE bit of the descriptor header is set to 1 (link end).) (When the WBD bit of the descriptor header is set to 0, this bit is cleared upon the completion of writeback.)
• When bus error occurs [Note]</td></tr>
</table>

Note: A bus error occurs during access to an undefined area, etc.

Cautions 1. If the ER bit is set to 1 for a transfer, the series of the associated DMA transfers should be handled as an invalid transaction.

2. To stop the DMA transaction (the series of DMA transfers), mask or clear the transfer request or clear the EN bit (follow the procedure described in section 18.8.13, Suspending Transfer).

3. Although setting up the use of transfer requests in the form of both a DMA transfer request signal and the software trigger for the same DMA channel is possible (by setting the RTCHCRTL.STG bit to 1), the source of a request is then not identifiable. Only use one of the two possible transfer requests at a time.

4. When starting transfer by software, check with the Current register or other data to confirm that the last requested DMA transfer has been completed, before manipulating the RTCHCTRL.STG bit.

(b) Channel control register (RTCHCTRL)

This register controls the DMA transfer operation of the DMA controller for real-time ports.

- Access The register can only be written in units of 32 bits. Any bit of the register does not affect the operation if 0 is written to it. A read operation results in 0 being read from all the bits.

(1/3)

RTCHCTRL — Address: 400A 2C28H — Initial value: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCHCTRL	0	0	0	0	0	0	0	0	0	0	0	0	CLRDMARQM	SETDMARQM	CLRINTM	SETINTM	0	SETSSWPRQ	0	SETREN	0	0	CLRSUS	SETSUS	CLRDER	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
R/W	0	0	0	0	0	0	0	0	0	0	0	0	W	W	W	W	0	W	0	W	0	0	W	W	W	W	W	W	W	W	W	W

Bit Position	Bit Name	Description
31 to 20	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
19	CLRDMARQM	Clears the temporary mask status for DMA transfer request input. When this bit is set to 1, the temporary mask status for hardware DMA transfer requests is cleared. This clears the RTCHSTAT.DMARQM (temporary mask status for DMA transfer requests) bit to 0. 0: Does not affect the operation. 1: Releases the temporary mask status for hardware DMA transfer requests enabled by setting SETDMARQM to 1.
18	SETDMARQM	Sets the mask status for DMA transfer request input. When this bit is set to 1, the temporary mask status is set for hardware DMA transfer requests. This sets the RTCHSTAT.DMARQM (temporary mask status for DMA transfer requests) bit to 1. 0: Does not affect the operation. 1: Masks hardware DMA transfer requests temporarily.
17	CLRINTM	Clears the mask status for INTRTDMA output. When this bit is set to 1, the mask status INTDMAn output is released. This clears the RTCHSTAT.INTM (temporary mask status for INTRTDMA interrupt output) bit to 0. If the mask is released when the DMA transfer has been completed, INTRTDMA is not output. 0: Does not affect the operation. 1: Releases the mask status for INTRTDMA output enabled by setting SETINTM to 1

(2/3)

Bit Position	Bit Name	Description
16	SETINTM	Sets the mask status for INTRTDMA output. When this bit is set to 1, the temporary mask status is set for INTRTDMA output. This sets the RTCHSTAT.INTM (temporary mask status for INTRTDMA output) bit to 1. 0: Does not affect the operation. 1: Masks INTRTDMA output.
15	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.
14	SETSSWPRQ	Forces the buffer to dump data. When this bit is set to 1, the buffer is forced to dump the data stored in it (see section 18.8.7, Forced Dumping), Note that, when RTCHCFG.REQD is set to 1 and RTDMAACKZ is asserted at the time of writing, forced dumping cannot be used. 0: Does not affect the operation. 1: Forces the buffer data not yet written to the destination to be written (dumped) to the destination.
13	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.
12	SETREN	Set this bit to 1 to proceed to the next DMA transfer using the Next register set specified by the RTCHCFG.RSEL bit after a DMA transaction (the series of DMA transfers) is completed in register mode. This sets the RTCHCFG.REN bit to 1. For details, see the description of the REN bit of the channel configuration register (RTCHCFG). 0: Does not affect the operation. 1: Sets RTCHCFG.REN to 1.
11, 10	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
9	CLRSUS	Releases the suspension of the ongoing DMA transfer. If this bit is set to 1 while RTCHSTAT.SUS is set to 1, the DMA channel is released from the suspended state. 0: Does not affect the operation. 1: Releases the suspension of the ongoing DMA transfer.
8	SETSUS	Sets the ongoing DMA transfer to be suspended. If this bit is set to 1 while RTCHSTAT.EN is set to 1 (the operation of the DMA channel is enabled), the active DMA channel is placed in the suspended state. 0: Does not affect the operation. 1: Suspends the ongoing DMA transfer.
7	CLRDER	Clears the descriptor error in link mode. When this bit is set to 1, the RTCHSTAT.DER (descriptor error) bit is cleared to 0. 0: Does not affect the operation. 1: Clears the RTCHSTAT.DER (descriptor error) bit to 0.

(3/3)

Bit Position	Bit Name	Description
6	CLRTC	Clears the terminal count (DMA transaction (the series of DMA transfers) completion) status. When this bit is set to 1, the RTCHSTAT.TC (terminal count) bit is cleared to 0. 0: Does not affect the operation. 1: Clears the RTCHSTAT.TC (terminal count) bit to 0.
5	CLREND	Clears the RTCHSTAT.END bit, which is set at the same time a DMA transaction (the series of DMA transfers) is completed and INTRTDMA occurs. When this bit is set to 1, the RTCHSTAT.END bit is cleared to 0. 0: Does not affect the operation. 1: Clears the RTCHSTAT.END bit to 0.
4	CLRRQ	Clears the DMA transfer request. When this bit is set to 1, the RTCHSTAT.RQST (DMA transfer request) bit is cleared to 0. 0: Does not affect the operation. 1: Clears the RTCHSTAT.RQST (DMA transfer request) bit to 0.
3	SWRST	Executes software reset for DMA channel. When this bit is set to 1, software reset is executed and each bit of the channel status register (RTCHSTAT) for which this operation is defined as the clearing condition is cleared to 0. Set this bit to 1 when the transfer on DMA channel n is completely stopped. To see whether the DMA channel transfer is completely stopped, check that both RTCHSTAT.EN and RTCHSTAT.TACT are set to 0. 0: Does not affect the operation. 1: Clear each bit of the RTCHSTAT register to 0 for which SWRST is defined as the clearing condition
2	STG	Serves as a software trigger for starting a DMA transfer by software. When this bit is set to 1, an internal transfer request is set (software trigger). If this bit is set to 1 at the same time as the SWRST bit, setting of the SWRST bit (software reset) is given priority. 0: Does not affect the operation. 1: Sets a transfer request by software (sets the RTCHSTATn.RQST bit to 1).
1	CLREN	Stops the operation of DMA channel. When this bit is set to 1, the RTCHSTAT.EN bit is cleared to 0 and the operation of DMA channel n is stopped (for details, see section 18.8.13, Suspending Transfer). 0: Does not affect the operation. 1: Stops the operation of DMA channel (clears the RTCHSTAT.EN bit to 0).
0	SETEN	Enables the operation of DMA channel n. When this bit is set to 1, the RTCHSTAT.EN bit is set to 1 and the operation of DMA channel n is enabled. If this bit is set to 1 at the same time as the SWRST bit, setting of the SWRST bit (software reset) is given priority. 0: Does not affect the operation. 1: Enables the operation of DMA channel n (sets the RTCHSTAT.EN bit to 1).

(c) Channel configuration register (RTCHCFG)

This register sets the DMA operation mode of the DMA controller for real-time ports.

- Access The register can be read or written in units of 32 bits.

(1/7)

	31	30	29	28	27	26	25	24	23	22	21	20	19–16	15–12	11	10–8	7	6	5	4	3	2–0	Address
RTCHCFG	DMS	REN	RSW	RSEL	SBE	DIM	TCM	DEM	WONLY	TM	DAD	SAD	DDS3–DDS0	SDS3–SDS0	DRRP	AM2–AM0	0	LVL	HEN	LEN	REQD	SEL2–SEL0	400A 2C2CH Initial value 0000 0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W R/W R/W R/W	R/W R/W R/W R/W	R/W	R/W R/W R/W	0	R/W	R/W	R/W	R/W	R/W R/W R/W	

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="2">Description</th></tr>
<tr><td>31</td><td>DMS</td><td colspan="2">Selects the DMA operation mode.
0: Register mode (initial value)
1: Link mode</td></tr>
<tr><td rowspan="3">30</td><td rowspan="3">REN</td><td colspan="2">Selects whether to proceed to the next DMA transfer following the completion of the DMA transaction (the series of DMA transfers). When proceeding to the next DMA transfer, the Next register set selected by the RSEL bit is used to perform the DMA transfer.
This setting is valid only in register mode.
When this bit is set to 1 during the DMA transaction, we recommend using the SETERN bit of the RTCHCTRL register.
0: Does not proceed to the next transfer.
1: Proceed to the next transfer (the Next register set selected by the RSEL bit is used).</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• This bit is set to 1.
• The RTCHCTRL.SETREN bit is set to 1.</td><td>• This bit is cleared to 0.
• When REN is set to 1, a DMA transaction (the series of DMA transfers) is completed.</td></tr>
<tr><td>29</td><td>RSW</td><td colspan="2">Selects whether to invert the RSEL (Next register set selection) bit when a DMA transaction (the series of DMA transfers) is completed.
This setting is valid only in register mode.
0: Does not invert RSEL after a DMA transaction (the series of DMA transfers) is completed (initial value).
1: Inverts RSEL after a DMA transaction (the series of DMA transfers) is completed.</td></tr>
</table>

(2/7)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="2">Description</th></tr>
<tr><td>28</td><td>RSEL</td><td colspan="2">Selects the Next register set to be used for the next DMA transfer.
This setting is valid only in register mode.
When RSW is set to 1, the bit is automatically inverted upon the completion of a DMA transaction (the series of DMA transfers).
0: Uses the Next 0 register set (initial value).
1: Uses the Next 1 register set</td></tr>
<tr><td>27</td><td>SBE</td><td colspan="2">Selects how to handle the data already read into the buffer, if the operation of DMA channel n is stopped by clearing RTCHCTRL.CLREN to 0 during a DMA transaction (the series of DMA transfers). Note that, if REQD is set to 1 and the mode is selected in which RTDMAACKZ is output at the time of writing, this bit cannot be set to 1.
0: Stops the transfer without dumping (writing) the buffer data (initial value).
1: Stops the transfer after dumping (writing) the buffer data.</td></tr>
<tr><td>26</td><td>DIM</td><td colspan="2">Selects how the descriptor error interrupt (INTRTDMAERR) behaves if the LV bit of the descriptor header is set to 0 in link mode.
0: Does not mask INTRTDMAERR (initial value).
1: Masks INTDMAERR.</td></tr>
<tr><td rowspan="3">25</td><td rowspan="3">TCM</td><td colspan="2">Masks terminal count output (RTDMATCZ).
If this bit is set to 1 when the terminal count is output, RTDMATCZ is not output. RTCHSTAT.TC is not set to 1, either. In this case, the bit is automatically cleared to 0 in register mode or not cleared to 0 in link mode.
Use this bit when controlling DMA transfers by software.
0: Does not mask (enables terminal count output (RTDMATCZ); initial value).
1: Masks (disables terminal count output (RTDMATCZ).)</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• This bit is set to 1.</td><td>• This bit is cleared to 0.
• The DMA transaction (the series of DMA transfers) is completed when this bit is set to 1 in register mode.</td></tr>
<tr><td rowspan="3">24</td><td rowspan="3">DEM</td><td colspan="2">Selects how INTRTDMA behaves when a DMA transaction (the series of DMA transfers) is completed.
If this bit is set to 1 when INTRTDMA occurs, INTRTDMA is not output. RTCHSTAT.END is not set to 1, either. In this case, the bit is automatically cleared to 0 in register mode or not cleared to 0 in link mode.
0: Does not mask (enable INTRTDMA output, initial value).
1: Masks (disables INTRTDMA output).</td></tr>
<tr><td>Condition for setting this bit to 1</td><td>Condition for clearing this bit to 0</td></tr>
<tr><td>• This bit is set to 1.</td><td>• This bit is cleared to 0.
• The DMA transaction (the series of DMA transfers) is completed when this bit is set to 1 in register mode.</td></tr>
</table>

(3/7)

Bit Position	Bit Name	Description
23	WONLY	Selects normal mode or write-only mode. In write-only mode, the data set in the Next source address register (RTN0SA or RTN1SA) is written to the address indicated by the Next destination address register (RTN0DA or RTN1DA). Use write-only mode to perform write operations continuously with the same value. 0: Normal mode (initial value) 1: Write-only mode.
22	TM	Selects the DMA transfer mode. 0: Single transfer mode (performs a single transfer for each DMA transfer request; initial value). 1: Block transfer mode (transfers the number of bytes set in the transaction byte register for a DMA transfer request).
21	DAD	Sets the counting direction of the destination address of DMA channel n. 0: Increment (initial value). 1: Fixed. **Caution: Do not select 1 (fixed) in DAD when the destination is using skip mode or the beats are not aligned on the destination side.**
20	SAD	Sets the counting direction of the source address of DMA channel n. 0: Increment (initial value) 1: Fixed **Caution: Do not select 1 (fixed) in SAD when the source is using skip mode or the beats are not aligned on the source side.**
19	DDS3	Selects normal mode or skip mode for DMA destination addressing. 0: Normal mode (initial value) 1: Skip mode

(4/7)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>18 to 16</td><td>DDS2–
DDS0</td><td>Sets the transfer size of the DMA destination.
<table><tr><th>DDS2</th><th>DDS1</th><th>DDS0</th><th>DMA destination transfer size</th></tr><tr><td>0</td><td>0</td><td>0</td><td>8 bits (initial value)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>16 bits</td></tr><tr><td>0</td><td>1</td><td>0</td><td>32 bits</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>128 bits Note</td></tr><tr><td colspan="3">Other than the above</td><td>Setting prohibited</td></tr></table>
Note: These bits can be set only when addresses are aligned in units of the transfer size.</td></tr>
<tr><td>15</td><td>SDS3</td><td>Selects normal mode or skip mode for DMA source addressing.
0: Normal mode (initial value)
1: Skip mode</td></tr>
<tr><td>14 to 12</td><td>SDS2–
SDS0</td><td>Sets the transfer size of the DMA source.
<table><tr><th>SDS2</th><th>SDS1</th><th>SDS0</th><th>DMA source transfer size</th></tr><tr><td>0</td><td>0</td><td>0</td><td>8 bits (initial value)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>16 bits</td></tr><tr><td>0</td><td>1</td><td>0</td><td>32 bits</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>128 bits Note</td></tr><tr><td colspan="3">Other than the above</td><td>Setting prohibited</td></tr></table>
Note: These bits can be set only when addresses are aligned in units of the transfer size.</td></tr>
</table>

(5/7)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>11</td><td>DRRP</td><td>Selects the operation if the descriptor header is disabled (LV = 0) in link mode.
0: Sets the RTCHSTAT.DER (descriptor error) bit to 1 and stops the operation (initial value).
1: Continues to read the same descriptor until LV becomes 1. When LV becomes 1, a DMA transfer is started by using that descriptor. To set the interval at which the descriptor is to be read, use the descriptor interval register (RTDSCITVL).</td></tr>
<tr><td>10 to 8</td><td>AM2–AM0</td><td>Selects the output mode of the DMA acknowledge signal.
<table>
<tr><th>AM2</th><th>AM1</th><th>AM0</th><th>DMA acknowledge signal (RTDMAACKZ) output mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Pulse mode Note 1 (initial value)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Level mode
The active level is maintained until the DMA transfer request (RTDMAREQZ) becomes inactive.</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>Bus cycle mode Note 2
The active level is maintained during a DMA transfer bus cycle.</td></tr>
<tr><td>1</td><td>X</td><td>X</td><td>DMA acknowledge signal (RTDMAACKZ) output disabled.</td></tr>
</table>
Notes 1. A pulse of one BUSCLK cycle is output as the RTDMAACKZ signal.
2. In bus cycle mode, the DMA acknowledge signal is output following the point at which acquisition of bus mastership is requested. For this reason, the DMA acknowledge signal is output earlier than the actual DMA bus cycle, and a bus cycle of an internal master which has previously acquired mastership of the same bus may proceed at this time.

Caution: The settings of AM2 to AM0 may duplicate those of the RTDMAIFC register. In general, however, when the RTDMAACKZ signal is set to the level mode by using AM2 to AM0, the RTDMAIFC register should be left at its initial value. Conversely, when the RTDMAIFC register is used to extend the RTDMAACK pulse width or for the RTDMAREQZ mask function, set AM2 to AM0 to select the pulse mode.

Remark: X: Don’t Care</td></tr>
<tr><td>7</td><td>—</td><td>Reserved. When writing to this bit, write 0. When read, 0 is returned.</td></tr>
</table>

(6/7)

Bit Position	Bit Name	Description
6	LVL	Selects the method of detecting a DMA transfer request signal. A DMA transfer request is chosen by using the DMA trigger source register 4 (RTDTFR). The procedure for detecting a DMA transfer request signal differs with the selected DMA transfer request.
5	HEN	
4	LEN	(see below)
3	REQD	Selects when RTDMAACKZ is to become active. Usually, set this bit so that RTDMAACKZ is output to the side on which RTDMAREQZ is asserted. 0: Makes RTDMAACKZ active when reading (RTDMAREQZ is the source). 1: Makes RTDMAACKZ active when writing (RTDMAREQZ is the destination).

[In the case where the DMA transfer request signals are the DMA request signals of the external pins]

The internal DMA interface is positive logic.

The DMA interface pins (RTDMAREQZ, RTDMAACKZ, and RTDMATCZ) are negative logic.

Since the signals of the DMA interface pins are inverted at the connection to the system bus DMAC signals, the opposite logic to that selected by the settings of the HENn and LENn bits is chosen.

LVL	HEN	LEN	Detection Method of DMA Transfer Request Signal (RTDMAREQZ)		
				Internal Signal	External Pin
0	0	0	Edge detection	Detection disabled	
0	0	1		Falling edge detection	Rising edge detection
0	1	0		Rising edge detection	Falling edge detection
0	1	1		Rising/falling edge detection (Does not recommend)	
1	0	0	Level detection	Detection disabled	
1	0	1		Low level detection	High level detection
1	1	0		High level detection	Low level detection
1	1	1		Setting prohibited	

[In the case where the DMA transfer request signals are interrupt signals (signals which start with INT).]

LVL	HEN	LEN	Detection Procedure of the DMA Transfer Request Signal by an Interrupt Signal	
0	0	0	Edge detection	Detection disabled
0	0	1		Low level detection
0	1	0		High level detection
0	1	1		Setting prohibited
1	×	×	Level detection	Setting prohibited

(7/7)

Bit Position	Bit Name	Description
2 to 0	SEL2–SEL0	Selects the DMA interface signal for each channel. Since the DMA controller for real-time ports (unit 1) only has one channel, the only available setting is for RTDTFR.

SEL2	SEL1	SEL0	DMA interface signal selection
0	0	0	The DMA transfer source selected by RTDTFR is chosen.
Other than the above			Setting prohibited

(d) Channel interval register (RTCHITVL)

This register sets the DMA transfer interval of the DMA controller for real-time ports.

The specifiable interval values are the internal system bus clock (HCLK) cycle × the value of ITVL15–ITVL0.

- Access The register can be read or written in units of 32 bits.

For details, see section 18.8.9, Interval Counting.

	31 to 16	15 to 0	Address	Initial value
RTCHITVL	0	ITVL15–ITVL0	400A 2C30H	0000 0000H
R/W	0	R/W		

Bit Position	Bit Name	Description
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 0	ITVL15–ITVL0	Set the DMA transfer interval of DMA channel.

(5) Link Register Set

This is a register set that indicates the link addresses in link mode.

When the DMA controller is started by setting a descriptor address in the RTNXLA register, the hardware loads the value of the RTNXLA register to the RTCRLA register and the descriptor is read. The DMAC starts a DMA transaction based on that descriptor value. The RTNXLA register is automatically updated based on the link address value in the read descriptor, and its value is used as the descriptor address for the next DMA transaction.

(a) Next link address register (RTNXLA)

This register sets the link address of the DMA controller for real-time ports.

Set the address to which the descriptor in link mode is allocated.

- Access The register can be read or written in units of 32 bits.

For information about link mode, see section 18.7.3, Link Mode.

	31 to 2	1	0	Address	Initial value
RTNXLA	NXLA31–NXLA2	0	0	400A 2C38H	0000 0000H
R/W	R/W	0	0		

Initial Value	Bit Name	Description
31 to 0	NXLA31–NXLA2	Sets the link address in link mode. Only an address aligned by word (32 bits) can be set. The lower-order two bits are fixed at 0.

(b) Current link address register (RTCRLA)

This register indicates the address of the descriptor currently executed in link mode.

- Access The register can only be read in units of 32 bits.

	31 to 0	Address	Initial value
RTCRLA	CRLA31–CRLA0	400A 2C3CH	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRLA31–CRLA0	Indicates the address of the descriptor currently executed in link mode.

(c) Continuous space source size register (RTSCNT)

This register sets the size of the continuous access space for access to the source by the DMA controller for real-time ports in bytes. The register is used in combination with the skip space source size register (RTSSKP).

To use skip mode for the source address, set the SDS3 bit of the channel configuration register (RTCHCFG) to 1.

Do not set the SAD bit of the channel configuration register (RTCHCFG) to 1 (fixed at the source address).

Moreover, please do not set "0000 0000 H" to this register in skip mode.

- Access The register can be read or written in units of 32 bits.

	31 … 0	Address	Initial value
RTSCNT	SCNT31–SCNT0	400A 2E00H	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	SCNT31–SCNT0	Specifies the size of the continuous access space for the source address in bytes in skip mode.

(d) Skip space source size register (RTSSKP)

This register sets the size of the skip space for access to the source by the DMA controller for real-time ports in bytes. The register is used in combination with continuous space source size register (RTSCNT).

To use skip mode for the source address, set the SDS3 bit of the channel configuration register (RTCHCFG) to 1.

Do not set the SAD bit of the channel configuration register (RTCHCFG) to 1 (fixed at the source address).

- Access The register can be read or written in units of 32 bits.

	31 … 0	Address	Address
RTSSKP	SSKP31–SSKP0	400A 2E04H	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	SSKP31–SSKP0	Specifies the size of the skip space for the source address in bytes in skip mode.

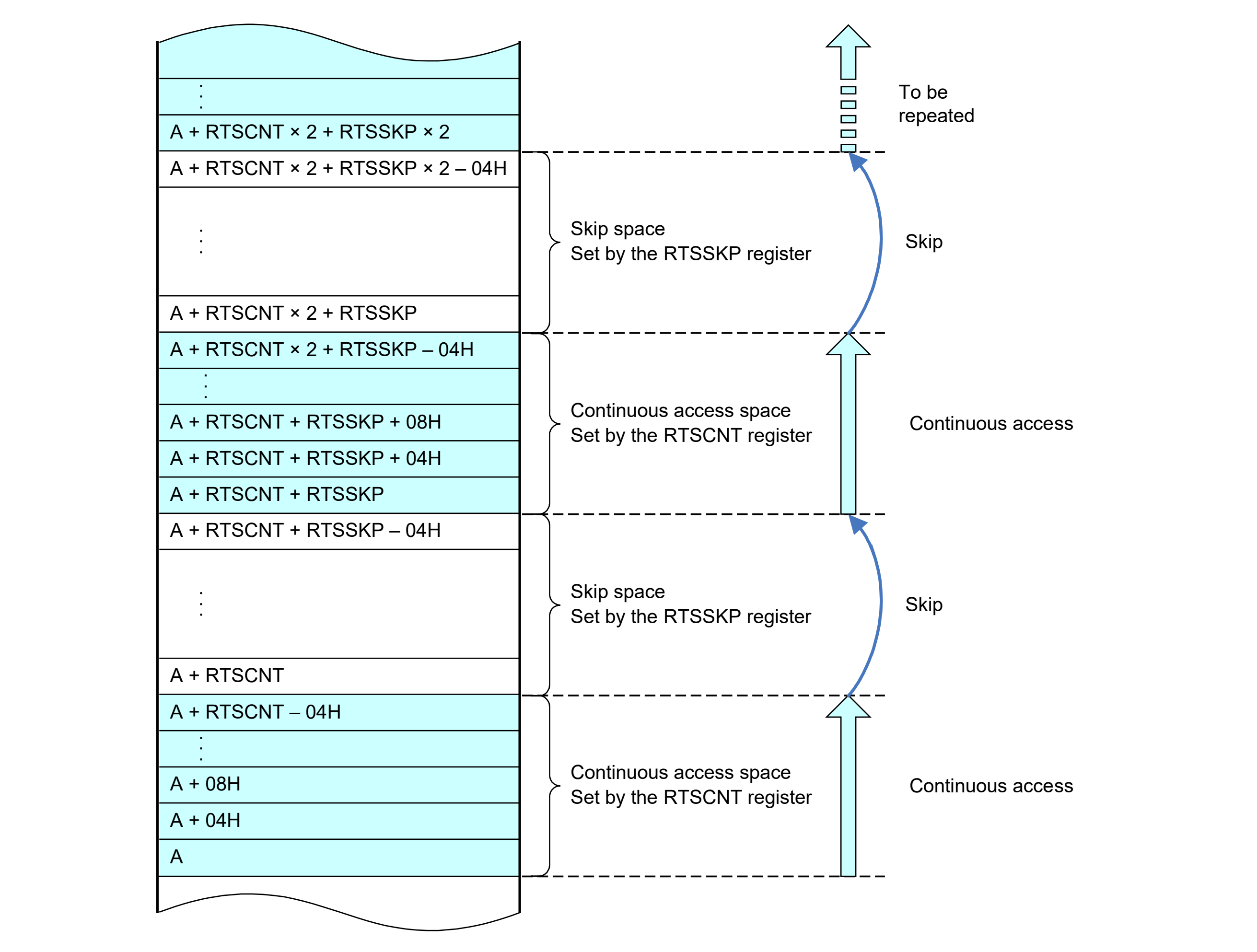


Figure 18.6 Relationship between the RTSSKP and RTSCNT Registers in Skip Mode

Remark: The values of RTSCNT and RTSSKP can be set independently of the source address and the value of the SDS2–SDS0 bit (source data size) of the channel configuration register (RTCHCFG).
The DMA controller is accessed in the size set by SDS2 to SDS0, and only valid data is retrieved into the buffer.

(e) Continuous space destination size register (RTDCNT)

This register sets the size of the continuous access space for access to the destination by the DMA controller for real-time ports in bytes. The register is used in combination with the skip space destination size register (RTDSKP).

To use skip mode for the destination address, set the DDS3 bit of the channel configuration register (RTCHCFG) to 1.

Do not set the DAD bit of the channel configuration register (RTCHCFG) to 1 (fixed at the destination address).

Also, do not set 0000 0000H in this register in skip mode.

- Access The register can be read or written in units of 32 bits.

	31–0	Address	Initial value
RTDCNT	DCNT31–DCNT0	400A 2E08H	0000 0000H
R/W	R/W		

Bit Position	Bit Position	Description
31 to 0	DCNT31–DCNT0	Specifies the size of the continuous access space for the destination address in bytes in skip mode.

(f) Skip space destination size register (RTDSKP)

This register sets the size of the skip space for access to the destination by the DMA controller for real-time ports in bytes.
The register is used in combination with the continuous space destination size register (RTDCNT).

To use skip mode for the destination address, set the DDS3 bit of the channel configuration register (RTCHCFG) to 1.

Do not set the DAD bit of the channel configuration register (RTCHCFG) to 1 (fixed at the destination address).

- Access The register can be read or written in units of 32 bits.

	31–0	Address	Initial value
RTDSKP	DSKP31–DSKP0	400A 2E0CH	0000 0000H
R/W	R/W		

Bit Position	Bit Name	Description
31 to 0	DSKP31–DSKP0	Specifies the size of the skip space for the destination address in bytes in skip mode.

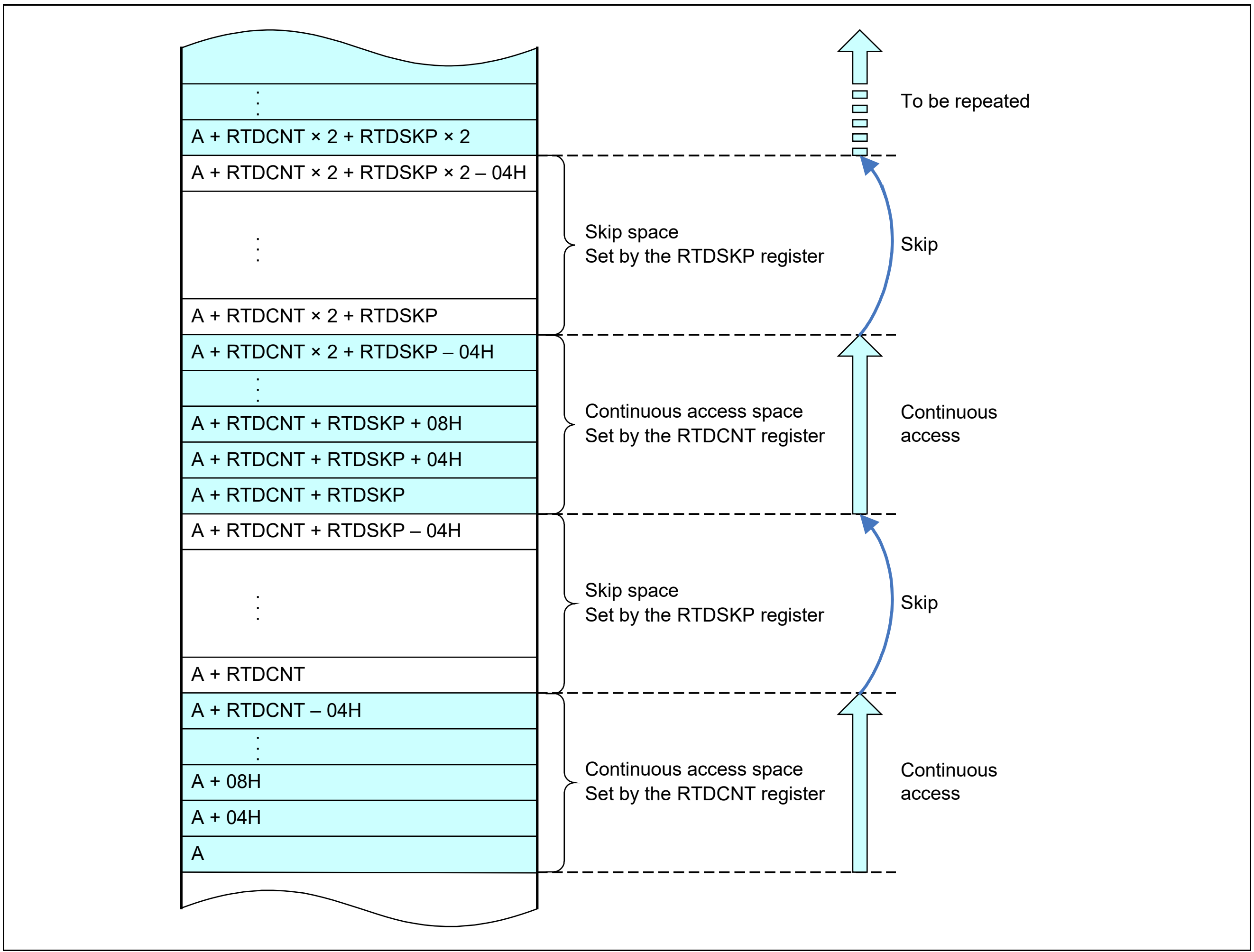


Figure 18.7 Relationship between the RTDSKP and RTDCNT Registers in Skip Mode

Remark: The values of RTDCNT and RTDSKP can be set independently of the destination address and the value of the DDS2–DDS0 bit (destination data size) of the channel configuration register (RTCHCFG). The DMA controller only writes to the specified space in combinations of sizes equal to or smaller than that set by DDS2 to DDS0.

(6) DMA Control Registers

(a) DMAC control register (RTDCTRL)

This register selects the transfer priority control mode.

Since the DMAC for real-time ports only has one channel, the setting of this register has no effect.

Be sure to set bits 31 to 1 to 0.

- Access The register can be read or written in units of 32 bits.

	31 … 1	0	Address	Initial value
RTDCTRL	0	PR	400A 2F00H	0000 0000H
R/W	0	R/W		

Bit Position	Bit Name	Description
31 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	PR	Selects the transfer priority control mode. 0: Fixed priority mode. 1: Round robin mode.

(b) Descriptor interval register (RTDSCITVL)

If the descriptor header is read in link mode when the DRRP bit of the channel configuration register (RTCHCFG) set to 1, and if the LV bit is set to 0 (descriptor disabled), the descriptor continues to be read until LV becomes 1.

This register sets the interval at which the descriptor is to be read in such a case. It can be set in units of the internal system bus clock (HCLK) cycle × 256.

- Access The register can be read or written in units of 32 bits.

	31 … 16	15 … 8	7 … 0	Address	Initial value
RTDSCITVL	0	DITVL15–DITVL8	0	400A 2F04H	0000 0000H
R/W	0	R/W	0		

Bit Position	Bit Name	Description
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 8	DITVL15–DITVL8	Sets the interval at which the descriptor header continues to be read until the LV bit becomes 1. The descriptor is read in the (DITVL15–DITVL8 value) × 256 × internal system bus clock (HCLK) cycles.
7 to 0	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.

(c) DMAC enable status register (RTDSTEN)

This register indicates the state of the EN (enable) bit.

- Access The register can only be read in units of 32 bits.
  Writing to the register does not change the value of any of its bits.

To set EN to 1 (enable DMA channel n), set the SETEN bit of the channel control register (RTCHCTRL) to 1.
To set EN to 0 (disable DMA channel n), set the CLREN bit of the channel control register (RTCHCTRL) to 1.

Remark: The EN bit is the 0th bit of the channel status register (RTCHSTAT).

RTDSTEN — Address 400A 2F10H, Initial value 0000 0000H

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

Bit Position	Bit Name	Description
31 to 1	—	Reserved. These bits return 0 when read.
0	EN	Indicates the EN (DMA enable) bit of DMA channel.

(d) DMAC error status register (RTDSTER)

This register indicates the state of the ER (error) bit.

- Access The register can only be read in units of 32 bits.
  Writing to this register does not change the value of the bits.

If an error occurs in a DMA transfer bus cycle, 1 is set. To clear the bit to 0, the SWRST bit of the channel control register (RTCHCTRL) needs to be set to 1. While the ER bit is set to 1, use processing to handle the series of associated DMA transfers as invalid.

Remark: The ER bit is the 4th bit of the channel status register (RTCHSTAT).

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

Address 400A 2F14H
Initial value 0000 0000H

Bit Position	Bit Name	Description
31 to 1	—	Reserved. These bits return 0 when read.
0	ER	Indicates the ER (DMA transfer error) bit of DMA channel.

(e) DMAC end status register (RTDSTEND)

This register indicates the state of the END bit (indicating generation of INTRTDMA on completion of the DMA transaction (the series of DMA transfers)).

- Access The register can only be read in units of 32 bits.
  Writing to the register does not change the value of any of its bits.

For information about the setting and clearing conditions, see the description of the END bit of the channel status register (RTCHSTAT).

Remark: The END bit is the 5th bit of the channel status register (RTCHSTAT).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 400A 2F18H
RTDSTEND	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	END	Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	

Bit Position	Bit Name	Description
31 to 1	—	Reserved. These bits return 0 when read.
0	END	Indicates the END (DMA transaction completion) bit of DMA channel.

Table 18.8 Correspondence between DMA End Status Register and Interrupt Signal

Register Name	Bit Name	Corresponding Transfer Completion Interrupt Signal
DSTEND	END	INTRTDMA

(f) DMAC terminal count status register (RTDSTTC)

This register indicates the state of the TC bit (indicating the completion of the DMA transaction (the series of DMA transfers)).

- Access The register can only be read in units of 32 bits.
  Writing to the register does not change the value of any of its bits.

For information about the setting and clearing conditions, see the description of the TC bit of the channel status register (RTCHSTAT).

Remark: The TC bit is the 6th bit of the channel status register (RTCHSTAT).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 400A 2F1CH
RTDSTTC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TC	Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	

Bit Position	Bit Name	Description
31 to 1	—	Reserved. These bits return 0 when read.
0	TC	Indicates the TC (DMA transaction completion) bit of DMA channel.

(g) DMAC suspend status register (RTDSTSUS)

This register indicates the state of the SUS (suspended state) bit.

- Access The register can only be read in units of 32 bits.
  Writing to the register does not change the value of any of its bits.

To set SUS to 1 (set the suspended state), set the SETSUS bit of the channel control register (RTCHCTRL) to 1.
To set SUS to 0 (release from the suspended state), set the CLRSUS bit of the channel control register (RTCHCTRL) to 1.

Remark: The SUS bit is the 3rd bit of the channel status register (RTCHSTAT).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
RTDSTSUS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SUS	400A 2F20H Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	

Bit Position	Bit Name	Description
31 to 1	—	Reserved. These bits return 0 when read.
0	SUS	Indicates the SUS (suspended state) bit of DMA channel.

18.4.5 DMA Transfer Interface Signal Control Registers (DMAIFC0, DMAIFC1, RTDMAIFC)

These registers set the active level width of the DMA acknowledge output signal (DMAACKZp, RTDMAACKZ) and the mask width of the DMA transfer request input signal (DMAREQZp, RTDMAREQZ) in units of bus clock BUSCLK cycles. The registers can be read or written in units of 32 bits.

DMA transfer requests (DMAREQZp or RTDMAREQZ) are acknowledged following the input of at least one cycle of BUSCLK. When a DMA transfer request is acknowledged, the active level of the DMA acknowledge signal (DMAACKZp or RTDMAACKZ) is output for at least one cycle of BUSCLK.

Generally, the circuit should be designed so that the DMA acknowledge signal is detected based on BUSCLK, making the DMA transfer request inactive. If BUSCLK is fast, the timing design is difficult to create. Therefore, there is also a built-in mechanism for setting the active level width arbitrarily and masking the DMA transfer request signal when the DMA acknowledge signal returns to the inactive state, so as to allow the DMA acknowledge signal to be detected easily by an external circuit.

This enables an external circuit estimate to be made based on BUSCLK, making it easy to connect a low-speed device.

Cautions 1. Two sets of DMA input and output pin functions are usable by the general-purpose DMAC (unit 0) and one set is usable by the DMAC for real-time ports (unit 1).

2. The operation mode of DMAACKZp/RTDMAACKZ output can also be controlled by using the channel configuration register (CHCFGn, RTCHCFG). In addition, the DMA transfer interface signal control register (DMAIFCp, RTDMAIFC) has a mask function for preventing a DMA transfer request overrun due to DMAREQZp/RTDMAREQZ input.

3. The settings of the AM2 to AM0 bits of the CHCFGn and RTCHCFG registers may duplicate those of the DMAIFCp and RTDMAIFC registers. In general, however, when the DMAACKZp or RTDMAACKZ signal is set to the level mode by using AM2 to AM0 of the CHCFGn or RTCHCFG register, the DMAIFC1, DMAIFC0, or RTDMAIFC register should be left at its initial value. Conversely, when the DMAIFC1, DMAIFC0, or RTDMAIFC register is used to extend the DMAACKZp or RTDMAACKZ pulse width or for the DMAREQZp or RTDMAREQZ mask function, set the AM2 to AM0 bits of the CHCFGn or RTCHCFG register to select the pulse mode.

4. An external pin's minimum acknowledge time of the DMA transfer request signal (DMAREQZp/RTDMAREQZ) is 1 × BUSCLK.

5. An external pin's minimum output period of the DMA acknowledge signal (DMAACKZp/RTDMAACKZ) is 1 × BUSCLK.

6. Only when a priority needs to be replaced within the channel of an external DMA transfer request (DMAREQZp, RTDMAREQZ), please change a DMA trigger by SEL1/SEL0 bit.

7. These registers are only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

Remark: n = 0 to 3; p = 0, 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
DMAIFC0 DMAIFC1	DIF EN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4001 0720H 4001 0724H
R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
	0	0	0	RQ MK4	RQ MK3	RQ MK2	RQ MK1	RQ MK0	0	0	0	AK WD4	AK WD3	AK WD2	AK WD1	AK WD0	0000 0000H
R/W	0	0	0	R/W	R/W	R/W	R/W	R/W	0	0	0	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31	DIFEN	Selects whether to enable or disable the DMA transfer interface signal control function. 0: Disables the function (initial value). 1: Enables the function.
30 to 13	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
12 to 8	RQMK4–RQMK0	Sets the mask width Note 1 of the DMA transfer request signal (DMAREQZp) in units of BUSCLK.
7 to 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4 to 0	AKWD4–AKWD0	Sets the active level width Note 2 of the DMA acknowledge signal (DMAACKZp) in units of BUSCLK.

RQ MK4	RQ MK3	RQ MK2	RQ MK1	RQ MK0	DMAREQZp signal mask width Note 1
0	0	0	0	0	0 BUSCLK cycles (initial value)
0	0	0	0	1	1 BUSCLK cycle
0	0	0	1	0	2 BUSCLK cycles
:					:
1	1	1	1	0	30 BUSCLK cycles
1	1	1	1	1	31 BUSCLK cycles

AK WD4	AK WD3	AK WD2	AK WD1	AK WD0	DMAACKZp signal active level width Note 2
0	0	0	0	0	+ 0 BUSCLK cycles (initial value)
0	0	0	0	1	+ 1 BUSCLK cycle
0	0	0	1	0	+ 2 BUSCLK cycles
:					:
1	1	1	1	0	+ 30 BUSCLK cycles
1	1	1	1	1	+ 31 BUSCLK cycles

Notes 1. The mask starts at the rising edge (change to inactive) of DMAACKZp.

2. The active level width of DMAACKZp is based on the acknowledge signal specified by the AM2 to AM0 bits of the CHCFGn register. The AM0 bit of the CHCFGn register allows the DMAACKZp output mode to be selected from pulse mode and level mode.

3. Only a single DMA_IF circuit is present for each pin.

Remark: p = 0, 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
RTDMAIFC	DIF EN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4001 0728H
R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
	0	0	0	RQ MK4	RQ MK3	RQ MK2	RQ MK1	RQ MK0	0	0	0	AK WD4	AK WD3	AK WD2	AK WD1	AK WD0	0000 0000H
R/W	0	0	0	R/W	R/W	R/W	R/W	R/W	0	0	0	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31	DIFEN	Selects whether to enable or disable the DMA transfer interface signal control function. 0: Disables the function (initial value). 1: Enables the function.
30 to 13	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
12 to 8	RQMK4–RQMK0	Sets the mask width Note 1 of the DMA transfer request signal (RTDMAREQZ) in units of BUSCLK.
7 to 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4 to 0	AKWD4–AKWD0	Sets the active level width Note 2 of the DMA acknowledge signal (RTDMAACKZ) in units of BUSCLK.

RQMK4–RQMK0:

RQ MK4	RQ MK3	RQ MK2	RQ MK1	RQ MK0	RTDMAREQZ Signal Mask Width Note 1
0	0	0	0	0	0 BUSCLK cycles (initial value)
0	0	0	0	1	1 BUSCLK cycle
0	0	0	1	0	2 BUSCLK cycles
:					:
1	1	1	1	0	30 BUSCLK cycles
1	1	1	1	1	31 BUSCLK cycles

AKWD4–AKWD0:

AK WD4	AK WD3	AK WD2	AK WD1	AK WD0	RTDMAACKZ Signal Active Level Width Note 2
0	0	0	0	0	+ 0 BUSCLK cycles (initial value)
0	0	0	0	1	+ 1 BUSCLK cycle
0	0	0	1	0	+ 2 BUSCLK cycles
:					:
1	1	1	1	0	+ 30 BUSCLK cycles
1	1	1	1	1	+ 31 BUSCLK cycles

Notes 1. The mask starts at the rising edge (change to inactive) of RTDMAACKZ.

2. The active level width of RTDMAACKZ is based on the acknowledge signal specified by the AM2 to AM0 bits of the RTCHCFG register. The AM0 bit of the RTCHCFG register allows the RTDMAACKZ output mode to be selected from pulse mode and level mode.

3. Only a single DMA_IF circuit is present for each pin.

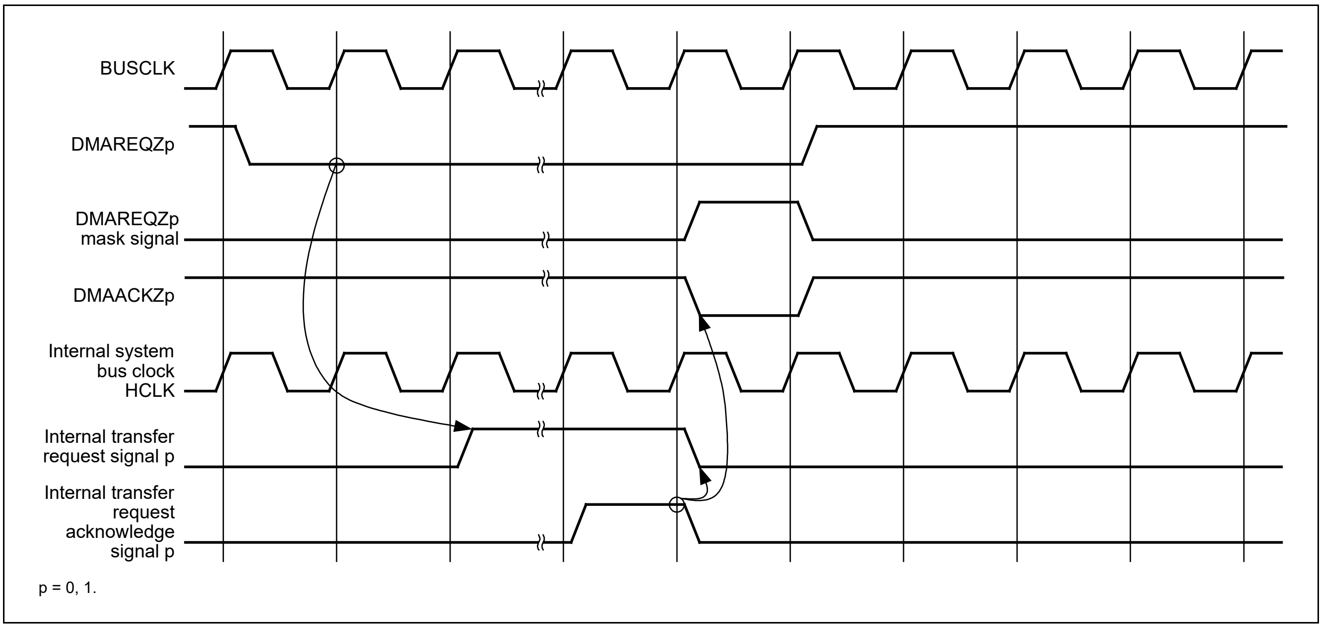


Figure 18.8 DMA Pin Signals and Internal Signals (1) (DMAIFCp = 8000 0000H)

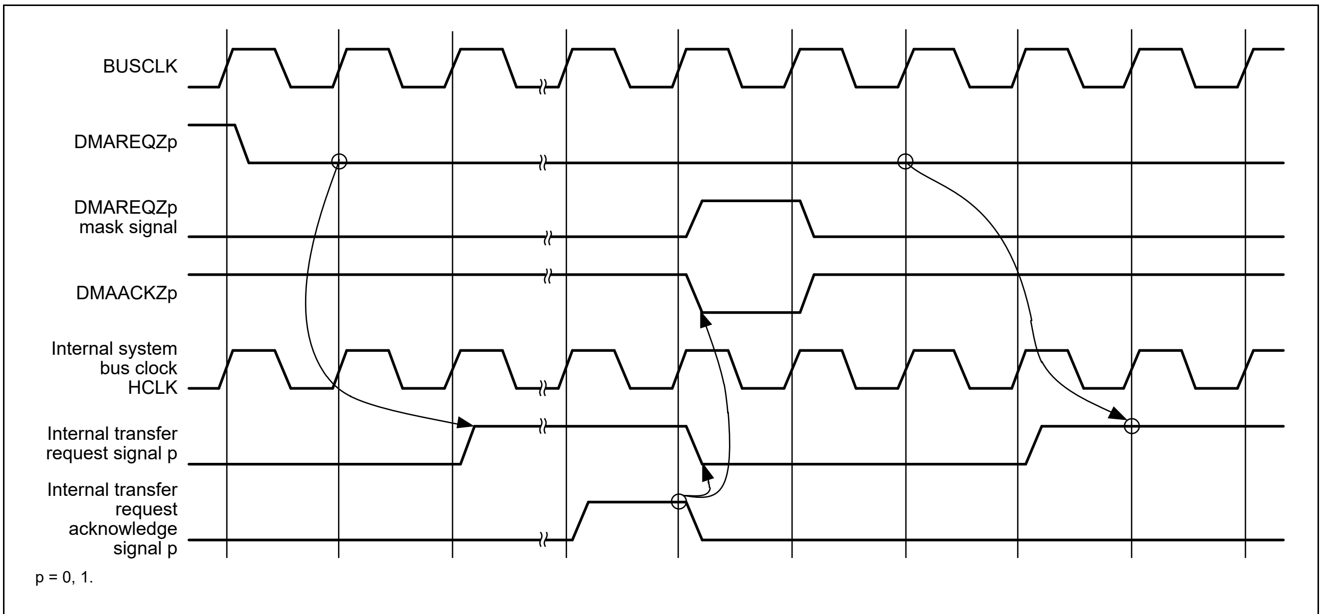


Figure 18.9 DMA Pin Signals and Internal Signals (2) (DMAIFCp = 8000 0000H)

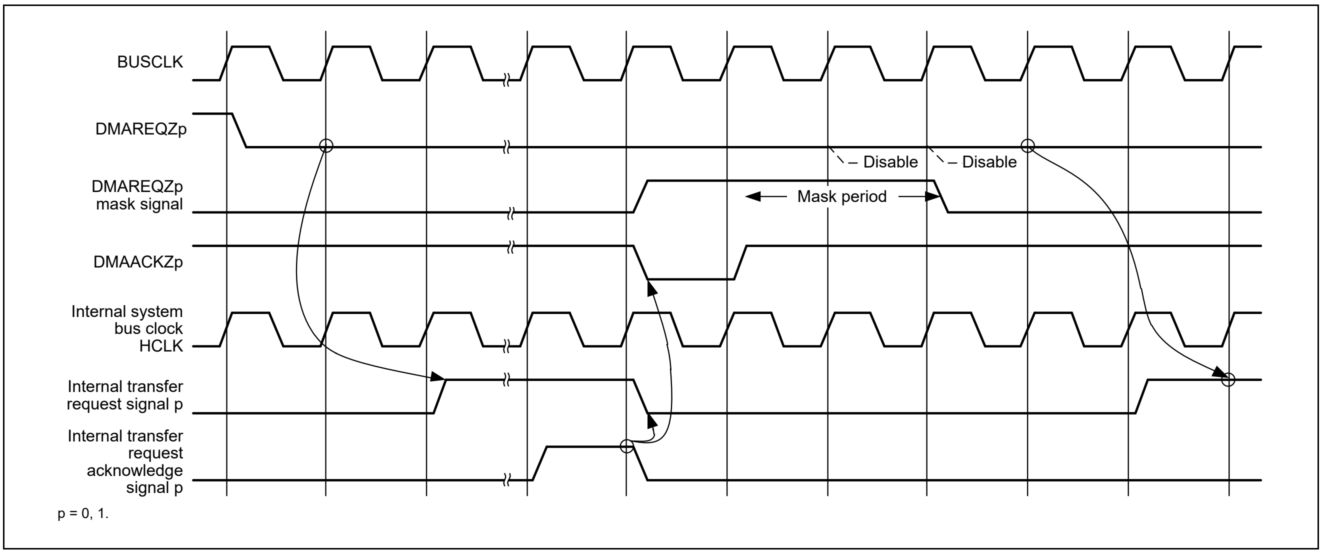


Figure 18.10 DMA Pin Signals and Internal Signals (3) (DMAIFCp = 8000 0200H)

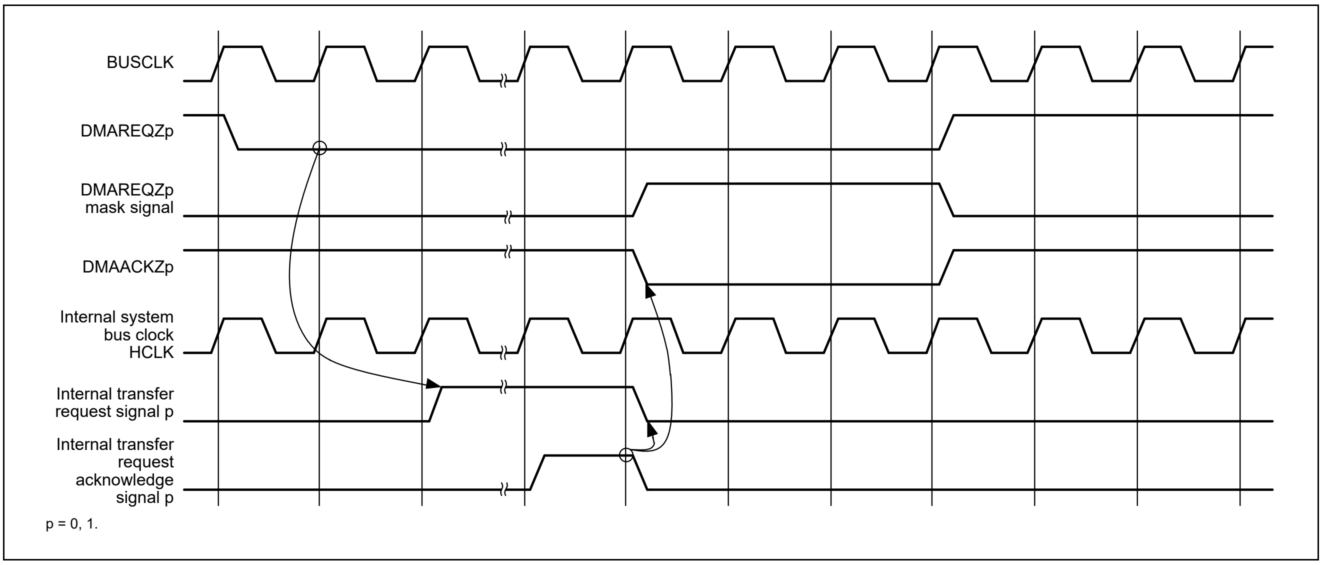


Figure 18.11 DMA Pin Signals and Internal Signals (4) (DMAIFCp = 8000 0002H)

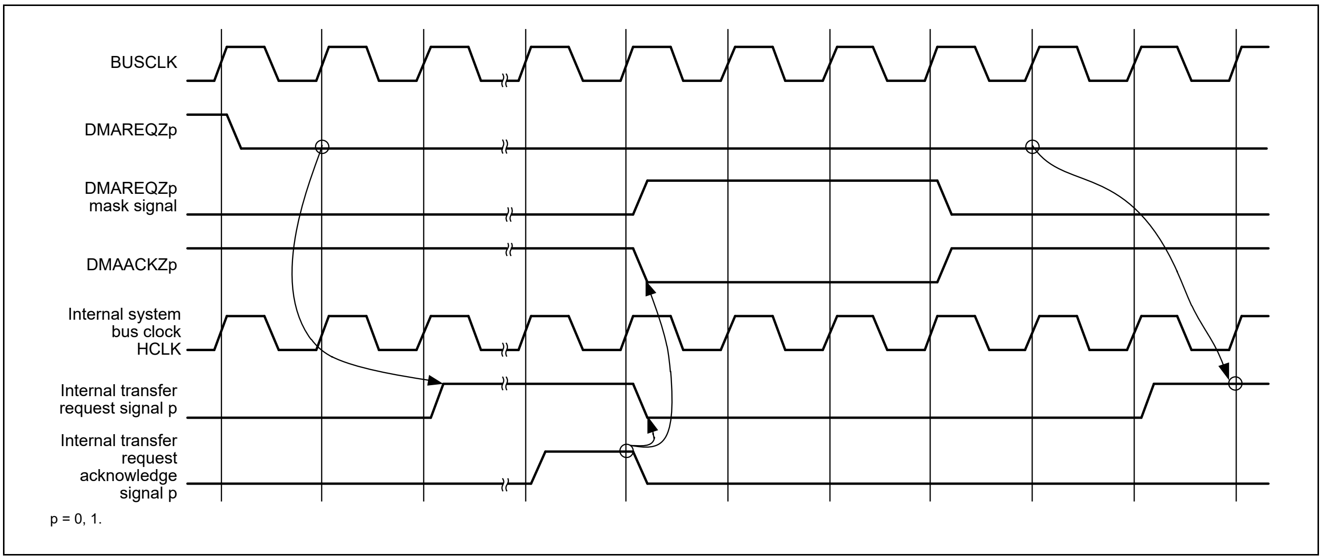


Figure 18.12 DMA Pin Signals and Internal Signals (5) (DMAIFCp = 8000 0002H)

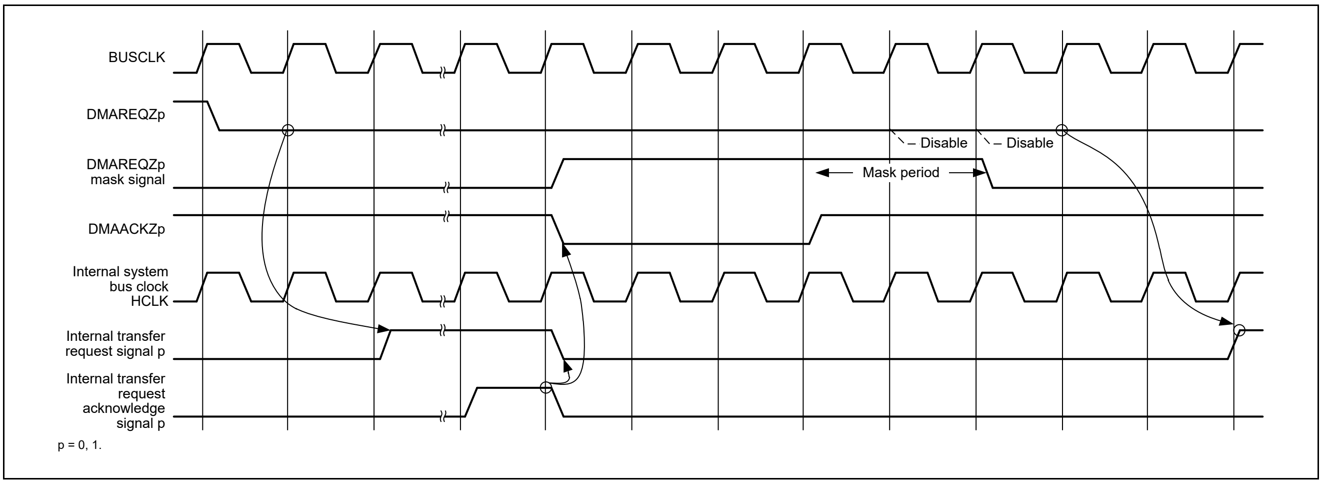


Figure 18.13 DMA Pin Signals and Internal Signals (6) (DMAIFCp = 8000 0202H)

18.4.6 DMA Trigger Source Registers (DTFRn, RTDTFR)

These registers are used to select the DMA transfer request by the interrupt from the DMAREQZp and RTDMAREQZ (DMA transfer request pin) internal peripheral function, and the interrupt from the external interrupt pin input. The interrupt source selected by these registers is a trigger for starting DMA transfer.

There are a total of five DTFRn and RTDFTR registers, which equals the number of system bus DMAC channels, and they are assigned to the individual DMA channels according to the setting of the SEL2 to SEL0 bits in the channel control registers (CHCFGn and RTCHCFG).

The following triggers can be selected for DMA transfer requests.

- Access These registers can be read or written in units of 32 bits.

Cautions 1. When you change the setting of the DTFRn register, do so after stopping operation of the DMA controller.

2. These registers are only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

Remark: All the interrupt request signals are resynchronized with the internal system bus clock (HCLK). n = 0 to 3, p = 0, 1

DTFRn — Address 4001 0730H + 4n, Initial value 0000 0000H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IFC7	IFC6	IFC5	IFC4	IFC3	IFC2	IFC1	IFC0
0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTDTFR — Address 4001 0740H, Initial value 0000 0000H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IFC7	IFC6	IFC5	IFC4	IFC3	IFC2	IFC1	IFC0
0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Remark: n = 0 to 3

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>7 to 0</td><td>IFC7–IFC0</td><td>Select the trigger source of the DMA channel (four channels for the general-purpose, and one channel for the real-time port).
<table>
<tr><th>IFCn7–IFCn0</th><th>Selection of a DMA Transfer Trigger Source</th></tr>
<tr><td>00H</td><td>Mask the DMA transfer trigger source (transmission request is not generated).</td></tr>
<tr><td>01H</td><td>DMAREQZ0 pin (DMA transfer request) input Note</td></tr>
<tr><td>02H</td><td>DMAREQZ1 pin (DMA transfer request) input Note</td></tr>
<tr><td>03H</td><td>RTDMAREQZ0 pin (DMA transfer request) input Note</td></tr>
<tr><td>04H</td><td>TAUJ2 channel 0 interrupt</td></tr>
<tr><td>05H</td><td>TAUJ2 channel 1 interrupt</td></tr>
<tr><td>06H</td><td>TAUJ2 channel 2 interrupt</td></tr>
<tr><td>07H</td><td>TAUJ2 channel 3 interrupt</td></tr>
<tr><td>08H</td><td>UARTJ0 transmission interrupt</td></tr>
<tr><td>09H</td><td>UARTJ0 reception interrupt</td></tr>
<tr><td>0AH</td><td>UARTJ1 transmission interrupt</td></tr>
<tr><td>0BH</td><td>UARTJ1 reception interrupt</td></tr>
<tr><td>0CH</td><td>CSIH0 communications status interrupt</td></tr>
<tr><td>0DH</td><td>CSIH0 reception status interrupt</td></tr>
<tr><td>0EH</td><td>CSIH0 end of job interrupt</td></tr>
<tr><td>0FH</td><td>CSIH1 communications status interrupt</td></tr>
<tr><td>10H</td><td>CSIH1 reception status interrupt</td></tr>
<tr><td>11H</td><td>CSIH1 end of job interrupt</td></tr>
<tr><td>12H</td><td>IICB0 data transmission/reception interrupt</td></tr>
<tr><td>13H</td><td>IICB1 data transmission/reception interrupt</td></tr>
<tr><td>14H</td><td>FCN0 reception completion interrupt</td></tr>
<tr><td>15H</td><td>FCN0 transmission completion interrupt</td></tr>
<tr><td>16H</td><td>FCN0 sleep and wakeup / transmission suspension interrupt</td></tr>
<tr><td>17H</td><td>FCN1 reception completion</td></tr>
<tr><td>18H</td><td>FCN1 transmission completion</td></tr>
<tr><td>19H</td><td>FCN1 sleep and wakeup / transmission suspension interrupt</td></tr>
</table>
</td></tr>
</table>

Note: External DMA transfer request inputs on the DMAREQZ0, DMAREQZ1, and RTDMAREQZ pins can be individually set as DMA transfer trigger requests for the corresponding registers listed below. Do not use other combinations.

DMA Transfer Trigger Source	**Selectable DMA Trigger Source Register**
DMAREQZ0 pin input	**DTFR0**
DMAREQZ1 pin input	**DTFR1**
RTDMAREQZ pin input	**RTDTFR**

Bit Position	Bit Name	Description
7 to 0	IFC7–IFC0	Select the trigger source of the DMA channel (four channels for the general-purpose, and one channel for the real-time port).

IFCn7–IFCn0	Selection of a DMA Transfer Trigger Source
1AH	General-purpose DMAC channel 0 transfer completion interrupt
1BH	General-purpose DMAC channel 1 transfer completion interrupt
1CH	General-purpose DMAC channel 2 transfer completion interrupt
1DH	General-purpose DMAC channel 3 transfer completion interrupt
1EH	Real-time ports DMAC transfer completion interrupt
1FH	TAUD channel 0 interrupt
20H	TAUD channel 1 interrupt
21H	TAUD channel 2 interrupt
22H	TAUD channel 3 interrupt
23H	TAUD channel 4 interrupt
24H	Inter-buffer DMA transfer completion interrupt
25H	Gigabit Ethernet PHY port 0 interrupt
26H	Gigabit Ethernet PHY port 1 interrupt
27H	Ethernet MII management access completion interrupt
28H	Ethernet pause packet transmission completion interrupt
29H	Ethernet transmission completion interrupt
2AH	Ethernet SWITCH interrupt
2BH	Ethernet SWITCH DLR interrupt
2CH	Ethernet SWITCH SEC interrupt
2DH–2EH	Reserved (setting prohibited)
2FH	Ethernet MACDMA reception completion interrupt
30H	Ethernet MACDMA transmission completion interrupt
31H	Receive frame successfully interrupt
32H	Reserved (setting prohibited)
33H	INTPZ0 input [Note 1]
34H	INTPZ1 input [Note 1]
35H	INTPZ2 input [Note 1]
36H	INTPZ3 input [Note 1]
37H	INTPZ4 input [Note 1]
38H	INTPZ5 input [Note 1]
39H	INTPZ6 input [Note 1]
3AH	INTPZ7 input [Note 1]

Note 1. When the external interrupt is used as the DMA trigger source, be sure to specify edge. (Do not set level detection.)

Bit Position	Bit Name	Description
7 to 0	IFC7–IFC0	Select the trigger source of the DMA channel (four channels for the general-purpose, and one channel for the real-time port).

IFCn7–IFCn0	Selection of a DMA Transfer Trigger Source
3BH	INTPZ8 input Note 1
3CH	INTPZ9 input Note 1
3DH	INTPZ10 input Note 1
3EH	INTPZ11 input Note 1 /TAUD channel 5 interrupt Note 2
3FH	INTPZ12 input Note 1 /TAUD channel 6 interrupt Note 2
40H	INTPZ13 input Note 1 /TAUD channel 7 interrupt Note 2
41H	INTPZ14 input Note 1 /TAUD channel 8 interrupt Note 2
42H	INTPZ15 input Note 1 /TAUD channel 9 interrupt Note 2
43H	INTPZ16 input Note 1 /TAUD channel 10 interrupt Note 2
44H	INTPZ17 input Note 1 /TAUD channel 11 interrupt Note 2
45H	INTPZ18 input Note 1 /TAUD channel 12 interrupt Note 2
46H	INTPZ19 input Note 1 /TAUD channel 13 interrupt Note 2
47H	INTPZ20 input Note 1 /TAUD channel 14 interrupt Note 2
48H	INTPZ21 input Note 1 /TAUD channel 15 interrupt Note 2
49H	INTPZ22 input Note 1
4AH	INTPZ23 input Note 1
4BH	INTPZ24 input Note 1
4CH	INTPZ25 input Note 1
4DH	INTPZ26 input Note 1
4EH	INTPZ27 input Note 1
4FH	INTPZ28 input Note 1
50H–62H	Reserved (setting prohibited)
63H	Gigabit Ethernet PHY LED0_PHY0 input interrupt
64H	Gigabit Ethernet PHY LED0_PHY1 input interrupt
65H–6CH	Reserved (setting prohibited)

Notes 1. When the external interrupt is used as the DMA trigger source, be sure to specify edge. (Do not set level detection.)

2. The INTPZ/TAUD interrupt is selected by using the INTSEL register. For details, see section 28.18, INTPZ/Timer Interrupt Select Register (INTSEL).

Bit Position	Bit Name	Description
7 to 0	IFC7–IFC0	Select the trigger source of the DMA channel (four channels for the general-purpose, and one channel for the real-time port).

IFCn7–IFCn0	Selection of a DMA Transfer Trigger Source
6DH–6EH	Reserved (setting prohibited)
6FH	CC-Link IE Field Network NMIZ interrupt
70H	CC-Link IE Field Network WDTZ interrupt
71H	CC-Link IE Field Network INTZ interrupt
72H	CC-Link IE Field Network CLKLOSSZ interrupt
73H–7CH	Reserved (setting prohibited)
7DH	Gigabit Ethernet PHY LED1_PHY0 input interrupt
7EH	Gigabit Ethernet PHY LED1_PHY1 input interrupt
7FH	INTPZ29 input [Note1]

Note 1. When the external interrupt is used as the DMA trigger source, be sure to specify edge. (Do not set level detection.)

Bit Position	Bit Name	Description
7 to 0	IFC7–IFC0	Select the trigger source of the DMA channel (four channels for the general-purpose, and one channel for the real-time port).

IFCn7–IFCn0	Selection of a DMA Transfer Trigger Source
80H–83H	Reserved (setting prohibited)
84H	Bus error response interrupt (HSNGNREG)
85H	Bus error response interrupt (HSNGN)
86H	Bus error response interrupt (HSNGN)
87H	Reserved (setting prohibited)
88H	Watchdog timer error interrupt for external CPU
89H–97H	Reserved (setting prohibited)
98H	Built-in regulator interrupt
99H–FFH	Reserved (setting prohibited)

18.5 DMA Interface Pins

18.5.1 BUSCLK Synchronization

All DMA interface signals are synchronized with BUSCLK output. BUSCLK is a signal of the same phase as the internal system bus clock HCLK. The timing of the input of the DMA transfer request input signal (DMAREQZp or RTDMAREQZ) must meet the BUSCLK setup and hold requirements.

Remark: p = 0, 1

18.5.2 Transfer Request and Acknowledge

For DMA transfer requests (DMAREQZp and RTDMAREQZ), the following detection methods are supported.

- Rising edge detection
- Falling edge detection
- Transition point detection
- High level detection
- Low level detection
- Mask (DMAREQZp and RTDMAREQZ are not used as a trigger source.)

The following DMA acknowledge (DMAACKZp and RTDMAACKZ) output modes are supported.

- Assert a pulse when the transfer starts
- Continue to assert pulses until the DMA transfer request signal is deasserted
- Continue to assert pulses during the bus cycle
- Mask (DMAACKZp and RTDMAACKZ are not output)

Generally, the circuit should be designed so that the DMA acknowledge signal is detected based on BUSCLK, making the DMA transfer request inactive. If BUSCLK is fast, the timing design is difficult to create. Therefore, there is also a built-in mechanism for setting the active level width arbitrarily and masking the DMA transfer request signal when the DMA acknowledge signal returns to the inactive state, so as to allow the DMA acknowledge signal to be detected easily by an external circuit (DMA_IF module built-in function).

Remark: p = 0, 1

18.6 Interrupt Output

When a DMA transaction is completed, or when an invalid descriptor is read in link mode (when DIM in the header is set to 0, LV in the read descriptor header is set to 0), the transfer completion interrupt is asserted. Also, if an error response is returned in response to a transfer request issued by the master interface, the error response interrupt is asserted.

Table 18.9 General DMA Controller Interrupt Output

Interrupt Signal	Interrupt Source	Interrupt Detection Mask	Interrupt Output Mask
INTDMAn	The DMA transaction is completed.	CHCFGn register DEM = 1	CHSTATn. INTM = 1
	An invalid descriptor is read in link mode.	DIM in the header = 1	
INTDMEERR	An error response is returned in response to a transfer request issued by the master interface.	— (Not available)	— (Not available)

Remark: n = 0 to 3

Table 18.10 Interrupt Output of DMA Controller for Real-Time Ports

Interrupt Signal	Interrupt Source	Interrupt Detection Mask	Interrupt Output Mask
INTRTDMA	The DMA transaction is completed.	RTCHCFG register DEM = 1	RTCHSTAT. INTM = 1
	An invalid descriptor is read in link mode.	DIM in the header = 1	
INTRTDMEERR	An error response is returned in response to a transfer request issued by the master interface.	— (Not available)	— (Not available)

18.7 DMAC Operation Setting

Caution: This section explains only operation of the general-purpose DMAC since the specifications of operations of the general-purpose DMAC and the DMAC for real-time ports are the same.

18.7.1 Register Mode and Link Mode Selection

By using the DMS bit (bit 31) of the channel configuration register (CHCFGn), select register mode or link mode.

Table 18.11 Register Mode and Link Mode

CHCFGn Register DMS Bit	Mode	Operation
0	Register mode	Performs DMA transfer by using the values set in the Next register set.
1	Link mode	Performs DMA transfer by setting a descriptor in the Current register. The process of loading a descriptor and performing DMA transfer is repeated unless it is stopped by a descriptor setting or the channel control register (CHCTRLn).

Remark: n = 0 to 3

18.7.2 Register Mode

In register mode, DMA transfer is performed by using the values set in the Next register set.

Two types of source address, destination address, and number of transfer bytes (Next 0 register set and Next 1 register set) can be set.

DMA transfer is possible by selecting the Next register to be used and by using the two Next register sets consecutively (execution of the DMA transaction by using the Next 1 register after the completion of the DMA transaction by using the Next 0 register, etc.).

The figure below shows examples of loading the registers when Next 0 and Next 1 are used.

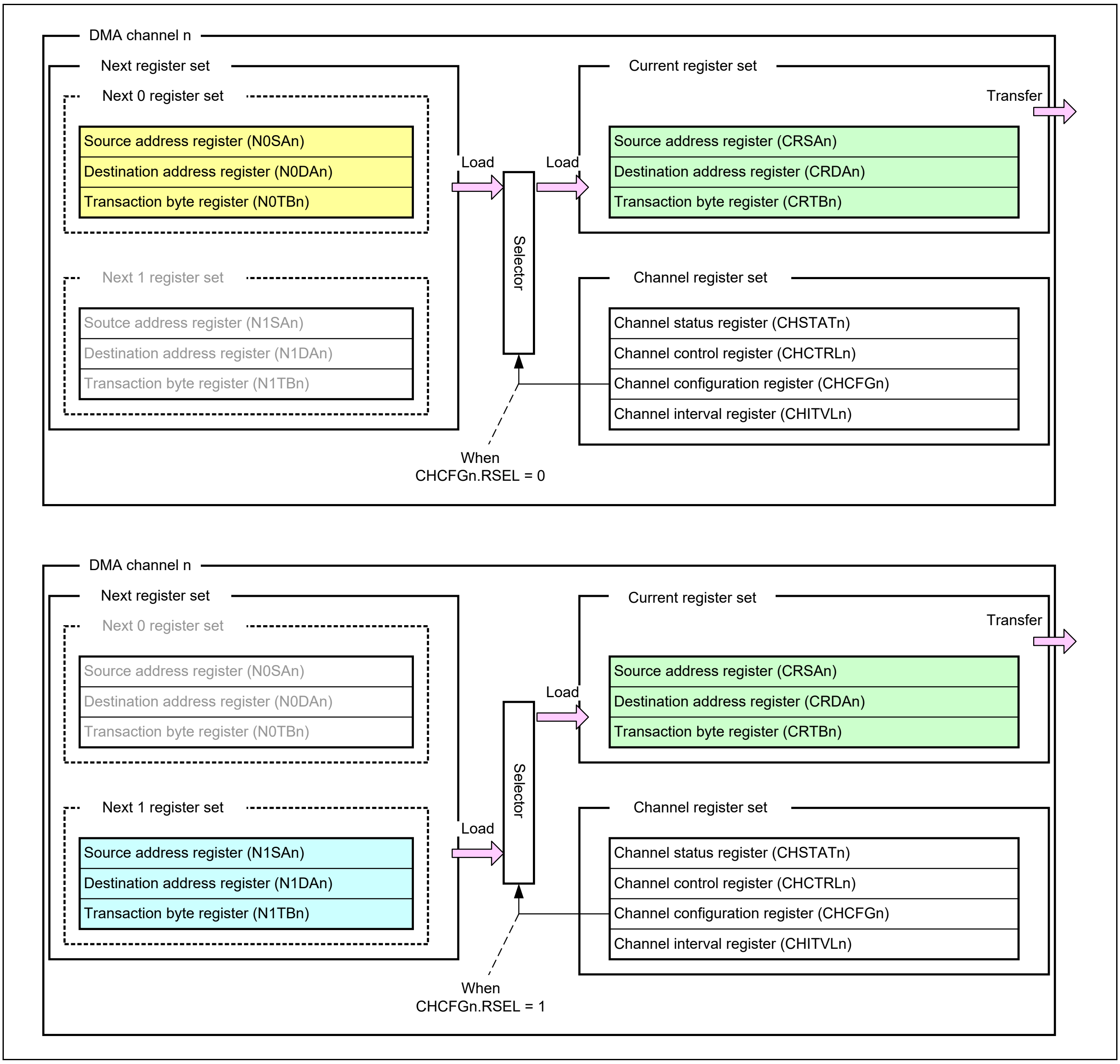
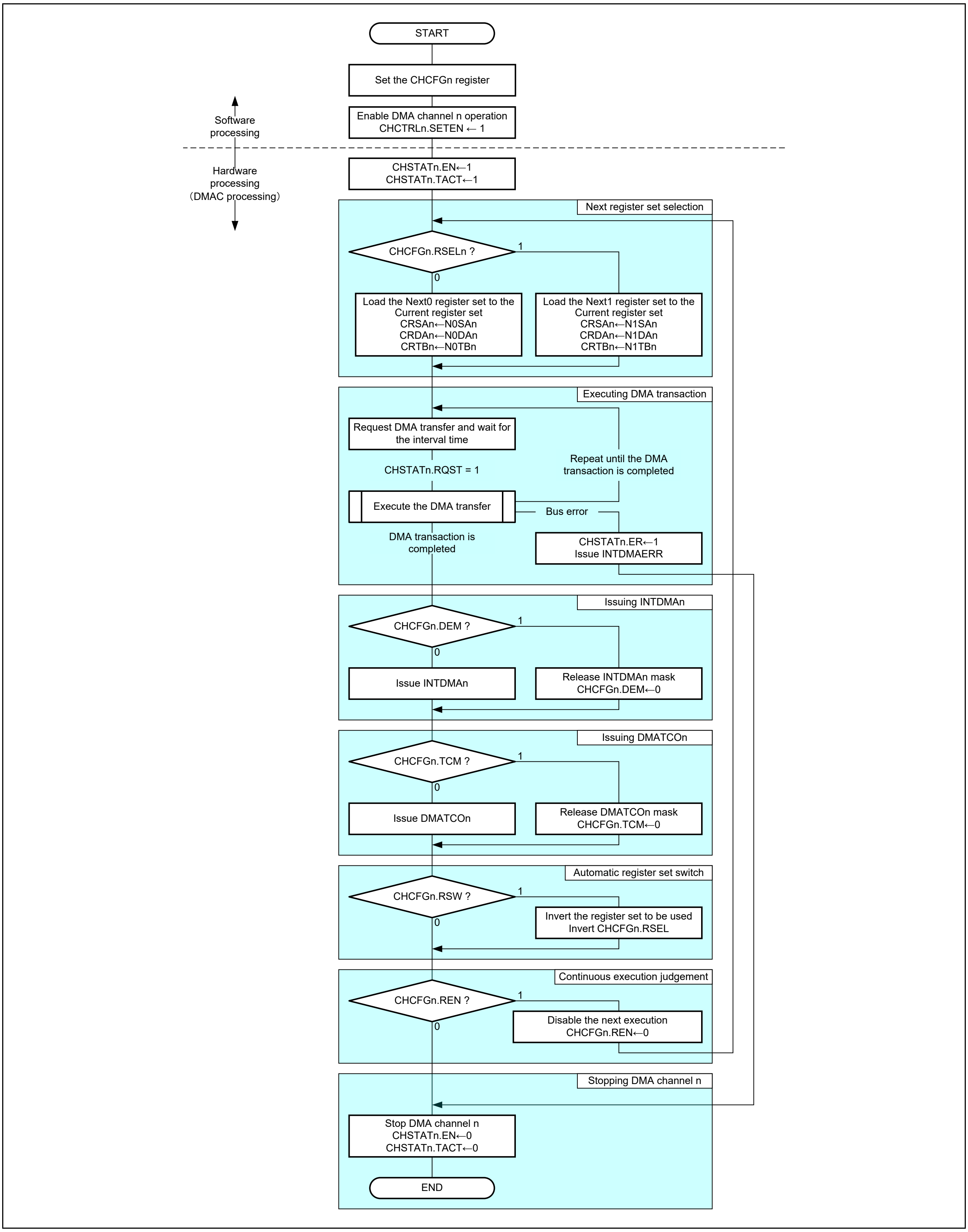


Figure 18.14 Outline of the Register Mode Operation

(1) Register Mode Operation Flow



<1> Channel configuration

Set the Next 0 or Next 1 register set (destination address, source address, and total number of transfer bytes).
Set the operation mode by using the channel configuration register (CHCFGn).

<2> Next register set selection

When the SETEN bit of the channel control register (CHCTRLn) register is set to 1, the EN and TACT bits of the channel status register (CHSTATn) are set to 1 and the values set in the Next register set selected by the CHCFGn.RSEL bit are loaded to the Current register set.

<3> Executing DMA transaction

Execute DMA transfer according to the settings. For details of the transfer, see section 18.8, DMAC Operation.
If a DMA transfer error during this process, INTDMAERRn is issued and the DMA transfer ends.

<4> Issuing INTDMAn

According to the value set in the CHCFGn.DEM bit, INTDMAn is masked.
When DEM is set to 1, INTDMAn is not issued. Also, the DEM bit is automatically cleared to 0 immediately after that.

<5> Issuing DMATCZp

According to the value set in the CHCFGn.TCM bit, DMATCZp output is masked.
When TCM is set to 1, DMATCZp is not output. Also, the TCM bit is automatically cleared to 0 immediately after that.

<6> Automatic register set switch

According to the value set in the CHCFGn.RSW bit, it is determined whether to use the other Next register set.

<7> Continuous execution judgment

According to the value set in the CHCFGn.REN bit, it is determined whether to continue the DMA transfer.
When REN is set to 0, the EN and TACT bits of the CHSTATn register are cleared to 0 and the DMAC stops operation.
When REN is set to 1, the DMA transaction continues to be executed. Also, the REN bit is automatically cleared to 0 immediately after that.

Remark: n = 0 to 3; p = 0, 1

(2) Register Settings

(a) Register set selection (CHCFGn.DMS)

By using the RSEL bit (bit 28) of the channel configuration register (CHCFGn), select the register set to be executed.

Table 18.12 Register Mode Setting

CHCFGn.DMS	CHCFGn.RSEL	Operation
0 (register mode selection)	0	Uses the Next 0 register set.
	1	Uses the Next 1 register set.

Remark: n = 0 to 3

(b) INTDMAn operation selection (CHCFGn.DEM)

By using the DEM bit (bit 24) of the channel configuration register (CHCFGn), select the operation of INTDMAn when the DMA transaction (the series of DMA transfers) is completed in register mode.

Table 18.13 INTDMAn Operation Selection

CHCFGn.DEM	Operation	
0	Enables INTDMAn (INTDMAn is not masked.)	Outputs INTDMAn when the DMA transaction (the series of DMA transfers) is completed.
1	Disables INTDMAn (INTDMAn is masked.)	Does not output INTDMAn when the DMA transaction (the series of DMA transfers) is completed. After that, the DEM bit is automatically cleared to 0 and INTDMAn output is enabled again.

Remark: n = 0 to 3

(c) Terminal count output (DMATCZp) mask setting (CHCFGn.TCM)

By using the TCM bit (bit 25) of the channel configuration register (CHCFGn), set whether to mask the terminal count DMATCZp output when the DMA transaction (the series of DMA transfers) is completed in register mode.

Table 18.14 Terminal Count Output (DMATCZp) Mask Setting

CHCFGn.TCM	Operation	
0	Enables terminal count output (DMATCZp) (DMATCZp is not masked.)	Outputs DMATCZp when the DMA transaction (the series of DMA transfers) is completed.
1	Disables terminal count output (DMATCZp) (DMATCZp is masked.)	Does not output DMATCZp when the DMA transaction (the series of DMA transfers) is completed. After that, the TCMn bit is automatically cleared to 0 and DMATCZp output is enabled again.

Remark: n = 0 to 3; p = 0, 1

(d) Continuous execution setting (CHCFGn.REN)

By using the REN bit (bit 30) of the channel configuration register (CHCFGn), select whether to proceed to the next DMA transfer following the completion of the DMA transaction (the series of DMA transfers).

To proceed to the next transfer, use the Next register set selected by the RSEL bit of the channel configuration register (CHCFGn).

Table 18.15 Continuous Execution Set

CHCFGn.REN	Operation	Remark
0	Clears the EN bit to 0 and ends the DMA operation when a DMA transaction (the series of DMA transfers) using the register set selected by RSEL is completed.	Set this value when executing the DMA transaction (the series of DMA transfers) once.
1	Executes DMA transfer to transfer the content of the selected register set after the DMA transaction (the series of DMA transfers) is completed. After that, REN is automatically cleared to 0.	Set this value when executing the DMA transaction (the series of DMA transfers) consecutively according to the content of the register set.

Remark: n = 0 to 3

(e) Automatic register set switch setting (CHCFGn.RSW)

By using the RSW bit (bit 29) of the channel configuration register (CHCFGn), select whether to invert the value of the RSEL (Next 0/Next 1 register set selection) bit when the DMA transaction (the series of DMA transfers) is completed.

Table 18.16 Automatic Register Set Switch Setting

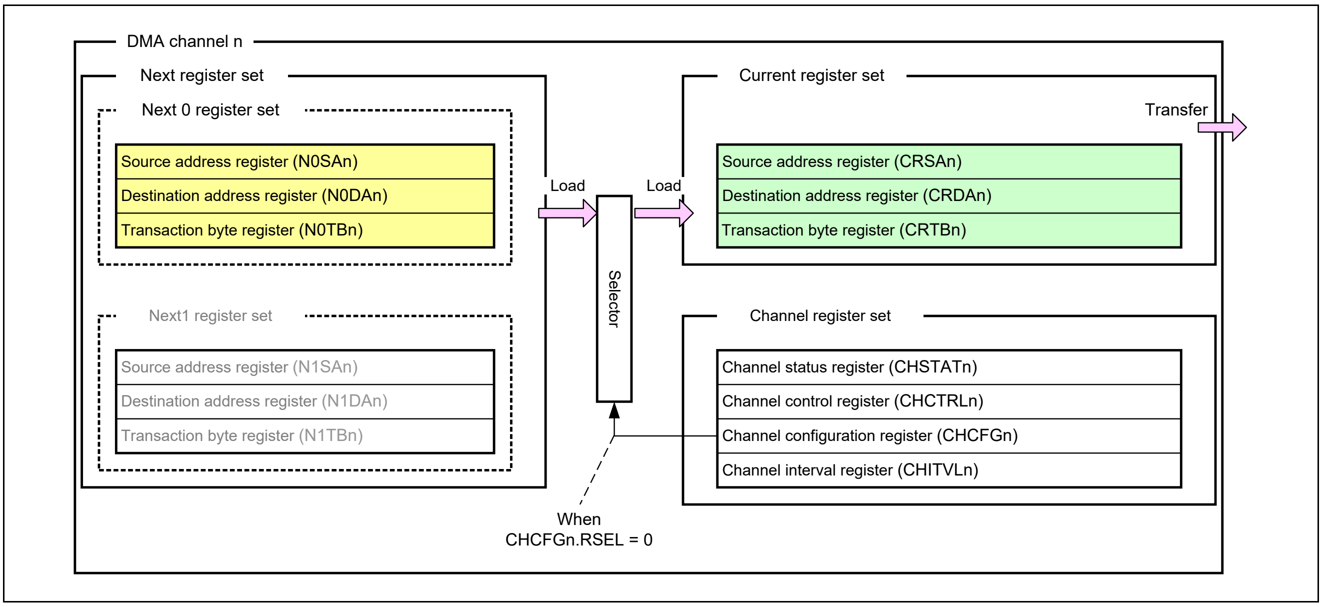
CHCFGn.RSW	Operation	Remark
0	Does not switch the register (invert the RSEL bit) when the DMA transaction (the series of DMA transfers) is completed while REN is set to 1 (continuous execution enabled).	Select this value when using only one register set.
1	Switches the register (inverts the RSEL bit) when the DMA transaction (the series of DMA transfers) is completed while REN is set to 1 (continuous execution enabled), and selects the other register set for continuous execution.	Select this value when switching register sets for continuous execution.

Remark: n = 0 to 3

(3) Register Setting Examples

(a) When only the Next 0 register set is used

CHCFGn.DMS	CHCFGn.RSEL	CHCFGn.DEM	CHCFGn.TCM	CHCFGn.RSW	CHCFGn.REN
0	0	0	0	0	0
Register mode	Next 0 register set	INTDMAn not masked	DMATCZp not masked	No register switching	No continuous execution



<1> When CHCTRLn.SETEN is set to 1, both CHSTATn.EN and CHSTATn.TACT are set to 1 and the Next 0 register set is loaded to the Current register set.

<2> A DMA transaction (the series of DMA transfers) is executed according to the values of the Current register set and channel register set.

<3> Since CHCFGn.DEM is set to 0, INTDMAn is issued after the DMA transaction (the series of DMA transfers) is completed.

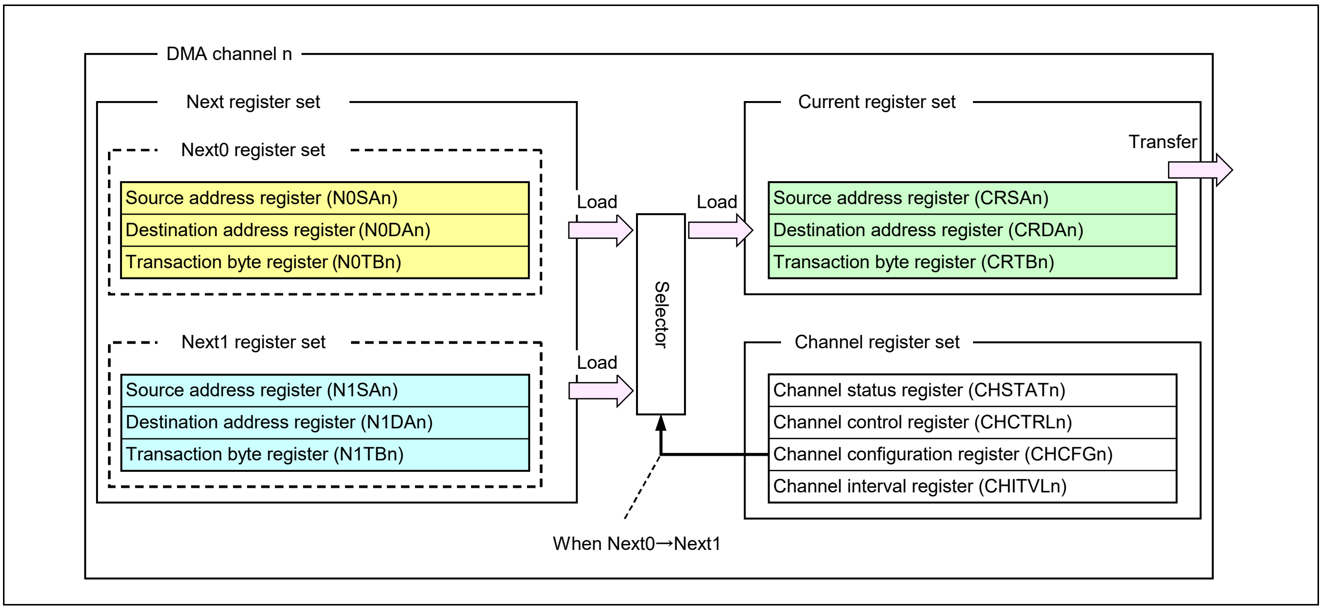
<4> Since CHCFGn.TCM is set to 0, DMATCZp is issued after the DMA transaction (the series of DMA transfers) is completed.

<5> Since CHCFGn.REN is set to 0, the EN and TACT bits are cleared to 0 and the processing ends.

Remark: n = 0 to 3; p = 0, 1

(b) When two register sets are used for continuous execution

CHCFGn.DMS	CHCFGn.RSEL	CHCFGn.DEM	CHCFGn.TCM	CHCFGn.RSW	CHCFGn.REN
0	0	1	0	1	1
Register mode	Next 0 register set	INTDMAn masked	DMATCZp not masked	Register switching selected	Continuous execution selected



<1> When CHCTRLn.SETEN is set to 1, both CHSTATn.EN and CHSTATn.TACT are set to 1 and the Next 0 register set is loaded to the Current register set.

<2> A DMA transaction (the series of DMA transfers) is executed according to the values of the Current register set and channel register set.

<3> Since CHCFGn.DEM is set to 1, INTDMAn is not issued after the DMA transaction (the series of DMA transfers) is completed. Also, the DEM bit is automatically cleared to 0. This means that, when the continuously executed DMA transaction is completed, INTDMAn is issued.

<4> Since CHCFGn.REN is set to 1, execution is continued. Also, the REN bit is automatically cleared to 0.

<5> Since CHCFGn.RSW is set to 1, the register set to be executed next is switched (RSEL = 0→1).

<6> The Next 1 register set is loaded to the Current register set.

<7> The DMA transaction (the series of DMA transfers) is executed according to the values of the Current register set and channel register set.

<8> Since CHCFGn.DEM is set to 0, INTDMAn is issued after the DMA transaction (the series of DMA transfers) is completed.

<9> Since CHCFGn.TCM is set to 0, DMATCZp is issued after the DMA transaction (the series of DMA transfers) is completed.

<10> Since CHCFGn.REN is set to 0, the EN and TACT bits are cleared to 0 and the processing ends.

Remark: n = 0 to 3; p = 0, 1

18.7.3 Link Mode

In link mode, the “descriptor” stored in memory is loaded as the set value to execute a DMA transaction (the series of DMA transfers).

In the DMAC, there is a pair of a Next link address register and a Current link address register for each channel.

The Next link address register is used to set the address of the descriptor to be executed next. The Current link address register is used to indicate the address of the descriptor for the currently executed DMA transaction (the series of DMA transfers).

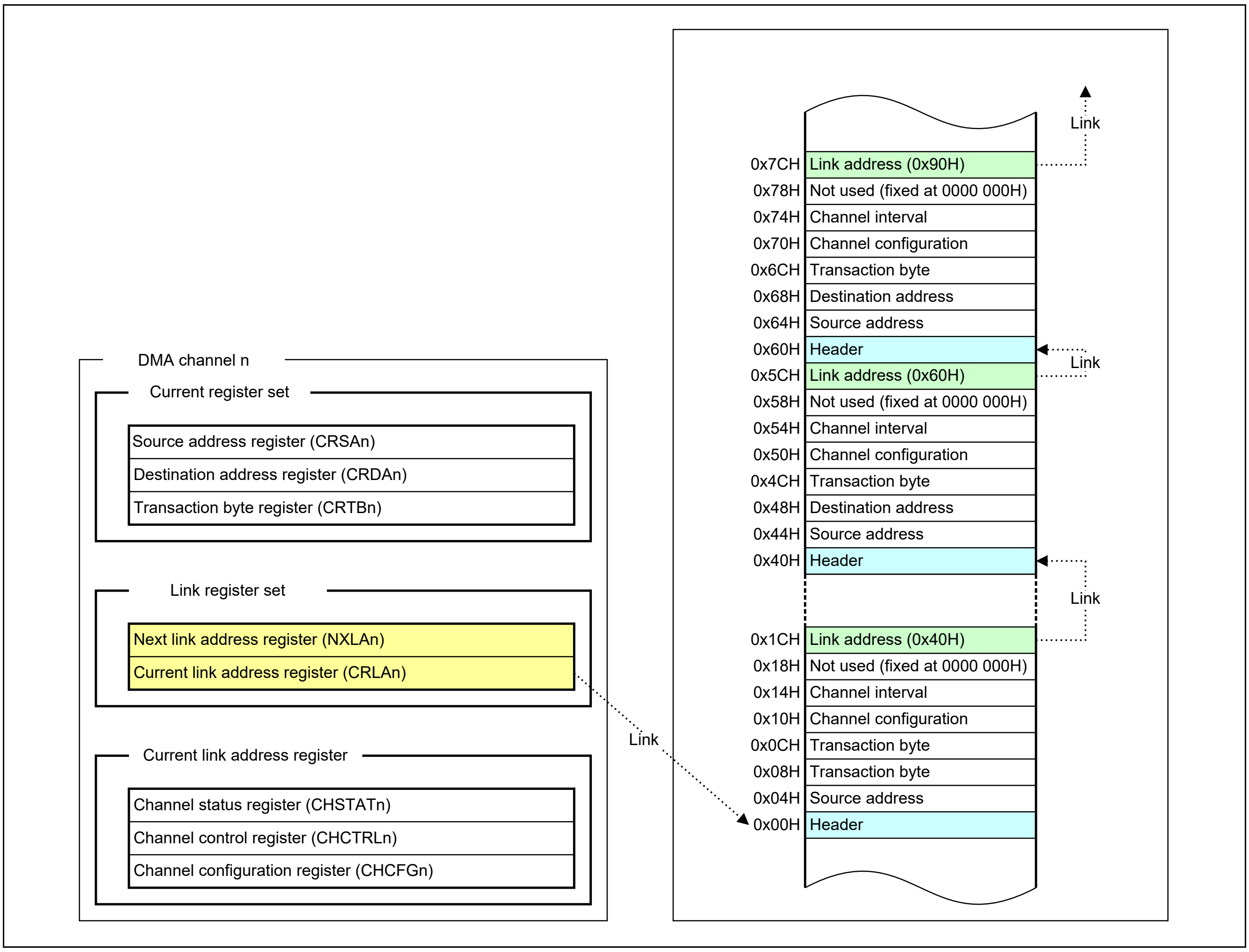
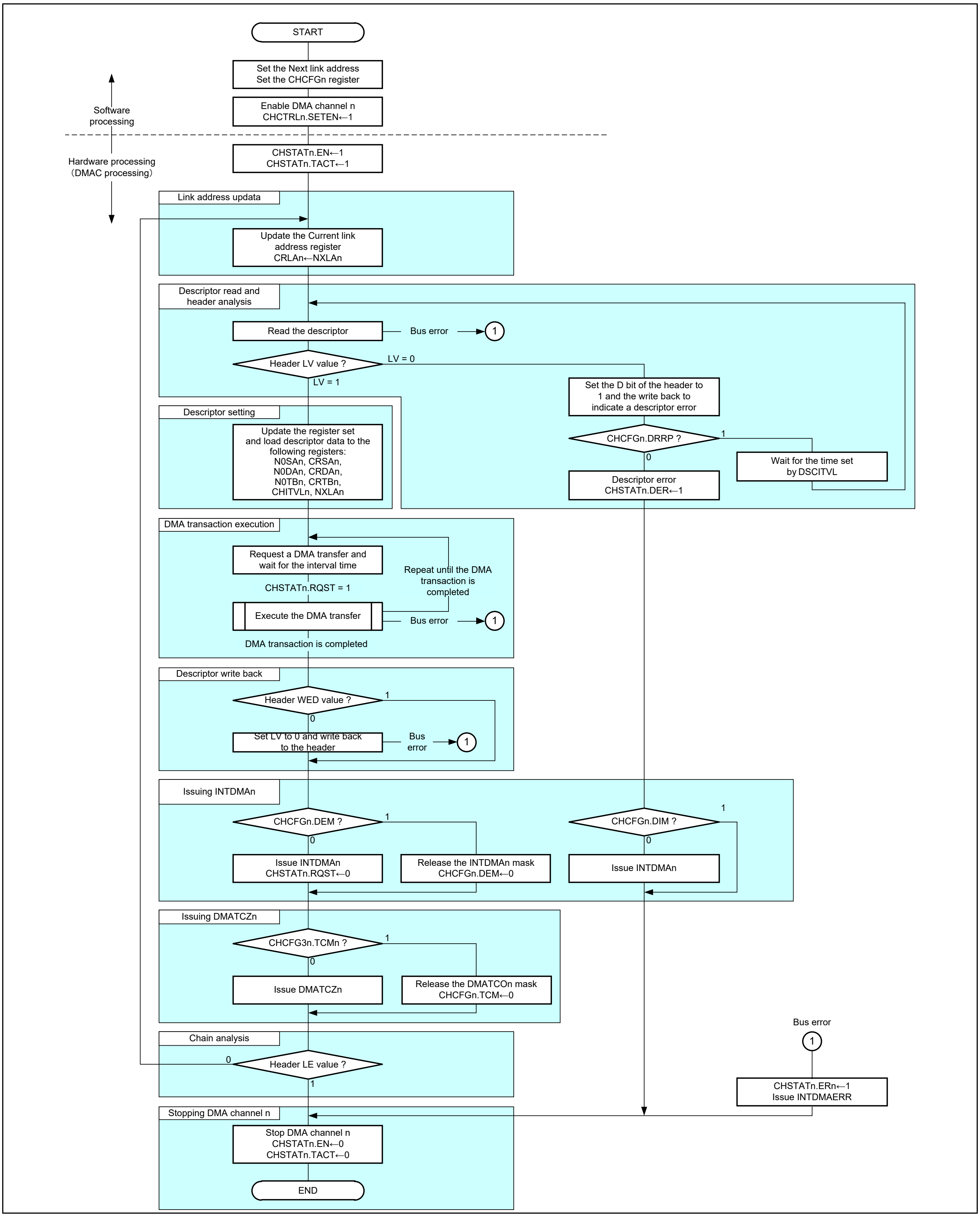


Figure 18.15 Outline of Link Mode

(1) Link Mode Operation Flow



<1> Channel configuration
Set the start link address in NXLAn.

<2> Link address update
When the CHCTRLn.SETEN bit is set to 1, both CHSTATn.EN and CHSTATn.TACT are set to 1 and the link address set in the NXLAn register is loaded to CRLAn.

<3> Descriptor read and header analysis
The loading of the descriptor starts, and the DMAC checks the content of the “header”.
When LV is set to 0, the D bit of the header is set to 1 and written back.
When CHCFGn.DRRP is set to 1, the same descriptor is read again after the lapse of the time set in the DSCITVL register.
When CHCFGn.DRRP is set to 0, CHSTATn.DER is set to 1, resulting in a completion status (EN = 0 and TACT = 0). In this case, when CHCFGn.DIM is set to 0, INTDMAn is issued.

<4> Descriptor setting
The loaded descriptor is set in both the Current register set and channel register set. Also, the next link address is set in NXLAn.

<5> DMA transaction execution
A DMA transaction is executed according to the set values.
If a DMA transfer error occurs during this process, INTDMAERRn is issued and the DMA transfer ends.

<6> Header writeback
When the WBD bit of the header is set to 0, the DMAC clears the LV bit of the header to 0 and writes back to the header.

<7> Issuing INTDMAn
The INTDMAn is masked according to the value set by the CHCFGn.DEM bit. INTDMAn is not issued if DEM = 1.

<8> Issuing DMATCZp
DMATCZp output is masked according to the value set by the CHCFGn.TCM bit.
When TCM is set to 1, DMATCZp is not output.

<9> Link end judgment
When the LE bit of the header is set to 1, EN and TACT are cleared to 0 after the DMA transaction set with the descriptor and the processing ends. When LE is set to 0, the Current register is updated and the next descriptor starts to be loaded.

Remark: n = 0 to 3; p = 0, 1

(2) Register Settings

(a) Link mode selection (CHCFGn.DMS)

By using the DMS bit (bit 31) of the channel configuration register (CHCFGn), select link mode.

The DMS bit cannot be rewritten by using a descriptor.

Table 18.17 Link Mode Selection

CHCFGn.DMS	Operation
1 (link mode selection)	Operates in link mode.

(b) Link address setting (NXLAn)

There are two registers that indicate the link address: Next link address register (NXLAn) and Current link address register (CRLAn).

To start link mode, set the link address in the Next link address register (NXLAn).

After a descriptor is loaded, the Next link address register (NXLAn) indicates the link address described below.

The Current link address register (CRLAn) indicates the currently executed link address.

Table 18.18 Link Address Register Set

Register	Operation
Next link address register (NXLAn)	Indicates the next link address. Before starting link mode, set the link address in this register.
Current link address register (CRLAn)	Indicates the currently executed link address. This register is read only.

Caution: In link mode, the settings can be changed by reading a descriptor. However, the timing of the setting change cannot be synchronized with a hardware-initiated DMA transfer request (DMAREQZp or interrupt signal). Therefore, when using a hardware-triggered DMA transfer request, set the AM2–AM0, LVL, HIEN, LOEN, and SEL2–SEL0 bits of the CHCFGn register before setting the EN bit to 1, and do not change these bits with the descriptor.

Remark: n = 0 to 3; p = 0, 1

(c) Descriptor settings

The DMAC supports two descriptor formats.

To switch the format, use the DSCFM field of the 31 to 28 bits of the first word (header) of the descriptor.

The following table shows the relationship between the DSCFM value and the descriptor format.

Table 18.19 Descriptor Format

DSCFM Field Value	0001B	0011B
Descriptor size	8 words	4 words
Link address	✓	✓
Channel interval	✓	— (reload)
Channel configuration	✓	— (reload)
Transaction size	✓	— (Header)
Destination address	✓	✓
Destination address	✓	✓
Header	✓	✓ (STS)

Cautions 1. Do not set any value other than the above in the DSCFM field.

2. The setting of the DMS bit of the channel configuration register (CHCFGn) cannot be changed by using the descriptor (fixed at link mode).

3. The settings of the REN and RSW bits of the channel configuration register (CHCFGn) can be changed by using the descriptor. However, these bits are intended for use in register mode, such changes do not affect the operation.

4. The setting of the RSEL bit of the channel configuration register (CHCFGn) can be changed by using the descriptor. However, only the Next 0 register set is used in link mode.

Remark: n = 0 to 3

Table 18.20 Description of Each Field of the Descriptor

Field	Symbol	Description
Link address	✓	Specifies the address (link address) of the next descriptor to be read after the DMA transfer executed with the current descriptor.
Channel interval, Channel configuration	✓	Specifies the channel interval and channel configuration.
	— (reload)	Omits the specification of the channel interval and channel configuration and continues to use the last settings.
Transaction size	✓	Specifies the transaction byte size.
	— (Header)	Omits the specification of the transaction byte size and uses the value of the STS field of the header as the total number of transfer bytes. Since the STS field is 16 bits long, the maximum specifiable size is 65,536 bytes.
Destination address	✓	Specifies the destination address.
Source address	✓	Specifies the source address.
Header	✓ (noSTS)	The STS field of the 15 to 0 bits of the header is invalid. The transaction size of the descriptor is used as the total number of transfer bytes.
	✓ (STS)	The STS field of the 15 to 0 bits of the header is valid. The value set in the STS field is used as the total number of transfer bytes.

(d) Header settings

The header indicates the state of the descriptor, etc.

The header is read before DMA transfer starts in link mode.

After the DMA transaction (the series of DMA transfers) is completed, the values are written back to the header.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
DSCFM				0	WBD	LE	LV	D	0	0	0	0	0	0	0	+00H

When WBD is set to 0, only the LV bit is cleared to 0 and written back, after the DMA transaction (the series of DMA transfers) using this descriptor is completed.

If LV is set to 0 when the header is read, a descriptor error is recognized and only the D bit is set to 1 and written back.

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

Bit Position	Bit Name	Description
31 to 28	DSCFM	(Descriptor Format) Specifies one of the two descriptor formats shown in Table 18.19.
27	Reserved	Set this bit 0.
26	WBD	(Write Back Disable) Sets the writeback operation of the LV bit. 0: Writes the LV bit back to 0 after the DMA transaction (the series of DMA transfers) is completed. 1: Does not write the LV bit back to 0 after the DMA transaction (the series of DMA transfers) is completed.
25	LE	(Link End) Indicates the link continuation status of the DMA transaction (the series of DMA transfers) of this descriptor. Set this bit to 1 at the end of the link. 0: The link continues. 1: The link ends.
24	LV	(Link Valid) Indicates whether the descriptor is valid or invalid. When WBD is set to 0, the LV bit is cleared to 0 and written back, after a DMA transaction (the series of DMA transfers) using the descriptor is completed. When setting the header, set 1 in this bit. 0: The descriptor is invalid. 1: The descriptor is valid.
23	D	(Descriptor Error) This is the descriptor error bit. When LV is set to 0 (descriptor is invalid) when the header is loaded, the DMAC sets this bit to 1 and writes it back. 0: No error. 1: Descriptor error
22 to 16	Reserved	Set these bits 0.
15 to 0	STS	(Short Transaction Size) When 0011b is set in the DSCFM field, set the total number of bytes of the DMA transfer in this field. The maximum specifiable value is 65,536 bytes. In this case, 0 cannot be set in STS.

Caution: When adding descriptors sequentially during DMAC operation, make byte access to write 1

to the LV bit. The DMAC writes back the D bit through byte access. Therefore, this prevents a contention between setting the LV bit to 1 by software and writing back the D bit by the DMAC.

(e) Descriptor settings other than the header

The descriptor data other than the header has the same specifications as the registers in the DMAC.

Table 18.21 shows their correspondence. For information about the specifications of the registers in the DMAC, see section 18.4, DMA Controller Registers.

Table 18.21 Correspondence between the Descriptors Other Than the Header and the DMAC Internal Registers

Descriptor Offset Address	Descriptor	DMAC Internal Register
+ 04H	Descriptor offset address	Source address register (CRSAn)
+ 08H	Destination address	Destination address register (CRDAn)
+ 0CH	Destination address	Transaction byte register (CRTBn)
+ 10H	Channel configuration	Channel configuration register (CHCFGn)
+ 14H	Channel configuration	Channel interval register (CHITVLn)
+ 18H	Be sure to set 0000 0000H.	—

Caution: The DMS bit of the CHCFGn register cannot be rewritten by using a descriptor.

Remark: n = 0 to 3

(3) Outline of the Descriptor Area and DMA Transfer Area

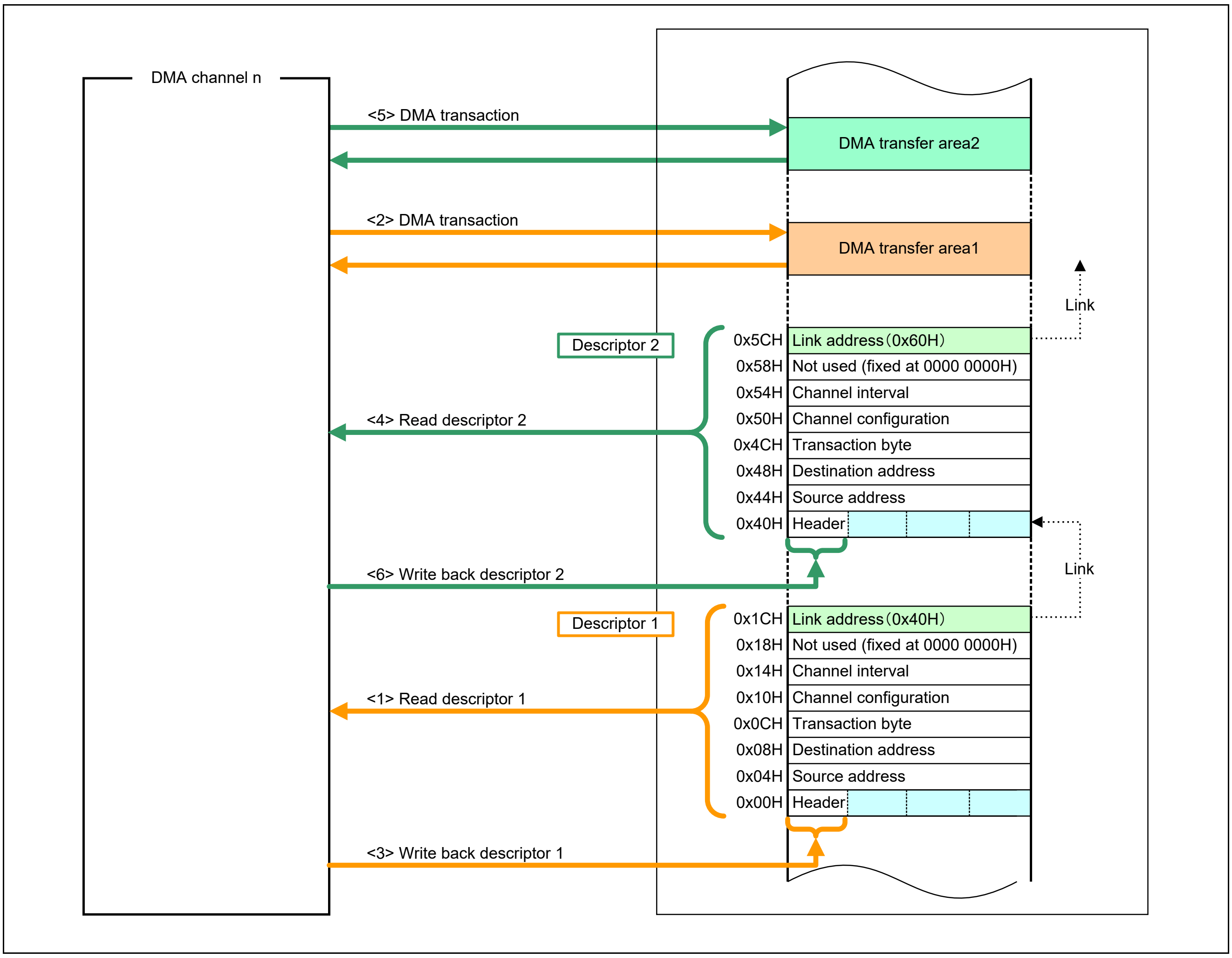


Figure 18.16 Outline of the Descriptor Area and DMA Transfer Area

<1> Descriptor read

The value set in the Next link address register (NXLAn) in the DMAC is loaded to the Current link address register (CRLAn), and a descriptor is read from “descriptor 1” in the memory space indicated by the CRLAn register.

<2> DMA transfer (DMA transaction)

When LV of the descriptor header is set to 1, a DMA transfer is executed according to the descriptor information.

<3> Descriptor writeback

After a DMA transaction of the set number of bytes is completed, when WBD of the header is set to 0, LV is cleared to 0 and written back to bits 31 to 24 of the descriptor 1 header. For the other fields, the values read in <1> are written back through byte write.

<4> Descriptor read

When LE of the descriptor header read in <1> is set to 0, the next descriptor is read from the address (descriptor 2) indicated by the Next link address in the descriptor.

<5> DMA transfer (DMA transaction)

When LV of the descriptor header is set to 1, a DMA transfer is executed according to the descriptor information.

<6> Descriptor writeback

After a DMA transaction of the set number of bytes is completed, when WBD of the header is set to 0, LV is cleared to 0 and written back to bits 31 to 24 of the descriptor 2 header. For the other fields, the values read in <4> are written back as write data through byte access.

Repeat steps <4> to <6>.

Remarks 1. When LE is set to 1 and WBD is set to 0 in the header, a DMA transaction is executed by using the settings of the descriptor and LV is cleared to 0 and written back to end the transaction.

2. When both LE and WBE is set to 1 in the header, a DMA transaction is executed by using the settings of the descriptor and the transaction ends. No data is written back.

3. When LV is set to 0 in the header, 1 is written back to the D bit of the header. After that, when CHCFGn.DRRP is set to 1, a descriptor is read again after the interval set in the DITVL field of the DSCITVL register. When CHCFGn.DRRP is set to 0, the DMA controller is stopped.

4. n = 0 to 3

(4) Notes on the Descriptor

- In link mode, the settings can be changed by reading a descriptor. However, the timing of the setting change cannot be synchronized with a hardware transfer request. Therefore, when using a hardware transfer request, set the AM2–AM0, LVL, HEN, LEN, and SEL2–SEL0 bits of the CHCFGn register before setting the CHCTRLn.SETEN bit to 1, and be careful not to change these bits with the descriptor.
- The DMS bit of the CHCFGn register cannot be changed by using a descriptor (fixed at link mode). Also, while the REN, RSW, and RSEL bits of the CHCFGn register can be changed by using a descriptor, such changes do not affect the operation.
- The DMAC references the DSCFM and LV bits of the header of a descriptor to determine whether that descriptor is valid or invalid. Therefore, initialize the memory area corresponding to the DSCFM and LV bits of the descriptor (DSCFM = 0001b or 0011b and LV = 0) before the DMAC accesses it.
- When the next descriptor is set in memory during DMA operation, write 1 in the LV bit after setting the descriptors following the header (source address, destination address, next link address, etc.). This is intended to prevent DMA from being executed by using the previously set descriptor values if a conflict occurs between descriptor setting by software and descriptor read by the DMAC, in which case the descriptor read attempt by the DMAC may interrupt the descriptor setting by the CPU.
- To leave the write-back information of the D bit of the header, write 1 in the LV bit of the header through byte access.

Remark: n = 0 to 3

(5) Link Configuration Examples

In link mode, descriptors can have a "list configuration" or "loop configuration", as described below.

(a) List configuration

Setting LE of the header of the last descriptor to 1 ends the link.

For list configuration, set the LE bit of the last descriptor to 1.

(b) Loop configuration

Setting the address of the first descriptor as the link address for the last descriptor configures the descriptors in a loop. To end the loop, either overwrite the LE bit of the header with 1 before the DMAC reads the descriptor or stop the DMAC according to the transfer suspension procedure.

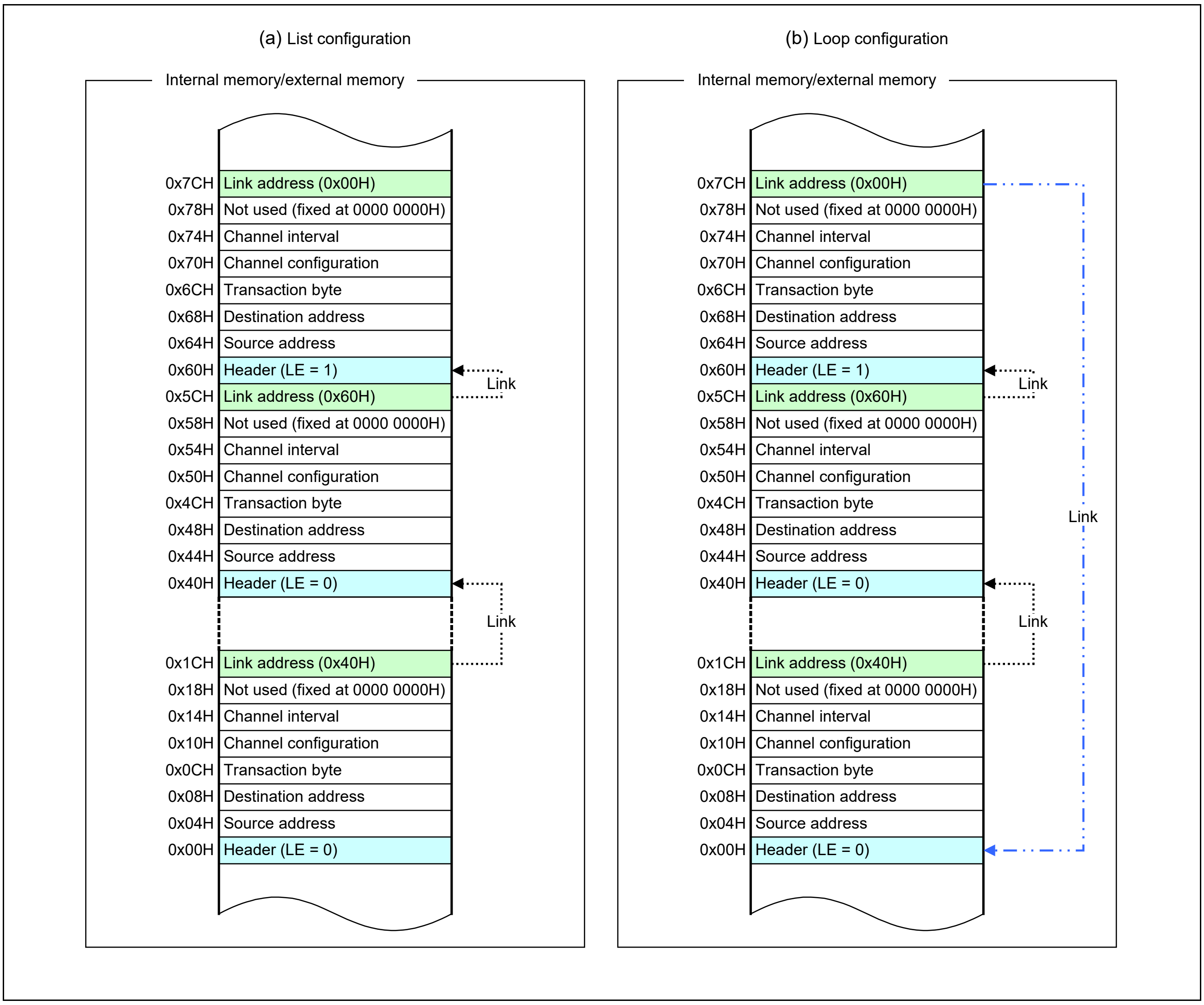


Figure 18.17 Link Mode Configuration Example

18.7.4 Write-Only Mode

When the WONLY bit of the channel configuration register (CHCFGn) is set to 1, write-only mode is entered.

In write-only mode, DMA read transfers are not executed. However, descriptors are read in link mode. The values set in the NxSAn register (x = 0 when CHCFGn.RSEL = 0; x = 1 when CHCFGn.RSEL = 1) are used as write data.

Use write-only mode for initialization of the memory area, etc.

Table 18.22 Setting for Write-Only Mode

CHCFGn.WONLY	Mode	Operation
0	Normal mode	Executes DMA transfer by using the values set in the Next register set.
1	Write-only mode	Does not execute DMA read transfers; only DMA write transfers are executed.

Remark: n = 0 to 3

18.8 DMAC Operation

Caution: This section explains only operation of the general-purpose DMAC since the specifications of operations of the general-purpose DMAC and the DMAC for real-time ports are the same.

18.8.1 Transfer Mode

The DMAC supports single transfer mode and block transfer mode.

Select one of these modes for each channel by using the TM bit of the channel configuration register (CHCFGn).

Table 18.23 DMA Transfer Mode Selection

CHCFGn.TMn	Mode	Operation
0	Single transfer mode	A single transfer proceeds in response to the request for a single DMA transfer.
1	Block transfer mode	Transfer proceeds until completion of the DMA transaction (the series of DMA transfers) in response to the request for a single DMA transfer.

Caution: When the interrupt request signal from internal peripheral modules is selected, detection of the DMA transfer request signal should be selected as follows.

DMA Transfer Request Source	DMA Transfer Request Signal Detection
Interrupt request signal from internal peripheral modules	Rising edge detection CHCFGn.LVL = 0 CHCFGn.LEN = 0 CHCFGn.HEN = 1
DMA transfer request input from external pins	As desired

Remark: n = 0 to 3

(1) Single Transfer Mode

When a DMA transfer request is acknowledged, a DMA transfer is executed once on the side (source or destination) indicated by the REQD bit of the channel configuration register (CHCFGn) and DMAACKZp is asserted at the timing specified by the AM2 to AM0 bits of the CHCFGn register.

A transfer is executed every time a transfer request is acknowledged. This operation is repeated as many times as the number of bytes loaded to the Current transaction byte register (CRTBn) (inter-channel arbitration is performed for each DMA transfer).

The DMAACKZp output timing and the CRTBn register count timing differ depending on the settings of the REQD bit of the CHCFGn register and the transfer size (DDS or SDS). For details, see section 18.8.10, Differences in Operation by Transfer Size.

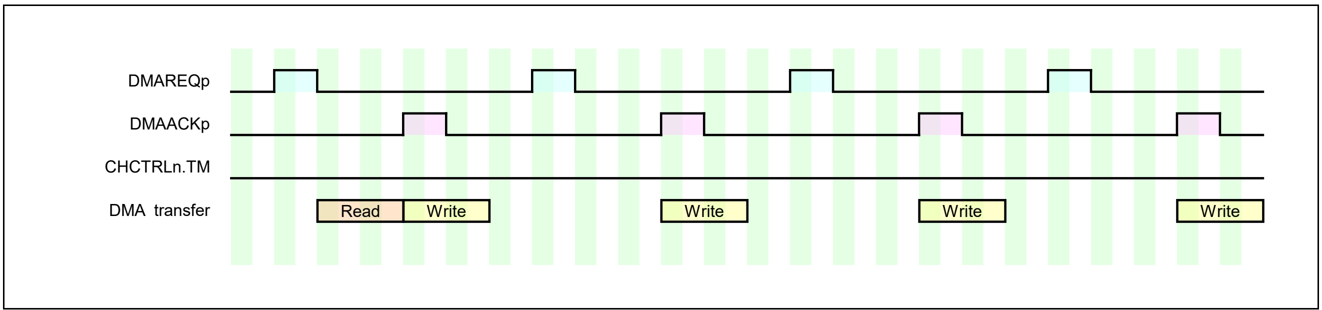


Figure 18.18 Single Transfer Mode Example

DMA transfer request: Rising edge detection, request from the destination.
DMA acknowledge output: Pulse mode.
SDS[3:0]>DDS[3:0] (In this example, the transfer size of the source is four times that of the destination.)

Remarks 1. The DMA interface signals (DMAREQZp, DMAACKZp, and DMATCZp) of the external pins are negative logic.

2. n = 0 to 3; p = 0, 1

(2) Block Transfer Mode

Once a DMA transfer request is acknowledged, a DMA transfer is repeated until as many transfers as the number of bytes loaded to the Current transaction byte register (CRTBn) are completed (a DMA transaction is completed) (inter-channel arbitration is performed for each DMA transfer).

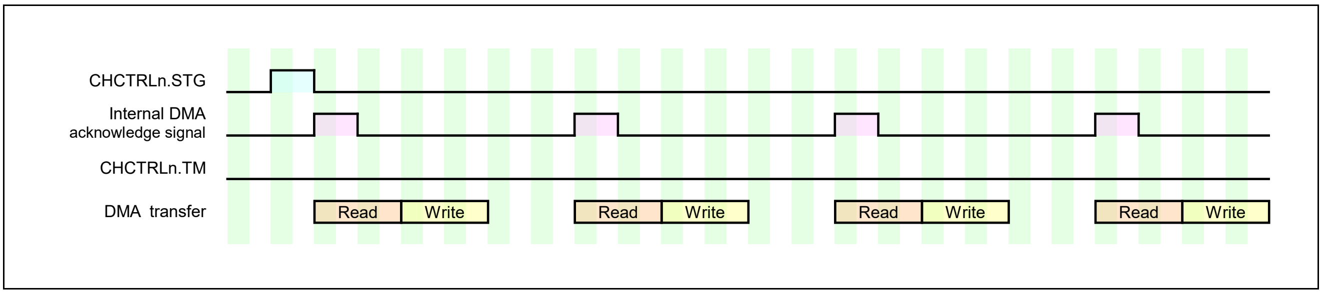


Figure 18.19 Block Transfer Mode Example

DMA transfer request: Software trigger DMA acknowledge output: Pulse mode SDS[3:0] = DDS[3:0] (In this example, the transfer size of the source is the same as that of the destination.)

Remark: n = 0 to 3

18.8.2 DMA Unit Priority Control

Since the general DMA controller and the DMA controller for real-time ports use the individual AHB layers, when the same slave is accessed, arbitration proceeds according to the priority decision system in Table 6.1, AHB Internal Buses of an R-IN32M4-CL3.

In addition, the priority between the channels of general DMA controller supports fixed priority mode and round-robin mode. Selection of the mode can be set by using the PR bit of the DMA control register (DCTRL0 register) in each DMAC. When the PR bit is 0, fixed priority mode is entered, and when the PR bit is 1, round-robin mode is entered.

Table 18.24 DMA Channel Priority Control Selection

DCTRL0.PR	Mode	Operation
0	Fixed priority	Controls in order of the fixed priority (high: CH0 > CH1 > CH2 > CH3: low). Use this mode when the channels have an order of priority.
1	Round robin	Controls in a round robin fashion. Use this mode when you wish equal handling of the execution of DMA transfer by all channels.

(1) Fixed Priority Mode

In fixed priority mode, the order of priority for the channels is fixed as follows.

High priority CH0 > CH1 > CH2 > CH3 Low priority

If a DMA transfer request is generated on multiple channels simultaneously, priority is given to the DMA transfer request of the channel having the smallest number. The following figure shows an example when a DMA transfer request is generated on a higher-priority channel while a DMA transfer is being executed in fixed priority mode.

Caution: DMA inter-channel priority control is also performed between the source read cycle and the destination write cycle.

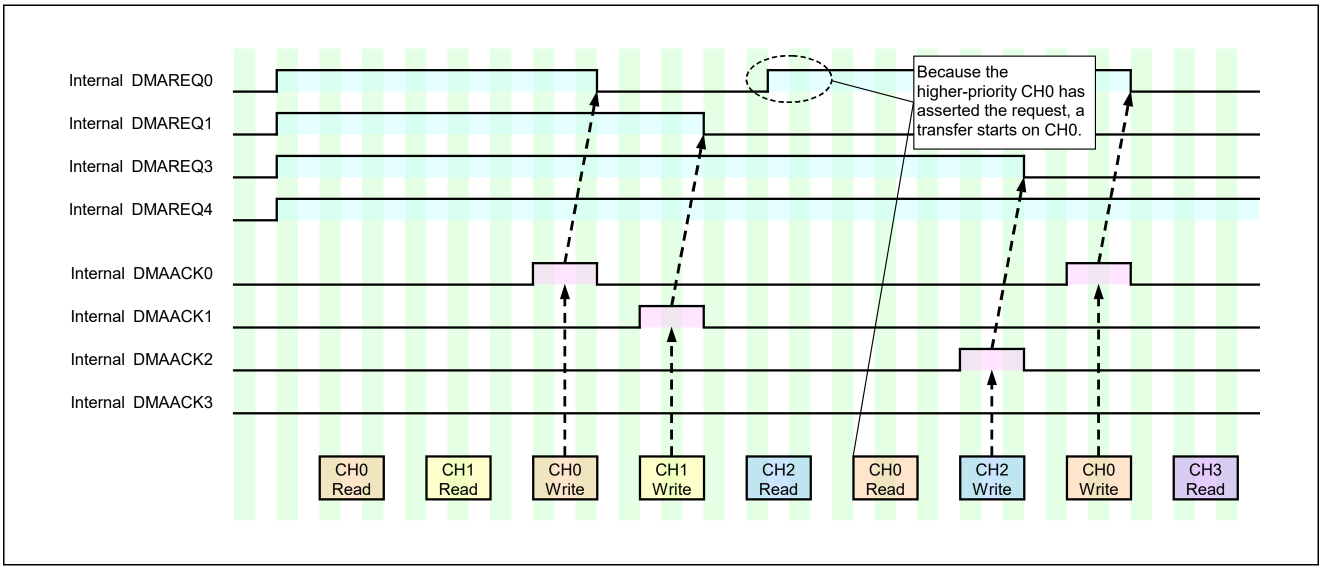


Figure 18.20 Fixed Priority Mode Example

DMA transfer request: High level detection, request from the destination. DMA acknowledge output: Level mode

Remark: As the internal DMA signals, the DMA transfer request of each channel is represented as "Internal DMAREQn" and the DMA acknowledge output is represented as "Internal DMAACKn" (n = 0 to 3).

(2) Round Robin Mode

In round robin mode, the order of priority is changed every time a DMA transfer request from a channel is acknowledged so that the lowest priority is given to the channel that last executed a transfer.

As in fixed priority mode, the order of priority immediately after deasserting the reset signal is as follows.

High priority CH0 > CH1 > CH2 > CH3 Low priority

In this state, if there is no transfer request for DMA channel 0 while there is a transfer request for DMA channel 2, a transfer is executed on DMA channel 2. After the transfer is completed, the order of priority is as follows.

High priority CH3 > CH0 > CH1 > CH2 Low priority

The following example shows how DMA transfers are executed in round robin mode.

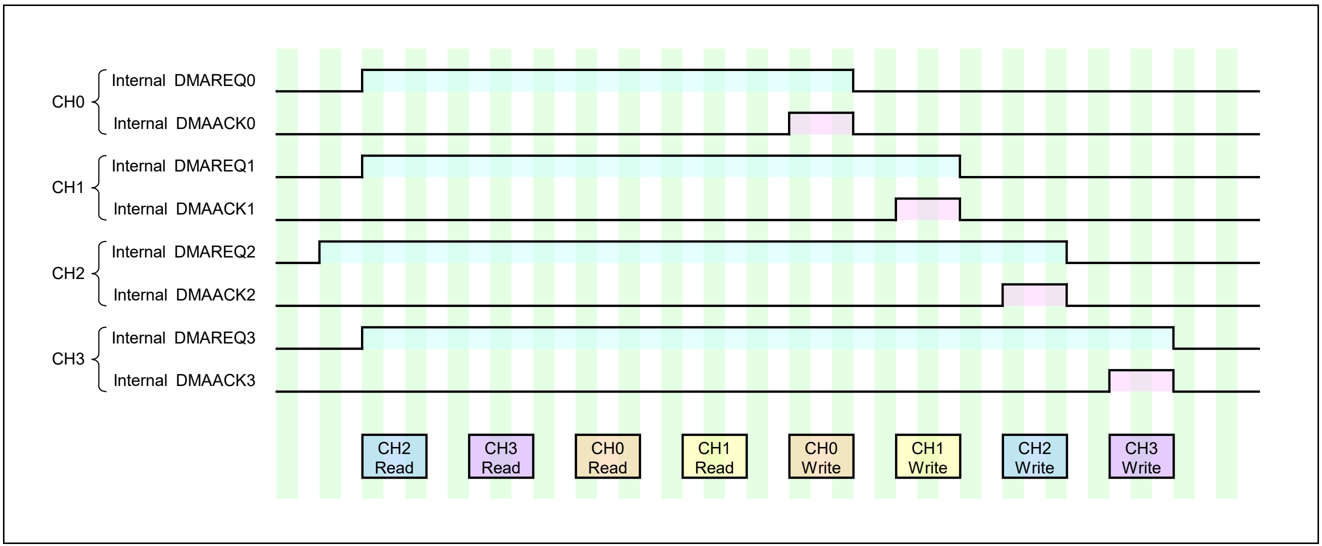


Figure 18.21 Round Robin Mode

DMA transfer request: High level detection, request from the source DMA acknowledge output: Level mode

Arbitration proceeds between read channels and between write channels, respectively.

Remark: As the internal DMA signals, the DMA transfer request of each channel is represented as "Internal DMAREQn" and the DMA acknowledge output is represented as "Internal DMAACKn" (n = 0 to 3).

18.8.3 DMA Transfer Request

DMA transfer requests are fixed for each DMA unit. Which channel to use to select the DMA transfer requests for each unit can be selected for each channel by using the SEL2 to SEL0 bits of the channel configuration register (CHCFGn).

Remark: n = 0 to 3

(1) Specification of Detection for Each DMA Transfer Request Source

For some DMA transfer requests, the method of detection is specified for each source.

Based on the following table, specify the proper detection method for each DMA transfer request source by using the LVL, LEN, and HEN bits of the channel configuration register (CHCFGn).

Table 18.25 Specification of the Detection for Each DMA Transfer Request Source

DMA Transfer Request Source	DMA Transfer Request Detection Specification (CHCFGn.LVL, LEN, HEN)	DMA Acknowledge Signal Specification (CHCFGn.AM2–AM0)
Interrupt request from external pins (INTPZ0–INTPZ29)	Rising edge detection	The DMAACKZ0–1 and RTDMAACKZ pins cannot be used.
Interrupt request from internal peripheral modules	Rising edge detection	The DMAACKZ0–1 and RTDMAACKZ pins cannot be used.
DMA transfer request from external pins (DMAREQZ0, DMAREQZ1, RTDMAREQZ)	To be set arbitrarily according to the specification of the DMA transfer request source.	To be set arbitrarily according to the specification of the DMA transfer request source.

Table 18.26 DMA Transfer Request Signal Detection Method

LVL	HEN	LEN	DMA Transfer Request Signal Detection Method	
0	0	0	Edge detection	Detection disabled
0	0	1		Falling edge detection
0	1	0		Rising edge detection
0	1	1		Rising/falling edge detection
1	0	0	Level detection	Detection disabled
1	0	1		Low level detection
1	1	0		High level detection
1	1	1		A DMA transfer is started when the SETENn bit of the CHCTRLn register is set to 1, regardless of the input level of the DMA transfer request.

Remark: n = 0 to 3

(2) Edge Detection

Edge detection is selected when the LVL bit of the CHCFGn register is set to 0.

When the HEN bit of the CHCFGn register is set to 1, rising edge detection is performed. When the LEN bit is set to 1, falling edge detection is performed.

When the DMAREQZ0–DMAREQZ1 signal is used as a DMA transfer request, make sure that the next DMA transfer request (DMAREQZ0–DMAREQZ1) is issued after the DMA acknowledge signal (DMAACKZ0–DMAACKZ1) is detected.

When an interrupt signal is used as a DMA transfer request, it is not recognized as a DMA transfer request if the next interrupt signal is generated before the DMA transfer is completed. Take care on the interval of the interrupt signal.

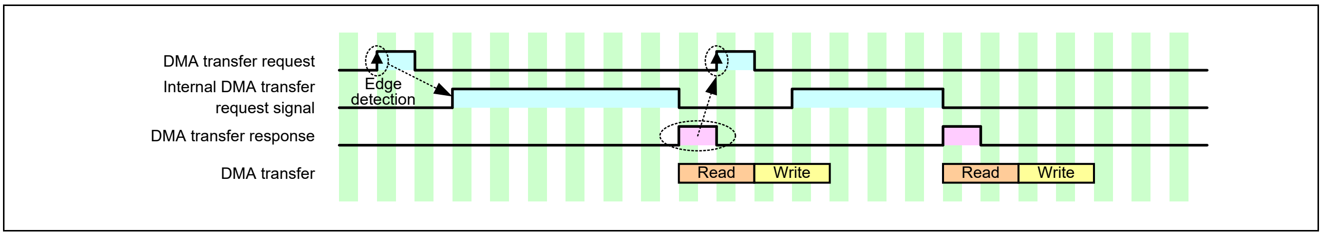


Figure 18.22 Edge Detection Mode Operation Example 1

DMA transfer request: Rising edge detection Request from the source (CHCFGn.REQD = 0) DMA acknowledge output: Pulse mode

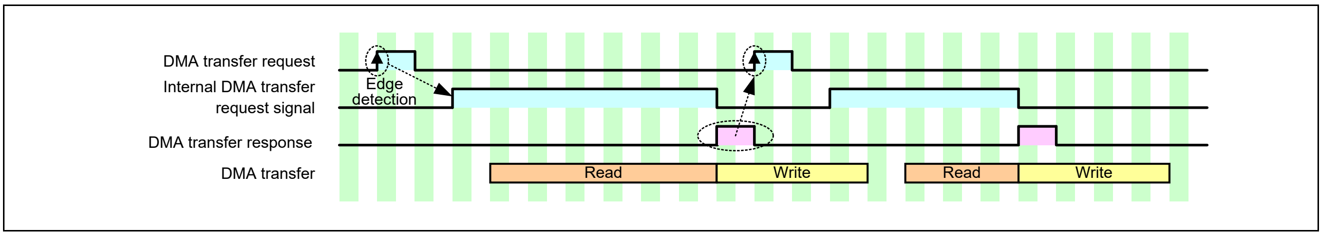


Figure 18.23 Edge Detection Mode Operation Example 2

DMA transfer request: Rising edge detection Request from the destination (CHCFGn.REQD = 1) DMA acknowledge output: Pulse mode

(3) Level Detection

When the LVL bit of the CHCFGn register is set to 1, level detection is selected.

When the DMAREQZp signal is used as a DMA transfer request, it is recognized as a DMA transfer request if a valid level of a width of BUSCLK × 2 is input (specified by HEN and LEN of the CHCFGn register).

When the level mode is selected for the DMA acknowledge signal, DMAACKZp remains at the high level until DMAREQZp is deasserted. When the pulse mode is selected, DMAACKZp is output in response to a pulse of 1 × BUSCLK.

When the DMAREQZp signal is used as a DMA transfer request, make sure that the next DMA transfer request (DMAREQZp) is issued after the DMA acknowledge signal (DMAACKZp) is detected.

When an interrupt signal is used as a DMA transfer request, it is not recognized as a DMA transfer request if the next interrupt signal is generated before the DMA transfer is completed. Take care on the interval of the interrupt signal.

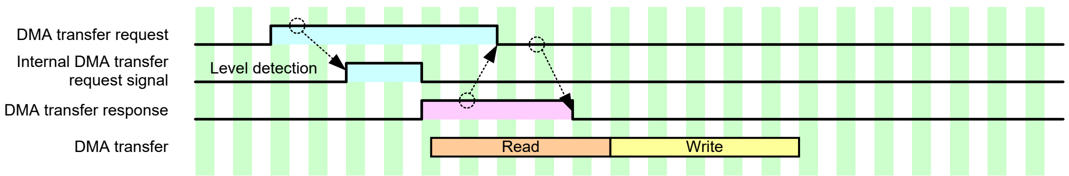


Figure 18.24 Level Detection Mode Operation Example 1

DMA transfer request: High level detection Request from the source (CHCFGn.REQD = 0) DMA acknowledge output: Level mode

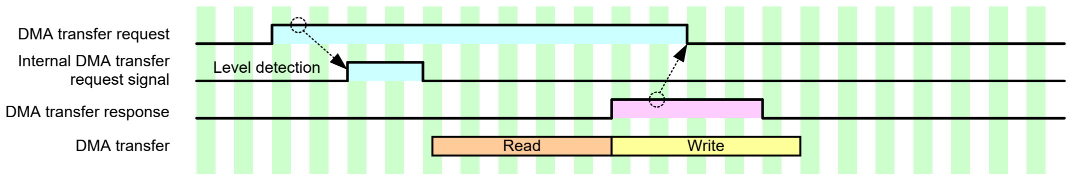


Figure 18.25 Level Detection Mode Operation Example 2

DMA transfer request: High level detection Request from the destination (CHCFGn.REQD = 1) DMA acknowledge output: Level mode.

Remark: n = 0 to 3; p = 0, 1

18.8.4 DMA Acknowledge Output

The DMA acknowledge signal is output as an acknowledge response signal for a DMA transfer request.

When the DMAREQZp signal is used as a DMA transfer request, use DMAACKZp as the DMA acknowledge signal. The signal is output from DMA unit. Set the output mode by using the AM2 to AM0 bits of the channel configuration register (CHCFGn).

When an external interrupt or an interrupt request from an internal peripheral module is used as a DMA transfer request, the DMA acknowledge signal is not used.

The DMA transfer requests assigned to the individual channels can be changed by using the SEL2 to SEL0 bits of the channel configuration register (CHCFGn). For information about the relationship between DMA transfer requests and DMA units, see Figure 18.1, Relation between DMA Units/Channels and DMA Trigger.

Remark: n = 0 to 3; p = 0, 1

(1) Specification of the Acknowledge Signal Mode for Each DMA Transfer Request Source

For some DMA acknowledge signals, the output mode is specified for each source.

Based on the following table, specify the proper detection method for each DMA transfer request source by using the AM2 to AM0 bits of the channel configuration register (CHCFGn).

Table 18.27 Specification of the Acknowledge Signal Mode for Each DMA Transfer Request Source

DMA Transfer Request Source	DMA Transfer Request Detection Mode Specification (CHCFGn.LVL, LEN, HEN)	DMA Acknowledge Signal Specification (CHCFGn.AM2–AM0)
Interrupt request from external pins (INTPZ0–INTPZ29)	Rising edge detection	The DMAACKZp and RTDMAACKZ pins cannot be used.
Interrupt request from internal peripheral modules	Rising edge detection	The DMAACKZp and RTDMAACKZ pins cannot be used.
DMA transfer request from external pins (DMAREQZp, RTDMAREQZ)	To be set arbitrarily according to the specification of the DMA transfer request source.	To be set arbitrarily according to the specification of the DMA transfer request source.

Table 18.28 DMA Acknowledge Signal (DMAACKZp) Output Mode

AMn2	AMn1	AMn0	DMA Acknowledge Signal (DMAACKZp) Output Mode
0	0	0	Pulse mode Note 1 (initial value)
0	0	1	Level mode The active level is maintained until the DMA transfer request (DMAREQZp) becomes inactive.
0	1	X	Bus cycle mode Note 2 The active level is maintained during the DMA transfer bus cycle.
1	X	X	The output of the DMA acknowledge signal (DMAACKZp) is disabled.

Notes 1. A pulse of 1 BUSCLK cycle is output as the DMAACKZp signal.

2. In bus cycle mode, the DMA acknowledge signal is output following the point at which acquisition of bus mastership is requested. For this reason, the DMA acknowledge signal is output earlier than the actual DMA bus cycle, and a bus cycle of an internal master which has previously acquired mastership of the same bus may proceed at this time.

Cautions 1. When the interrupt request signal of internal peripheral modules or external interrupt input is selected, the settings of AM2 to AM0 do not affect the operation.

2. The settings of AM2 to AM0 may duplicate those of the DMAIFCn register. In general, however, when the DMAACKZp signal is set to the level mode by using AM2 to AM0, the DMAIFCn register should be left at its initial value. Conversely, when the DMAIFCn register is used to extend the DMAACKZp pulse width or for the DMAREQZp mask function, set AM2 to AM0 to select the pulse mode.

Remark: n = 0 to 3; p = 0, 1

(2) Pulse Output

When the AM2 to AM0 bits of the channel configuration register (CHCFGn) are set to 000B, pulse output is selected for the DMA acknowledge signal (DMAACKZp).

A high-level pulse of 1 × BUSCLK is output.

If the pulse width is insufficient for the DMA transfer request source, the width of DMAACKZp can be set from 1 × BUSCLK to 32 × BUSCLK by using the AKWD4 to AKWD0 bits of the DMA transfer interface signal control registers 0 to 3 (DMAIFC0 to DMAIFC3).

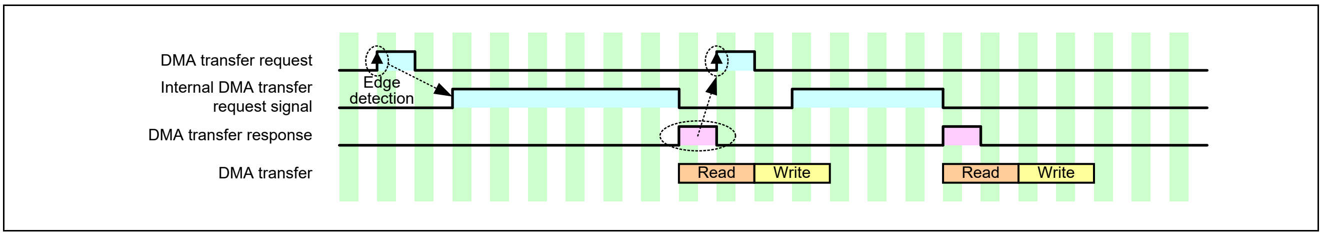


Figure 18.26 Pulse Output Mode Operation Example 1

DMA transfer request: Rising edge detection Request from the source (CHCFGn.REQD = 0) DMA acknowledge output: Pulse mode

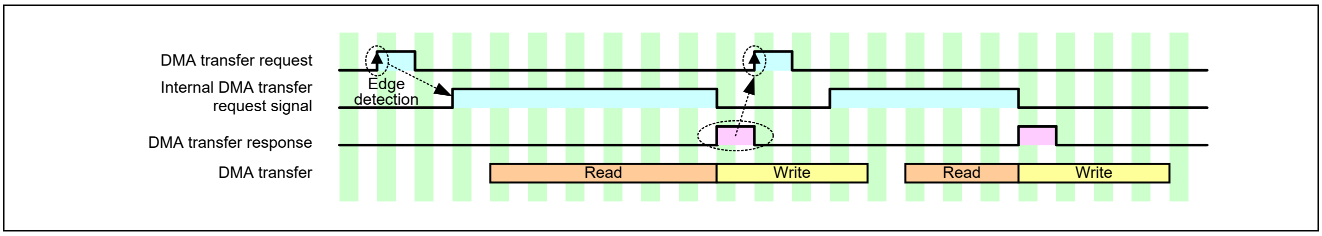


Figure 18.27 Pulse Output Mode Operation Example 2

DMA transfer request: Rising edge detection Request from the destination (CHCFGn.REQD = 1) DMA acknowledge output: Pulse mode

Remark: n = 0 to 3; p = 0, 1

(3) Level Output

When the AM2 to AM0 bits of the channel configuration register (CHCFGn) are set to 001B, level output is selected for the DMA acknowledge signal (DMAACKZp). The DMAACKZp signal continues to be asserted until the DMAREQZp signal is deasserted.

When level output is selected for the DMA acknowledge signal, the DMA transfer interface signal control registers 0 to 3 (DMAIFCp) should be left at their initial value and extension of the DMAACKZp width should not be used.

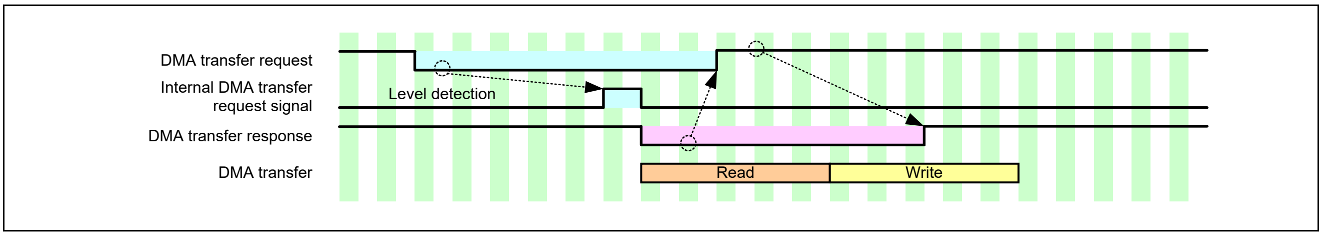


Figure 18.28 Level Output Mode Operation Example 1

DMA transfer request: High level detection Request from the source (CHCFGn.REQD = 0) DMA acknowledge output: Level mode

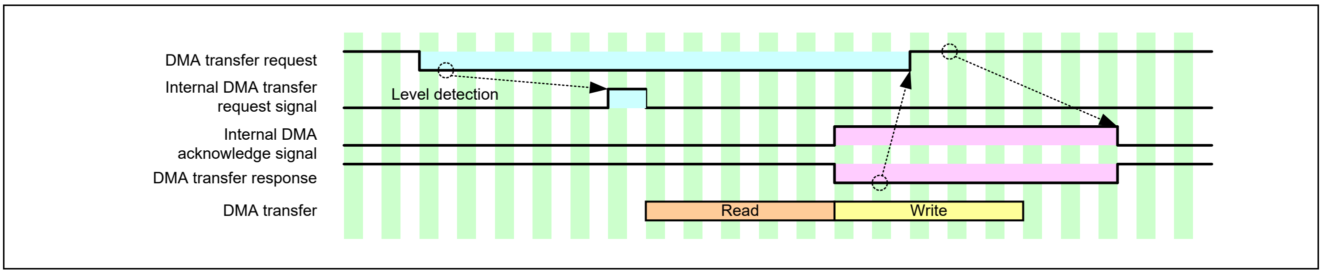


Figure 18.29 Level Output Mode Operation Example 2

DMA transfer request: High level detection Request from the destination (CHCFGn.REQD = 1) DMA acknowledge output: Level mode

Remark: n = 0 to 3; p = 0, 1

(4) Bus Cycle Output

When the AM2 to AM0 bits of the channel configuration register (CHCFGn) are set to 010B, bus cycle output is selected for the DMA acknowledge signal (DMAACKZp).

The DMAACKZp signal remains active (low level) during the bus cycle. Depending on which side (source or destination) has issued the DMA transfer request, the DMA acknowledge signal is output to either the read cycle (in the case of the source) or the write cycle (in the case of the destination). If a DMA transfer (transactions) involves more than one read or write due to a bus size difference between the source and destination or for some other reason, DMAACKZp is asserted during that period.

When bus cycle output is selected for the DMA acknowledge signal, the DMA transfer interface signal control registers 0 to 3 (DMAIFCp) should be left at their initial value and extension of the DMAACKZp width should not be used.

Caution: In bus cycle output mode, the DMAREQZp signal is not accepted for a period of one BUSCLK cycle after the DMA transfer bus cycle is completed.

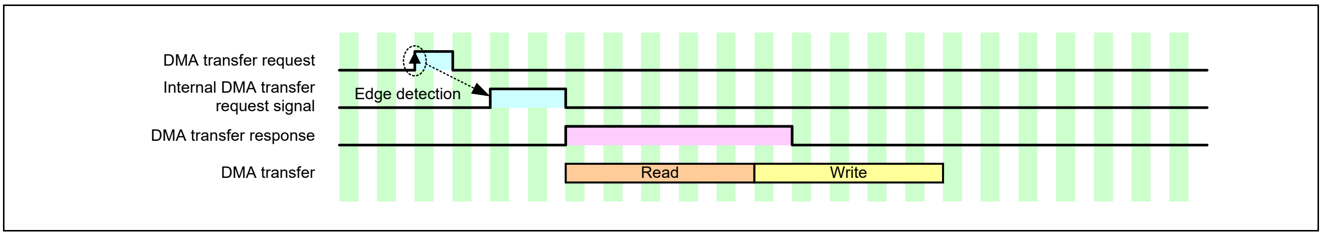


Figure 18.30 Bus Cycle Output Mode Operation Example 1

DMA transfer request: Rising edge detection Request from the source (CHCFGn.REQD = 0) DMA acknowledge output: Level mode SDS[3:0] = DDS[3:0] (In this example, the transfer size of the source is the same as that of the destination.)

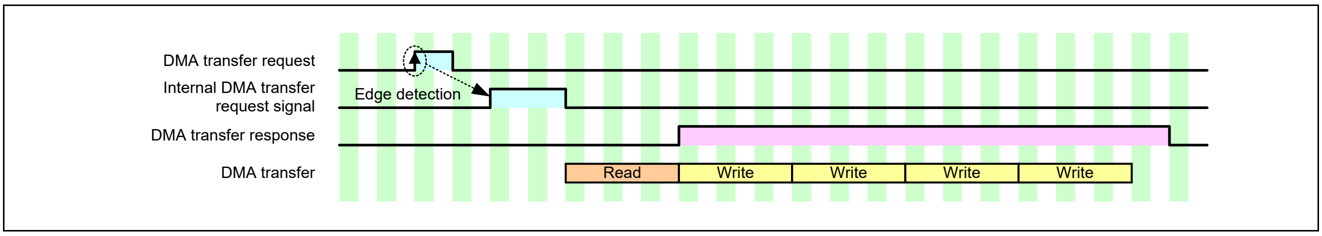


Figure 18.31 Bus Cycle Output Mode Operation Example 2

DMA transfer request: Rising edge detection Request from the destination (CHCFGn.REQD = 1) DMA acknowledge output: Level mode SDS[3:0]>DDS[3:0] (In this example, the transfer size of the source is four times that of the destination.)

Remark: n = 0 to 3; p = 0, 1

18.8.5 DMA Transfer Completion Interrupt

When a DMA transaction (the series of DMA transfers) is completed, the pulse output of INTDMAn occurs. For the relationship between INTDMAn and units/channels, see Table 18.29, Relationship between DMA Transfer Completion Interrupts and Units/Channels.

When the total numbers of transfer bytes loaded to the Current transaction byte register (CRTBn) have successfully been transferred, the END bit of the channel status register (CHSTATn) is set to 1. In this case, when the DEM bit of the channel configuration register (CHCFGn) is cleared to 0, INTDMAn occurs.

When writeback is performed in link mode, INTDMAn occurs after the writeback operation. Also, when a descriptor is read in link mode and CHCFGn.DRRP is set to 0, the CHSTATn.DER bit is set to 1 if LV of the read descriptor header is set to 0. In this case, when CHCFGn.DIM is set to 0, INTDMAn occurs.

Table 18.29 Relationship between DMA Transfer Completion Interrupts and Units/Channels

Unit	Channel	Corresponding Transfer Completion Interrupt Signal
DMA0 (General-purpose DMAC)	CH0	INTDMA0
	CH1	INTDMA1
	CH2	INTDMA2
	CH3	INTDMA3
DMA1 (DMAC for real-time ports)	CH0	INTRTDMA

Table 18.30 DMA Transfer Completion Interrupt Asserting Conditions

Source	Condition	INTDMAn Mask Setting Bit
DMA transaction completion	The total numbers of transfer bytes loaded to the Current transaction byte register (CRTBn) have been successfully transferred. (When writeback is performed in link mode, the interrupt occurs after the writeback operation.)	CHCFGn.DEM
Invalid descriptor (LV of the header = 0)	When DRRP and DIM of the channel configuration register (CHCFGn) are set to 0 in link mode, the LV of the read descriptor header is set to 0.	CHCFGn.DIM

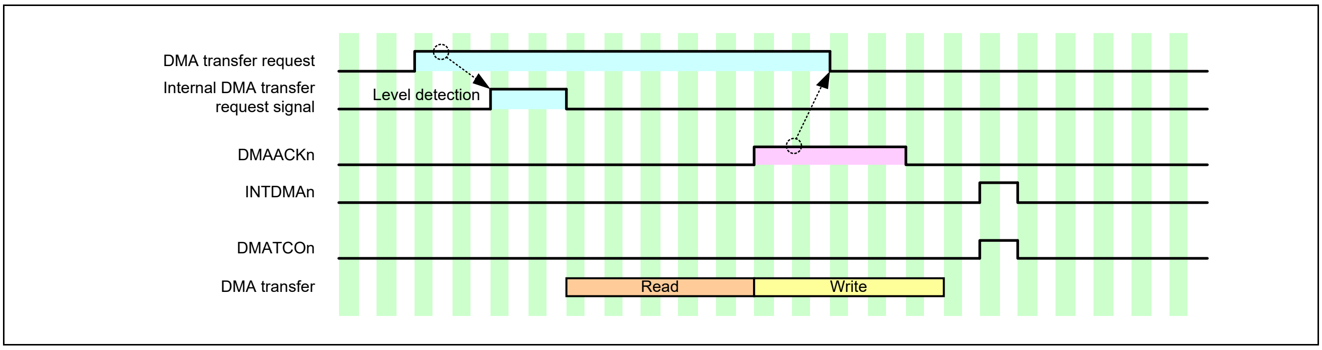


Figure 18.32 DMA Transfer Completion Interrupt Output Operation Example

DMA transfer request: High level detection, request from the destination.
DMA acknowledge output: Pulse mode

Remark: n = 0 to 3

18.8.6 DMA Terminal Count Output

As a DMA transaction (the series of DMA transfers) completion signal, the DMA terminal count signal is output.

When the DMAREQZp signal is used as a DMA transfer request, DMATCZp is used as the DMA terminal count signal. When an external interrupt or an interrupt request from an internal peripheral module is used as a DMA transfer request, the DMA terminal count signal is not used.

When the total numbers of transfer bytes loaded to the Current transaction byte register (CRTBn) have successfully been transferred, the TC bit of the channel status register (CHSTATn) is set to 1, and the DMA terminal count signal (DMATCZp) is output as a high-level signal that lasts for 1 BUSCLK cycle.

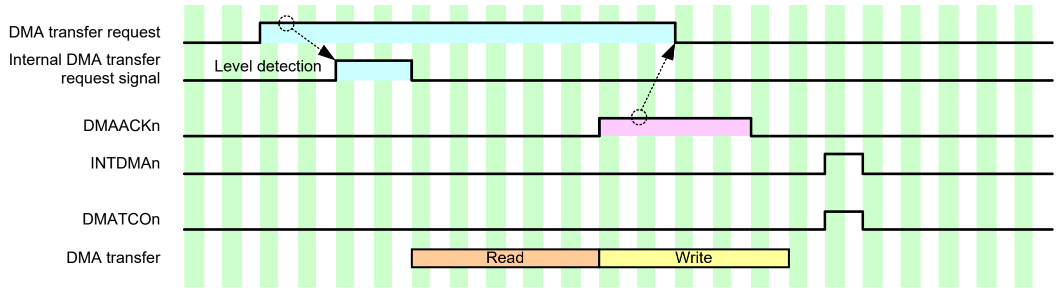


Figure 18.33 DMA Terminal Count Output Operation Example

DMA transfer request: High level detection, request from the destination. DMA acknowledge output: Pulse mode

Remark: n = 0 to 3; p = 0, 1

(1) DMA Terminal Count Signal Mask Function

The DMA terminal count signal can be masked by using the TCM bit of the CHCFGn register. Generally, in the case of a software-triggered DMA transfer (the STG bit of the channel control register (CHCTRLn) is set to 1), mask the DMA terminal count signal.

The DMA transfer requests assigned to the individual channels can be changed by using the SEL2 to SEL0 bits of the channel configuration register (CHCFGn). The output of the DMA terminal count signal depends on the selection of these bits.

Table 18.31 DMA Terminal Count Output Setting

CHCFGn.TCM	Operation	Use
0	Enables the DMA terminal count output.	Use for hardware-triggered DMA transfers. This is intended to detect: • End of counting • End of link mode
1	Masks the DMA terminal count output.	Use for software-triggered DMA transfers. After the DMA transaction (the series of DMA transfers), TCM is cleared to 0 and the DMA terminal count output is enabled.

Remark: n = 0 to 3

(2) Assignment of DMA Channels and DMA Terminal Count Output Signals

In round-robin mode, where control is exerted so that all DMA channels have equal priority, make sure that channels correspond to pin names by using the SEL2 to SEL0 bits of the CHCFGn register.
For example, select DMAREQZ1, DMAACKZ1, or DMATCZ1 for the DMA interface signal of channel 1.

In fixed priority mode, change the relationship between DMA channels and DMA interface signals, by using the SEL2 to SEL0 bits of the CHCFGn register in accordance with the requirement for the priority of the DMA transfer request. For the configuration of allocation of DMA channels and DMA terminal output signals, see Figure 18.1, Relation between DMA Units/Channels and DMA Trigger.

Remark: n = 0 to 3

18.8.7 Forced Dumping

When the SETSSWPRQ bit of the channel control register (CHCTRLn) is set to 1, the DMAC forces the buffer to dump (write) its data to the transfer destination. After that, the DMA transfer continues.

If the DMA transfer request and forced dumping are in contention, forced dumping is given priority and then the DMA transfer is executed.

When the REQD bit of the channel configuration register (CHCFGn) is set to 1 and DMAACKZp is set to become active at the time of writing, forced dumping cannot be used. This is because a malfunction may occur at the destination if data is transferred while the destination does not assert the DMA transfer request (DMAREQZp).

Data is also dumped when the SBE bit of the channel configuration register (CHCFGn) is set to 1. In this case, however, the EN bit of the channel status register (CHSTATn) is cleared to 0 and the DMA controller is stopped after dumping. In the case of forced dumping using the SETSSWPRQ bit, the DMA transfer continues even after the dump.

Remark: n = 0 to 3; p = 0, 1

18.8.8 DMA Error Interrupt

If an error occurs during DMA transfer or access to the descriptor, DMA transfer is stopped.

If an error occurs, the EN bit of the channel status register (CHSTATn) is cleared to 0 and the ER bit is set to 1. Also, INTDMAERRn occurs.

The validity of data cannot be guaranteed for the series of transfers that had an error. To restart DMA transfer, set the SWRST bit of the channel configuration register (CHCTRLn) to 1 to reset DMA channel n and set each register again.

Remark: n = 0 to 3

18.8.9 Interval Counting

The DMA transfer interval can be adjusted by setting the ITVL bit of the channel interval register (CHITVL).

The interval of the internal system bus clock (HCLK) cycle × the value set in ITVL15–ITVL0 can be set. This allows the bus occupancy ratio of the DMAC to be adjusted. When a single read or write operation is completed, counting down starts from the value set in CHITVL and the next internal DMA transfer request is put on hold until the count value becomes 0.

Remark: n = 0 to 3

18.8.10 Differences in Operation by Transfer Size

(1) When the Transfer Size of the Source Is Smaller Than That of the Destination

When data of the data size set in the DDS3 to DDS0 bits of the channel configuration register (CHCFGn) has been read, it is written to the destination. The number of read operations corresponds to the destination size divided by the source size.

The following timing chart shows the waveforms generated when the source size is 16 bits and the destination size is 64 bits.

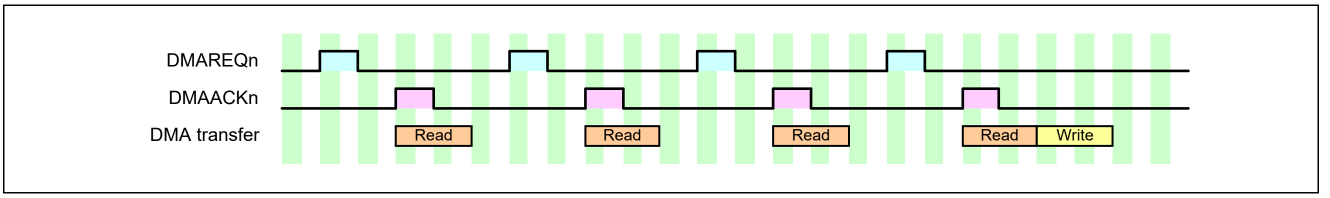


Figure 18.34 When the Transfer Size of the Source Is Smaller Than That of the Destination

DMA transfer request: Edge detection, request from the source DMA acknowledge output: Pulse mode Source: 16 bits; Destination: 64 bits

Remark: n = 0 to 3

(2) When the Transfer Size of the Destination Is Smaller Than That of the Source

Since the transfer size of the destination is smaller, the number of write operations corresponds to the source size divided by the destination size.

The following timing chart shows the waveforms generated when the source size is 64 bits and the destination size is 16 bits.

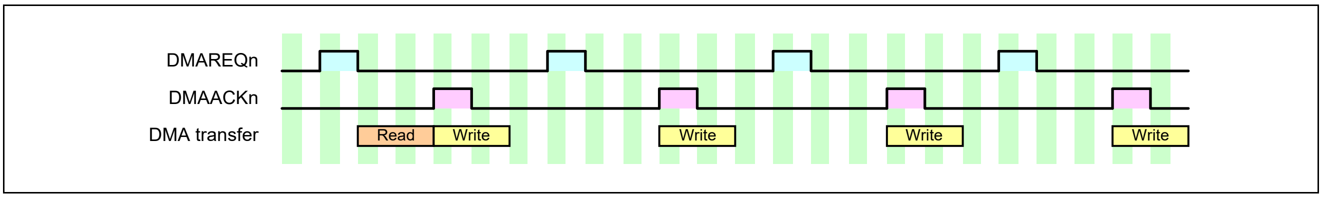


Figure 18.35 When the Transfer Size of the Destination Is Smaller Than That of the Source

DMA transfer request: Edge detection, request from the source DMA acknowledge output: Pulse mode Source: 64 bits; Destination: 16 bits

Remark: n = 0 to 3

(3) When the Transfer Size of the Destination Is Equal to That of the Source

Each time a DMA transfer request is detected, reading from the source and writing to the destination are performed.

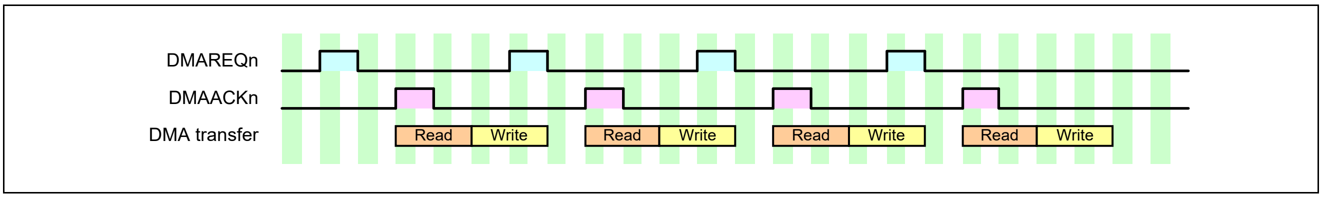


Figure 18.36 When the Transfer Size of the Destination Is Equal to That of the Source

DMA transfer request: Rising edge detection, request from the source DMA acknowledge output: Pulse mode

Remark: n = 0 to 3

(4) In Addition, Transmission in Case an Address Differs from Transfer Size

Read access proceeds as follows according to the data size set in the SDS2–SDS0 bits of the channel configuration register (CHCFG).

- If the data size set in the SDS2–SDS0 bits is 32 bits or less
  Access size: setting of SDS2–SDS0
- If the data size set in the SDS2–SDS0 bits is 128 bits or 256 bits
  Access size: 32-bit units

If a source address is not aligned with a 32-bit boundary, it is still accessed in a 32-bit unit, but only the required portion is loaded into the buffer of the DMAC through the system bus. This may lead to access to addresses which are not within the range of source addresses.

Write access proceeds as follows according to the data size set in the DDS2–DDS0 bits of the channel configuration register (CHCFG).

- If the data size set in the DDS2–DDS0 bits is 32 bits or less
  Access size: setting of DDS2–DDS0
- If the data size set in the DDS2–DDS0 bits is 128 bits or 256 bits
  Access size: 32-bit units

In write access, there is no access to locations other than those in space specified by the settings. Moreover, access is in combinations of sizes that include smaller ones than that set in the DDS2–DDS0 bits in the following cases.

- When a destination address is not aligned to the data size set in the DDS2–DDS0 bits.
- When a skip boundary is crossed in access of the data size set in the DDS2–DDS0 bits.
- When the number of remaining transfer bytes is smaller than the data size set in the DDS2–DDS0 bits.
- When the number of transfer bytes is smaller than the data size set in the DDS2–DDS0 bits.

Remark: n = 0 to 3

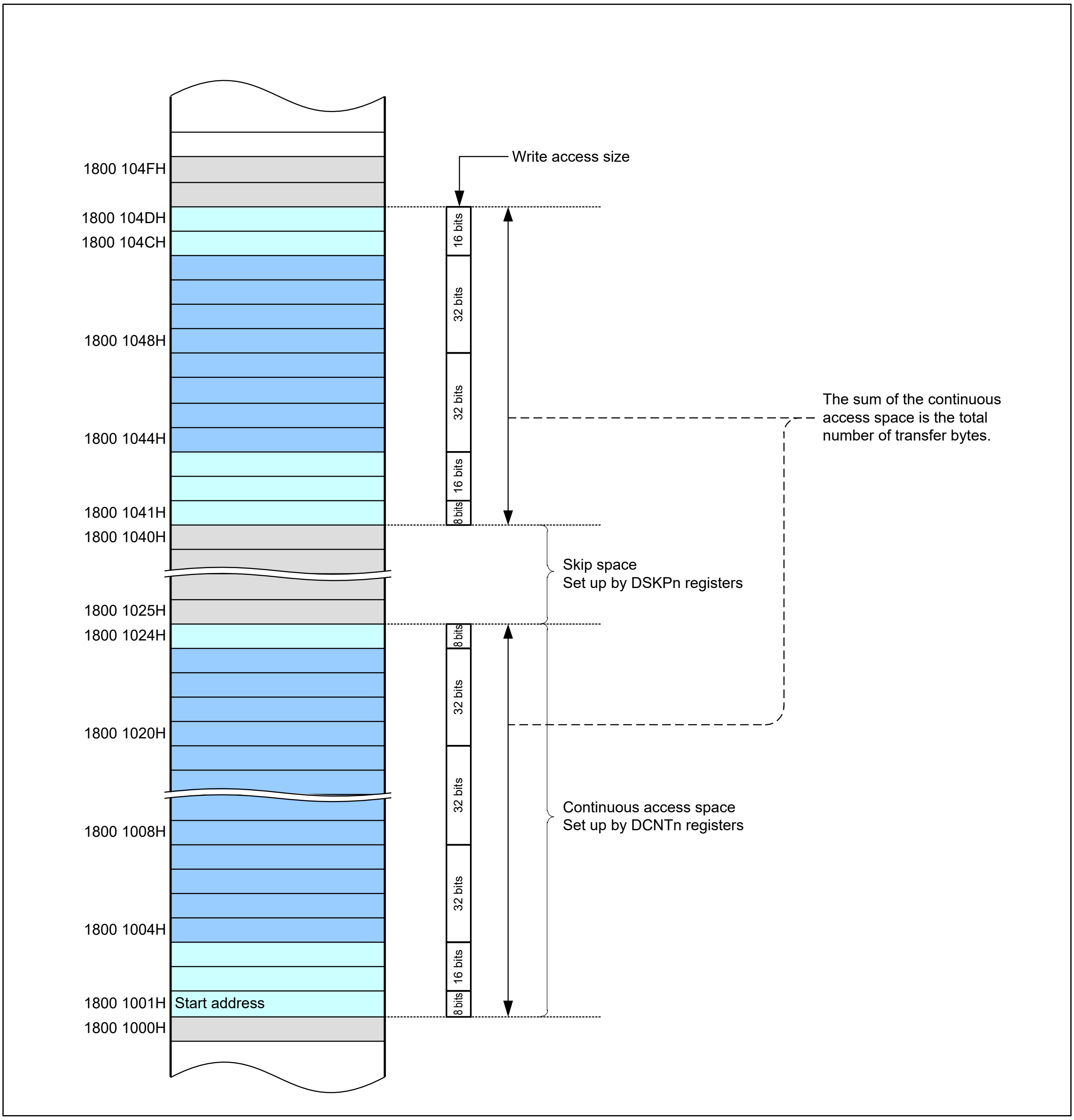


Figure 18.37 DMA Write Access and Access Type Example

18.8.11 Transfer Status

The state of transfer by DMA channel n can be checked by using the channel status register (CHSTATn).

The TACT bit of the CHSTATn register indicates whether channel n is active. When the SETEN bit of the channel control register (CHCTRLn) is set to 1, the TACT bit is set to 1. The setting of the TACT bit is also 1 during access to the descriptor in link mode or in the DMA transfer request wait state.

The TACT bit is cleared to 0 when the clearing condition for the EN bit of the CHSTATn register is met and the DMA transfer is completed. Even after the DMA transaction is completed, the TACT bit is not cleared to 0, if the clearing condition for the EN bit of the CHSTATn register is not met (for example, CHCFGn.REN is set to 1 in register mode or the descriptor is accessed in link mode).

The transfer status is updated individually for each DMA transfer.

Remark: n = 0 to 3

18.8.12 Suspension

DMA transfer can be suspended by setting the SETSUS bit of the channel control register (CHCTRLn) to 1.
In this case, if there is any bus cycle already being executed, the transfer is suspended after that bus cycle is completed. Setting the CLRSUS bit of the channel control register (CHCTRLn) to 1 leads to release from the suspended state.

To check the suspended state, set SETSUS to 1 and then read the CHSTATn register or DSTSUS register to see that the SUS bit is set to 1 for the relevant channel.

Remark: n = 0 to 3

18.8.13 Suspending Transfer

A DMA transaction (the series of DMA transfers) on a specific DMA channel can be suspended by setting the CLREN bit of the channel control register (CHCTRLn) to 1 during the DMA transaction (the series of DMA transfers).

For the processing after the suspension, one of two modes can be selected. One mode dumps the data remaining in the buffer when the transaction is suspended (the SBE bit of the channel configuration register (CHCFGn) is set to 1), and the other mode does not (the SBE bit is cleared to 0).

If there is any data remaining in the DMAC buffer when a DMA transaction (the series of DMA transfers) is suspended with dump mode enabled and CLREN set to 1, the data is dumped and then the DMA transaction is completed.

If a DMA transfer is interrupted, INTDMAn does not occur.

After DMA transfer is suspended, be sure to set the SWRST bit of the channel control register (CHCTRLn) to 1 to reset the internal state of the DMA channel before setting the next transfer.

Caution: A DMA transfer may be in progress even if the CLREN bit of the channel control register (CHCTRLn) set to 1 and the EN bit is cleared to 0. To make sure that the DMA channel is stopped, check that the EN and TACT bits of the channel status register (CHSTATn) are cleared to 0.

Remark: n = 0 to 3

(1) When Buffer Dumping is Disabled (SBE = 0)

When CLREN is set to 1 during the DMA transaction (the series of DMA transfers), the transaction is stopped by suspending the DMA transfer.

According to the setting of the REQD bit of the channel configuration register (CHCFGn), the transaction is stopped after the read cycle in the case of a DMA transfer request from the source or after the write cycle in the case of a DMA transfer request from the destination.

(2) When Buffer Dumping is Enabled (SBE = 1)

When CLREN is set to 1 during a DMA transaction (the series of DMA transfers), the transaction is stopped by suspending the DMA transfer. When REQD is set to 0, the data already read is dumped (written) and then the DMA transfer is stopped.
When REQD is set to 1 and a DMA transfer request from hardware is used, do not use dump mode.

18.9 DMA Transfer Setting Examples

The conditions for transfer in the individual setting examples are as follows.

Caution: This section explains only operation of the general-purpose DMAC since the specifications of operations of the general-purpose DMAC and the DMAC for real-time ports are the same.

Table 18.32 Conditions for Transfer in DMA Transfer Setting Examples

Setting Example	DMA Mode	Transfer Mode	Transfer Request
Setting example 1	Register mode	Single transfer mode	Hardware
Setting example 2	Register mode	Block transfer mode	Software
Setting example 3	Register mode (continuous execution)	Block transfer mode	Software
Setting example 4	Link mode	Block transfer mode	Software

18.9.1 Setting Example 1 (Register Mode, Single Transfer Mode, and Hardware Trigger)

Shown below are the setting examples applicable when performing a DMA transfer based on the settings in Table 18.33.

Table 18.33 DMA Transfer Setting Example 1

Item	Description	
Unit used	Unit 0 (General-purpose DMAC)	
Channel used	Channel 1	
Priority control	Fixed priority	
DMA mode	Register mode	
Transfer mode	Single transfer mode	
Register set used	Next 0 register set	
Source/destination	Source	Destination
Start address	1000 0000H	2000 0000H
Address counting direction	Increment	Increment
Transfer data size	32 bits	32 bits
DMA transaction data size	64 bytes	
DMA interface pin	DMAREQZ1, DMAACKZ1, DMATCZ1	
DMA transfer request	Hardware (Rising edge detection using DMAREQZ1 of the source)	
DMA acknowledge signal	Hardware (output when read due to a request from the source)	
INTDMA mask function	Not applicable	

Table 18.34 Register Settings of Setting Example 1

Register	Set Value	Set Content
N0SA1	1000 0000H	Source address
N0DA1	2000 0000H	Destination address
N0TB1	0000 0040H	Number of transaction data bytes
CHCFG1	0002 2021H	Channel configuration
CHITVL1	0000 0000H	Minimum transfer interval
DTFR1	0000 0002H	DMAREQZ1 pin input is set up.

Table 18.35 Channel Configuration Register (CHCFG1) Settings of Setting Example 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCFG1	DMS	REN	RSW	RSEL	SBE	DIM	TCM	DEM	WONLY	TM	DAD	SAD	DDS3–DDS0				SDS3–SDS0				DRRP	AM2–AM0			0	LVL	LEN	HEN	REQD	SEL2–SEL0		
Set value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1

Address 400A 286CH
Initial value 0000 0000H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>31</td><td>DMS</td><td>0: Register mode</td></tr>
<tr><td>30</td><td>REN</td><td>0: Does not execute continuously.</td></tr>
<tr><td>29</td><td>RSW</td><td>0: Does not invert RSEL after a DMA transaction (the series of DMA transfers) is completed.</td></tr>
<tr><td>28</td><td>RSEL</td><td>0: Uses the Next 0 register set for the next DMA transfer.</td></tr>
<tr><td>27</td><td>SBE</td><td>0: Stops the transfer without dumping (writing) buffer data if the operation is stopped.</td></tr>
<tr><td>26</td><td>DIM</td><td>0: Does not mask INTDMA01 when LV is set to 0 in link mode.</td></tr>
<tr><td>25</td><td>TCM</td><td>0: Does not mask (enables terminal count output (DMATCZ1)).</td></tr>
<tr><td>24</td><td>DEM</td><td>0: Enables INTDMA01 output when a DMA transaction is completed.</td></tr>
<tr><td>23</td><td>WONLY</td><td>0: Normal mode.</td></tr>
<tr><td>22</td><td>TM</td><td>0: Single transfer mode.</td></tr>
<tr><td>21</td><td>DAD</td><td>0: Increments the destination address.</td></tr>
<tr><td>20</td><td>SAD</td><td>0: Increments the source address.</td></tr>
<tr><td>19</td><td>DDS3</td><td>0: Uses the normal addressing mode for the destination.</td></tr>
<tr><td>18 to 16</td><td>DDS2–
DDS0</td><td><table><tr><th>DDS2</th><th>DDS1</th><th>DDS0</th><th>DMA transfer destination transfer size</th></tr><tr><td>0</td><td>1</td><td>0</td><td>32 bits</td></tr></table></td></tr>
<tr><td>15</td><td>SDS3</td><td>0: Uses the normal addressing mode for the source.</td></tr>
<tr><td>14 to 12</td><td>SDS2–
SDS0</td><td><table><tr><th>SDS2</th><th>SDS1</th><th>SDS0</th><th>Uses the normal addressing mode for the source</th></tr><tr><td>0</td><td>1</td><td>0</td><td>32 bits</td></tr></table></td></tr>
<tr><td>11</td><td>DRRP</td><td>0: Stops the operation by setting the CHSTAT1.DER bit to 1 when LV is set to 0 in link mode.</td></tr>
<tr><td>10 to 8</td><td>AM2–
AM0</td><td><table><tr><th>AM2</th><th>AM1</th><th>AM0</th><th>DMA acknowledge signal (DMAACKZ1) output mode</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Pulse mode (initial value)</td></tr></table></td></tr>
<tr><td>6</td><td>LVL</td><td rowspan="3"><table><tr><th>LVL</th><th>HEN</th><th>LEN</th><th>DMA transfer request signal detection method</th></tr><tr><td>0</td><td>1</td><td>0</td><td>Rising edge detection</td></tr></table></td></tr>
<tr><td>5</td><td>LEN</td></tr>
<tr><td>4</td><td>HEN</td></tr>
<tr><td>3</td><td>REQD</td><td>0: DMAACKZ1 becomes active when read (DMAREQZ1 is the source)</td></tr>
<tr><td>2 to 0</td><td>SEL2–
SEL0</td><td><table><tr><th>SEL2</th><th>SEL1</th><th>SEL0</th><th>DMA interface signal selection</th></tr><tr><td>0</td><td>0</td><td>1</td><td>The DMA transfer source selected by DTFR1 is chosen.</td></tr></table></td></tr>
</table>

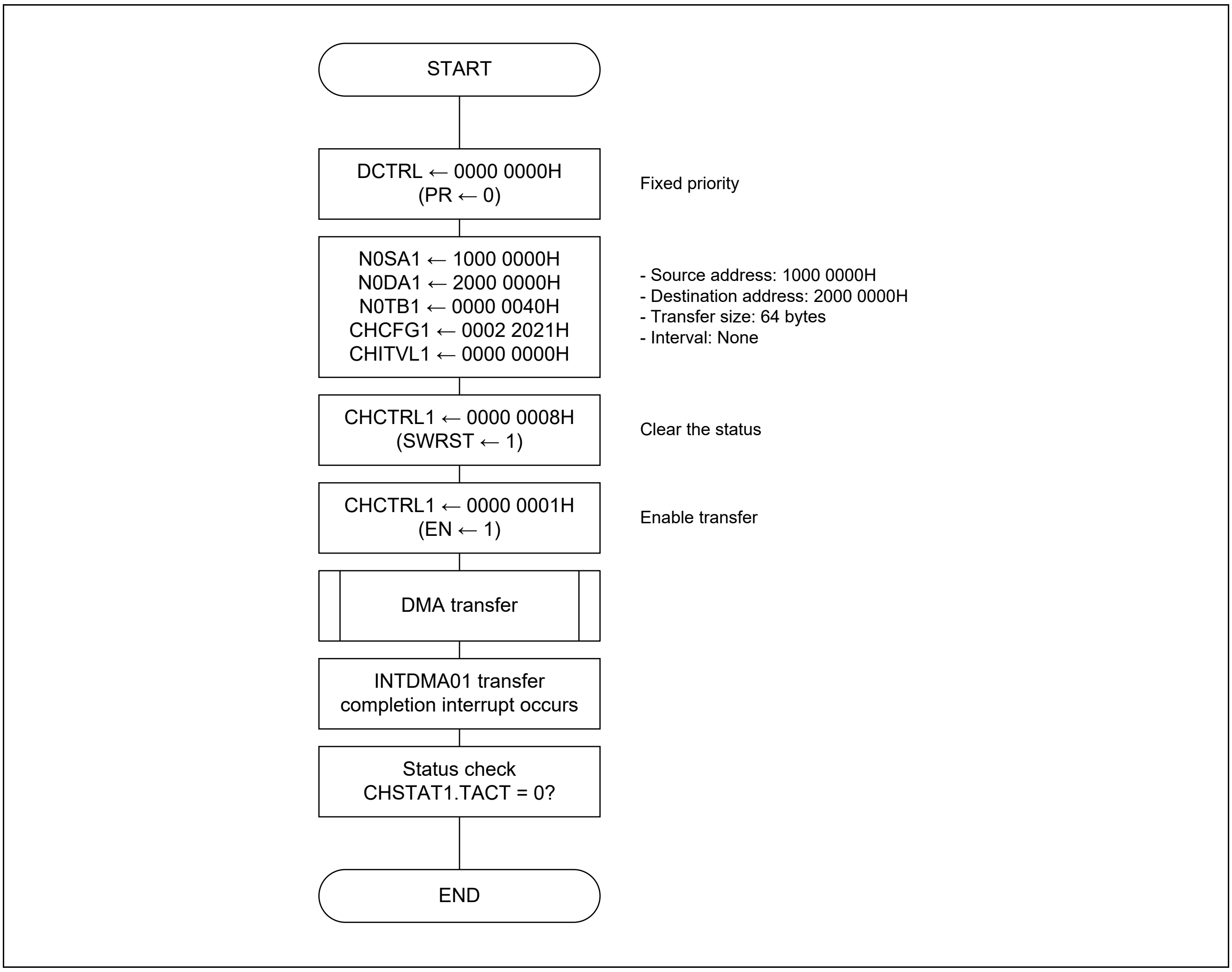


Figure 18.38 Operation Flow of Setting Example 1

18.9.2 Setting Example 2 (Register Mode, Block Transfer Mode, and Software Trigger)

Shown below are the setting examples applicable when performing a DMA transfer based on the settings in Table 18.36.

Table 18.36 DMA Transfer Setting Example 2

Item		Description	
Unit used		Unit 0 (General-purpose DMAC)	
Channel used		Channel 2	
Priority control		Round robin mode	
DMA mode		Register mode	
Transfer mode		Block transfer mode	
Register set used		Next 1 register set	
Source/destination		Source	Destination
	Start address	1100 0000H	2007 0000H
	Address counting direction	Increment	Increment
	Transfer data size	8 bits	256 bits
DMA transaction data size		128 bytes	
DMA interface pin		DMA transfer source selected by DTFR2 is chosen.	
DMA transfer request		Software	
DMA acknowledge signal		Masks the DMA acknowledge signal.	
INTDMA mask function		Not applicable	

Table 18.37 Register Settings of Setting Example 2

Register	Set Value	Set Content
DCTRL	0000 0001H	Set the order of priority (round robin mode).
N1SA2	1100 0000H	Source address
N1DA2	2007 0000H	Destination address
N1TB2	0000 0080H	Number of transaction data bytes
CHCFG2	1245 0402H	Channel configuration
CHITVL2	0000 0000H	Minimum transfer interval
DTFR2	0000 0000H	Hardware trigger mask

Table 18.38 Channel Configuration Register (CHCFG2) Settings of Setting Example 2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCFG2	DMS	REN	RSW	RSEL	SBE	DIM	TCM	DEM	WONLY	TM	DAD	SAD	DDS3–DDS0				SDS3–SDS0				DRRP	AM2–AM0			0	LVL	LEN	HEN	REQD	SEL2–SEL0		
Set value	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

Address 400A 28ACH

Initial value 0000 0000H

Initial Value	Bit Name	Description
31	DMS	0: Register mode
30	REN	0: Does not execute continuously.
29	RSW	0: Does not invert RSEL after a DMA transaction (the series of DMA transfers) is completed.
28	RSEL	1: Uses the Next 1 register set for the next DMA transfer.
27	SBE	0: Stops the transfer without dumping (writing) buffer data if the operation is stopped.
26	DIM	0: Does not mask INTDMA02 when LV is set to 0 in link mode.
25	TCM	0: Masks terminal count output.
24	DEM	0: Enables INTDMA02 output when a DMA transaction is completed.
23	WONLY	0: Normal mode
22	TM	1: Block transfer mode
21	DAD	0: Increments the destination address.
20	SAD	0: Increments the source address.
19	DDS3	0: Uses the normal addressing mode for the destination.
18 to 16	DDS2–DDS0	(see table below)
15	SDS3	0: Uses the normal addressing mode for the source.
14 to 12	SDS2–SDS0	(see table below)
11	DRRP	0: Stops the operation by setting the CHSTAT2.DER bit to 1 when LV is set to 0 in link mode.
10 to 8	AM2–AM0	(see table below)
6	LVL	(see table below)
5	LEN	
4	HEN	
3	REQD	0: Acknowledge becomes active when read.
2 to 0	SEL2–SEL0	(see table below)

DDS2–DDS0:

DDS2	DDS1	DDS0	DMA transfer destination transfer size
1	0	1	256 bits

SDS2–SDS0:

SDS2	SDS1	SDS0	DMA transfer source transfer size
0	0	0	8 bits

AM2–AM0:

AM2	AM1	AM0	DMA acknowledge signal output mode
1	X	X	Disables DMA acknowledge signal output.

LVL, LEN, HEN:

LVL	HEN	LEN	DMA transfer request signal detection method
0	0	0	Disables detection.

SEL2–SEL0:

SEL2	SEL1	SEL0	DMA interface signal selection
0	1	0	DMA transfer source selected by DTFR2 is chosen.

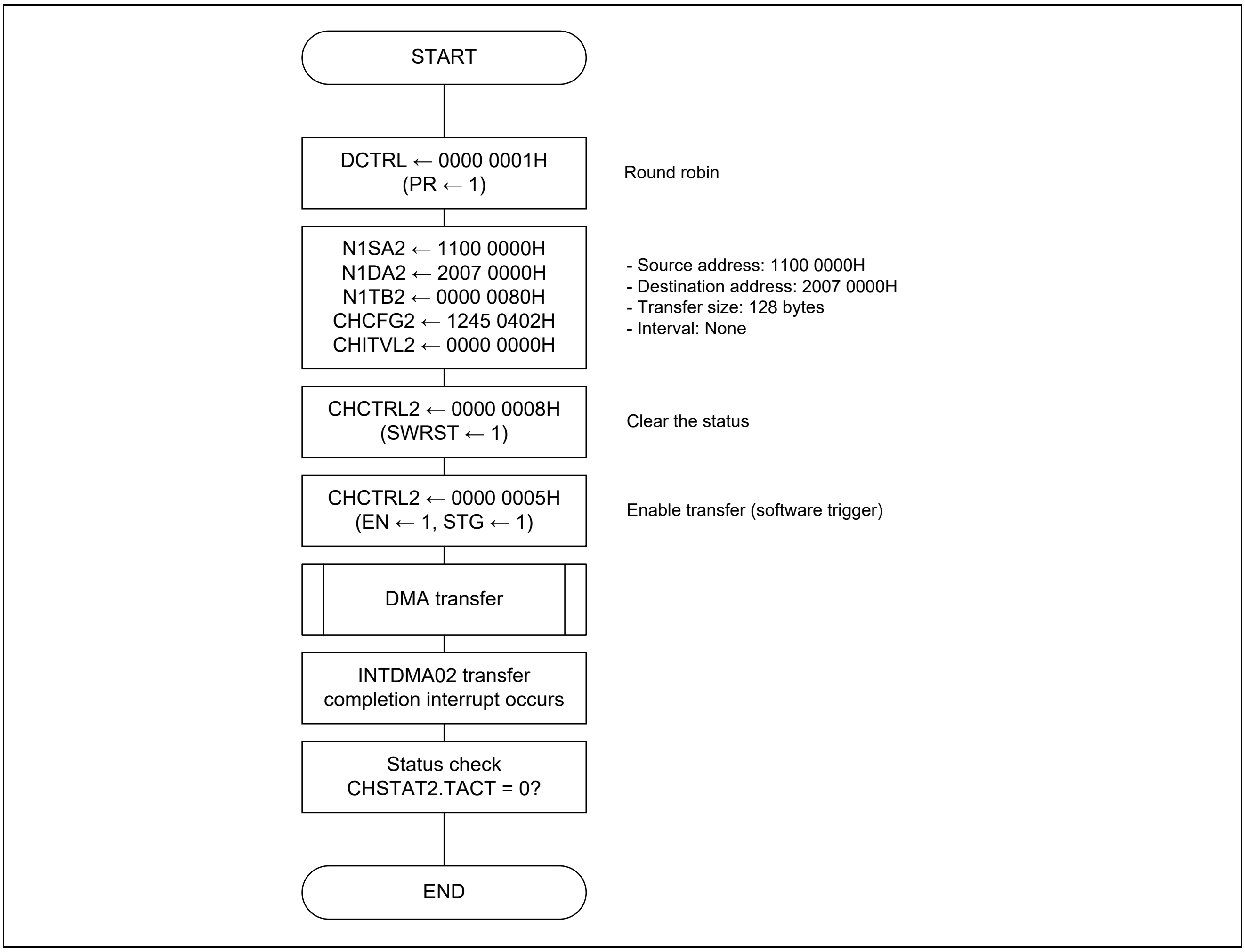


Figure 18.39 Operation Flow of Setting Example 2

18.9.3 Setting Example 3 (Register Mode: Continuous Execution, Block Transfer Mode, and Software Trigger)

Shown below are the setting examples applicable when performing a DMA transfer based on the settings in Table 18.39.

Table 18.39 DMA Transfer Setting Example 3

Item		Description	
Unit used		Unit 0 (General-purpose DMAC)	
Channel used		Channel 1	
Priority control		Round robin mode	
DMA mode		Register mode	
DMA mode		Block transfer mode	
Register set used		Uses the Next 0 register set and then the Next 1 register set continuously.	
Next 0 source/destination		Source	Destination
	Start address	2000 1000H	0800 0000H
	Address counting direction	Fixed	Fixed
	Transfer data size	32 bits	512 bits
	DMA transaction data size	512 bytes	
Next 1 source/destination		Source	Destination
	Start address	0800 0000H	1100 0000H
	Address counting direction	Fixed	Fixed
	Transfer data size	32 bits	512 bits
	DMA transaction data size	2,048 bytes	
DMA interface pin		DMAREQZ1, DMAACKZ1, DMATCZ1	
DMA transfer request		Software	
DMA acknowledge signal		Masks the DMA acknowledge signal.	
INTDMA mask function		Enables the mask when a DMA transaction is completed for the Next 0 register set.	

Table 18.40 Register Settings of Setting Example 3

Register	Set Value	Settings, etc.
DCTRL1	0000 0001H	Set the order of priority (round robin mode)
N0SA1	2000 1000H	Next 0 source address
N0DA1	0800 0000H	Next 0 destination address
N0TB1	0000 0200H	Number of transaction data bytes for Next 0
N1SA1	0800 0000H	Next 1 source address
N1DA1	1100 0000H	Next 1 destination address
N1TB1	0000 0800H	Number of transaction data bytes for Next 1
CHCFG1	6176 2001H	Channel configuration
CHITVL1	0000 0000H	Minimum transfer interval
DTFR1	0000 0000H	Mask hardware trigger

Table 18.41 Channel Configuration Register (CHCFG1) Settings of Setting Example 3

CHCFG1 — Address 400A 286CH, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMS	REN	RSW	RSEL	SBE	DIM	TCM	DEM	WONLY	TM	DAD	SAD	DDS3–DDS0				SDS3–SDS0				DRRP	AM2–AM0			0	LVL	LEN	HEN	REQD	SEL2–SEL0		
Set value	0	1	1	0	0	0	0	1	0	1	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit Position	Bit Name	Description
31	DMS	0: Register mode
30	REN	1: Executes continuously (uses the Next register set selected by the RSEL bit).
29	RSW	1: Inverts RSEL after a DMA transaction (the series of DMA transfers) is completed.
28	RSEL	0: Uses the Next 0 register set for the next DMA transfer.
27	SBE	0: Stops the transfer without dumping (writing) buffer data if the operation is stopped.
26	DIM	0: Does not mask INTDMA01 when LV is set to 0 in link mode.
25	TCM	0: Does not mask (enables terminal count output (DMATCZ1)).
24	DEM	1: Masks INTDMA01 output when a DMA transaction is completed.
23	WONLY	0: Normal mode
22	TM	1: Block transfer mode
21	DAD	1: The destination address is fixed.
20	SAD	1: The source address is fixed.
19	DDS3	0: Uses the normal addressing mode for the destination.
18 to 16	DDS2–DDS0	DDS2 = 1, DDS1 = 1, DDS0 = 0: DMA Destination Transfer Size 512 bits
15	SDS3	0: Uses the normal addressing mode for the source.
14 to 12	SDS2–SDS0	SDS2 = 0, SDS1 = 1, SDS0 = 0: DMA Source Transfer Size 32 bits
11	DRRP	0: Stops the operation by setting the CHSTAT1.DER bit to 1 when LV is set to 0 in link mode.
10 to 8	AM2–AM0	AM2 = 0, AM1 = 0, AM0 = 0: DMA Acknowledge Signal (DMAACKZ1) Output Mode: Pulse mode
6	LVL	LVL = 0, HEN = 0, LEN = 0: DMA Transfer Request Signal Detection Method: Disables detection.
5	LEN	
4	HEN	
3	REQD	0: DMAACKZ1 becomes active when read (DMAREQZ1 is the source).
2 to 0	SEL2–SEL0	SEL2 = 0, SEL1 = 0, SEL0 = 1: DMA Interface Signal Selection: DMA transfer source selected by DTFR1 is chosen.

Sub-tables:

DDS2	DDS1	DDS0	DMA Destination Transfer Size
1	1	0	512 bits

SDS2	SDS1	SDS0	DMA Source Transfer Size
0	1	0	32 bits

AM2	AM1	AM0	DMA Acknowledge Signal (DMAACKZ1) Output Mode
0	0	0	Pulse mode

LVL	HEN	LEN	DMA Transfer Request Signal Detection Method
0	0	0	Disables detection.

SEL2	SEL1	SEL0	DMA Interface Signal Selection
0	0	1	DMA transfer source selected by DTFR1 is chosen.

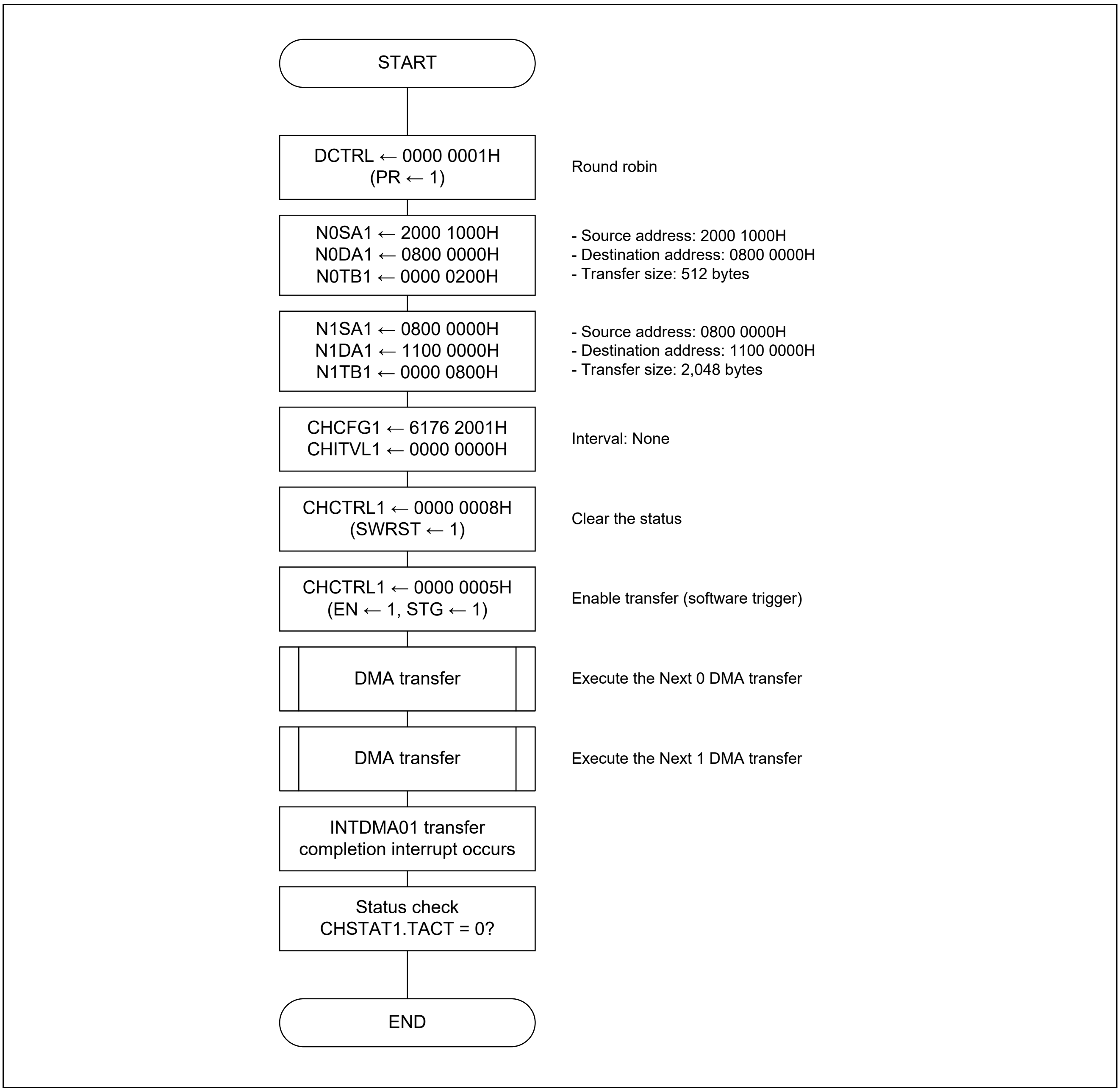


Figure 18.40 Operation Flow of Setting Example 3

18.9.4 Setting Example 4 (Link Mode, Block Transfer Mode, and Software Trigger)

Shown below are the setting examples applicable when performing a DMA transfer based on the settings in Table 18.42 to Table 18.47.

Table 18.42 DMA Transfer Setting Example 4

Item	Description
Unit used	Unit 0 (General-purpose DMAC)
Channel used	Channel 0
Priority control	Round robin mode
DMA mode	Link mode
Transfer mode	Block transfer mode
Descriptor start address	2001 1000H

Table 18.43 Descriptor 1 Settings of DMA Transfer Setting Example 4

Item		Descriptor 1 Settings	
Descriptor start address		2001 1000H	
Next descriptor start address		2001 2000H	
Transfer mode		Block transfer mode	
Next 0 source/destination		Source	Destination
	Start address	0900 0000H	2000 0000H
	Address counting direction	Increment	Increment
	Transfer data size	32 bits	32 bits
	DMA transaction data size	2,048 bytes	
DMA interface pin		DMAREQZ0, DMAACKZ0, DMATCZ0	
DMA transfer request		Software	
DMA acknowledge signal		Masks the DMA acknowledge signal.	
INTDMA mask function		Enables the mask when a DMA transaction is completed for descriptor 1.	
Descriptor format		1 (8 words)	
Descriptor header			
	LV bit writeback	Enabled (WBD = 0)	
	Next link destination	Available (LE = 0)	
	Descriptor enable status	Enabled (LV = 1)	

Table 18.44 Descriptor 2 Settings of DMA Transfer Setting Example 4

Item		Descriptor 2 Settings	
Descriptor start address		2001 2000H	
Next descriptor start address		2001 5000H	
Transfer mode		Block transfer mode	
Next 0 source/destination		Source	Destination
	Start address	0800 0000H	1800 0000H
	Address counting direction	Increment	Increment
	Transfer data size	256 bits	256 bits
	DMA transaction data size	1,024 bytes	
DMA interface pin		DMAREQZ0, DMAACKZ0, DMATCZ0	
DMA transfer request		Software	
DMA acknowledge signal		Masks the DMA acknowledge signal.	
INTDMA mask function		Enables the mask when a DMA transaction is completed for descriptor 2.	
Descriptor format		1 (8 words)	
Descriptor header			
	LV bit writeback	Enabled (WBD = 0)	
	Next link destination	Available (LE = 0)	
	Descriptor enable status	Enabled (LV = 1)	

Table 18.45 Descriptor 3 Settings of DMA Transfer Setting Example 4

Item		Descriptor 3 Settings	
Descriptor start address		2001 5000H	
Next descriptor start address		—	
Transfer mode		Block transfer mode	
Next 0 source/destination		Source	Destination
	Start address	2000 0000H	1400 0000H
	Address counting direction	Increment	Increment
	Transfer data size	512 bits	512 bits
	DMA transaction data size	4,096 bytes	
DMA interface pin		DMAREQZ0, DMAACKZ0, DMATCZ0	
DMA transfer request		Software	
DMA acknowledge signal		Masks the DMA acknowledge signal.	
INTDMA mask function		Does not mask.	
Descriptor format		1 (8 words)	
Descriptor header			
	LV bit writeback	Enabled (WBD = 0)	
	Next link destination	Not available (LE = 1)	
	Descriptor enable status	Enabled (LV = 1)	

Table 18.46 Register Settings of Setting Example 4

Register	Set Value	Settings, etc.
DCTRL	0000 0001H	Set the order of priority (round robin mode).
NXLA0	2001 1000H	Descriptor start address.
CHCFG0	8000 0000H	Channel configuration.

Table 18.47 Descriptor Settings of Setting Example 4

Item	Descriptor 1	Descriptor 2	Descriptor 3
Header	1100 0000H	1100 0000H	1300 0000H
Source address	0900 0000H	0800 0000H	2000 0000H
Destination address	2000 0000H	1800 0000H	1400 0000H
Transaction byte	0000 0800H	0000 0400H	0000 1000H
Channel configuration	8342 2008H	8345 5008H	8246 6008H
Channel interval	0000 0000H	0000 0000H	0000 0000H
Next link address	2001 2000H	2001 5000H	0000 0000H

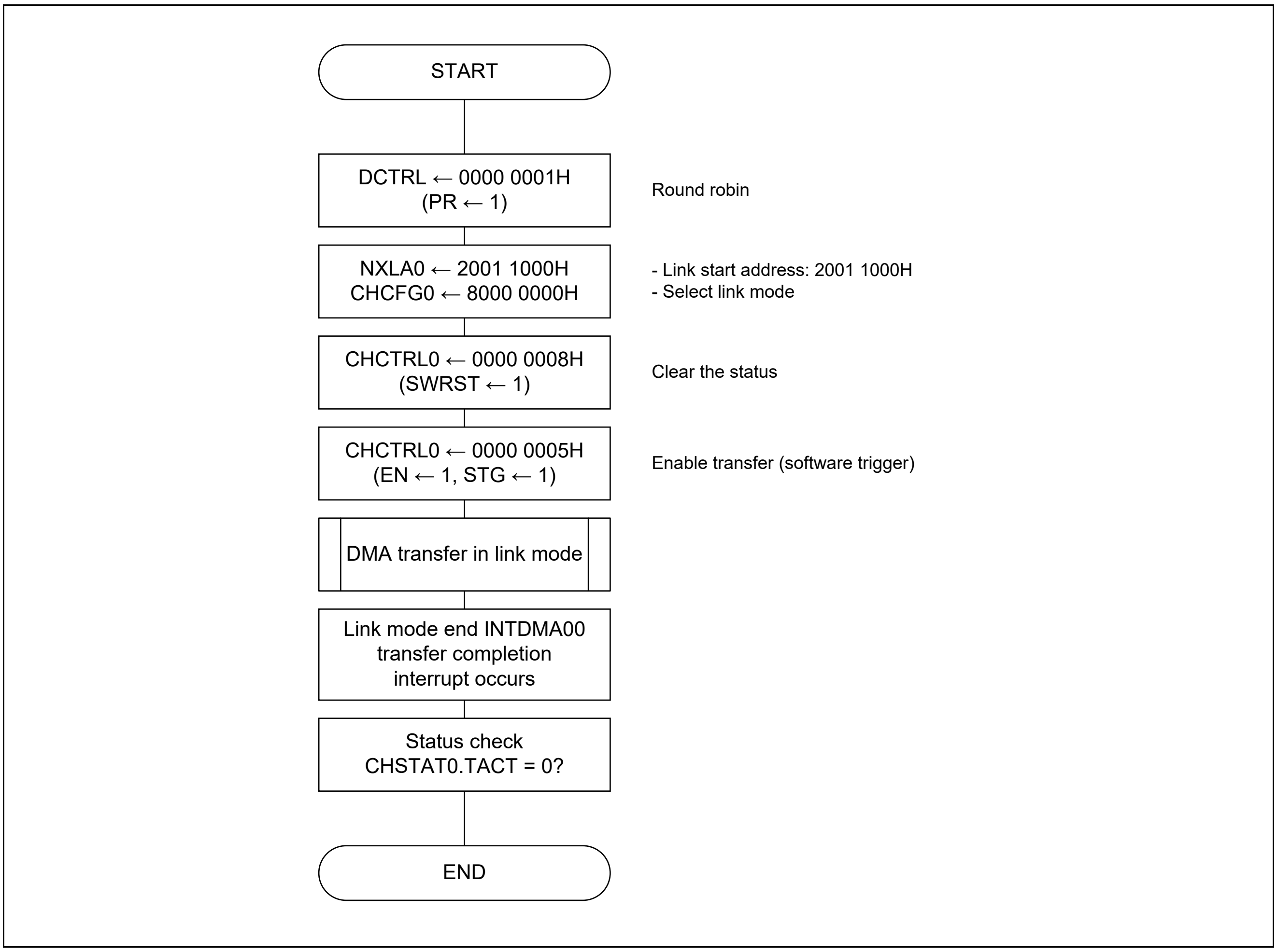


Figure 18.41 Operation Flow of Setting Example 4

18.10 Notes

1. Data consistency cannot be guaranteed if the source and destination of the transfer use the same area or share part of the same area. Therefore, make sure that the address area of the data transfer source does not overlap that of the data transfer destination.
2. If the source address is fixed (SAD of the CHCFGn register is set to 1), skip mode cannot be specified for the transfer source.
3. If the destination address is fixed (DAD of the CHCFGn register is set to 1), skip mode cannot be specified for the transfer destination.
4. If the source address is fixed (SAD of the CHCFGn register is set to 1), the source data needs to be aligned with the address of the transfer size selected by SDS3 to SDS0 of the CHCFGn register. For example, when 32 bits is selected, allocate data to an address whose value is divisible by 4.
5. If the destination address is fixed (DAD of the CHCFGn register is set to 1), the destination data needs to be aligned with the address of the transfer size selected by DDS3 to DDS0 of the CHCFGn register.

   For example, when 32 bits is selected, allocate data to an address whose value is divisible by 4.
6. When a hardware trigger is used, dump mode (SBE of the CHCFGn register is set to 1) cannot be used if a DMA transfer request is issued from the destination (REQD of the CHCFGn register is set to 1).
7. When a bus cycle output is chosen as a DMA acknowledge output, the bus cycle output is based on the read/write cycle of the internal system bus.

   In the bus cycle of the external bus interface, the DMA acknowledge signal may be output faster than the actual read/write cycle proceeds due to settings for bus conversion in the memory controller, waiting, etc.
8. Usually arrange the descriptor to the data RAM.

   A descriptor cannot be arranged in the area which cannot be specified as a slave.

   In the area which cannot be specified as a slave, a bus error occurs when the descriptor is read.

Remark: n = 0 to 3

19. 32-Bit Timer Array Unit (TAUJ2)

This section explains the 32-bit timer array unit (TAUJ2).

19.1 Features of TAUJ2

- Number of units: 1
- Channel index m: TAUJ2 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.

Caution: Since TINJm and TOUTJm of TAUJ2 (m = 0 to 3) are multiplexed with the same port pins as TINDm and TOUTDm of TAUD, they cannot be used at the same time. Select the pin functions to be used by using the timer I/F selection register described in section 28.17, Timer Interface Select Register (TMISEL).
Channels of both TAUJ2 and TAUD can be used at the same time in roles which do not involve external pins, for example, as interval timers driven by the internal clock. .

- I/O signals: The I/O signals of TAUJ2 are listed in the following table.

Table 19.1 TAUJ2 I/O Signals

TAUJ2 Signal	Function	Connected To
TAUJ2TTIN0	Channel 0 to 3 input ports	TIN0 (multiplexed with port P27)
TAUJ2TTIN1		TIN1 (multiplexed with port P26)
TAUJ2TTIN2		TIN2 (multiplexed with port P57)
TAUJ2TTIN3		TIN3 (multiplexed with port P52)
TAUJ2TTOUT0	Channel 0 to 3 output ports	TOUT0 (multiplexed with port P27)
TAUJ2TTOUT1		TOUT1 (multiplexed with port P26)
TAUJ2TTOUT2		TOUT2 (multiplexed with port P57)
TAUJ2TTOUT3		TOUT3 (multiplexed with port P52)

Caution: Since TINm and TOUTm are multiplexed on the same port pins, the input pin function for TINm must be set to a pin other than a port pin (m = 0 to 3).
For details, see section 28.9, Timer Input Function Select Register (SELCNT, SELCNTD).

- Interrupts and Peripheral Modules:

The following interrupt requests from TAUJ2 can be used as triggers for interrupt service routines or hardware ISRs (where listed as such), for DMA transfer (by the general-purpose DMAC or real-time port DMAC), for capture by a timer (TAUJ2 or TAUD), and for updating the real-time port pins (RP00–RP37).

Table 19.2 TAUJ2 Interrupt Signals and Requests for Peripheral Modules

TAUJ2 Signal	Function	Connected To
INTTAUJ2I0	Channel 0 interrupt	• Interrupt controller INTTAUJ2I0 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUJ2I1	Channel 1 interrupt	• Interrupt controller INTTAUJ2I1 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUJ2I2	Channel 2 interrupt	• Interrupt controller INTTAUJ2I2 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUJ2I3	Channel 3 interrupt	• Interrupt controller INTTAUJ2I3 • DMA Controller trigger (DTFR) • Timer Capture trigger (TTFR) • Real-time port trigger (RPTFR)

19.1.1 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by operating combinations of multiple channels.

19.1.1.1 Functional List

Caution: TAUJ2 only supports usage described in Table 19.3, TAUJ2 Operations. Settings of the registers for usage other than those listed in Table 19.3 are prohibited.

Table 19.3 TAUJ2 Operations

Operation	Functional Description
Independent channel operation	
19.7.1 "Interval Timer"	An interrupt is output at a regular interval.
19.7.2 "TAUJ2TTINm Input Interval Timer"	An interrupt is output at a regular interval or in response to an effective edge of an externally input signal.
19.7.3 "External Event Counting"	This is used as an event timer. It outputs an interrupt in response to the detection of an effective edge of an externally input signal.
19.7.4 "Delay Counting"	Interrupts which have a defined delay relative to input of the effective edge of an externally input signal are output.
19.7.5 "TAUJ2TTINm Input Pulse Interval Measurement"	The time of the input interval of an externally input signal is measured.
19.7.6 "TAUJ2TTINm Input Signal Width Measurement"	The signal width of an externally input signal is measured.
19.7.7 "TAUJ2TTINm Input Position Detection"	The time from the start of counting until an effective edge of an externally input signal is measured.
Synchronous channel operation	
19.8.1 "PWM Output"	PWM waveform is output.

19.2 Functional Overview

The TAUJ2 has the following functions:

- 4 channels
- 32-bit counter and 32-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Counter can be triggered by external signal
- Interrupt generation

19.2.1 Terms

In this section, the following terms are used.

- Independent/synchronous channel operation

  Independent or synchronous channel operation shows the dependency of channels on each other:

  - If a channel operates independent 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.

- Operating mode

  The operating mode can be selected for every channel m. The operating 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.

- Upper/lower channel

  Depending on the channel number m, a neighboring channel can be referred to as "upper" or "lower" channel:

  - Upper channel: Channel that has a smaller number
  - Lower channel: Channel that has a greater number

  Example: For channel 2, channel 1 is an upper channel and channel 3 is a lower channel.

The following figure shows the main components of the TAUJ2:

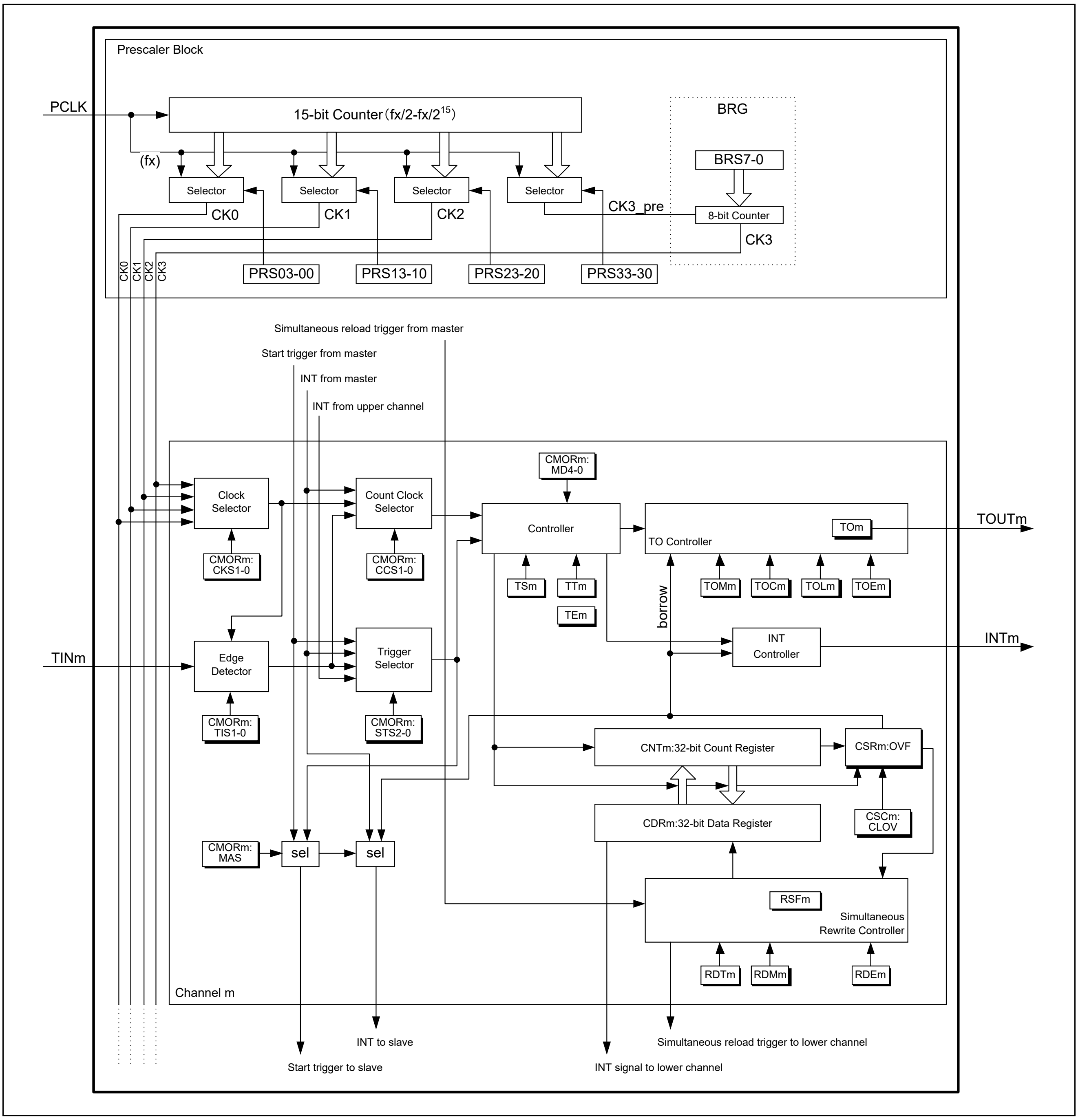


Figure 19.1 Block Diagram of the TAUJ2

19.2.2 Description of Blocks

The following describes operation of each control section of TAUJ2.

- Prescaler

  The prescaler can be used as the operating clock or count clock for all channels and up to four clock signals (CK0 to CK3) are selectable.

  Operating clocks CK0 to CK2 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}. The fourth count clock CK3 can be adjusted more precisely by using BRG to set an additional division factor that is not a power of 2.

- Clock selector

  Operation clocks for all channels (CK0–CK3) are selected.

- Count clock selector

  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)
  - INTTAUJ2Im from a master channel
  - Effective edge of TAUJ2TTINm input signal

- Controller

  The controller controls the main operations of the counter:

  - Operation mode (selected by the TAUJ2CMORm.TAUJ2MD [4:0] bits)
  - Count start enable (TAUJ2TS.TAUJ2TSm) and count stop (TAUJ2TT.TAUJ2TTm)

- Edge detection circuit

  This is for detecting an edge of TAUJ2TTINm. The type of edge to detect is selected by TAUJ2CMURm.TAUJ2TIS [1:0]. There are four types of edge which can be detected:

  - Rising edge detection
  - Falling edge detection
  - Rising and falling edge detection (width at low level)
  - Rising and falling edge detection (width at high level)

- Trigger register

  Depending on the selected operating mode, the counter starts automatically when it is enabled (TAUJ2TE.TAUJ2TEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

  - Effective edge of TAUJ2TTINm
  - INTTAUJ2Im from the master or any upper channel

- Simultaneous reload controller

  This controller controls the timing for simultaneous reloading of the values of the data registers of all channels in a channel group (TAUJ2CDRm) and the TAUJ2TOL.TAUJ2TOLm value.

- TAUJ2TO controller

  This controller controls output by each channel and generates output waveforms.

19.3 Registers

This section describes all the registers of the 32-bit TAUJ2.

Caution: TAUJ2 only supports usage described in Table 19.3, TAUJ2 Operations. Settings of the registers for usage other than those listed in Table 19.3 are prohibited.

19.3.1 TAUJ2 Registers Overview

The TAUJ2 is controlled and operated by the registers in the following table.

Table 19.4 TAUJ2 Registers Overview

(1/2)

Register Name	Symbol	Address
TAUJ2 prescaler registers		
TAUJ2 prescaler clock select register	TAUJ2TPS	4000 0090H
TAUJ2 prescaler baud rate setting register	TAUJ2BRS	4000 0094H
TAUJ2 control registers		
TAUJ2 channel data register 0	TAUJ2CDR0	4000 0000H
TAUJ2 channel data register 1	TAUJ2CDR1	4000 0004H
TAUJ2 channel data register 2	TAUJ2CDR2	4000 0008H
TAUJ2 channel data register 3	TAUJ2CDR3	4000 000CH
TAUJ2 channel counter register 0	TAUJ2CNT0	4000 0010H
TAUJ2 channel counter register 1	TAUJ2CNT1	4000 0014H
TAUJ2 channel counter register 2	TAUJ2CNT2	4000 0018H
TAUJ2 channel counter register 3	TAUJ2CNT3	4000 001CH
TAUJ2 channel mode OS register 0	TAUJ2CMOR0	4000 0080H
TAUJ2 channel mode OS register 1	TAUJ2CMOR1	4000 0084H
TAUJ2 channel mode OS register 2	TAUJ2CMOR2	4000 0088H
TAUJ2 channel mode OS register 3	TAUJ2CMOR3	4000 008CH
TAUJ2 channel mode user register 0	TAUJ2CMUR0	4000 0020H
TAUJ2 channel mode user register 1	TAUJ2CMUR1	4000 0024H
TAUJ2 channel mode user register 2	TAUJ2CMUR2	4000 0028H
TAUJ2 channel mode user register 3	TAUJ2CMUR3	4000 002CH
TAUJ2 channel status register 0	TAUJ2CSR0	4000 0030H
TAUJ2 channel status register 1	TAUJ2CSR1	4000 0034H
TAUJ2 channel status register 2	TAUJ2CSR2	4000 0038H
TAUJ2 channel status register 3	TAUJ2CSR3	4000 003CH
TAUJ2 channel status clear trigger register 0	TAUJ2CSC0	4000 0040H
TAUJ2 channel status clear trigger register 1	TAUJ2CSC1	4000 0044H
TAUJ2 channel status clear trigger register 2	TAUJ2CSC2	4000 0048H
TAUJ2 channel status clear trigger register 3	TAUJ2CSC3	4000 004CH
TAUJ2 channel start trigger register	TAUJ2TS	4000 0054H
TAUJ2 channel enable status register	TAUJ2TE	4000 0050H
TAUJ2 channel stop trigger register	TAUJ2TT	4000 0058H

(2/2)

Register name	Symbol	Address
TAUJ2 output registers		
TAUJ2 channel output enable register	TAUJ2TOE	4000 0060H
TAUJ2 channel output mode register	TAUJ2TOM	4000 0098H
TAUJ2 channel output configuration register	TAUJ2TOC	4000 009CH
TAUJ2 channel output register	TAUJ2TO	4000 005CH
TAUJ2 channel output active level register	TAUJ2TOL	4000 0064H
TAUJ2 reload data registers		
TAUJ2 channel reload data enable register	TAUJ2RDE	4000 00A0H
TAUJ2 channel reload data mode register	TAUJ2RDM	4000 00A4H
TAUJ2 channel reload data trigger register	TAUJ2RDT	4000 0068H
TAUJ2 channel reload status register	TAUJ2RSF	4000 006CH

19.3.2 TAUJ2 Prescaler Registers Details

(1) TAUJ2 Prescaler Clock Select Register (TAUJ2TPS)

This register specifies the PCLK prescalers for clocks CK0, CK1, CK2, and CK3_PRE for all channels. CK3 is generated by dividing CK3_PRE by the factor specified in TAUJ2BRS.

- Access This register can be read/written in 16-bit units. Writing is only possible while the counter is stopped (TAUJ2TE.TAUJ2TEm = 0).

(1/3)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TPS	TAUJ2PRS3[3:0]				TAUJ2PRS2[3:0]				TAUJ2PRS1[3:0]				TAUJ2PRS0[3:0]				4000 0090H	FFFFH
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Function
15 to 12	TAUJ2PRS3[3:0]	Specifies the CK3_PRE clock. Clock CK3_PRE is the input clock of the BRG unit. The BRG unit supplies the CK3 operation clock for all channels. (see table below) These bits can only be rewritten when all counters using CK3 are stopped (TAUJ2TE.TAUJ2TEm = 0).

TAUJ2PRS3[3:0]	CK3_PRE Clock
0000B	PCLK/2^{0}
0001B	PCLK/2^{1}
0010B	PCLK/2^{2}
0011B	PCLK/2^{3}
0100B	PCLK/2^{4}
0101B	PCLK/2^{5}
0110B	PCLK/2^{6}
0111B	PCLK/2^{7}
1000B	PCLK/2^{8}
1001B	PCLK/2^{9}
1010B	PCLK/2^{10}
1011B	PCLK/2^{11}
1100B	PCLK/2^{12}
1101B	PCLK/2^{13}
1110B	PCLK/2^{14}
1111B	PCLK/2^{15}

(2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUJ2PRS2[3:0]</td><td>Specifies the CK2 clock.
<table><tr><th>TAUJ2PRS2[3:0]</th><th>CK2 Clock</th></tr><tr><td>0000B</td><td>PCLK/2^0</td></tr><tr><td>0001B</td><td>PCLK/2^1</td></tr><tr><td>0010B</td><td>PCLK/2^2</td></tr><tr><td>0011B</td><td>PCLK/2^3</td></tr><tr><td>0100B</td><td>PCLK/2^4</td></tr><tr><td>0101B</td><td>PCLK/2^5</td></tr><tr><td>0110B</td><td>PCLK/2^6</td></tr><tr><td>0111B</td><td>PCLK/2^7</td></tr><tr><td>1000B</td><td>PCLK/2^8</td></tr><tr><td>1001B</td><td>PCLK/2^9</td></tr><tr><td>1010B</td><td>PCLK/2^10</td></tr><tr><td>1011B</td><td>PCLK/2^11</td></tr><tr><td>1100B</td><td>PCLK/2^12</td></tr><tr><td>1101B</td><td>PCLK/2^13</td></tr><tr><td>1110B</td><td>PCLK/2^14</td></tr><tr><td>1111B</td><td>PCLK/2^15</td></tr></table>
These bits can only be rewritten when all counters using CK2 are stopped (TAUJ2TE.TAUJ2TEm = 0).</td></tr>
<tr><td>7 to 4</td><td>TAUJ2PRS1[3:0]</td><td>Specifies the CK1 clock.
<table><tr><th>TAUJ2PRS1[3:0]</th><th>CK1 Clock</th></tr><tr><td>0000B</td><td>PCLK/2^0</td></tr><tr><td>0001B</td><td>PCLK/2^1</td></tr><tr><td>0010B</td><td>PCLK/2^2</td></tr><tr><td>0011B</td><td>PCLK/2^3</td></tr><tr><td>0100B</td><td>PCLK/2^4</td></tr><tr><td>0101B</td><td>PCLK/2^5</td></tr><tr><td>0110B</td><td>PCLK/2^6</td></tr><tr><td>0111B</td><td>PCLK/2^7</td></tr><tr><td>1000B</td><td>PCLK/2^8</td></tr><tr><td>1001B</td><td>PCLK/2^9</td></tr><tr><td>1010B</td><td>PCLK/2^10</td></tr><tr><td>1011B</td><td>PCLK/2^11</td></tr><tr><td>1100B</td><td>PCLK/2^12</td></tr><tr><td>1101B</td><td>PCLK/2^13</td></tr><tr><td>1110B</td><td>PCLK/2^14</td></tr><tr><td>1111B</td><td>PCLK/2^15</td></tr></table>
These bits can only be rewritten when all counters using CK1 are stopped (TAUJ2TE.TAUJ2TEm = 0).</td></tr>
</table>

(3/3)

Bit Position	Bit Name	Function
3 to 0	TAUJ2PRS0[3:0]	Specifies the CK0 clock.

TAUJ2PRS0[3:0]	CK0 Clock
0000B	PCLK/2^{0}
0001B	PCLK/2^{1}
0010B	PCLK/2^{2}
0011B	PCLK/2^{3}
0100B	PCLK/2^{4}
0101B	PCLK/2^{5}
0110B	PCLK/2^{6}
0111B	PCLK/2^{7}
1000B	PCLK/2^{8}
1001B	PCLK/2^{9}
1010B	PCLK/2^{10}
1011B	PCLK/2^{11}
1100B	PCLK/2^{12}
1101B	PCLK/2^{13}
1110B	PCLK/2^{14}
1111B	PCLK/2^{15}

These bits can only be rewritten when all counters using CK0 are stopped (TAUJ2TE.TAUJ2TEm = 0).

(2) TAUJ2 Prescaler Baud Rate Setting Register (TAUJ2BRS)

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 TAUJ2TPS.TAUJ2PRS3[3:0].

- Access This register can be read/written in 8-bit units. Writing is only possible while the counter is stopped (TAUJ2TE.TAUJ2TEm = 0).

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2BRS	TAUJ2BRS[7:0]								4000 0094H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 0	TAUJ2BRS[7:0]	Specifies the CK3_PRE clock division factor for generating CK3.

TAUJ2BRS[7:0]	CK3 clock
0000 0000B	CK3_PRE / 1
0000 0001B	CK3_PRE / 2
0000 0010B	CK3_PRE / 3
0000 0011B	CK3_PRE / 4
…	…
1111 1110B	CK3_PRE / 255
1111 1111B	CK3_PRE / 256

19.3.3 TAUJ2 Control Registers Details

(1) TAUJ2 Channel Data Register (TAUJ2CDRm)

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUJ2CMORm.TAUJ2MD[4:0].

- Access This register can be read/written in 32-bit units.
  In capture mode, only reading is possible. Write operation is ignored.
  In compare mode, reading and writing is possible.

	31 … 0	
TAUJ2CDRm	TAUJ2CDR[31:0]	Address 4000 0000H m × 4H; Initial Value 0000 0000H
R/W	R/W	

Bit Position	Bit Name	Function
31 to 0	TAUJ2CDR[31:0]	Data register for the capture/compare value

(2) TAUJ2 Channel Counter Register (TAUJ2CNTm)

This register is the channel m counter register.

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

	31 … 0	
TAUJ2CNTm	TAUJ2CNT[31:0]	Address 4000 0010H + m × 4H; Initial Value 0000 0000H or FFFF FFFFH Note
R/W	R	

Bit Position	Bit Name	Function
31 to 0	TAUJ2CNT[31:0]	32-bit counter value

Note: The initial value depends on the operating mode set by the TAUJ2 channel mode OS register. The initial value is FFFF_FFFFH in interval timer mode or one-count mode and it is 0000_0000H in other modes.
For details of the operating mode settings, see section 19.3.3(3), TAUJ2 Channel Mode OS Register (TAUJ2CMORm).

(3) TAUJ2 Channel Mode OS Register (TAUJ2CMORm)

This register controls channel m operation.

- Access This register can be read or written in 16-bit units. Writing is only possible while the counter is stopped (TAUJ2TE.TAUJ2TEm = 0).

(1/4)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2 CMORm	TAUJ2 CKS[1:0]		TAUJ2 CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2 COS[1:0]		0	TAUJ2MD[4:0]					4000 0080H + m × 4H	0000H
R/W	R/W		R/W		R/W	R/W			R/W		R	R/W						

Bit Position	Bit Name	Function
15, 14	TAUJ2CKS[1:0]	Selects the operation clock. The operation clock is used for the TAUJ2TTINm input edge detection circuit. It can also be used as the count clock depending on bits TAUJ2CMORm.CCS[1:0].
13, 12	TAUJ2CCS[1:0]	Selects the count clock for TAUJ2CNTm counter.
11	TAUJ2MAS	Specifies the channel as master or slave channel during synchronous channel operation. 0: Slave 1: Master This bit is only valid for even-numbered channels (CHm_even). For odd-numbered channels (CHm_odd), it is fixed to 0.

TAUJ2CKS[1:0]:

TAUJ2CKS1	TAUJ2CKS0	Selected Prescaler Output
0	0	CK0
0	1	CK1
1	0	CK2
1	1	CK3

TAUJ2CCS[1:0]:

TAUJ2CCS1	TAUJ2CCS0	Selected Count Clock
0	0	Prescaler output specified by TAUJ2CMORm.TAUJ2CKS[1:0]
0	1	Effective edge of TAUJ2TTINm input signal
1	0	Setting prohibited
1	1	

(2/4)

Bit Position	Bit Name	Function
10 to 8	TAUJ2STS[2:0]	Selects the external start trigger.

TAUJ2STS2	TAUJ2STS1	TAUJ2STS0	Description
0	0	0	Software trigger
0	0	1	Effective edge of the TAUJ2TTINm input signal. TAUJ2CMURm.TAUJ2TIS[1:0] specifies the effective edge.
0	1	0	Effective edge of the TAUJ2TTINm input signal is used as the start trigger, and the reverse edge is used as the stop trigger.
0	1	1	Setting prohibited
1	0	0	INT of the master channel
1	0	1	Setting prohibited
1	1	0	
1	1	1	

(3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>TAUJ2COS[1:0]</td><td>Specifies when the capture register TAUJ2CDRm and the overflow flag TAUJ2CSRm.TAUJ2OVF of channel m are updated.
These bits are only valid if channel m is in capture mode.
• Capture mode
• Capture & One Count mode
• Capture & Gate Count mode
• Count Capture mode
<table>
<tr><th>TAUJ2 COS1</th><th>TAUJ2 COS0</th><th>TAUJ2CDRm</th><th>TAUJ2CSRm.TAUJ2OVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of an effective edge of the TAUJ2TTINm input signal</td><td>Updated (cleared or set) upon detection of an effective edge of the TAUJ2TTINm input signal:
• If a counter overflow has occurred after the last effective edge detection, TAUJ2CSRm.TAUJ2OVF is set.
• If no counter overflow has occurred after the last effective edge detection, TAUJ2CSRm.TAUJ2OVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Invalid</td></tr>
<tr><td>1</td><td>0</td><td>Updated upon detection of an effective edge of the TAUJ2TTINm input signal and upon counter overflow:
• Detection of an effective edge of the TAUJ2TTINm input signal: Counter value is written to TAUJ2CDRm.
• Overflow: FFFF FFFFH is loaded to TAUJ2CDRm. Detection of the next effective edge of the TAUJ2TTINm input signal is ignored.</td><td>Invalid</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">Setting prohibited</td></tr>
</table>
</td></tr>
</table>

(4/4)

Bit Position	Bit Name	Function
5	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.
4 to 0	TAUJ2MD[4:0]	Specifies the operating mode. Settings not listed in the following table are prohibited.

TAUJ2 MD4	TAUJ2 MD3	TAUJ2 MD2	TAUJ2 MD1	TAUJ2 MD0	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	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 & One Count mode
0	1	1	1	1/0	Setting prohibited
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1	1/0	Count Capture mode
1	1	0	0	1/0	Setting prohibited
1	1	0	1	0	Capture & Gate Count mode

Mode	Role of the MD0 Bit
Interval Timer mode Capture mode Count Capture mode	Specifies whether the INTTAUJ2Im signal is output when the counter starts counting (when the start trigger is input). 0: Does not output INTTAUJ2Im. 1: Outputs INTTAUJ2Im
Event count mode	Set this bit to 0 (the INTTAUJ2Im signal is not output when the counter starts counting).
One Count mode	Enables/disables start trigger detection during counting: 0: Disabled 1: Enabled

Mode	Role of the MD0 Bit
Capture & One Count mode Capture & Gate Count mode	This bit must be set to 0. 0: Generation of INTTAUJ2Im is disabled

(4) TAUJ2 Channel Mode User Register (TAUJ2CMURm)

This register specifies the type of effective edge detection used for the TAUJ2TTINm input.

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

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2 CMURm	0	0	0	0	0	0	TAUJ2TIS[1:0]		4000 0020H + m × 4H	00H
R/W	0	0	0	0	0	0	R/W			

Bit Position	Bit Name	Function
7 to 2	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
1, 0	TAUJ2TIS[1:0]	(see table below) Edge detection for TAUJ2TTINm input signals is performed based on the prescaler output selected by TAUJ2CMORm.TAUJ2CKS[1:0].

TAUJ2TIS1	TAUJ2TIS0	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

(5) TAUJ2 Channel Status Register (TAUJ2CSRm)

This register indicates the overflow status of channel m.

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

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2 CSRm	0	0	0	0	0	0	RFU	TAUJ2OVF	4000 0030H + m × 4H	00H
R/W	0	0	0	0	0	0	R	R		

Bit Position	Bit Name	Function
7 to 2	—	Reserved. These bits are read as 0.
1	RFU	Reserved (don’t care)
0	TAUJ2OVF	Indicates the counter overflow status: 0: No overflow occurred 1: Overflow occurred This bit is only used in the following modes: • Capture mode • Capture & One Count mode • Count Capture mode • Capture & Gate Count mode The function of this bit depends on the setting of control bits TAUJ2CMORm.TAUJ2COS[1:0] In other modes, the value read is undefined.

(6) TAUJ2 Channel Status Clear Register (TAUJ2CSCm)

This register is a trigger register for clearing the overflow flag TAUJ2CSRm.TAUJ2OVF of a channel m.

- Access This register can be written in 8-bit units. It is always read as 00H.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2 CSCm	0	0	0	0	0	0	0	TAUJ2 CLOV	4000 0040H + m × 4H	00H
R/W	0	0	0	0	0	0	0	W		

Bit Position	Bit Name	Function
7 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	TAUJ2CLOV	Clears the overflow flag of channel m. Writing 1 to this bit clears the overflow flag TAUJ2CSRm.TAUJ2OVF. Writing 0 to this bit has no effect.

(7) TAUJ2 Channel Enable Status Register (TAUJ2TE)

This register indicates whether counter operation is enabled or disabled.

- Access This register indicates whether counter operation is enabled or disabled.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TE	0	0	0	0	TAUJ2 TE03	TAUJ2 TE02	TAUJ2 TE01	TAUJ2 TE00	4000 0050H	00H
R/W	0	0	0	0	R	R	R	R		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. These bits are read as 0.
3 to 0	TAUJ2TEm	These bits indicate whether counter operation for channel m is enabled or disabled. 0: Counter operation disabled 1: Counter operation enabled These bits are set to 1 when TAUJ2TSSTm (the synchronous channel start trigger signal) trigger input is detected or when TAUJ2TS.TAUJ2TSm is set to 1. Setting TAUJ2TT.TAUJ2TTm to 1 resets these bits to 0.

(8) TAUJ2 Channel Start Trigger Register (TAUJ2TS)

This register enables counter operation for each channel.

- Access This register can be written in 8-bit units. It is always read as 00H.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TS	0	0	0	0	TAUJ2 TS03	TAUJ2 TS02	TAUJ2 TS01	TAUJ2 TS00	4000 0054H	00H
R/W	0	0	0	0	W	W	W	W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2TSm	These bits enable counter operation for channel m. Writing 1 to these bits enables counter operation and sets TAUJ2TE.TAUJ2TEm = 1. Writing 0 to these bits has no effect.

(9) TAUJ2 Channel Stop Trigger Register (TAUJ2TT)

This register stops counter operation for each channel.

- Access This register can be written in 8-bit units. It is always read as 00H.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TT	0	0	0	0	TAUJ2 TT03	TAUJ2 TT02	TAUJ2 TT01	TAUJ2 TT00	4000 0058H	00H
R/W	0	0	0	0	W	W	W	W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2TTm	These bits stop counter operation of channel m. Writing 1 to these bits stops counter operation and clears TAUJ2TE.TAUJ2TEm. Writing 0 to these bits has no effect. TAUJ2CNTm, TAUJ2TO.TAUJ2TOm, and TAUJ2TTOUTm retain their values before the counter stopped.

19.3.4 TAUJ2 Simultaneous Reload Registers Details

(1) TAUJ2 Channel Reload Data Enable Register (TAUJ2RDE)

This register enables or disables simultaneous reloading of the data registers TAUJ2CDRm and TAUJ2TOLm.

- Access This register can be read/written in 8-bit units. Writing is only possible while TAUJ2TE.TAUJ2TEm is 0.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2RDE	0	0	0	0	TAUJ2 RDE03	TAUJ2 RDE02	TAUJ2 RDE01	TAUJ2 RDE00	4000 00A0H	00H
R/W	0	0	0	0	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2RDEm	These bits enable or disable simultaneous reloading of the data registers of channel m. 0: Disables simultaneous reloading. 1: Enables simultaneous reloading.

(2) TAUJ2 Channel Reload Data Mode Register (TAUJ2RDM)

This register selects the timing for generating a simultaneous reload control signal.

- Access This register can be read/written in 8-bit units. Writing is only possible while TAUJ2TE.TAUJ2TEm is 0.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2RDM	0	0	0	0	TAUJ2 RDM03	TAUJ2 RDM02	TAUJ2 RDM01	TAUJ2 RDM00	4000 00A4H	00H
R/W	0	0	0	0	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2RDMm	These bits select the timing for generating a simultaneous reload control signal. 0: When the master channel counter starts counting 1: No function (setting prohibited) The setting of these bits is only applied when TAUJ2RDE.TAUJ2RDEm = 1.

(3) TAUJ2 Channel Reload Data Trigger Register (TAUJ2RDT)

This register specifies the channel for simultaneous reloading when INTTAUJ2Im is generated.

- Access This register can be written in 8-bit units. It is always read as 00H.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2RDT	0	0	0	0	TAUJ2 RDT03	TAUJ2 RDT02	TAUJ2 RDT01	TAUJ2 RDT00	4000 0068H	00H
R/W	0	0	0	0	W	W	W	W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2RDTm	These bits specify the trigger for the channel for simultaneous reloading when a simultaneous reload trigger is generated. When 1 is written to these bits, the specified channel for simultaneous reloading is pending and the simultaneous reload pending flag (TAUJ2RSFm) is set to 1. The specified channel awaits the simultaneous reload trigger. Writing 0 to these bits has no effect.

(4) TAUJ2 Channel Reload Status Register (TAUJ2RSF)

This flag register indicates the state of simultaneous reloading.

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

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2RSF	0	0	0	0	TAUJ2 RSF03	TAUJ2 RSF02	TAUJ2 RSF01	TAUJ2 RSF00	4000 006CH	00H
R/W	0	0	0	0	R	R	R	R		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2RSFm	These bits indicate the state of simultaneous reloading. 0: Indicates the completion of simultaneous reloading when a simultaneous reload trigger is generated. 1: Indicates that the channel awaits the simultaneous reload trigger when simultaneous reloading is pending (TAUJ2RDTm = 1).

19.3.5 TAUJ2 Output Registers Details

(1) TAUJ2 Channel Output Enable Register (TAUJ2TOE)

This register enables or disables independent channel output mode controlled by software.

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

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TOE	0	0	0	0	TAUJ2 TOE03	TAUJ2 TOE02	TAUJ2 TOE01	TAUJ2 TOE00	4000 0060H	00H
R/W	0	0	0	0	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2TOEm	These bits enable or disable channel output by the timer. 0: Disables channel output. 1: Enables channel output. The TAUJ2TOm bits can only be written while timer output of a channel is disabled (TAUJ2TOEm = 0).

(a) TAUJ2TTOUTm pin output control

- TAUJ2TOE.TAUJ2TOEm = 0
  The TAUJ2TOm bits can only be written while timer output of a channel is disabled (TAUJ2TOEm = 0).
- TAUJ2TOE.TAUJ2TOEm = 1
  Output by TAUJ2TTOUTm in counting of a channel.

(b) Setting to specify channel output

Make settings while timer output is disabled (TAUJ2TOE.TAUJ2TOEm = 0)

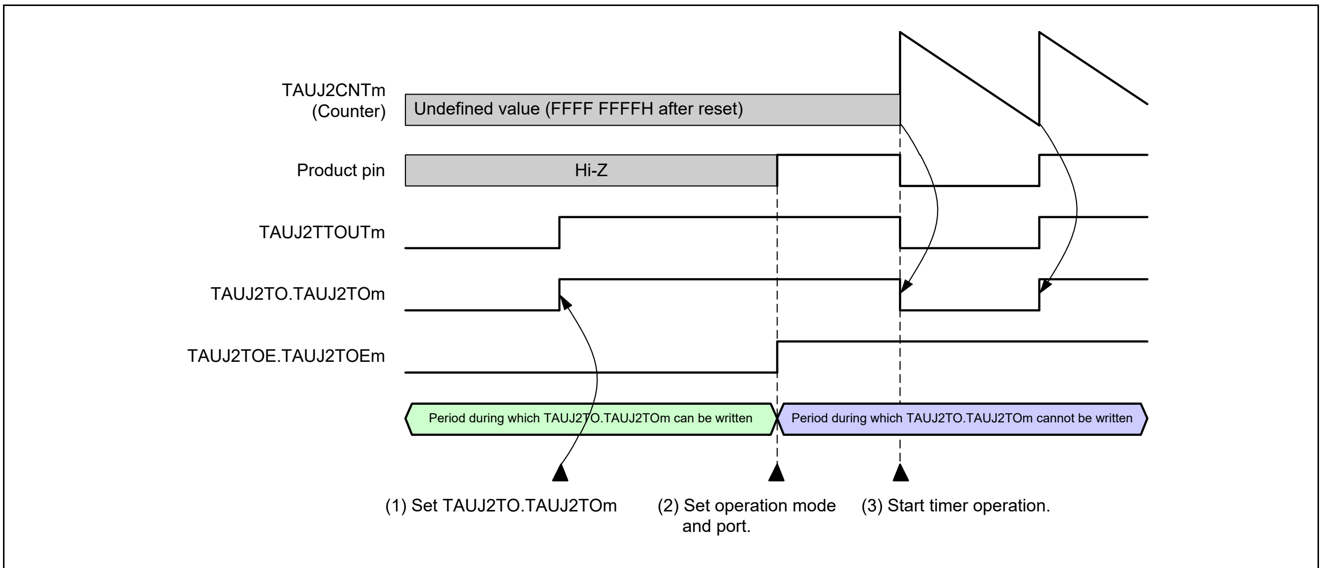


Figure 19.2 General Procedure for Specifying TAUJ2TTOUTm Channel Output Mode

(2) TAUJ2 Channel Output Register (TAUJ2TO)

This register specifies and reads the level of TAUJ2TTOUTm.

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

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TO	0	0	0	0	TAUJ2 TO03	TAUJ2 TO02	TAUJ2 TO01	TAUJ2 TO00	4000 005CH	00H
R/W	0	0	0	0	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2TOm	These bits specify and read the level of TAUJ2TTOUTm. 0: Low level 1: High level These bits can be written while TAUJ2TOE.TAUJ2TOEm = 0.

(3) TAUJ2 Channel Output Mode Register (TAUJ2TOM)

This register specifies the output mode of each channel.

- Access This register can be read/written in 8-bit units. Writing is only possible while the counter is stopped (TAUJ2TE.TAUJ2TEm = 0).

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TOM	0	0	0	0	TAUJ2 TOM03	TAUJ2 TOM02	TAUJ2 TOM01	TAUJ2 TOM00	4000 0098H	00H
R/W	0	0	0	0	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2TOMm	These bits specify the channel output mode. 0: Independent channel operation 1: Synchronous channel operation The output mode depends on the setting of the channel output control bits.

(4) TAUJ2 Channel Output Configuration Register (TAUJ2TOC)

This register specifies the output mode of each channel in combination with TAUJ2TOMm.

- Access This register can be read/written in 8-bit units. Writing is only possible while the counter is stopped (TAUJ2TE.TAUJ2TEm = 0).

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TOC	0	0	0	0	TAUJ2 TOC03	TAUJ2 TOC02	TAUJ2 TOC01	TAUJ2 TOC00	4000 009CH	00H
R/W	0	0	0	0	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2TOCm	These bits specify the output mode. 0: Independent timer output is disabled. 1: Setting prohibited • When TAUJ2nTOM.TAUJ2TOMm =0 Toggling proceeds when NTTAUJ2Im occurs. • When TAUJ2nTOM.TAUJ2TOMm =1 Set when INT occurs on the master channel and reset when INTTAUJ2Im occurs on the slave channel.

(5) TAUJ2 Channel Output Level Register (TAUJ2TOL)

This register specifies the output logic of the channel output bit (TAUJ2TO.TAUJ2TOm).

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

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUJ2TOL	0	0	0	0	TAUJ2 TOL03	TAUJ2 TOL02	TAUJ2 TOL01	TAUJ2 TOL00	4000 0064H	00H
R/W	0	0	0	0	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	TAUJ2TOLm	These bits specify the output logic of the channel m output bit (TAUJ2TO.TAUJ2TOm). 0: Positive logic (active high) 1: Inverted logic (active low) These bits are valid when TAUJ2TOE.TAUJ2TOEm = 1, TAUJ2nTOM.TAUJ2TOMm=1 and TAUJ2TOC.TAUJ2TOCm=0.

19.4 General Operating Procedure

The following lists the general operation procedure for the TAUJ2:

After release from the reset state, the operation of each channel is stopped. Writing to the registers is enabled when clock supply is started. The control register of TAUJ2TTOUTm is initialized and outputs the low level.

1. Set the TAUJ2TPS and TAUJ2BRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUJ2 function:
   - Set the operating mode (TAUJ2CMORm)
   - Set the channel output mode (TAUJ2TOE, TAUJ2TOM, etc)
   - Set any other control bits
3. Enable the counter by setting the TAUJ2TS.TAUJ2TSm bit to 1.
   The counter starts counting according to the bit settings.
4. To stop counting, set the TAUJ2TT.TAUJ2TTm bit to 1 to stop the function.

Remark: For details of the operation of the individual functions and register settings, see the description of the individual functions.

19.5 Overview of Synchronous Channel Operation

TAUJ2 consists of more than one channel, and handles independent channel operation whereby individual channels operate independently and synchronous channel operation whereby multiple channels operate in combination.
Independent channel operation can be used by any channel independently of all other channels.
Synchronous channel operation is realized by combining master and slave channels. Several rules apply to the settings of channels. The details of the rules are given below.

19.5.1 Basic Rules of Synchronous Channel Operation

(1) Only even-numbered channels (CH0, CH2) can be set as master channels.

(2) Any channel except CH0 can be set as a slave channel.

(3) Only channels lower than the master channel can be set as slave channels.

(4) Multiple slave channels can be set for one master channel.
Example: If CH0 is a master channel, CH1, CH2 and CH3 can be set as slave channels.

(5) 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 a slave channel for CH0, but CH3 cannot.

(6) The same operation clock should be set for the master channel and the synchronized slave channel. This is achieved by setting the same value in the TAUJnCMORm.TAUJnCKS[1:0] bits of the master and slave channels.

(7) Master channels can transfer INTTAUJ2lm and start trigger to lower channels.

(8) Slave channels can use INTTAUJ2lm and start trigger of the master channels but cannot transfer their INTTAUJ2lm and start trigger to the lower channels.

(9) A master channel cannot use INTTAUJ2lm and start trigger of the upper master channels.

(10) To simultaneously start the channels for synchronous operation, the TAUJ2TS.TAUJ2TSm bits for the target channels must be set at the same time.

(11) To simultaneously stop the channels for synchronous operation, the TAUJ2TT.TAUJ2TTm bits for the target channels must be set at the same time.

The basic concepts of usage of master and slave channels and operation clocks are illustrated in the following figure.

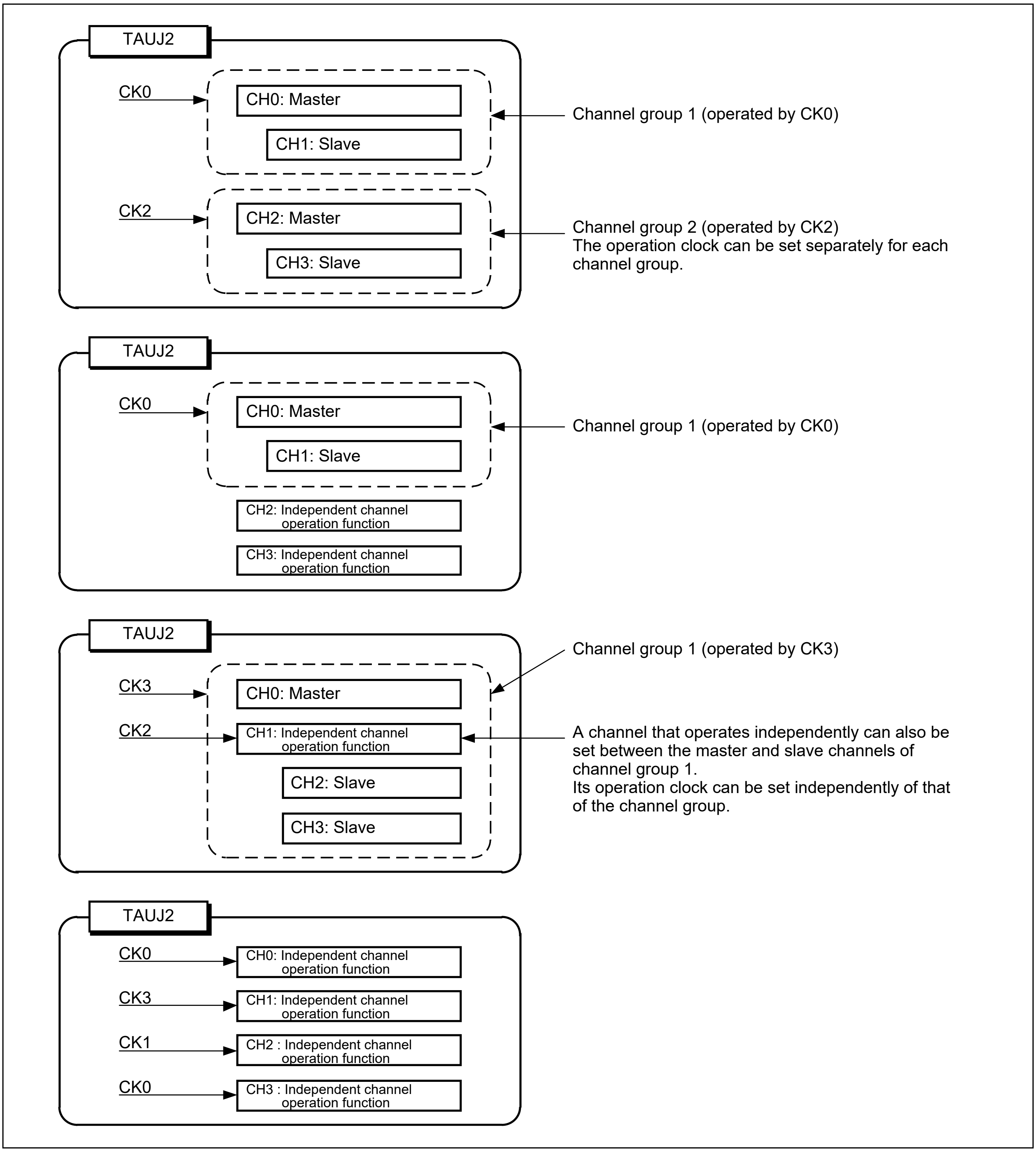


Figure 19.3 Grouping of the Channels and Assignment of Operation Clocks

19.6 Simultaneous Reloading

19.6.1 Outline of Operation

Simultaneous reloading refers to modifying the values of the data registers (TAUJ2CDRm) and output active level setting registers (TAUJ2TOL.TAUJ2TOLm) of the target channels all together. The new value does not affect the counter operation or the output signal until simultaneous reload trigger is enabled.

In TAUJ2, simultaneous reloading can proceed at two times.

- Start timing of a master channel
- Timing of interrupt output by the channel higher than the master channel.

19.6.2 How to Control Simultaneous Reloading (in Case of PWM Output)

The following figure shows the general procedure for simultaneous reloading. In TAUJ2, synchronous channel operation is only supported for PWM output.

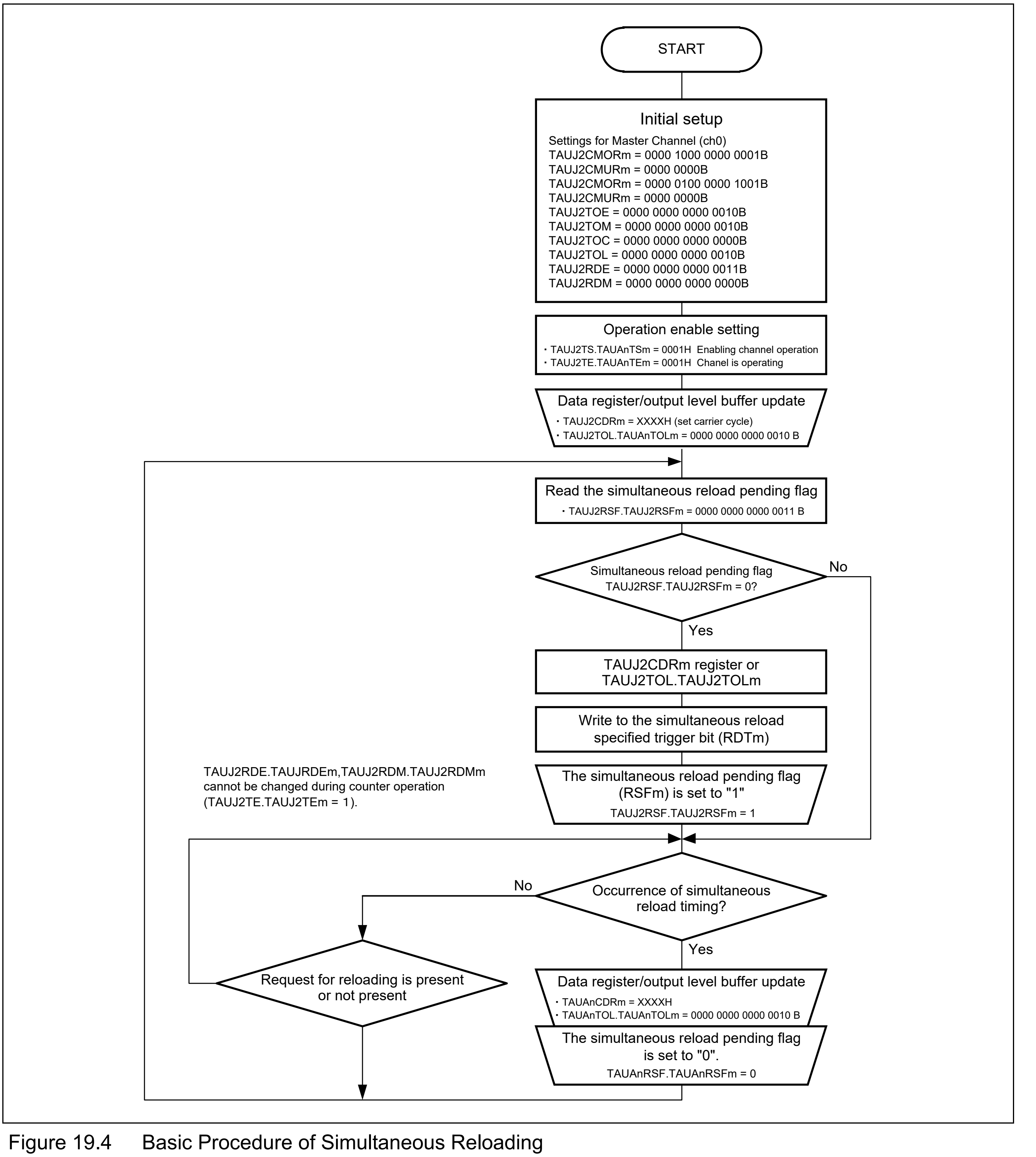


Figure 19.4 Basic Procedure of Simultaneous Reloading

19.6.3 Other General Rules of Simultaneous Reloading

(1) TAUJ2RDE.TAUJ2RDE for channels to be used is set to 1 to enable simultaneous reload operation.

(2) When TAUJ2TE.TAUJ2TEm = 0, set the following bits.
- TAUJ2RDE.TAUJ2RDEm
- TAUJ2RDM.TAUJ2RDMm

(3) Targets of simultaneous reloading in synchronous operation are TAUJ2CDRm and TAUJ2TOL.TAUJ2TOLm.

(4) The function which allows TAUJ2TOL.TAUJ2TOLm to be rewritten during operation is only PWM output. In other synchronous operations, rewriting is only possible at the timing of initial settings.

Cautions 1. Simultaneous reloading cannot be used in independent channel operation.

2. If TAUJ2RDT.TAUJ2RDTm is not set to 1, simultaneous reloading does not proceed.

3. When TAUJ2RDT.TAUJ2RDTm is set to 1, TAUJ2RSF.TAUJ2RSFm is set to 1 and generation of a simultaneous reload trigger leads to TAUJ2RSF.TAUJ2RSFm to be cleared; accordingly, before changing the value of the register, read TAUJ2RSF.TAUJ2RSFm and confirm that its value is 0.

19.7 Independent Channel Operation

The following describes the individual functions of independent channel operation.

- 19.7.1 "Interval Timer"
- 19.7.2 "TAUJ2TTINm Input Interval Timer"
- 19.7.3 "External Event Counting"
- 19.7.4 "Delay Counting"
- 19.7.5 "TAUJ2TTINm Input Pulse Interval Measurement"
- 19.7.6 "TAUJ2TTINm Input Signal Width Measurement"
- 19.7.7 "TAUJ2TTINm Input Position Detection"

19.7.1 Interval Timer

(1) Overview

This function generates a timer interrupt (INTTAUJ2Im) when the TAUJ2CDRm channel data register and TAUJ2CNTm channel counter register values match. When an interrupt occurs, the TAUJ2TTOUTm signal toggles and a square wave is output.

(2) Block Diagram

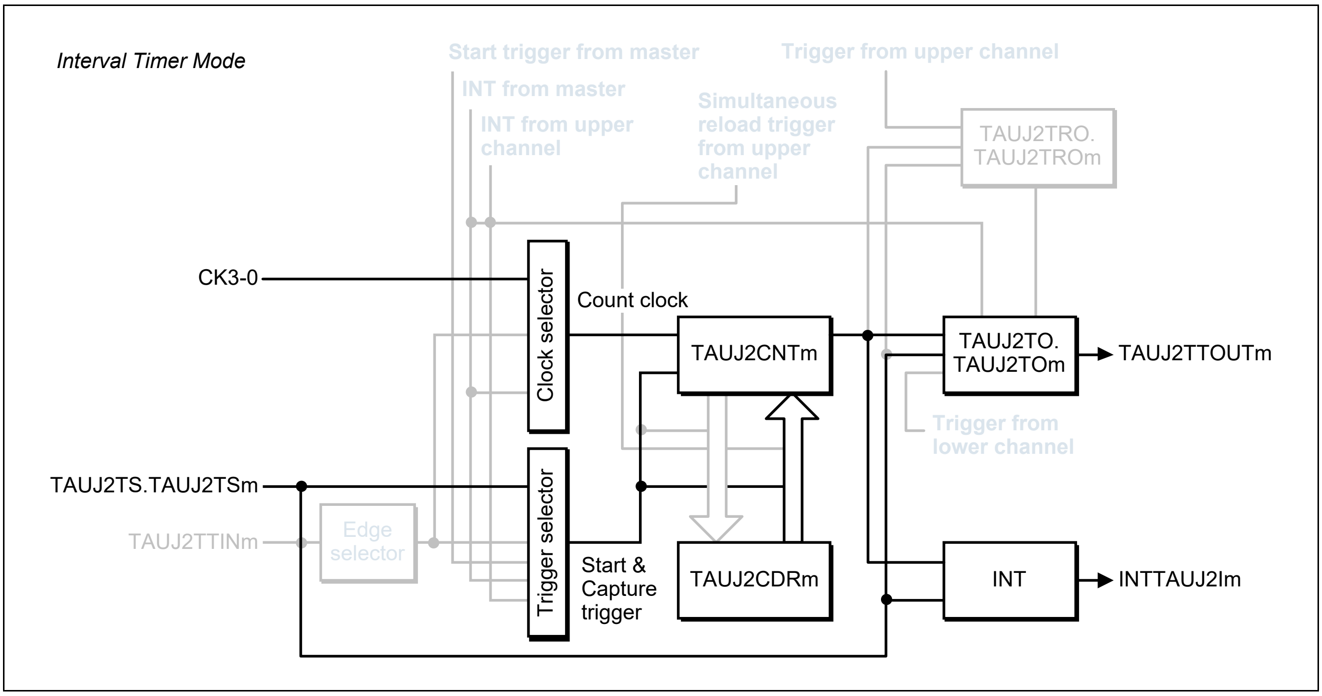


Figure 19.5 Block Diagram of Interval Timer

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- INTTAUJ2Im is generated at the start of operation (TAUJ2CMORm.TAUJ2MD0 = 1)

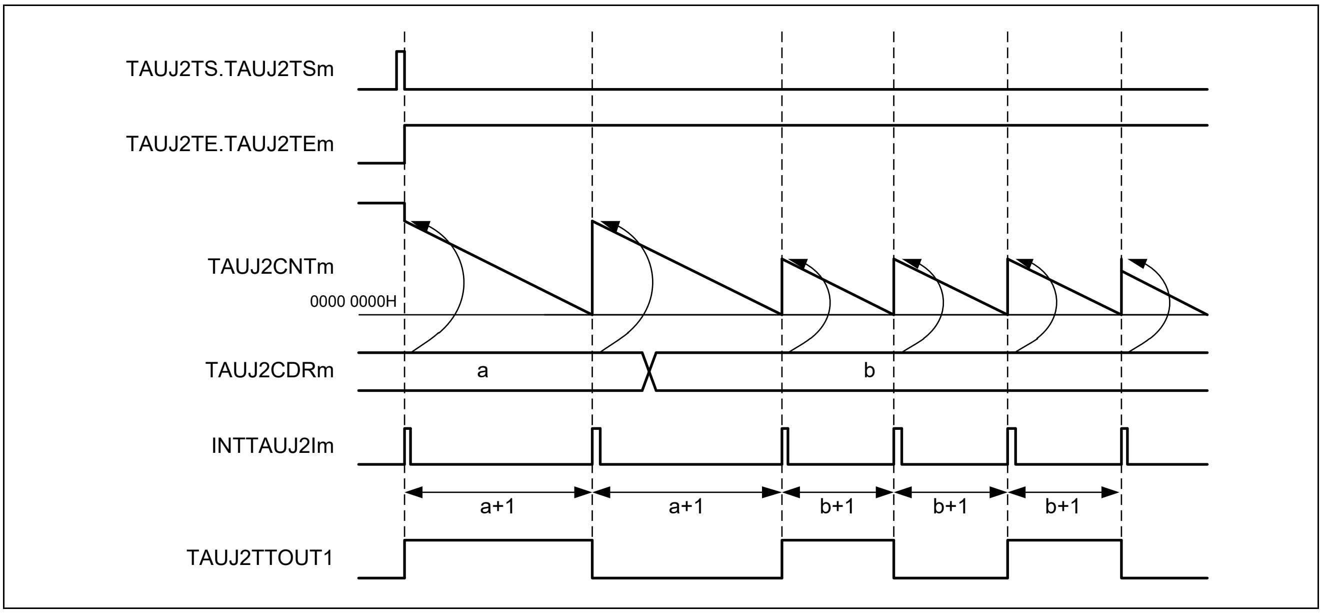


Figure 19.6 General Timing Diagram of Interval Timer

(4) Equations

INTTAUJ2Im cycle = count clock cycle × (TAUJ2CDRm + 1)

TAUJ2TTOUTm square wave cycle = count clock cycle × (TAUJ2CDRm + 1) × 2

(5) Register Settings

(a) TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.5 TAUJ2CMORm Settings for Interval Timer

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock that suits the application.
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	0: Independent operation
TAUJ2STS[2:0]	These bits select the external start trigger. 000: Software trigger
TAUJ2COS[1:0]	00: Not used (initial value)
TAUJ2MD[4:1]	These bits select the operating mode. 0000: Interval timer mode
TAUJ2MD0	This bit specifies whether an INTTAUJ2Im interrupt is generated when counting starts. 0: INTTAUJ2Im prohibited (TAUJ2TTOUTm output is not toggled) 1: INTTAUJ2Im permitted (TAUJ2TTOUTm output is toggled)

(b) TAUJ2CMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—														TAUJ2TIS[1:0]	

Table 19.6 TAUJ2CMURm Settings for Interval Timer

Bit Name	Setting
TAUJ2TIS[1:0]	00: Not used (initial value)

(c) Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with the interval timer. Therefore, these registers must be set to 0.

Table 19.7 Simultaneous Reload Settings for Interval Timer

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Set to 0 since this disables simultaneous reloading of channel m.
TAUJ2RDM.TAUJ2RDMm	0: Not used

(d) Channel output mode setting

Table 19.8 Control Bit Settings for Independent Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	Enables or disables TAUJ2TOm output operation 1: Operation enabled
TAUJ2TOM.TAUJ2TOMm	Specifies independent or synchronous channel operation. 0: Independent channel operation
TAUJ2TOC.TAUJ2TOCm	Specifies the operating mode of TAUJ2TOMm output by the channel. The setting of this bit depends on the setting of TAUJ2TOM.TAUJ2TOMm 0: Toggle mode (TAUJ2TOM.TAUJ2TOMm = 0)
TAUJ2TOL.TAUJ2TOLm	0: Setting is invalid in toggle mode (initial value).

(6) Operating Procedure for Interval Timer

Table 19.9 Operating Procedure

		Operation	Status of TAUJ2
	Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register. • Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output. • Set the interval time in the TAUJ2CDRm register. • Set the output level in the TAUJ2TOm register.	Channel operation is stopped.
Restart →	Start Operation	Set TAUJ2TS.TAUJ2TSm to 1. TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is set to 1 and the counter starts. The TAUJ2CDRm value is updated in TAUJ2CNTm. When TAUJ2CMORm.TAUJ2MD0 = 1: INTTAUJ2Im is generated and TAUJ2TTOUTm output is toggled. When TAUJ2CMORm.TAUJ2MD0 = 0: INTTAUJ2Im is not generated and TAUJ2TTOUTm output is not toggled.
	During Operation	Register whose value can be changed at any timing • TAUJ2CDRm Register which is readable at any timing • TAUJ2CNTm	TAUJ2CNTm counts down. When the counter reaches 0000H: • The TAUJ2CDRm value is updated in TAUJ2CNTm, INTTAUJ2Im is generated, INTTAUJ2Im is generated, and TAUJ2TTOUTm output is toggled. The counter continues counting again.
	Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm and TAUJ2TTOUTm stop and retain their current values.

(7) Specific Timing Diagrams

(a) Count clock = PCLK/2, TAUJ2CDRm = 0000 0000H, TAUJ2CMORm.TAUJ2MD0 = 1

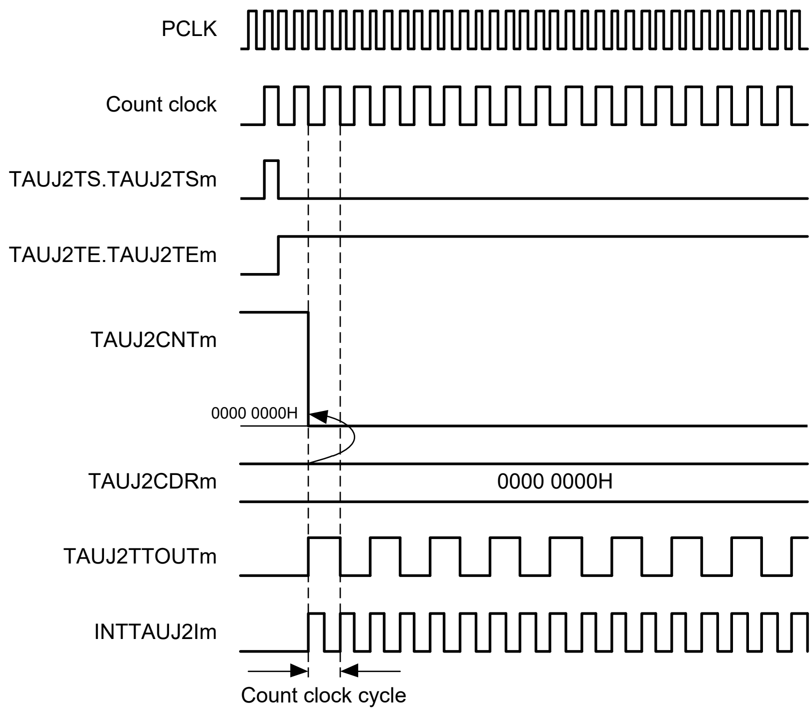


Figure 19.7 Count Clock = PCLK/2

- If TAUJ2CDRm = 0000 0000H and the count clock = PCLK/2, the TAUJ2CDRm value is updated in TAUJ2CNTm every count clock, meaning that TAUJ2CNTm is always 0000 0000H.
- INTTAUJ2Im is generated every count clock, resulting in TAUJ2TTOUTm toggling every count clock.

(b) Count clock = PCLK, TAUJ2CDRm = 0000 0000H, TAUJ2CMORm.TAUJ2MD0 = 1

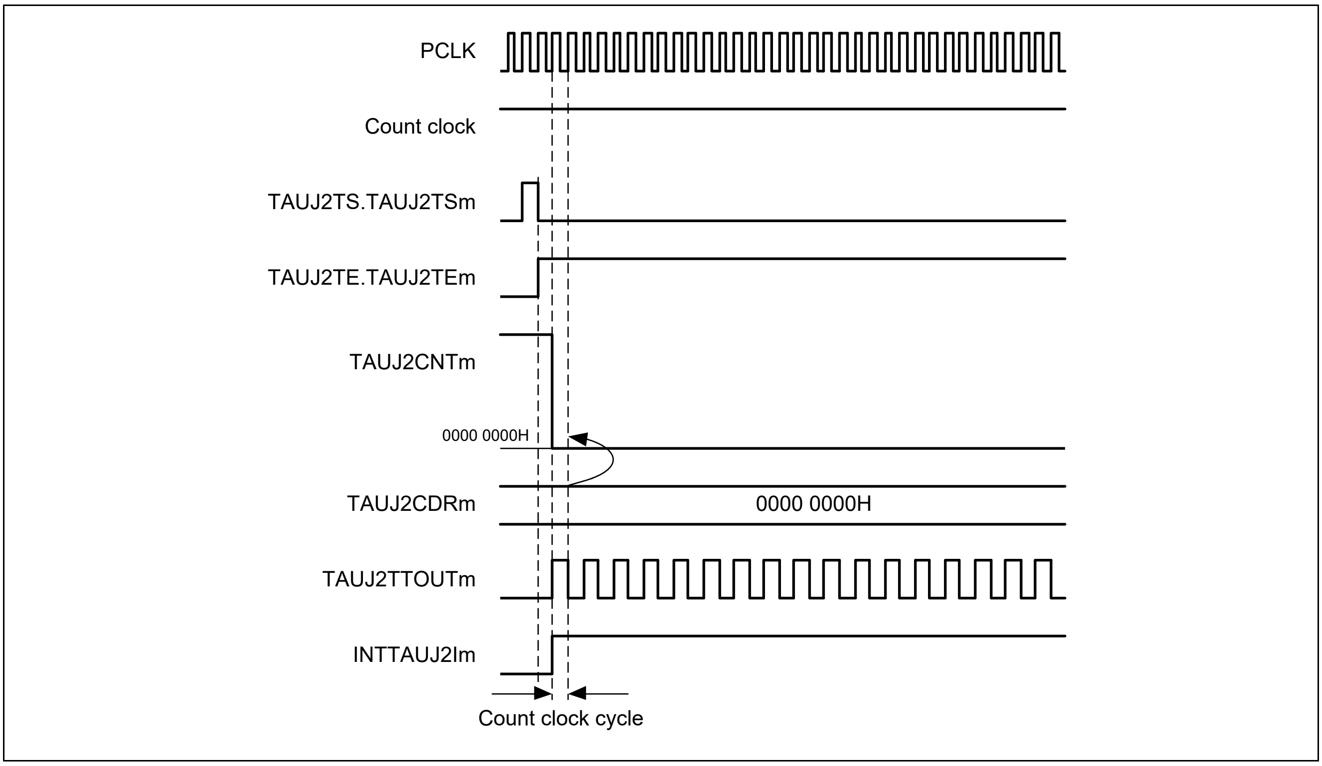


Figure 19.8 Count Clock = PCLK

- If TAUJ2CDRm = 0000 0000H and the counter clock = PCLK, the TAUJ2CDRm value is updated in TAUJ2CNTm every counter clock, meaning that TAUJ2CNTm is always 0000 0000H.
- INTTAUJ2Im is generated continuously, resulting in TAUJ2TTOUTm toggling every counter clock.

Caution: When the counter clock is PCLK, the interrupt request signal INTTAUJ2Im is fixed at the high level from the start of counter operation to stopping of the operation.
Therefore, INTTAUJ2Im interrupt output cannot be used when TAUJ2CDRm = 0000H.
However, timer (TAUJ2TTOUTm) output can be used. If timer output toggle mode is used for timer (TAUJ2TTOUTm) output, output is toggled every counter clock.

(c) Operation stop and restart

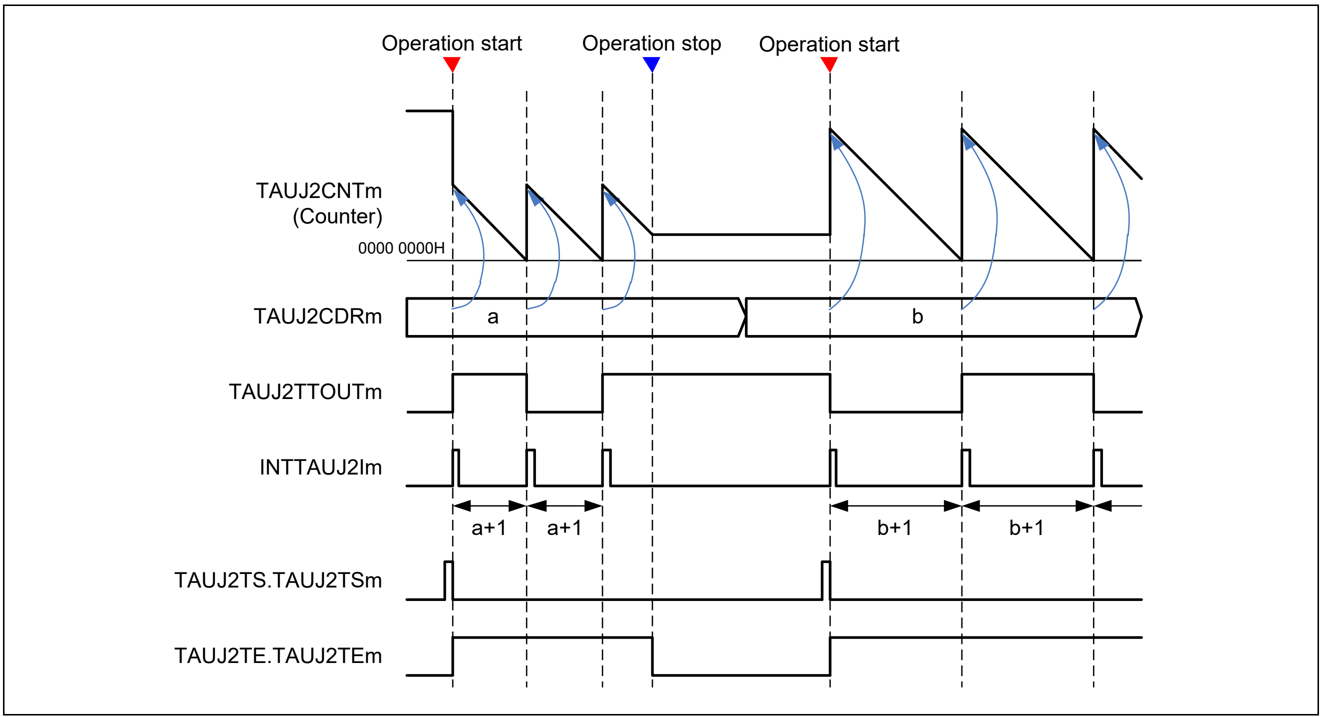


Figure 19.9 Operation Stop and Restart (TAUJ2CMORm.TAUJ2MD0 = 1)

- The counter can be stopped by setting TAUJ2TT.TAUJ2TTm to 1, which in turn sets TAUJ2TE.TAUJ2TEm to 0.
- TAUJ2CNTm and TAUJ2TTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJ2TS.TAUJ2TSm to 1.

(d) Forced restart

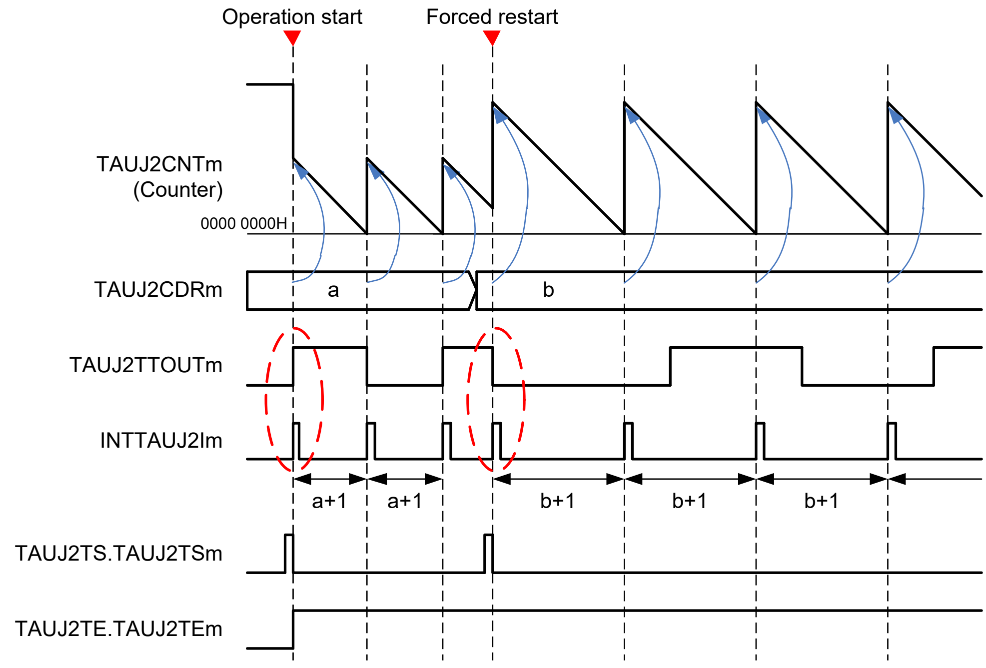


Figure 19.10 Forced Restart Operation, TAUJ2CMORm.TAUJ2MD0 = 1

- The counter can be forcibly restarted by setting TAUJ2TS.TAUJ2TSm to 1 during operation. When operation restarts, the TAUJ2CDRm register value is updated in the TAUJ2CNTm register and the counter starts.
- When the TAUJ2CMORm.TAUJ2MD0 bit is set to 1, the first interrupt after the start or restart of operation is generated and TAUJ2TTOUTm is toggled.

19.7.2 TAUJ2TTINm Input Interval Timer

(1) Overview

This function is used as a reference timer for generating timer interrupts (INTTAUJ2Im) at regular intervals or when a valid TAUJ2TTINm input edge is detected. When an interrupt is generated, the TAUJ2TTOUTm signal toggles, resulting in a square wave.

The type of edge for use as an effective trigger is selected from rising edges, falling edges or both (rising and falling) edges.

(2) Block Diagram

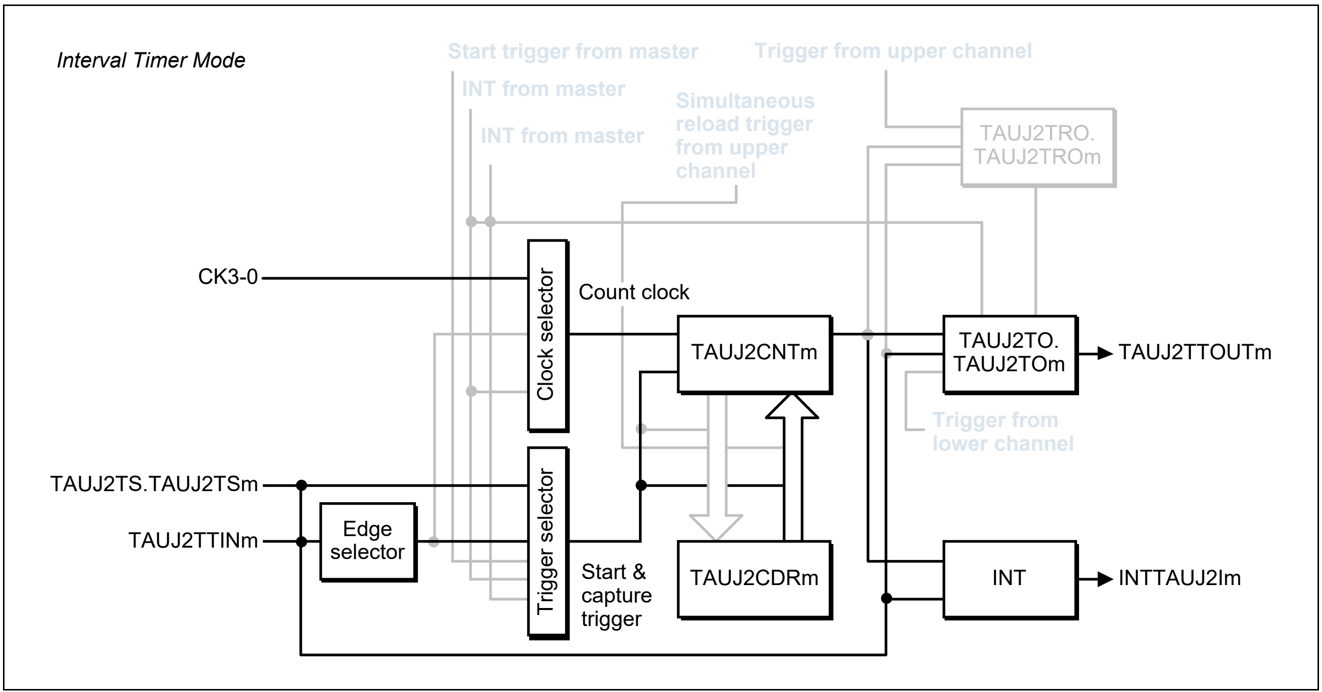


Figure 19.11 TAUJ2TTINm Block Diagram of TAUJ2TTINm Input Interval Timer

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- INTTAUJ2Im is generated at the start of operation (TAUJ2CMORm.TAUJ2MD0 = 1).
- Rising edge detection (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B)

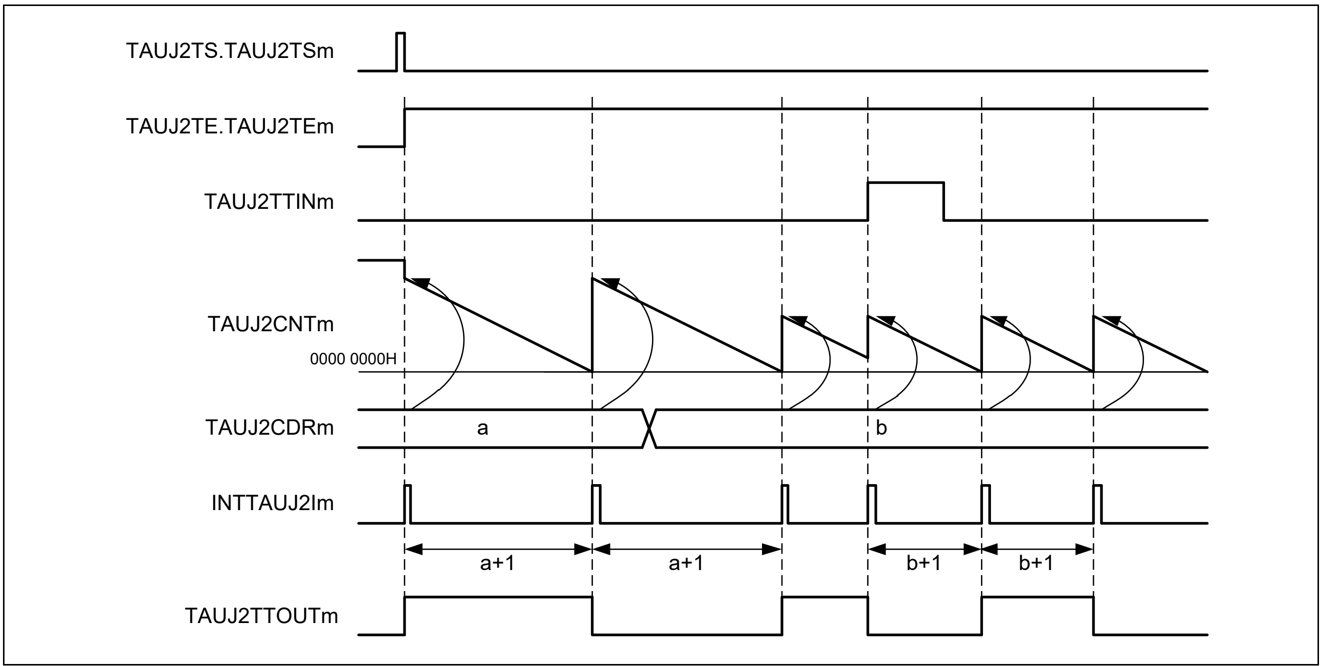


Figure 19.12 TAUJ2TTINm General Timing Diagram of TAUJ2TTINm Input Interval Timer

(4) Equations

INTTAUJ2Im cycle = count clock cycle × (TAUJ2CDRm + 1)

TAUJ2TTOUTm square wave cycle = count clock cycle × (TAUJ2CDRm + 1) × 2

(5) Register Settings

(a) TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.10 TAUJ2CMORm Settings

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock that suits the application.
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	0: Independent operation
TAUJ2STS[2:0]	These bits select the external start trigger. 001: Effective TAUJ2TTINm input edge signal is used as the external start trigger.
TAUJ2COS[1:0]	00: Not used (initial value)
TAUJ2MD[4:1]	These bits select the operating mode. 0000: Interval timer mode
TAUJ2MD0	This bit specifies whether an INTTAUJ2Im interrupt is generated when counting starts. 0: INTTAUJ2Im prohibited (TAUJ2TTOUTm output is not toggled) 1: INTTAUJ2Im permitted (TAUJ2TTOUTm output is toggled)

(b) TAUJ2CMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—														TAUJ2TIS[1:0]	

Table 19.11 TAUJ2CMURm Settings for TAUJ2TTINm Input Interval Timer

Bit Name	Setting
TAUJ2TIS[1:0]	These bits select the effective edge of the externally input signal. 00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (measurement of the width at low level) Select the effective edge to suit the application.

(c) Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with the TAUJ2TTINm input interval timer. Therefore, these registers must be set to 0.

Table 19.12 Simultaneous Reload Settings for TAUJ2TTINm Input Interval Timer

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Set to 0 since this disables simultaneous reloading of channel m.
TAUJ2RDM.TAUJ2RDMm	0: Not used (initial value)

(d) Register settings for channel output

Table 19.13 Control Bit Settings for Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	Enables or disables TAUJ2TOm output operation by counting. 1: Operation enabled
TAUJ2TOM.TAUJ2TOMm	Specifies independent or synchronous channel operation. 0: Independent channel output
TAUJ2TOC.TAUJ2TOCm	Specifies the operating mode for channel TAUJ2TOMm output. The setting of this bit depends on the setting of TAUJ2TOM.TAUJ2TOMm. 0: Toggle mode
TAUJ2TOL.TAUJ2TOLm	0: Setting has no effect in toggle mode (initial value).

(6) Operating Procedure for TAUJ2TTINm Input Interval Timer

Table 19.14 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register. • Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output. • Set the interval time in the TAUJ2CDRm register. • Set the output level in the TAUJ2TOm register.	Channel operation is stopped.
Start Operation	Set TAUJ2TS.TAUJ2TSm to 1. TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is set to 1 and the counter starts. The TAUJ2CDRm value is updated in TAUJ2CNTm. When TAUJ2CMORm.TAUJ2MD0 = 1: INTTAUJ2Im is generated and TAUJ2TTOUTm output is toggled. When TAUJ2CMORm.TAUJ2MD0 = 0: INTTAUJ2Im is not generated and TAUJ2TTOUTm output is not toggled.
During Operation	Edge detection by TAUJ2TTINm input Registers whose value can be changed at any time • TAUJ2CMURm.TAUJ2TIS[1:0] bits • TAUJ2CDRm register Register which is readable at any timing • TAUJ2CNTm register	TAUJ2CNTm counts down. When the counter reaches 0000 0000H or an effective edge of the TAUJ2TTINm input signal: • The TAUJ2CDRm value is updated in TAUJ2CNTm, INTTAUJ2Im is generated, INTTAUJ2Im is generated, and TAUJ2TTOUTm output is toggled. The counter continues counting again.
Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm and TAUJ2TTOUTm stop and retain their current values.

Restart (Stop Operation → Start Operation)

(7) Specific Timing Diagrams

For the operation of section 19.7.1, Interval Timer, the counter can also be restarted by an effective TAUJ2TTINm input edge.

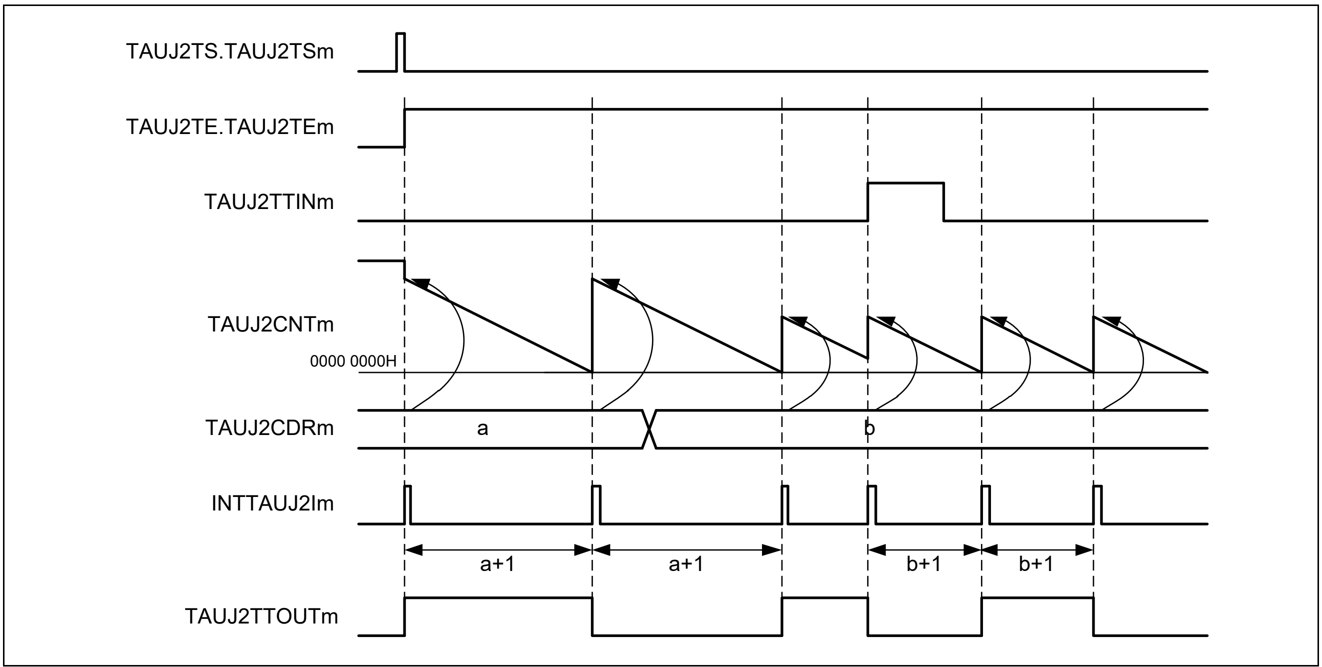


Figure 19.13 Counter Triggered by Rising TAUJ2TTINm Input Edge (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B), TAUJ2CMORm.TAUJ2MD0 = 1

- If an effective TAUJ2TTINm input edge is detected, an interrupt is generated which causes TAUJ2TTOUTm to toggle.

19.7.3 External Event Counting

(1) Overview

(a) Summary

This function is used as an event timer, which generates an interrupt (INTTAUJ2Im) when the specified number of effective edges of the TAUJ2TTINm input signal are detected.

(b) Prerequisites

- The operating mode should be set to event count mode (see Table 19.15, Contents of the TAUJ2CMORm Register for External Event Counting).
- TAUJ2TTOUTm is not used with this function.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUJ2TS.TAUJ2TSm) to 1. This in turn sets TAUJ2TE.TAUJ2TEm = 1, enabling counter operation. When the counter starts, the current value of TAUJ2CDRm is loaded into TAUJ2CNTm.

When an effective TAUJ2TTINm input edge is detected, the value of TAUJ2CNTm decrements by 1. TAUJ2CNTm retains this value until an effective TAUJ2TTINm input edge is detected or the counter is restarted.

When effective edges are detected (TAUJ2CDRm + 1) times, INTTAUJ2Im is generated. TAUJ2CNTm the loads the TAUJ2CDRm value and subsequently continues to operate.

The counter can be stopped by setting TAUJ2TT.TAUJ2TTm to 1. This in turn sets TAUJ2TE.TAUJ2TEm to 0. The counter can be restarted by setting TAUJ2TS.TAUJ2TSm to 1. The counter can also be restarted without stopping it first (forced start) by setting TAUJ2TS.TAUJ2TSm to 1 during operation.

The value of TAUJ2CDRm can be rewritten at any time, and the changed value of TAUJ2CDRm is applied the next time the counter starts to count down.

(d) Conditions

The type of edge for use as a trigger is specified by the TAUJ2CMURm.TAUJ2TIS[1:0] bits.

- When TAUJ2CMURm.TAUJ2TIS[1:0] = 00B, falling edges trigger the counter..
- When TAUJ2CMURm.TAUJ2TIS[1:0] = 01B, rising edges trigger the counter.
- When TAUJ2CMURm.TAUJ2TIS[1:0] = 10B, rising and falling edges trigger the counter.

(2) Equations

Number of effective edges detected before INTTAUJ2Im is generated = TAUJ2CDRm + 1

(3) Block Diagram and General Timing Diagram

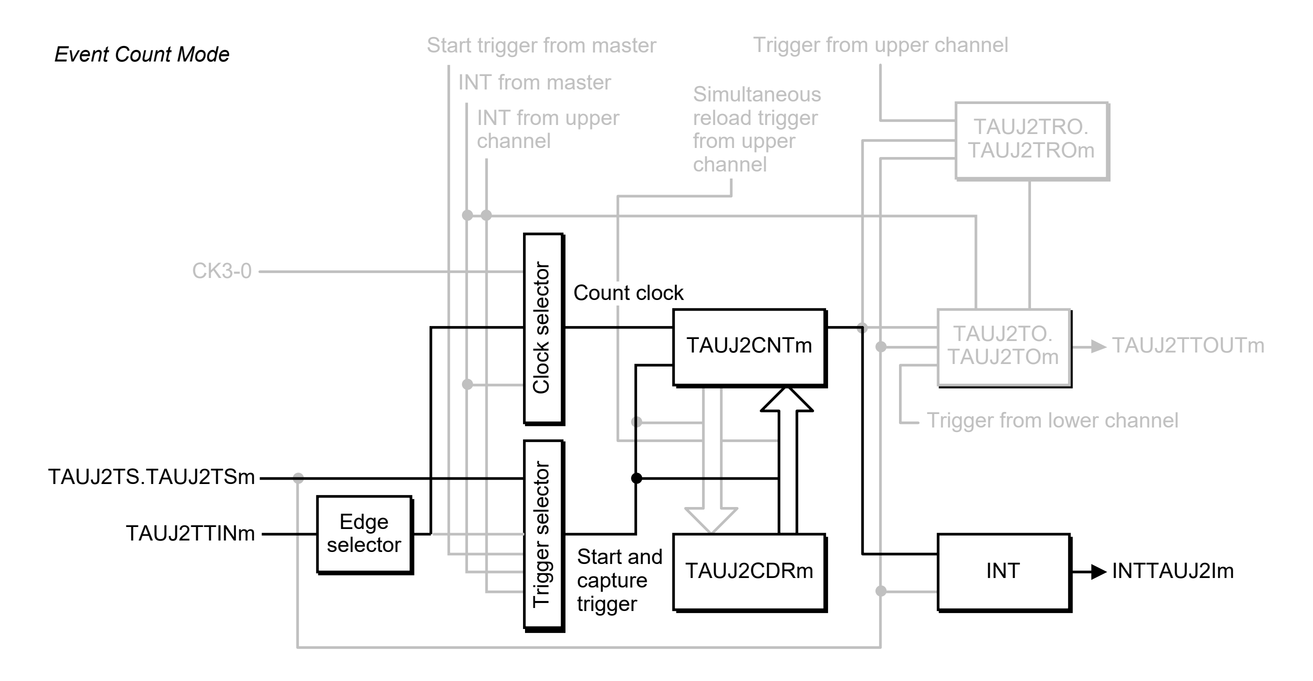


Figure 19.14 Block Diagram of External Event Counting

The following settings apply to the general timing diagram.

- Detection of rising edges (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B)

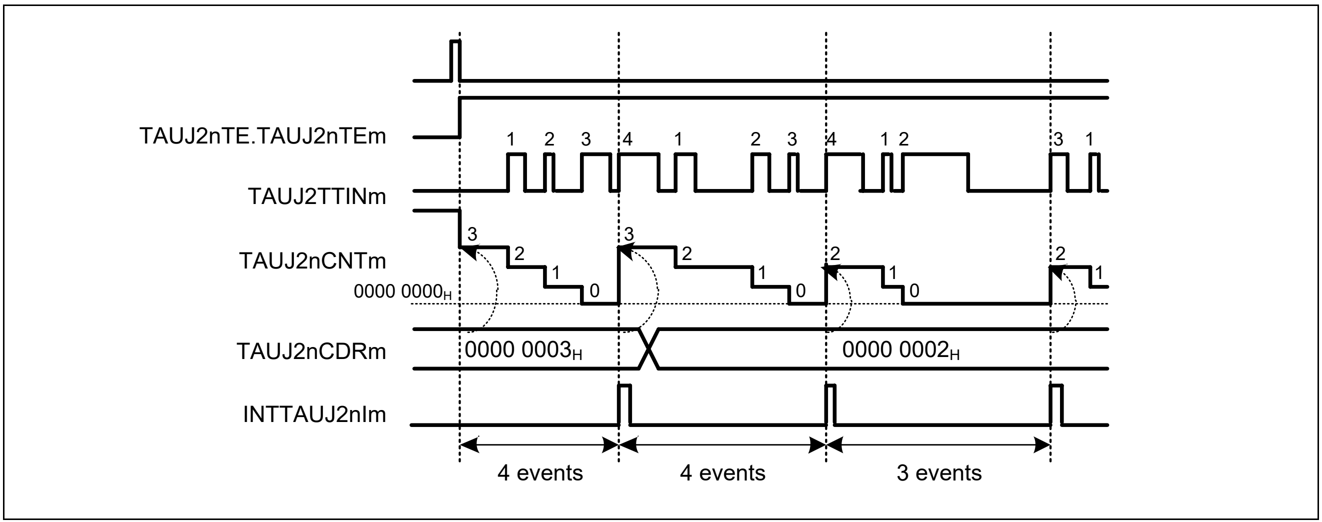


Figure 19.15 General Timing Diagram of External Event Counting

(4) Register Settings

(a) TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS [1:0]		TAUJ2CCS [1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS [1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.15 Contents of the TAUJ2CMORm Register for External Event Counting

Bit Position	Bit Name	Function
15, 14	TAUJ2CKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJ2CCS[1:0]	01: Effective TAUJ2TTINm input edge is used as a counter clock.
11	TAUJ2MAS	0: Independent operation. Set to 0.
10 to 8	TAUJ2STS[2:0]	000: Trigger the counter using software.
7, 6	TAUJ2COS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is read. When writing to this bit, write the value after reset.
4 to 1	TAUJ2MD[4:1]	0011: Event count mode.
0	TAUJ2MD0	0: INTTAUJ2Im not generated at the beginning of operation.

(b) TAUJ2CMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUJ2TIS[1:0]	

Table 19.16 Contents of the TAUJ2CMURm Register for External Event Counting

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is read. When writing to these bits, write the value after reset.
1, 0	TAUJ2TIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of falling and rising edges

(c) Channel output mode

Channel output mode is not used with this function.

(d) Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with this function. Therefore, these registers should be set to 0.

Table 19.17 Simultaneous Reload Settings for External Event Counting

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Disables simultaneous reloading.
TAUJ2RDM.TAUJ2RDMm	0: When simultaneous reloading is disabled (TAUJ2RDE.TAUJ2RDEm = 0), set these bits to 0.

(5) Operating Procedure for External Event Counting

Table 19.18 Operating Procedure for External Event Counting

	Operation	Status of TAUJ2
Initial Channel Setting	Set TAUJ2CMORm and TAUJ2CMURm registers as described in Table 19.15, Contents of the TAUJ2CMORm Register for External Event Counting and Table 19.16, Contents of the TAUJ2CMURm Register for External Event Counting. Set the value of TAUJ2CDRm register.	Channel operation is stopped.
Start operation	Set TAUJ2TS.TAUJ2TSm to 1. TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is set to 1 and the counter starts. TAUJ2CNTm loads the TAUJ2CDRm value and waits for detection of the TAUJ2TTINm input edge.
During Operation	Detection of TAUJ2TTINm edges The value of TAUJ2CDRm can be changed at any time. The TAUJ2CNTm register can be read at any time.	TAUJ2CNTm counts down each time a TAUJ2TTINm input edge is detected. When the counter reaches 0000 0000H: • TAUJ2CNTm loads the TAUJ2CDRm value and continues counting. • INTTAUJ2Im is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm stops and retains its current value.

Restart (from Stop Operation back to Start operation)

(6) Specific Timing Diagrams

(a) TAUJ2CDRm = 0000H

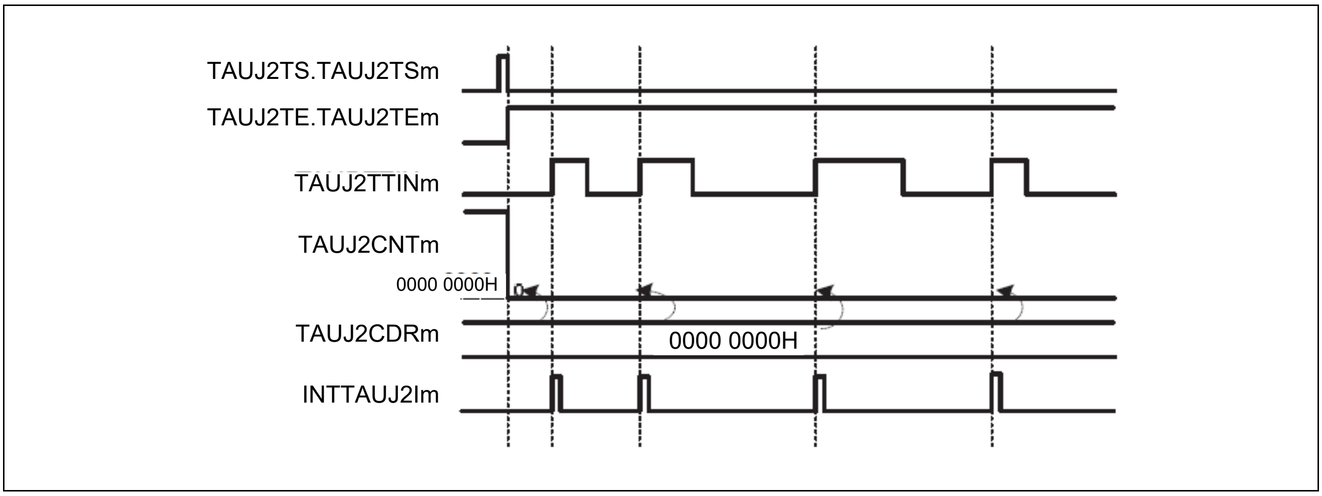


Figure 19.16 TAUJ2CDRm = 0000 0000H, TAUJ2CMURm.TAUJ2TIS[1:0] = 01B

- If 0000 0000H = TAUJ2CDRm, 0000 0000H is loaded to TAUJ2CNTm every time an effective TAUJ2TTINm input edge is detected. In other words, INTTAUJ2lm is generated every time an effective TAUJ2TTINm input edge is detected.

(b) Operation stop and restart

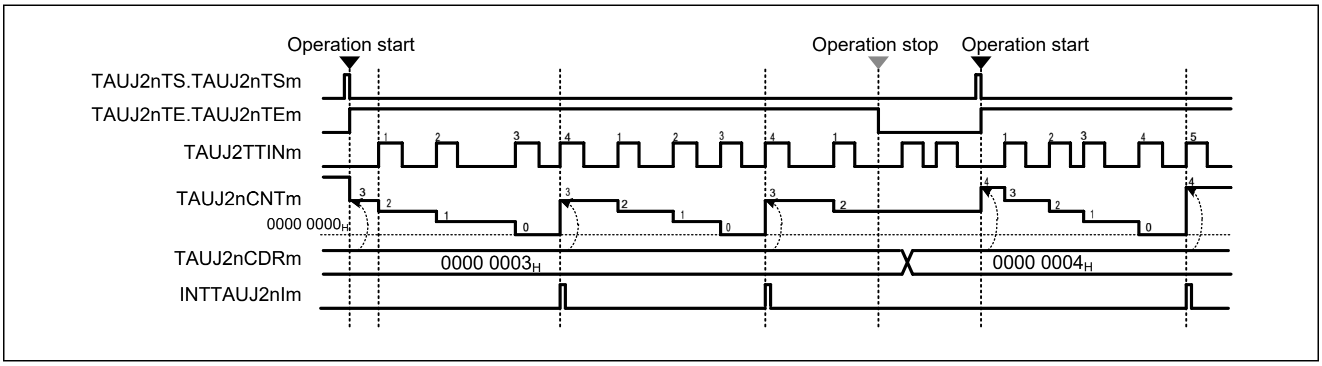


Figure 19.17 Operation Stop and Restart (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B)

- The counter can be stopped by setting TAUJ2TT.TAUJ2TTm to 1. This in turn sets TAUJ2TE.TAUJ2TEm to 0.
- TAUJ2CNTm stops and retains its current value. TAUJ2TTINm continues and TAUJ2CNTm ignores the effective edge.
- The counter can be restarted by setting TAUJ2TS.TAUJ2TSm to 1. TAUJ2CNTm loads the TAUJ2CDRm value and restarts counting.

(c) Forced restart

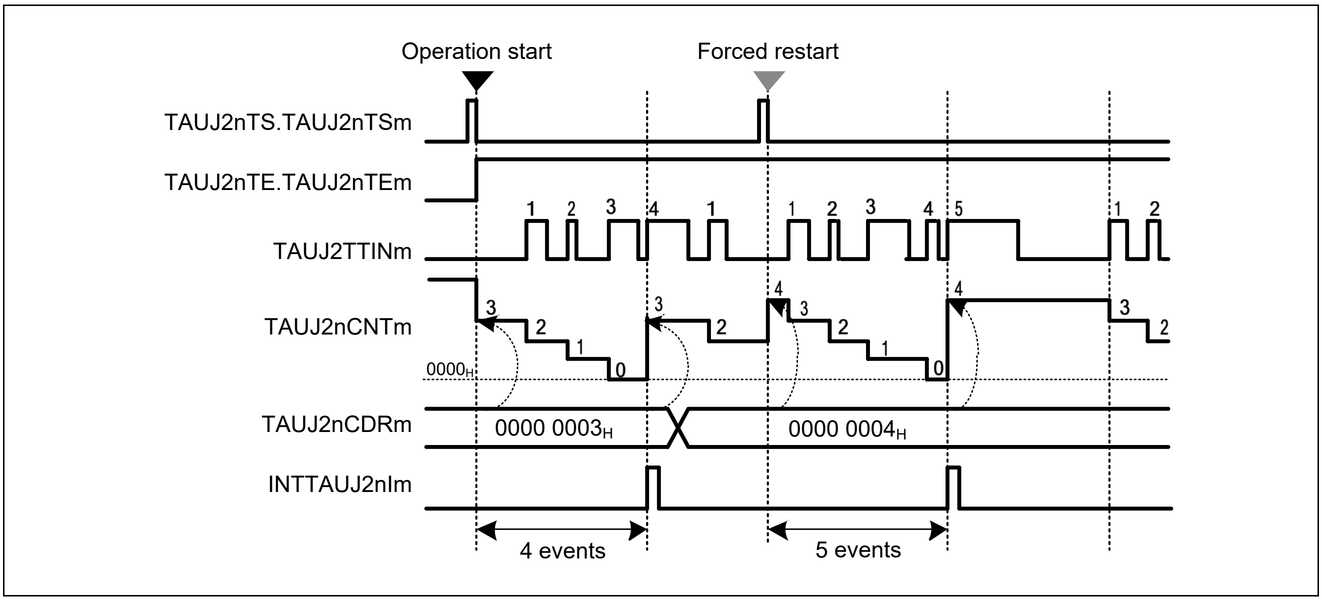


Figure 19.18 Forced Restart (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B)

When the counter is forcibly restarted, the changed TAUJ2CDRm value is applied to TAUJ2CNTm immediately.

- The counter can be restarted without stopping by setting TAUJ2TS.TAUJ2TSm to 1 during operation.
- The value of TAUJ2CDRm is loaded into TAUJ2CNTm and the counter awaits the next effective TAUJ2TTINm input edge.

19.7.4 Delay Counting

(1) Functional Description

This function generates interrupts (INTTAUJ2Im), which have a defined delay to the TAUJ2TTINm input signal. TAUJ2TTINm input signal pulses that occur within the delay period are ignored. The type of edge for use as an effective trigger is selectable from among rising edges, falling edges, or both (rising and falling) edges.

This function does not use TAUJ2TTOUTm.

(2) Block Diagram

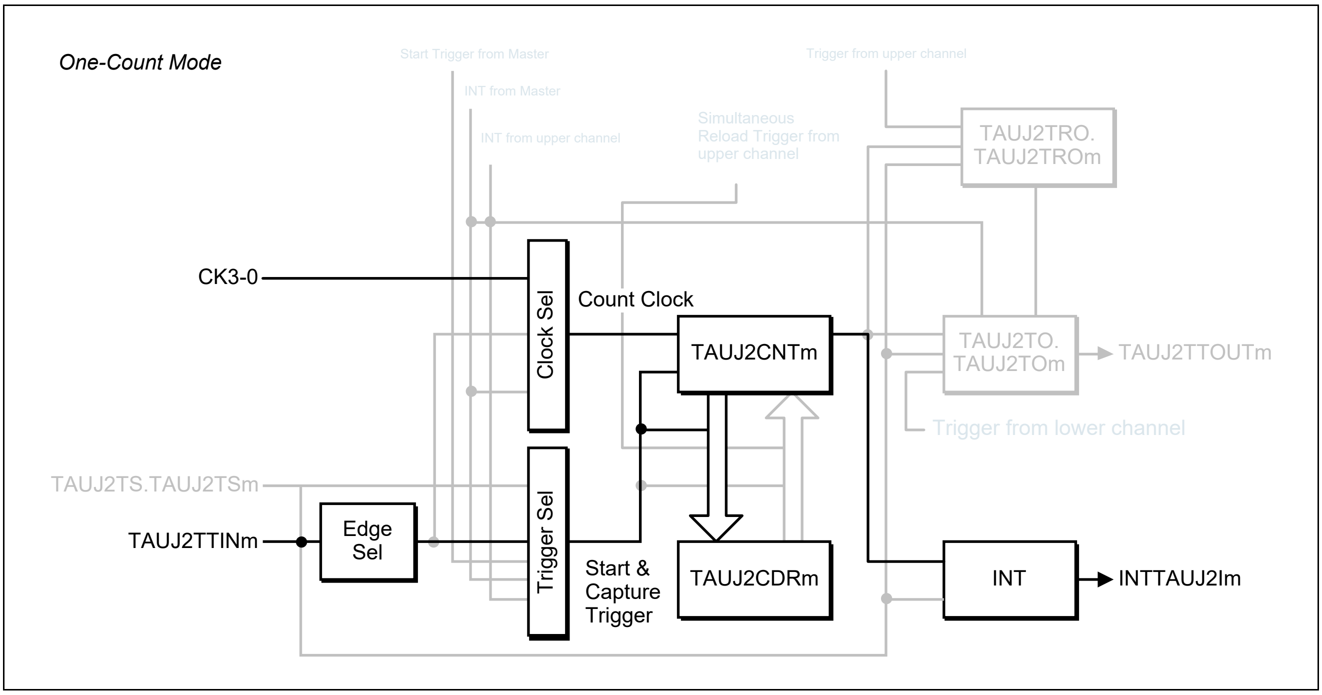


Figure 19.19 Block Diagram of Delay Counting

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- Falling edge detection (TAUJ2CMURm.TAUJ2TIS[1:0] = 00B)

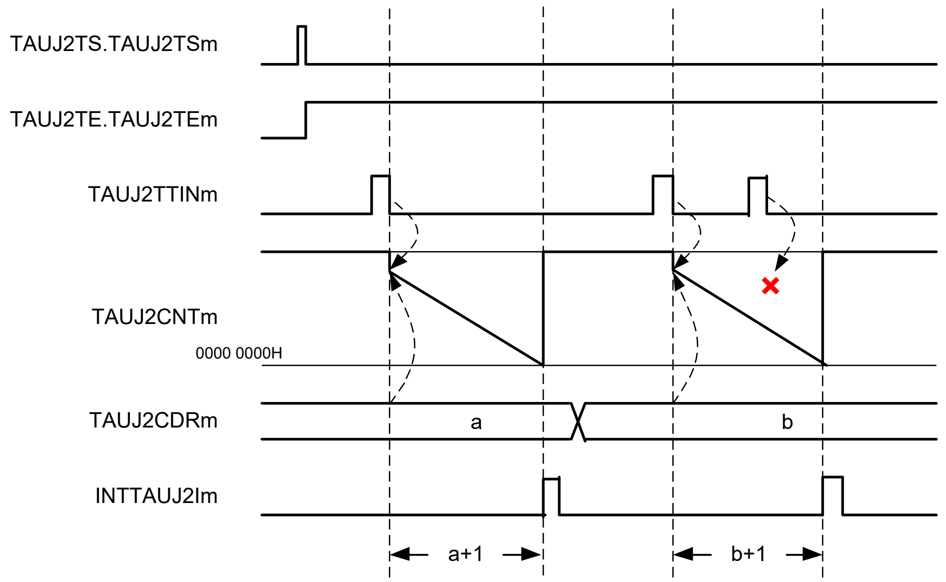


Figure 19.20 General Timing Diagram of Delay Counting

The value of TAUJ2CDRm can be rewritten at any time, and the changed value of TAUJ2CDRm is applied the next time the counter starts to count down.

(4) Equations

Delay between TAUJ2TTINm and INTTAUJ2Im = count clock cycle × (TAUJ2CDRm + 1)

(5) Register Settings

(a) TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.19 TAUJ2CMORm Settings for Delay Counting

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock that suits the application.
TAUJ2CCS[1:0]	These bits select the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	0: Independent operation
TAUJ2STS[2:0]	These bits select the external start trigger. 001: Effective TAUJ2TTINm input edge signal is used as the external start trigger.
TAUJ2COS[1:0]	00: Not used (initial value)
TAUJ2MD[4:1]	These bits select the operating mode. 0100: One-count mode
TAUJ2MD0	This bit enables or disables detection of a start trigger during counter operation. 0: Start trigger detection is disabled.

(b) TAUJ2CMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														TAUJ2TIS[1:0]	

Table 19.20 TAUJ2CMURm Settings for Delay Counting

Bit Name	Setting
TAUJ2TIS[1:0]	These bits select the effective edge of the externally input signal. 00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (measurement of the width at low level) Select the effective edge to suit the application.

(c) Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with the TAUJ2TTINm input interval timer. Therefore, these registers must be set to 0.

Table 19.21 Simultaneous Reload Settings for Delay Counting

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Set to 0 since this disables simultaneous reloading of channels.
TAUJ2RDM.TAUJ2RDMm	0: Not used (initial value)

(d) Register settings for channel output

Table 19.22 Control Bit Settings for Independent Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	0: Set to 0 since this disables output operation of channel m.
TAUJ2TOM.TAUJ2TOMm	0: Not used (initial value)
TAUJ2TOC.TAUJ2TOCm	0: Not used (initial value)
TAUJ2TOL.TAUJ2TOLm	0: Not used (initial value)

(6) Operating Procedure for Delay Counting

Table 19.23 Operating Procedure

		Operation	Status of TAUJ2
	Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register. • Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output. • Set the delay in the TAUJ2CDRm register.	Channel operation is stopped.
Restart	Start Operation	Set TAUJ2TS.TAUJ2TSm to 1. TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is set to 1 and TAUJ2CNTm waits for detection of the TAUJ2TTINm start edge.
	During Operation	Register whose value can be changed at any time • TAUJ2CDRm Register which is readable at any time • TAUJ2CNTm	When a start edge is detected, TAUJ2CNTm updates the value of TAUJ2CDRm and starts counting. When the counter reaches 0000 0000H (the delayed amount), INTTAUJ2Im occurs, and TAUJ2CNTm suspends counting and waits for a trigger. If a trigger occurs while TAUJ2CNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm stops and retains its value.

19.7.5 TAUJ2TTINm Input Pulse Interval Measurement

(1) Functional Description

This function captures the counter value TAUJ2CDRm and uses this value and the overflow bit TAUJ2CSRm.TAUJ2OVF to measure the interval of the TAUJ2TTINm input signal. The types of edge which can be used as effective triggers are rising edges, falling edges, and both (rising and falling) edges. This function does not use TAUJ2TTOUTm.

(2) Block Diagram

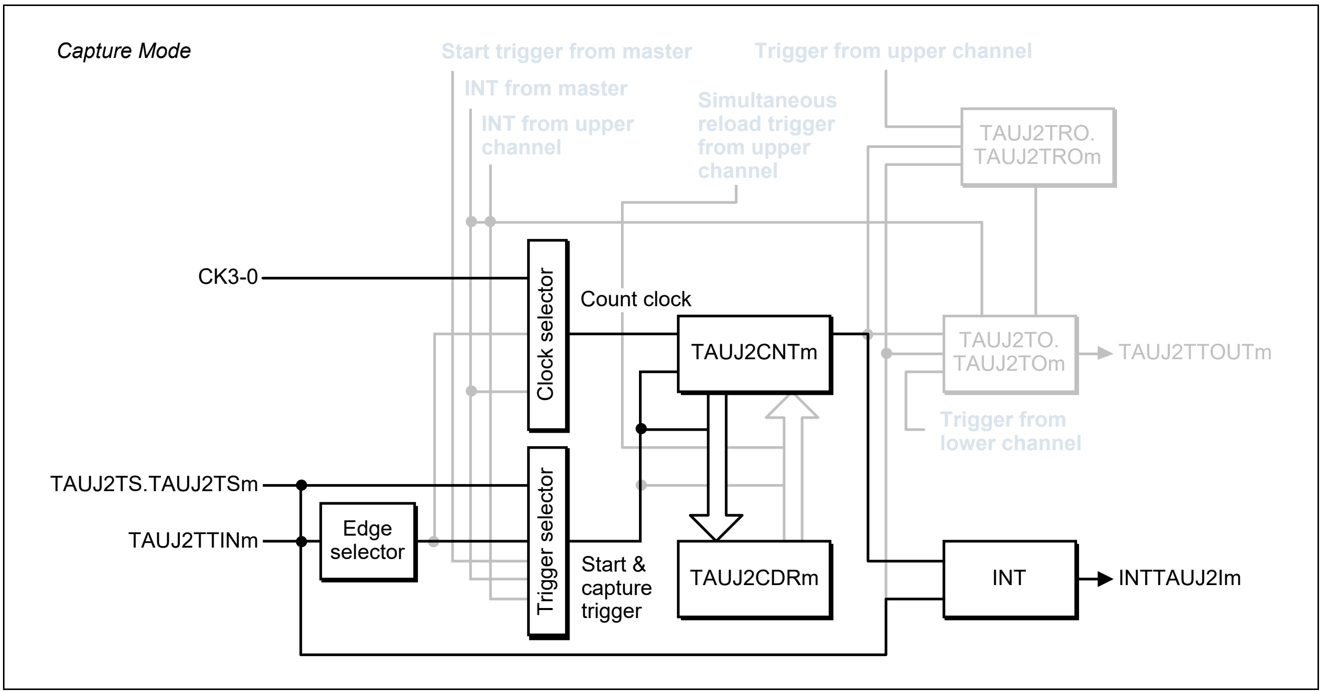


Figure 19.21 Block Diagram of TAUJ2TTINm Input Pulse Interval Measurement

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- INTTAUJ2Im not generated at the start of operation (TAUJ2CMORm.TAUJ2MD0 = 0)
- Falling edge detection (TAUJ2CMURm.TAUJ2TIS[1:0] = 00B)
- When an effective TAUJ2TTINm input is detected after the overflow, TAUJ2CDRm is changed and TAUJ2CSRm.TAUJ2OVF is set to 1 (TAUJ2CMORm.TAUJ2COS[1:0] = 00B)

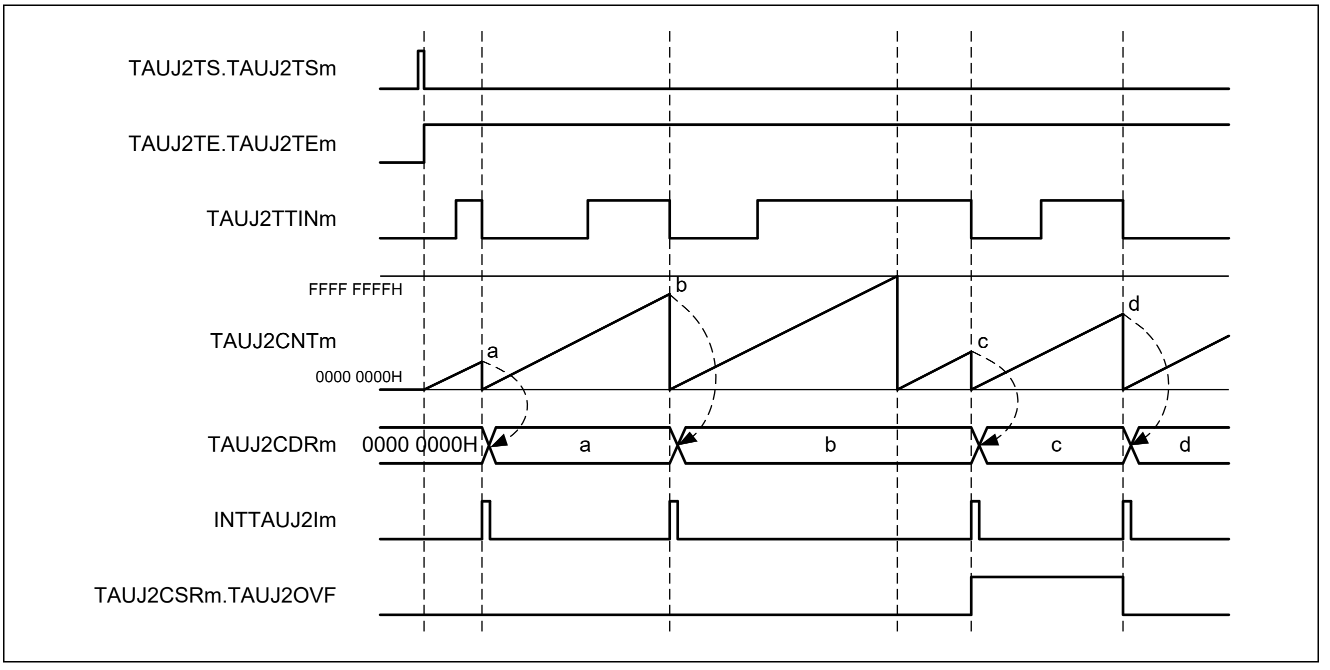


Figure 19.22 General Timing Diagram of TAUJ2TTINm Input Pulse Interval Measurement

(4) Equations

TAUJ2TTINm input pulse interval = count clock cycle ×
[(TAUJ2CSRm.TAUJ2OVF × (FFFF FFFFH + 1)) + TAUJ2CDRm capture value + 1]

(5) Register Settings

(a) TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.24 TAUJ2CMORm Settings for TAUJ2TTINm Input Pulse Interval Measurement

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock that suits the application.
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	0: Independent operation
TAUJ2STS[2:0]	These bits select the external start trigger. 001: Effective edge of the TAUJ2TTINm input signal is used as the external capture trigger.
TAUJ2COS[1:0]	These bits select operation of the data register and overflow flag when capturing is in use. 00: Setting/clearing TAUJ2CSRm.TAUJ2OVF in response to the detection of an effective edge of the capture input signal and capturing the counter value (TAUJ2CNTm) 10: When TAUJ2CSRm.TAUJ2OVF is set or cleared in response to the detection of an effective edge of the capture input signal and the counter overflows (FFFF FFFFH -> 0000 0000H), FFFF FFFFH is captured in TAUJ2CDRm and detection of the next effective edge of the capture input signal is ignored. Other than the above: Setting prohibited
TAUJ2MD[4:1]	These bits select the operating mode. 0010: Capture mode
TAUJ2MD0	This bit specifies whether an INTTAUJ2Im interrupt is generated when counting starts. 0: INTTAUJ2Im prohibited 1: INTTAUJ2Im permitted

(b) TAUJ2CMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														TAUJ2TIS[1:0]	

Table 19.25 TAUJ2CMORm Settings for TAUJ2TTINm Input Pulse Interval Measurement

Bit Name	Setting
TAUJ2TIS[1:0]	These bits select the effective edge of the externally input signal. 00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (measurement of the width at low level) Select the effective edge to suit the application.

(c) Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with this function. Therefore, these registers must be set to 0.

Table 19.26 Simultaneous Reload Settings for TAUJ2TTINm Input Pulse Interval Measurement

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Set to 0 since this disables simultaneous reloading of channels.
TAUJ2RDM.TAUJ2RDMm	0: Not used (initial value)

(d) Register settings for channel output

Table 19.27 Control Bit Settings for Independent Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	0: Set to 0 since this disables output operation of channel m.
TAUJ2TOM.TAUJ2TOMm	0: Not used (initial value)
TAUJ2TOC.TAUJ2TOCm	0: Not used (initial value)
TAUJ2TOL.TAUJ2TOLm	0: Not used (initial value)

(6) Operating Procedure for Input Pulse Interval Measurement by TAUJ2TTINm

Table 19.28 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register. • Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output. • TAUJ2CDRm operates as a capture register.	Channel operation is stopped.
Start Operation	Set TAUJ2TS.TAUJ2TSm to 1. TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is set to 1 and the counter starts. TAUJ2CNTm is cleared to 0000 0000H. INTTAUJ2Im is generated when TAUJ2CMORm.TAUJ2MD0 is set to 1.
During Operation	Register whose value can be changed at any time • TAUJ2CMURm.TAUJ2TIS[1:0] bits Registers which are readable at any time • TAUJ2CDRm register • TAUJ2CSRm register When clearing the TAUJ2CSRm.TAUJ2OVF bit, write 1 to the TAUJ2CSCm.TAUJ2CLOV bit.	TAUJ2CNTm starts counting up from 0000 0000H. When an effective edge of TAUJ2TTINm is detected, the counter is cleared to 0000 0000H and continues counting. When an effective edge of TAUJ2TTINm is detected, the value of TAUJ2CNTm is transferred to (captured in) TAUJ2CDRm and INTTAUJ2Im is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm stops and TAUJ2CNTm and TAUJ2CSRm.TAUJ2OVF retain their current values.

(Restart: from Stop Operation back to Start Operation)

(7) Specific Timing Diagrams: Overflow Behavior

The following describes the operation timing for each setting of TAUJ2CMORm.TAUJ2COS[1:0] when an overflow occurred.

(a) TAUJ2CMORm.TAUJ2COS[1:0] = 00B

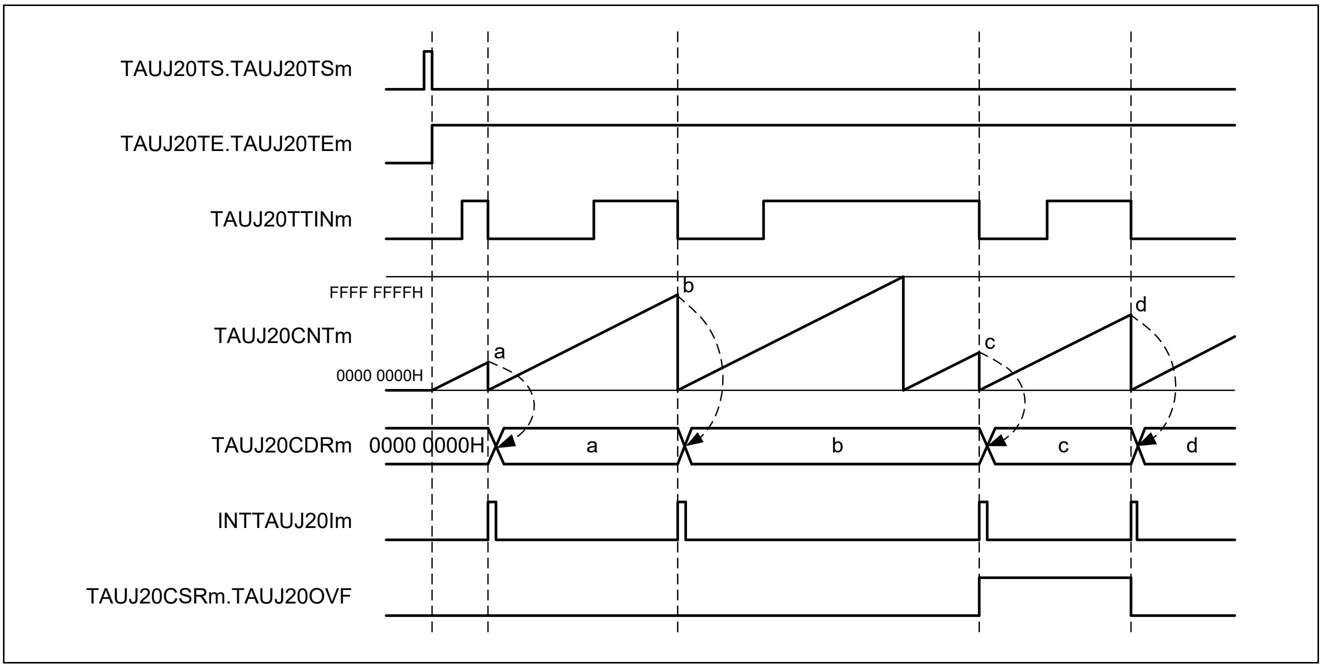


Figure 19.23 TAUJ2CMORm.TAUJ2COS[1:0] = 00B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B

- Even when an overflow occurred, the value of TAUJ2CDRm remains unchanged and TAUJ2CSRm.TAUJ2OVF remains 0.
- When an effective edge of the TAUJ2TTINm input signal is detected after the overflow, the value of TAUJ2CNTm is captured in TAUJ2CDRm and TAUJ2CSRm.TAUJ2OVF is set to 1.
- When an effective edge of the TAUJ2TTINm input signal is detected while no overflow has occurred, TAUJ2CSRm.TAUJ2OVF is cleared to 0.

(b) TAUJ2CMORm.TAUJ2COS[1:0] = 10B

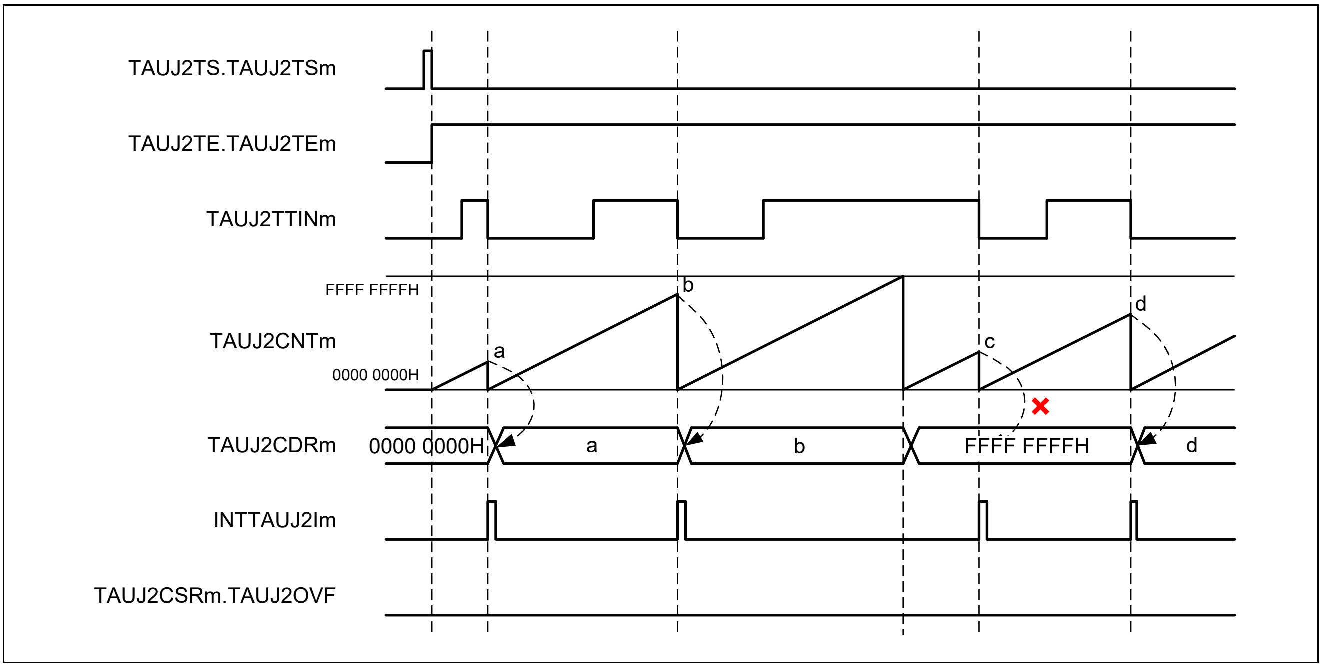


Figure 19.24 TAUJ2CMORm.TAUJ2COS[1:0] = 10B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B

- When an overflow occurred, TAUJ2CDRm is set to FFFF FFFFH and TAUJ2CSRm.TAUJ2OVF remains 0.
- Even when an effective edge of the TAUJ2TTINm input signal is detected, TAUJ2CSRm.TAUJ2OVF remains unchanged.
- An effective edge of the TAUJ2TTINm input signal being detected after the overflow is ignored.

19.7.6 TAUJ2TTINm Input Signal Width Measurement

(1) Overview

This function measures the width of the TAUJ2TTINm input signal. Counting starts on an effective edge (starting edge) of TAUJ2TTINm and stops on the opposite edge (stopping edge), and the input signal width is measured by capturing the number counted in that interval. When the counter reaches FFFF FFFFH before detecting a stop edge, the counter overflows. The types of input edge which can be used as effective triggers are the width at high level (from rising to falling) and the width at low level (from falling to rising).

(2) Block Diagram

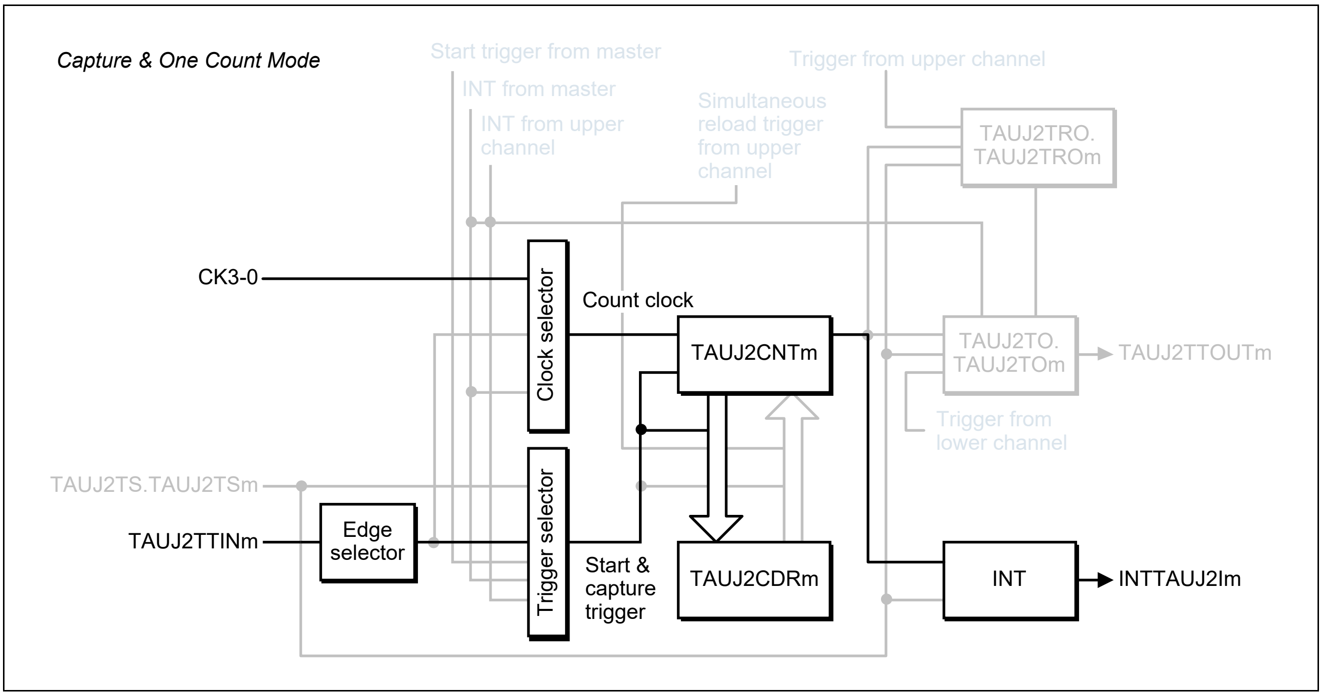


Figure 19.25 Block Diagram of TAUJ2TTINm Input Signal Width Measurement

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUJ2CMURm.TAUJ2TIS[1:0] = 11B)
- When an effective TAUJ2TTINm input is detected after the overflow, TAUJ2CDRm is changed and TAUJ2CSRm.TAUJ2OVF is set to 1 (TAUJ2CMORm.TAUJ2COS[1:0] = 00B)

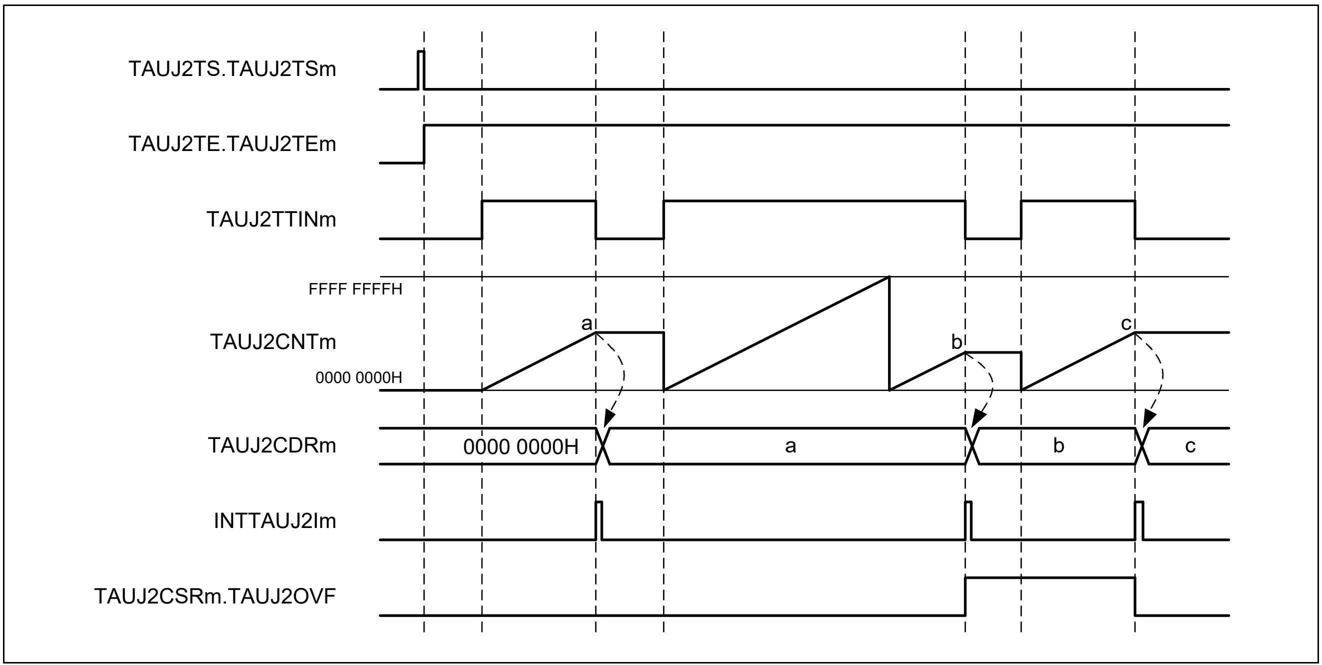


Figure 19.26 General Timing Diagram of TAUJ2TTINm Input Signal Width Measurement

(4) Equations

TAUJ2TTINm input signal width = count clock cycle ×
[(TAUJ2CSRm.TAUJ2OVF × (FFFF FFFFH + 1)) + TAUJ2CDRm capture value + 1]

(5) Register Settings

(a) TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.29 TAUJ2CMORm Settings for TAUJ2TTINm Input Signal Width Measurement

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock that suits the application.
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	0: Independent operation
TAUJ2STS[2:0]	These bits select the external start trigger. 001: An effective edge of the TAUJ2TTINm input signal is used as a start trigger and the opposite edge as a stop trigger.
TAUJ2COS[1:0]	These bits select operation of the data register and overflow flag when capturing is in use. 00: Setting/clearing TAUJ2CSRm.TAUJ2OVF in response to the detection of an effective edge of the capture input signal and capturing the counter value (TAUJ2CNTm) 10: When TAUJ2CSRm.TAUJ2OVF is set or cleared in response to the detection of an effective edge of the capture input signal and the counter overflows (FFFF FFFFH -> 0000 0000H), FFFF FFFFH is captured in TAUJ2CDRm and detection of the next effective edge of the capture input signal is ignored. Other than the above: Setting prohibited
TAUJ2MD[4:1]	These bits select the operating mode. 0010: Capture & one-count mode
TAUJ2MD0	This bit selects enabling or disabling of detection of a start trigger during counting. 0: Detection of a start trigger prohibited

(b) TAUJ2CMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														TAUJ2TIS[1:0]	

Table 19.30 TAUJ2CMURm Settings for TAUJ2TTINm Input Signal Width Measurement

Bit Name	Setting
TAUJ2TIS[1:0]	These bits select the width at low or high level of effective edges of the TAUJ2TTINm input signal. 10: Rising and falling edge detection (measurement of the width at low level) 11: Rising and falling edge detection (measurement of the width at high level) Select the effective edge to suit the application.

(c) Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with input signal width measurement by TAUJ2TTINm. Therefore, these registers must be set to 0.

Table 19.31 Simultaneous Reload Settings for TAUJ2TTINm Input Signal Width Measurement

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Set to 0 since this disables simultaneous reloading of channel m.
TAUJ2RDM.TAUJ2RDMm	0: Not used (initial value)

(d) Register settings for channel output

Table 19.32 Control Bit Settings for Independent Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	0: Set to 0 since this disables output operation of channel m.
TAUJ2TOM.TAUJ2TOMm	0: Not used (initial value)
TAUJ2TOC.TAUJ2TOCm	0: Not used (initial value)
TAUJ2TOL.TAUJ2TOLm	0: Not used (initial value)

(6) Operating Procedure for TAUJ2TTINm Input Signal Width Measurement

Table 19.33 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register. • Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output. • TAUJ2CDRm operates as a capture register.	Channel operation is stopped.
Start Operation (Restart)	Set TAUJ2TS.TAUJ2TSm to 1. TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is set to 1, and TAUJ2CNTm waits for TAUJ2TTINm start edge detection.
During Operation	Registers which are readable at any time: • TAUJ2CDRm register • TAUJ2CNTm register • TAUJ2CSRm register When clearing the TAUJ2CSRm.TAUJ2OVF bit, write 1 to the TAUJ2CSCm.TAUJ2CLOV bit.	When a start edge of TAUJ2TTINm is detected, TAUJ2CNTm starts counting from 0000 0000H. When a stop edge of TAUJ2TTINm is detected, it stops counting. When a stop edge of TAUJ2TTINm is detected, the value of TAUJ2CNTm is transferred to (captured in) TAUJ2CDRm and INTTAUJ2Im is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm stops and TAUJ2CNTm and TAUJ2CSRm.TAUJ2OVF retain their current values.

(7) Specific Timing Diagrams: Overflow Behavior

The following describes the operation timing for each setting of TAUJ2CMORm.TAUJ2COS[1:0] when an overflow occurred.

(a) TAUJ2CMORm.TAUJ2COS[1:0] = 00B

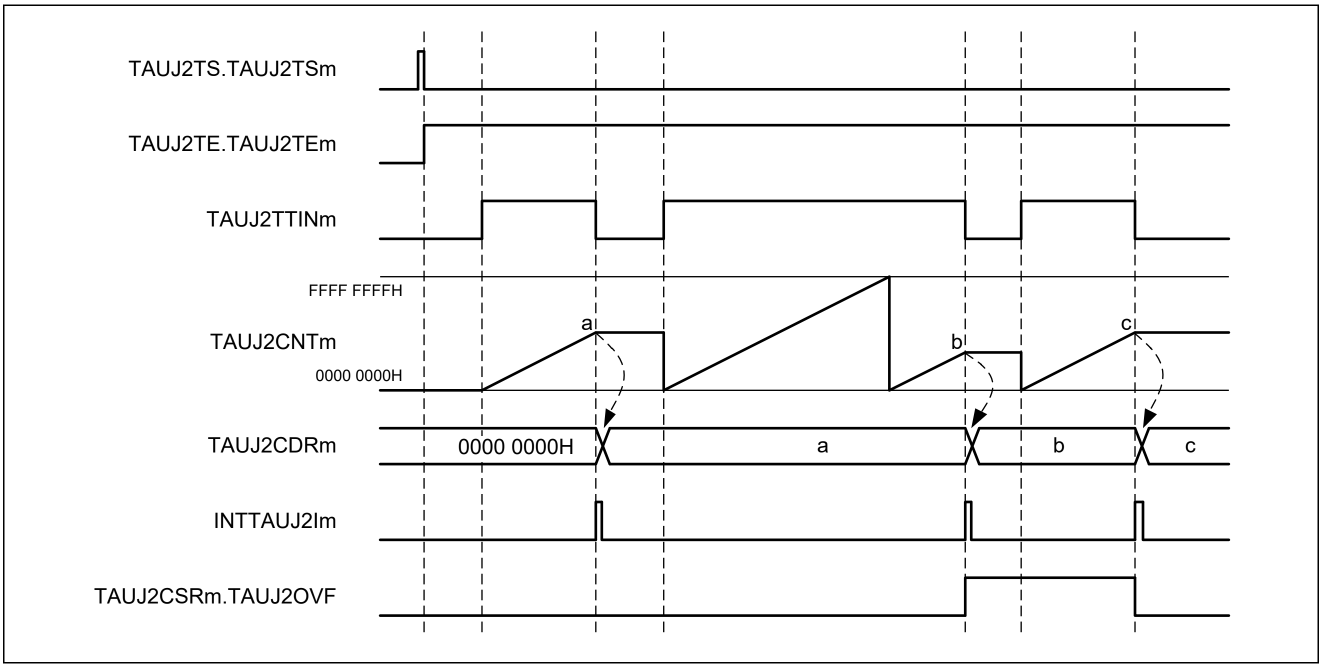


Figure 19.27 TAUJ2CMORm.TAUJ2COS[1:0] = 00B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 11B

- Even when an overflow occurred, the value of TAUJ2CDRm remains unchanged and TAUJ2CSRm.TAUJ2OVF remains 0.
- When an effective edge of the TAUJ2TTINm input signal is detected after the overflow, the value of TAUJ2CNTm is captured in TAUJ2CDRm and TAUJ2CSRm.TAUJ2OVF is set to 1.
- When an effective edge of the TAUJ2TTINm input signal is detected while no overflow has occurred, TAUJ2CSRm.TAUJ2OVF is cleared to 0.

(b) TAUJ2CMORm.TAUJ2COS[1:0] = 10B

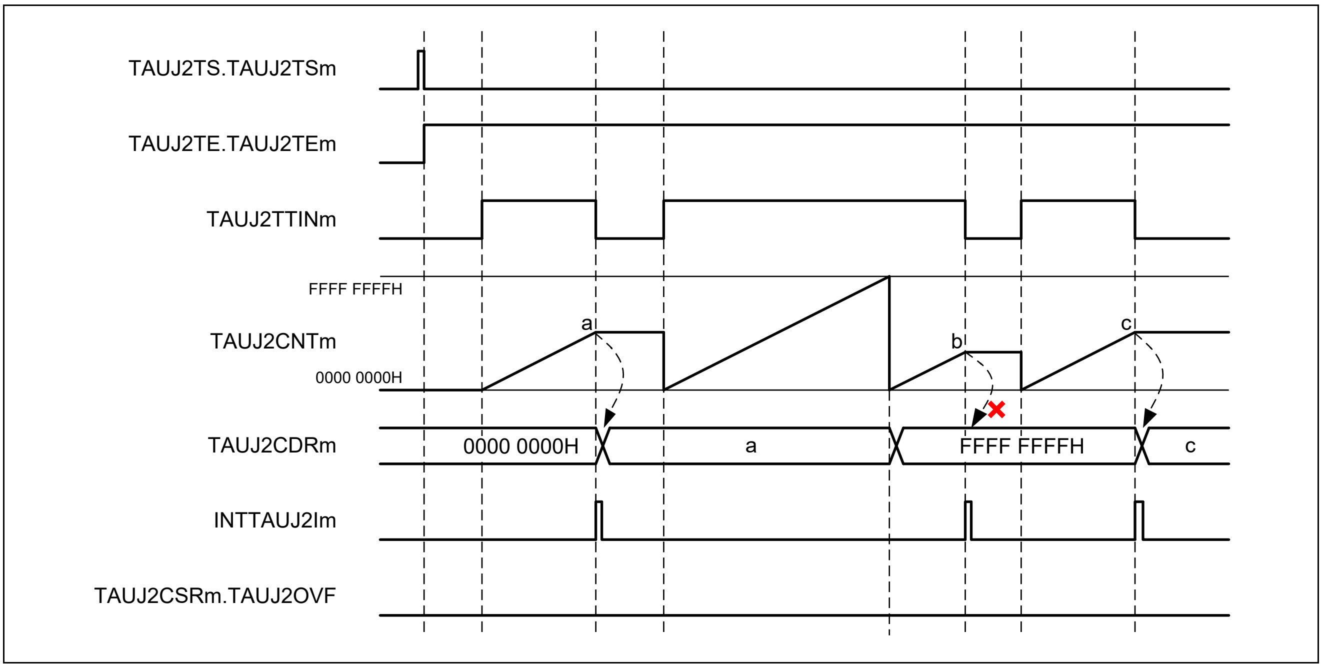


Figure 19.28 TAUJ2CMORm.TAUJ2COS[1:0] = 10B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 11B

- When an overflow occurred, TAUJ2CDRm is set to FFFF FFFFH and TAUJ2CSRm.TAUJ2OVF remains 0.
- Even when an effective edge of the TAUJ2TTINm input signal is detected, TAUJ2CSRm.TAUJ2OVF remains unchanged.
- An effective edge of the TAUJ2TTINm input signal being detected after the overflow is ignored.

(8) How to Output Overflow Interrupt

(a) Functional description

A channel for TAUJ2TTINm input signal width measurement and that for overflow interrupt output are combined to generate an overflow interrupt (two channels are required to generate an overflow interrupt).

See Figure 19.29, Block Diagram of Overflow Interrupt Output (for TAUJ2TTINm Width Measurement), for configuration of channels.

(b) Block diagram

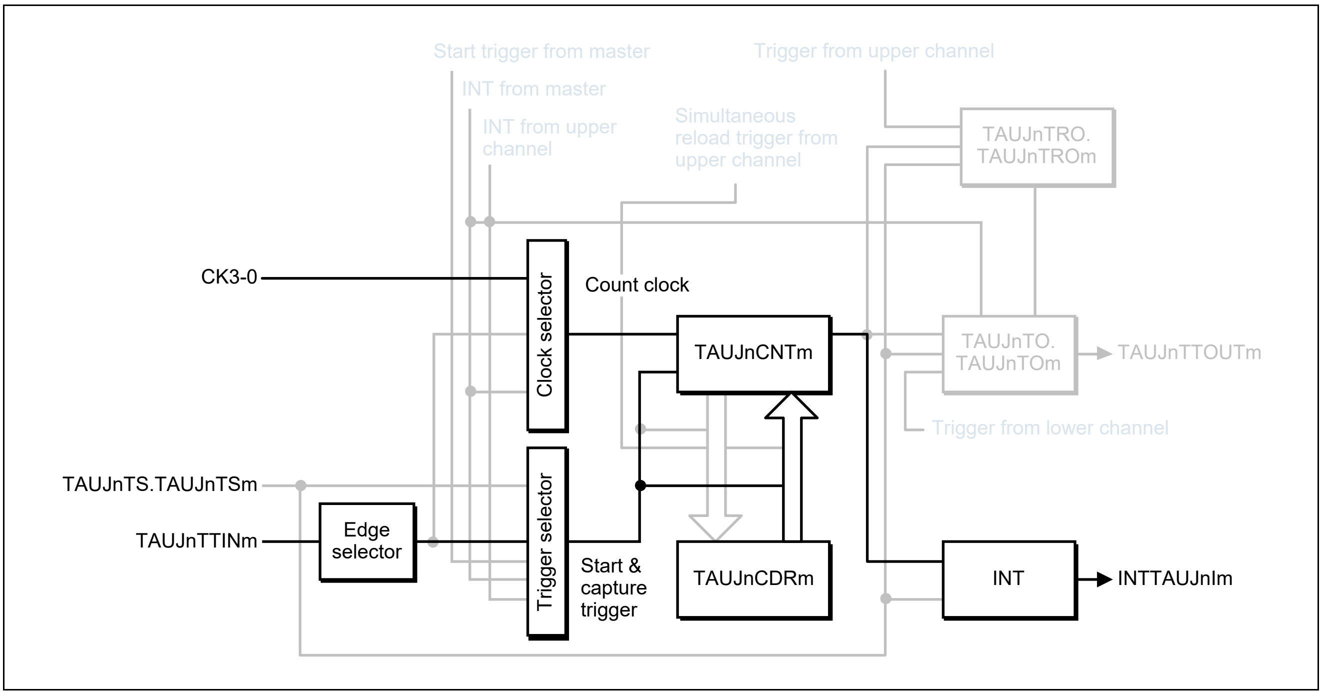


Figure 19.29 Block Diagram of Overflow Interrupt Output (for TAUJ2TTINm Width Measurement)

(c) General timing diagram

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement

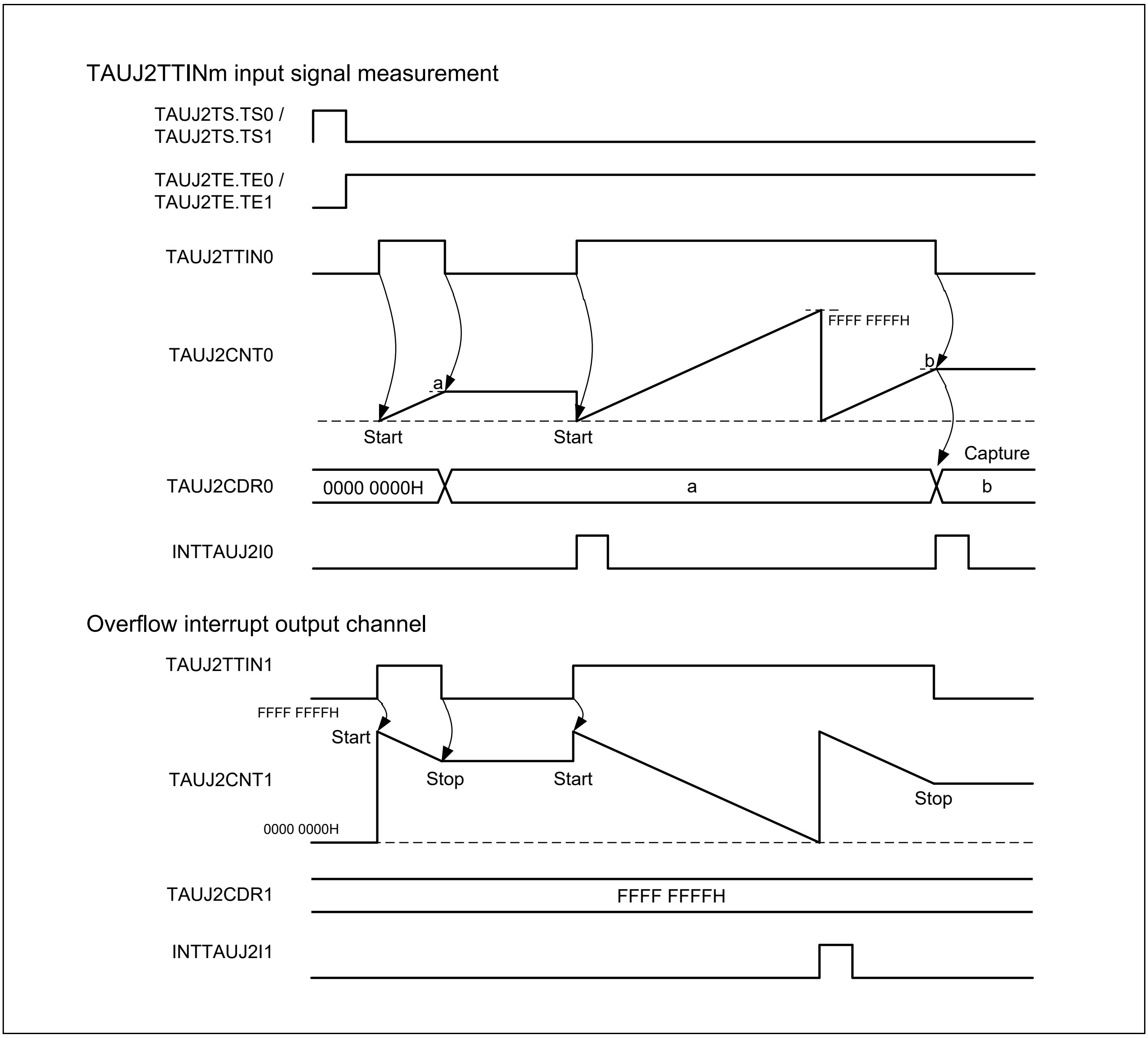


Figure 19.30 General Timing Diagram at the Time of Overflow Interrupt Output

(d) Register settings for TAUJ2TTINm input signal width measurement

Make settings for operation of TAUJ2TTINm signal width measurement.

(e) Register settings for overflow interrupt output channel

- TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.34 TAUJ2CMORm Settings

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock that suits the application.
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	0: Independent operation
TAUJ2STS[2:0]	These bits select the external start trigger. 001: An effective edge of the TAUJ2TTINm input signal is used as a start trigger and the opposite edge as a stop trigger.
TAUJ2COS[1:0]	00: Not used (initial value)
TAUJ2MD[4:1]	These bits select the operating mode. 0100: One-count mode
TAUJ2MD0	This bit selects enabling or disabling of detection of a start trigger during counting. 0: Detection of a start trigger prohibited

- TAUJ2CMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														TAUJ2TIS[1:0]	

Table 19.35 TAUJ2CMURm Settings for TAUJ2TTINm Input Signal Width Measurement

Bit Name	Setting
TAUJ2TIS[1:0]	These bits select the width at low or high level of effective edges of the TAUJ2TTINm input signal. 10: Rising and falling edge detection (measurement of the width at low level) 11: Rising and falling edge detection (measurement of the width at high level) Select the effective edge to suit the application.

- Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with input signal width measurement by TAUJ2TTINm. Therefore, these registers must be set to 0.

Table 19.36 Simultaneous Reload Settings for TAUJ2TTINm Input Signal Width Measurement

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Set to 0 since this disables simultaneous reloading of channel m.
TAUJ2RDM.TAUJ2RDMm	0: Not used (initial value)

- Register settings for channel output

Table 19.37 Control Bit Settings for Independent Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	0: Set to 0 since this disables output operation of channel m.
TAUJ2TOM.TAUJ2TOMm	0: Not used (initial value)
TAUJ2TOC.TAUJ2TOCm	0: Not used (initial value)
TAUJ2TOL.TAUJ2TOLm	0: Not used (initial value)

(9) Operating Procedure for Overflow Interrupt Output

Table 19.38 Operating Procedure

		Operation	Status of TAUJ2
	Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register (for 2 channels). • Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output (for 2 channels). • Set the TAUJ2CDRm register for TAUJ2TTINm signal width measurement to 0000 0000H and the TAUJ2CDRm register for overflow interrupt output channel to FFFF FFFFH.	Channel operation is stopped.
Restart	Start Operation	• Set TAUJ2TS.TAUJ2TSm to 1 for 2 channels at the same time. • TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0. • Detection of a start edge of TAUJ2TTINm	TAUJ2TE.TAUJ2TEm is set to 1, and TAUJ2CNTm waits for TAUJ2TTINm start edge detection. When a start edge is detected, the value of TAUJ2CDRm (FFFF FFFFH) is updated in TAUJ2CNTm.
	During Operation	No special notes	When a start edge of TAUJ2TTINm is detected, TAUJ2CNTm starts counting from FFFF FFFFH. When a stop edge of TAUJ2TTINm is detected, it stops counting. When the counter reaches 0000 0000H, INTTAUJ2Im is generated. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm stops and retains its current value.

19.7.7 TAUJ2TTINm Input Position Detection

(1) Functional Description

This function measures the time from the start of counting until an effective edge of the TAUJ2TTINm input signal. The counter operates as free running and the value counted is captured in TAUJ2CDRm when a further effective edge of TAUJ2TTINm is detected. The types of edge which can be used as effective triggers are rising edges, falling edges, and both (rising and falling) edges. This function does not use TAUJ2TTOUTm.

Remark: When the TAUJ2CMORm.TAUJ2MD0 bit is set to 0, the first interrupt after operation starts or is restarted is not generated.

Caution: In this function, an overflow is undetectable. If you need to detect an overflow, use this function in combination with interval timer mode. If you do not have two channels available, the same functionality can be achieved by using TAUJ2TTINm input signal width measurement and calculating the accumulated value of the result of capturing.

(2) Block Diagram

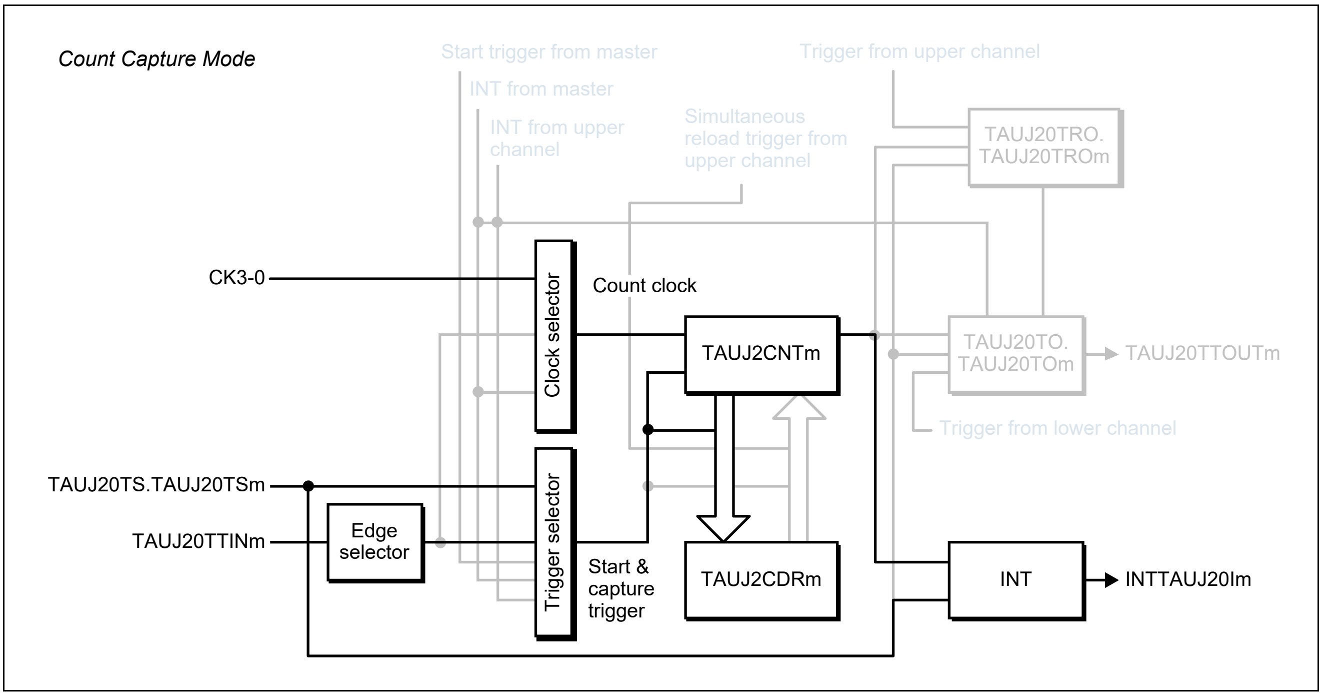


Figure 19.31 Block Diagram of TAUJ2TTINm Input Position Detection

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- INTTAUJ2Im is not generated at the start of operation (TAUJ2CMORm.TAUJ2MD0 = 0)
- Falling edge detection (TAUJ2CMURm.TAUJ2TIS[1:0] = 00B)

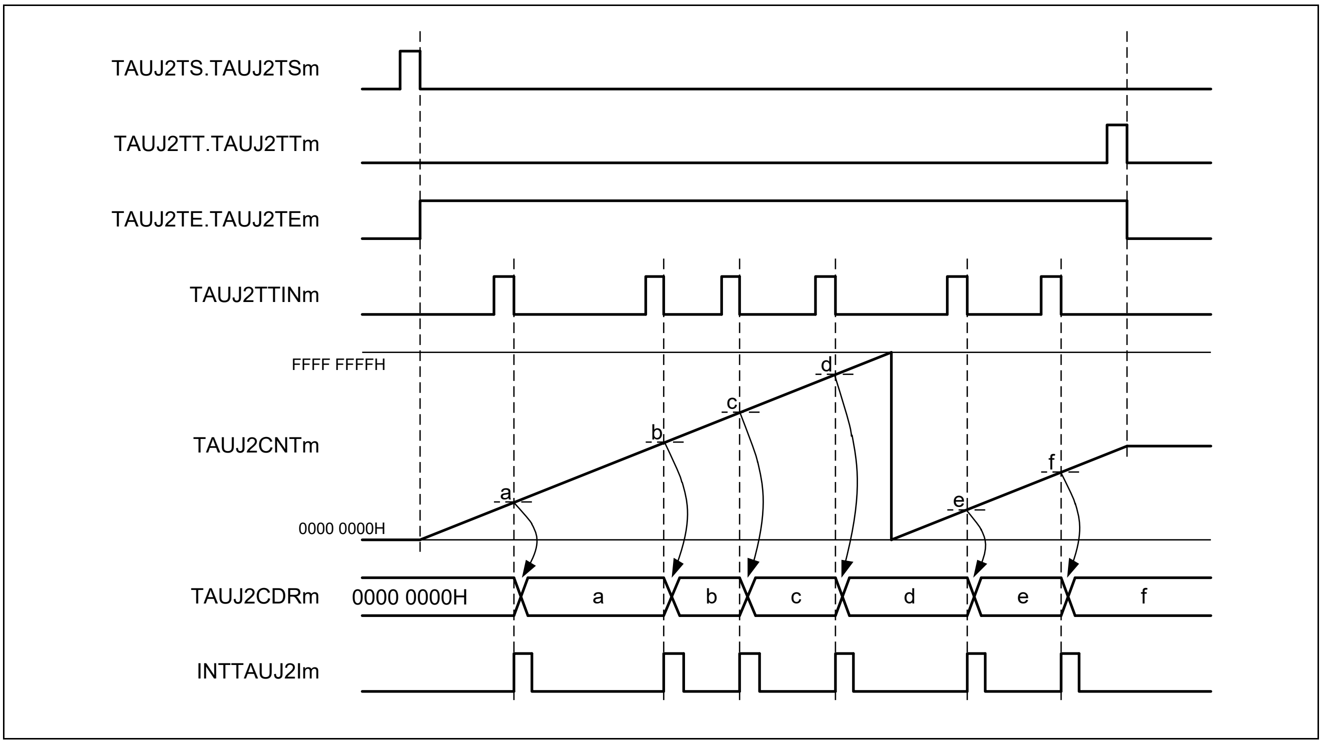


Figure 19.32 General Timing Diagram of TAUJ2TTINm Input Position Detection

(4) Equations

Function duration at a TAUJ2TTINm input pulse =
count clock cycle × [(FFFF FFFFH+1 × TAUJ2CSRm.TAUJ2OVF) + (TAUJ2CDRm capture value + 1)]

(5) Register Settings

(a) TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.39 TAUJ2CMORm Settings for TAUJ2TTINm Input Position Detection

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock that suits the application.
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	0: Independent operation
TAUJ2STS[2:0]	These bits select the external start trigger. 001: An effective edge of the TAUJ2TTINm input signal is used as an external capture trigger.
TAUJ2COS[1:0]	01: Fixed value setting
TAUJ2MD[4:1]	These bits select the operating mode. 1011: Count capture mode
TAUJ2MD0	This bit specifies whether an INTTAUJ2Im interrupt is generated when counting starts. 0: INTTAUJ2Im prohibited 1: INTTAUJ2Im permitted

(b) TAUJ2CMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														TAUJ2TIS[1:0]	

Table 19.40 TAUJ2CMURm Settings for TAUJ2TTINm Input Position Detection

Bit Name	Setting
TAUJ2TIS[1:0]	These bits select the effective edge of the TAUJ2TTINm input signal. 00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (measurement of the width at low level) 11: Rising and falling edge detection (measurement of the width at high level) Select the effective edge to suit the application.

(c) Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with the TAUJ2TTINm input interval timer. Therefore, these registers must be set to 0.

Table 19.41 Simultaneous Reload Settings for Delay Counting

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Set to 0 since this disables simultaneous reloading of channels.
TAUJ2RDM.TAUJ2RDMm	0: Not used (initial value)

(d) Register settings for channel output

Table 19.42 Control Bit Settings for Independent Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	0: Set to 0 since this disables output operation of channels.
TAUJ2TOM.TAUJ2TOMm	0: Not used (initial value)
TAUJ2TOC.TAUJ2TOCm	0: Not used (initial value)
TAUJ2TOL.TAUJ2TOLm	0: Not used (initial value)

(6) Operating Procedure for TAUJ2TTINm Input Position Detection

Table 19.43 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register. • Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output. • The TAUJ2CDRm register operates as a capture register.	Channel operation is stopped.
Start Operation	Set TAUJ2TS.TAUJ2TSm to 1. TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is set to 1 and the counter starts. INTTAUJ2Im is generated when TAUJ2CMORm.TAUJ2MD0 is set to 1.
During Operation	Register whose value can be changed at any time • TAUJ2CMURm.TAUJ2TIS[1:0] bits Registers which are readable at any time • TAUJ2CDRm register • TAUJ2CNTm register • TAUJ2CSRm register When clearing the TAUJ2CSRm.TAUJ2OVF bit, write 1 to the TAUJ2CSCm.TAUJ2CLOV bit.	TAUJ2CNTm starts counting up from 0000 0000H. When an effective edge of TAUJ2TTINm is detected, the value of TAUJ2CNTm is transferred to (captured in) TAUJ2CDRm to output INTTAUJ2Im The counter value is not cleared to 0000 0000H and counting continues. Afterwards, this procedure is repeated.
Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm stops and TAUJ2CNTm and TAUJ2CSRm.TAUJ2OVF retain their current values.

Restart (from Stop Operation back to Start Operation)

(7) Specific Timing Diagrams

(a) Operation stop and restart

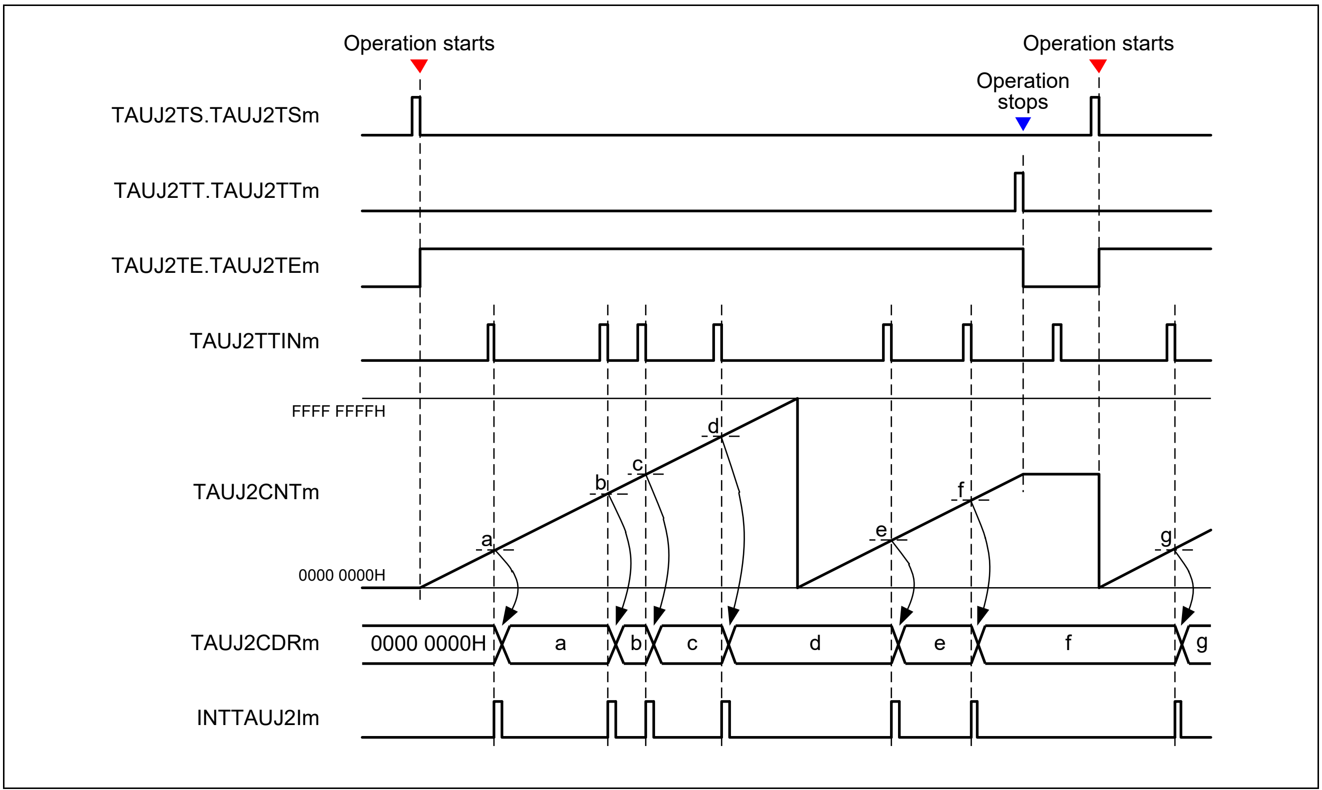


Figure 19.33 Operation Stop and Restart, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B

- The counter can be stopped by setting TAUJ2TT.TAUJ2TTm to 1, which in turn sets TAUJ2TE.TAUJ2TEm to 0.
- TAUJ2CNTm stops and the current value is retained.
- If the counter is stopped, effective TAUJ2TTINm input edges are ignored.
- The counter can be started by setting TAUJ2TS.TAUJ2TSm to 1.
  TAUJ2CNTm starts counting from 0000 0000H.

(8) Output of Overflow Interrupt

(a) Functional description

An overflow interrupt is generated by combining a TAUJ2TTINm input signal width measurement channel and a overflow interrupt output channel (generation of overflow interrupts requires two channels).

(b) Block diagram

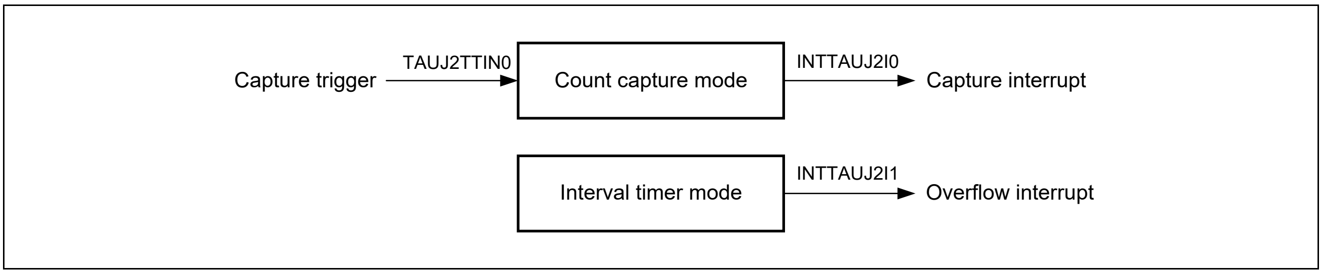


Figure 19.34 Block Diagram of Overflow Interrupt Output (when TAUJ2TTINm Input Position is Detected)

(c) General timing diagram

The following settings apply to the general timing diagram

- Falling edge detection

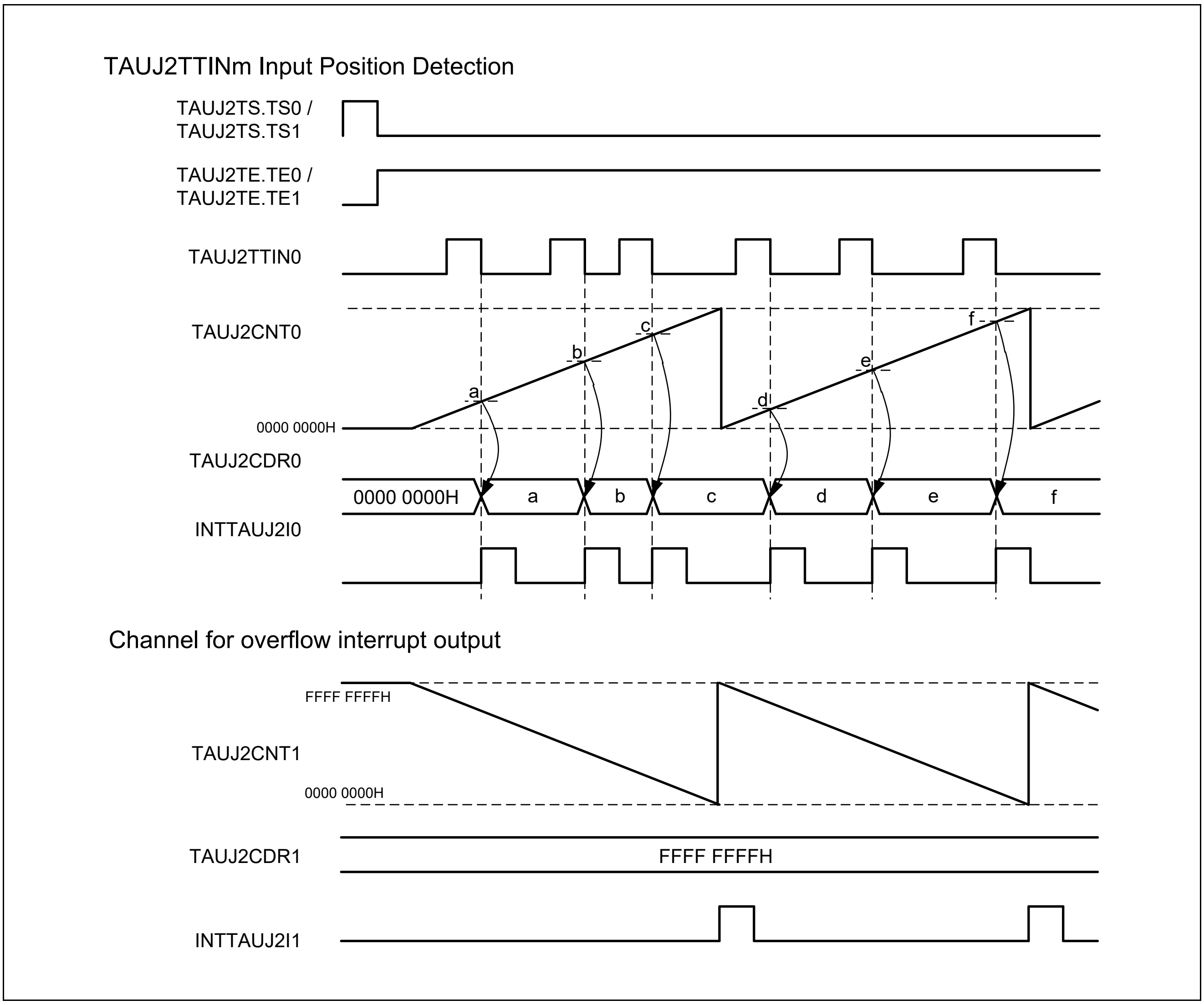


Figure 19.35 General Timing Diagram at the Time of Overflow Interrupt Output (when TAUJ2TTINm Input Position Detection is Used)

(d) Register settings for TAUJ2TTINm input position detection channel

Make settings for TAUJ2TTINm input position detection.

(e) Register settings for an overflow interrupt output channel

- TAUJ2CMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.44 TAUJ2CMORm Settings

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock that suits the application.
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	0: Independent operation
TAUJ2STS[2:0]	These bits select the external start trigger. 000: Software trigger
TAUJ2COS[1:0]	00: Not used (initial value)
TAUJ2MD[4:1]	These bits select the operating mode. 0000: Interval mode
TAUJ2MD0	This bit specifies whether an INTTAUJ2Im interrupt is generated when counting starts. 0: INTTAUJ2Im prohibited

- TAUJ2CMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														TAUJ2TIS[1:0]	

Table 19.45 TAUJ2CMURm Settings

Bit Name	Setting
TAUJ2TIS[1:0]	00: Not used (initial value)

(f) Simultaneous reloading

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with input signal width measurement by TAUJ2TTINm. Therefore, these registers must be set to 0.

Table 19.46 Simultaneous Reload Settings for TAUJ2TTINm Input Signal Width Measurement

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Set to 0 since this disables simultaneous reloading of channel m.
TAUJ2RDM.TAUJ2RDMm	0: Not used (initial value)

(g) Register settings for channel output

Table 19.47 Control Bit Settings for Independent Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	0: Set to 0 since this disables output operation of channel m.
TAUJ2TOM.TAUJ2TOMm	0: Not used (initial value)
TAUJ2TOC.TAUJ2TOCm	0: Not used (initial value)
TAUJ2TOL.TAUJ2TOLm	0: Not used (initial value)

(9) Operating Procedure for Overflow Interrupt Output

Table 19.48 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register (for 2 channels). • Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output (for 2 channels). • Set the value of the TAUJ2CDRm register for TAUJ2TTINm input position detection to 0000 0000H and the value of the TAUJ2CDRm register for the overflow interrupt output channel to FFFF FFFFH.	Channel operation is stopped.
Start Operation (Restart)	• Set TAUJ2TS.TAUJ2TSm for 2 channels to 1 simultaneously. • TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is set to 1 and counting starts. When the TAUJ2CDRm value (FFFF FFFFH) is updated in TAUJ2CNTm.
During Operation	No special notes	When TAUJ2CNTm counts down and the counter value reaches 0000 0000H, the value of TAUJ2CDRm is updated in TAUJ2CNTm and INTTAUJ2Im is generated. The counter resumes counting.
Stop Operation	Set TAUJ2TT.TAUJ2TTm to 1. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm stops and retains its current value.

19.8 Synchronous Channel Operation

19.8.1 PWM Output

(1) Overview

This function generates multiple PWM outputs by using a master and multiple slave channels.

The pulse cycle is set by a master channel and the duty cycle is set by a slave channel. This function requires at least two channels.

Caution: With this function, forced restarting cannot proceed.

(2) Block Diagram

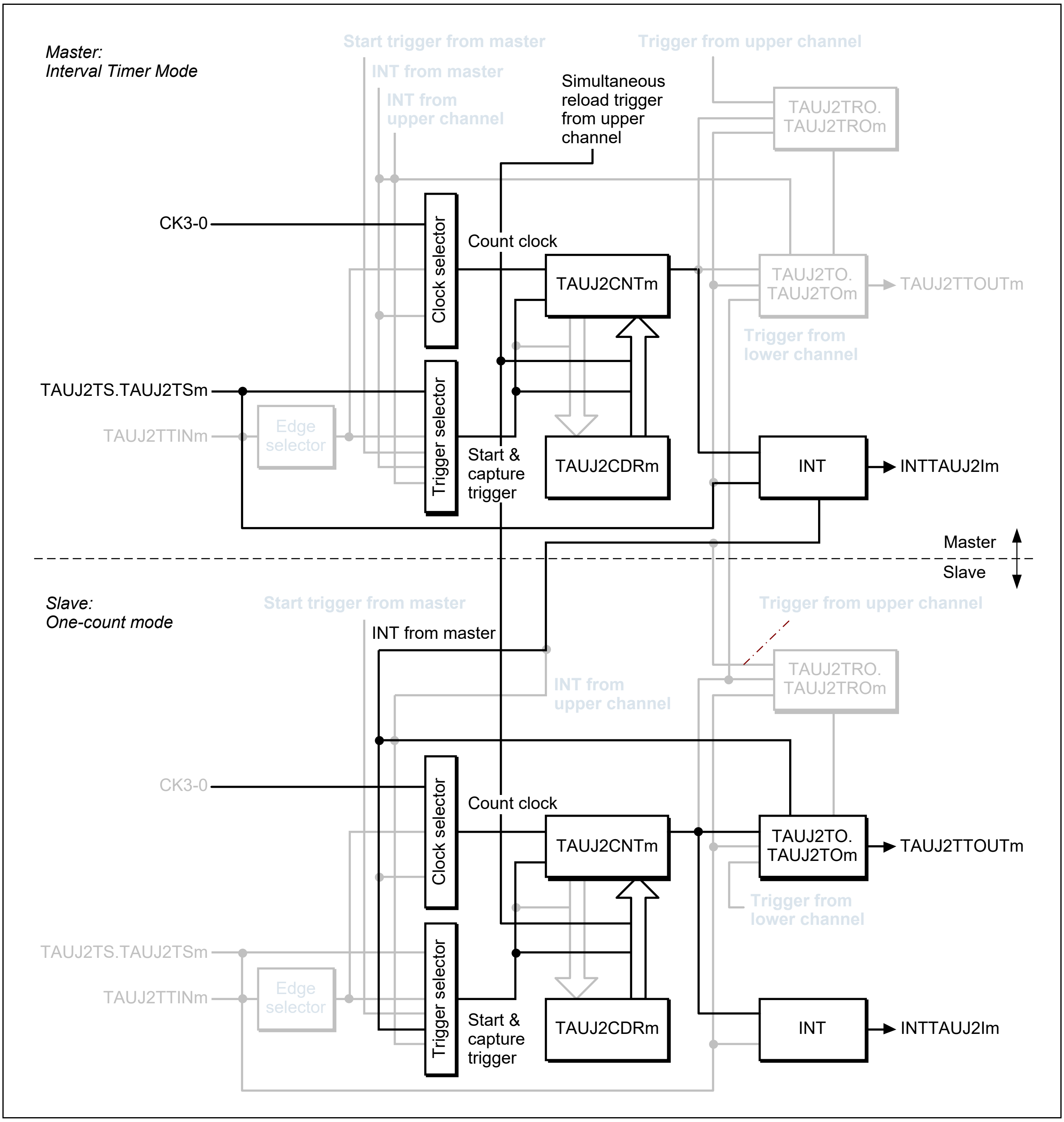


Figure 19.36 Block Diagram of PWM Output

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- Slave channel: Positive logic (TAUJ2TOL.TAUJ2TOLm = 0)

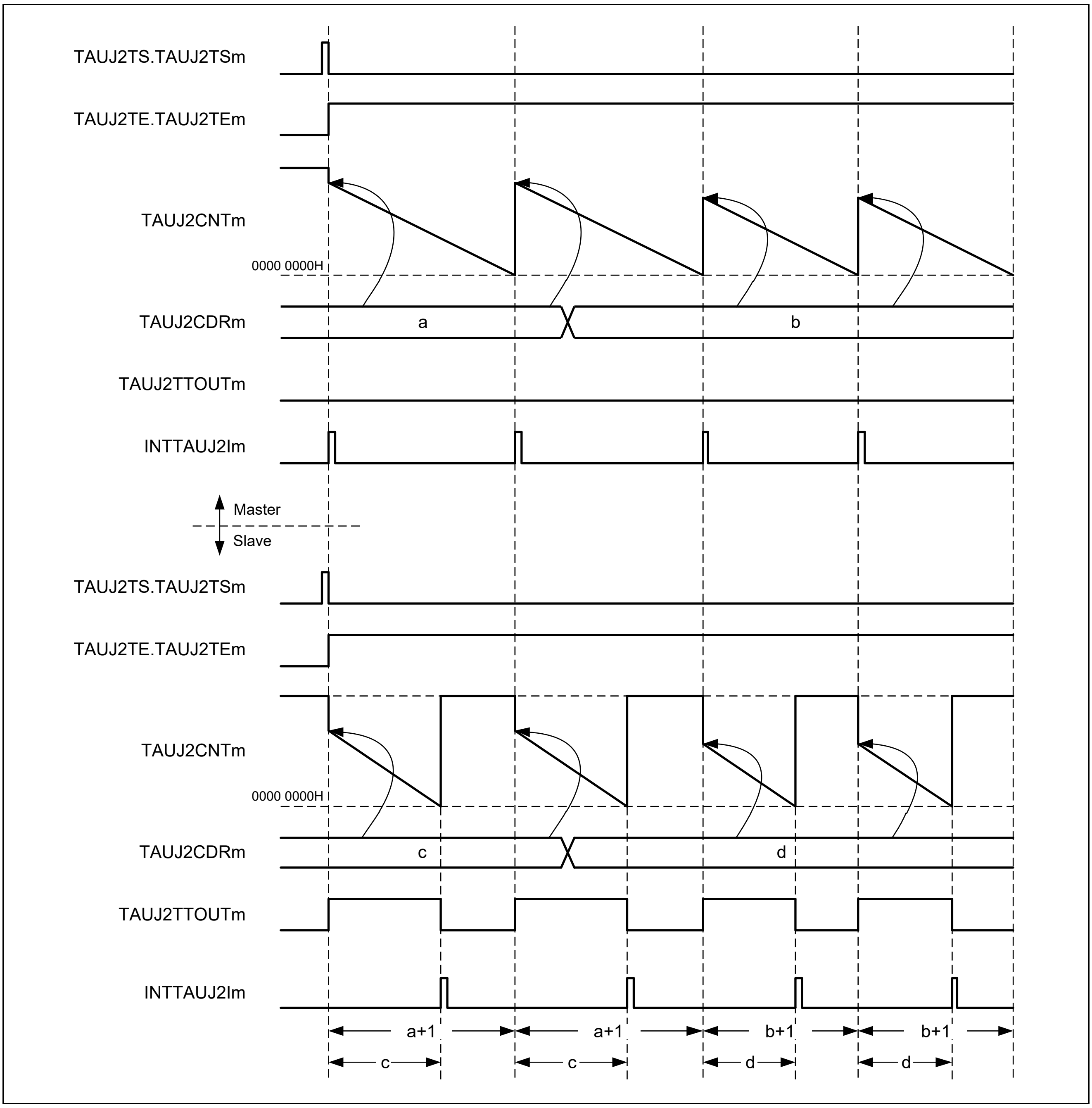


Figure 19.37 General Timing Diagram of PWM Output

Remark: The interval for the slave channels from the start of counting until an interrupt is generated is the corresponding TAUJ2CDRm value, whereas the interval for the master channel is the corresponding TAUJ2CDRm value + 1.

(4) Equations

Pulse cycle = (TAUJ2CDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUJ2CDRm (slave) / (TAUJ2CDRm (master) + 1)) × 100

- Duty cycle = 0 %
  TAUJ2CDRm (slave) = 0000 0000H
- Duty cycle = 100 %
  TAUJ2CDRm (slave) ≥ TAUJ2CDRm (master) + 1

(5) Register Settings for the Master Channel

(a) TAUJ2CMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.49 TAUJ2CMORm Settings for the Master Channel for PWM Output

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUJ2CKS[1:0] bits for the master and slave channel(s) must be identical.
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	These bits select the master/slave channel. 1: Master channel
TAUJ2STS[2:0]	These bits select the external start trigger. 000: Software trigger
TAUJ2COS[1:0]	00: Not used (initial value)
TAUJ2MD[4:1]	These bits select the operating mode. 0000: Interval timer mode
TAUJ2MD0	This bit specifies whether an INTTAUJ2Im interrupt is generated when counting starts. 1: INTTAUJ2Im permitted

(b) TAUJ2CMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														TAUJ2TIS[1:0]	

Table 19.50 TAUJ2CMURm Settings for the Master Channel for PWM Output

Bit Name	Setting
TAUJ2TIS[1:0]	00: Not used (initial value)

(c) Simultaneous Reloading of the Master Channel

The simultaneous reload registers (TAUJ2RDE and TAUJ2RDM) cannot be used with PWM output. Therefore, these registers must be set to 0.

Table 19.51 Simultaneous Reload Settings

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	This bit enables or disables simultaneous reloading of channels. 1: Enables simultaneous reloading
TAUJ2RDM.TAUJ2RDMm	This bit sets the timing for generating a simultaneous reload trigger. 0: The simultaneous reload trigger signal is generated when the master channel starts counting.

(d) Register settings for master channel output

Table 19.52 Control Bit Settings for Independent Channel Output

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	0: Set to 0 since this disables output operation of channel m.
TAUJ2TOM.TAUJ2TOMm	0: Not used (initial value)
TAUJ2TOC.TAUJ2TOCm	0: Not used (initial value)
TAUJ2TOL.TAUJ2TOLm	0: Not used (initial value)

(6) Register Settings for the Slave Channel(s)

(a) TAUJ2CMORm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJ2CKS[1:0]		TAUJ2CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2 MD0

Table 19.53 TAUJ2CMORm Settings for the Slave Channels for PWM Output

Bit Name	Setting
TAUJ2CKS[1:0]	These bits select prescaler output CK0 to CK3. 00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 Set the operation clock to the same setting as that for the slave channel(s).
TAUJ2CCS[1:0]	These bits set the counter clock. 00: Prescaler output (CK0 to CK3)
TAUJ2MAS	These bits select the master/slave channel. 0: Slave channel
TAUJ2STS[2:0]	These bits select the external start trigger. 100: Trigger for generating INTTAUJ2Im of the master channel
TAUJ2COS[1:0]	00: Not used (initial value)
TAUJ2MD[4:1]	These bits select the operating mode. 0100: One-count mode
TAUJ2MD0	This bit enables or disables detection of a start trigger during counting. 1: Enables detection of a start trigger.

(b) TAUJ2CMURm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														TAUJ2TIS[1:0]	

Table 19.54 TAUJ2CMURm Settings for the Slave Channel(s) for PWM Output

Bit Name	Setting
TAUJ2TIS[1:0]	00: Not used (initial value)

(c) Simultaneous reloading of the slave channel(s)

Table 19.55 Simultaneous Reload Settings

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	This bit enables or disables simultaneous reloading of channels. 1: Enables simultaneous reloading.
TAUJ2RDM.TAUJ2RDMm	This bit sets the timing for generating a simultaneous reload trigger. 0: The simultaneous reload trigger signal is generated when the master channel starts counting.

(d) Register settings for slave channel output

Table 19.56 Control Bit Setting in Independent Channel Output Mode 1

Bit Name	Setting
TAUJ2TOE.TAUJ2TOEm	This bit enables or disables TAUJ2TOm output operation by counting. 1: Enables the operation.
TAUJ2TOM.TAUJ2TOMm	This bit specifies independent or synchronous channel operation. 1: Synchronous channel operation
TAUJ2TOC.TAUJ2TOCm	This bit specifies the operating mode for TAUJ2TOm output of channels. The setting of this bit is as follows according to the setting of TAUJ2TOM.TAUJ2TOMm. 0: Synchronous operating mode 1 since TAUJ2TOM.TAUJ2TOMm = 1
TAUJ2TOL.TAUJ2TOLm	This bit sets the TAUJ2TOm output level of channels. 0: Positive logic output 1: Inverted logic output

(7) Operating Procedure for PWM Output

Table 19.57 Operating Procedure for PWM Output

		Operation	Status of TAUJ2
	Initial Channel Setting	• Use the TAUJ2TPS register to set the clock signal of the channel to be used. However, setting the clock signal of CK3 also requires setting the TAUJ2BRS register. • Master channel: Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output. • Slave channel: Set the TAUJ2CMORm and TAUJ2CMURm registers and the registers for channel output. • Set the carrier cycle in the TAUJ2CDRm register for a master channel and the duty cycle in the TAUJ2CDRm register for a slave channel.	Channel operation is stopped.
Restart	Start Operation	Set TAUJ2TS.TAUJ2TSm for the master and slave channels to 1 simultaneously. TAUJ2TS.TAUJ2TSm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TEm (for master/slave channels) is set to 1 and the master/slave channel counter starts. INTTAUJ2Im occurs on the master channel.
	During Operation	Registers whose value can be changed at any time • TAUJ2CDRm register • TAUJ2TOL.TAUJ2TOLm bit • TAUJ2RDT.TAUJ2RDTm bit (When simultaneous reloading is used) Register which is readable at any time • TAUJ2CNTm register	The master channel controls the cycle (TAUJ2CNTm register =0000 0000H). A slave channel controls the duty cycle and outputs a PWM waveform from TAUJ2TTOUTm.
	Stop Operation	Set TAUJ2TT.TAUJ2TTm for the master and slave channels to 1 simultaneously. TAUJ2TT.TAUJ2TTm is a trigger bit, so it is automatically cleared to 0.	TAUJ2TE.TAUJ2TEm is cleared to 0 and the counter stops. TAUJ2CNTm and TAUJ2TTOUTm stop and retain their current values.

(8) Specific Timing Diagrams

(a) Duty cycle = 0%

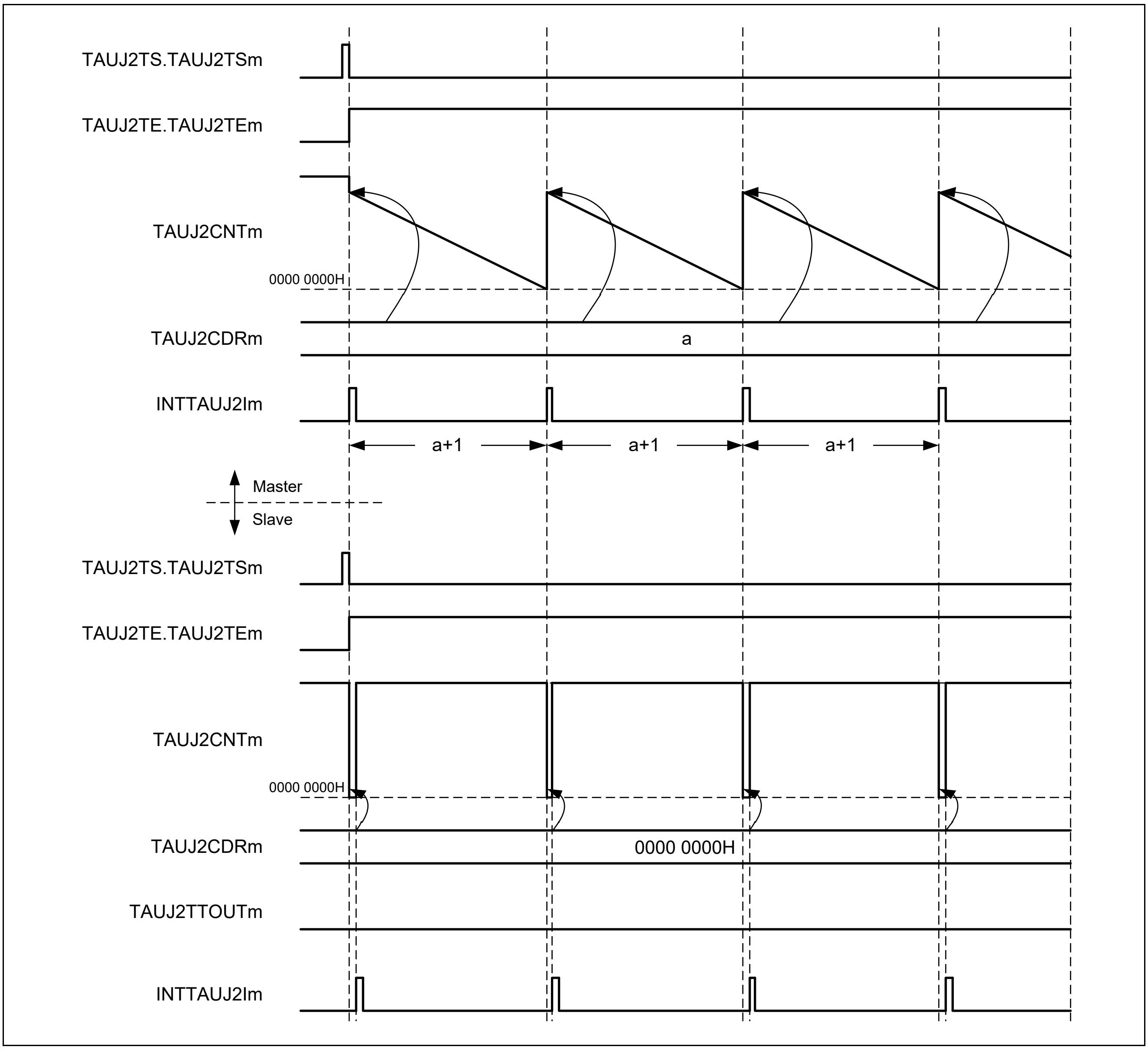


Figure 19.38 TAUJ2CDRm (Slave) = 0000 0000H, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0)

- The master channel generates an interrupt (INTTAUJ2Im) every a + 1 cycles. In response, TAUJ2CNTm (of the slave) is updated to 0000 0000H, and it also generates an interrupt and stops counting. TAUJ2TTOUTm remains at the low level.
- The TAUJ2CDRm value is updated to 0000 0000H in TAUJ2CNTm (of the slave) and an interrupt is generated.

(b) Duty cycle = 100%

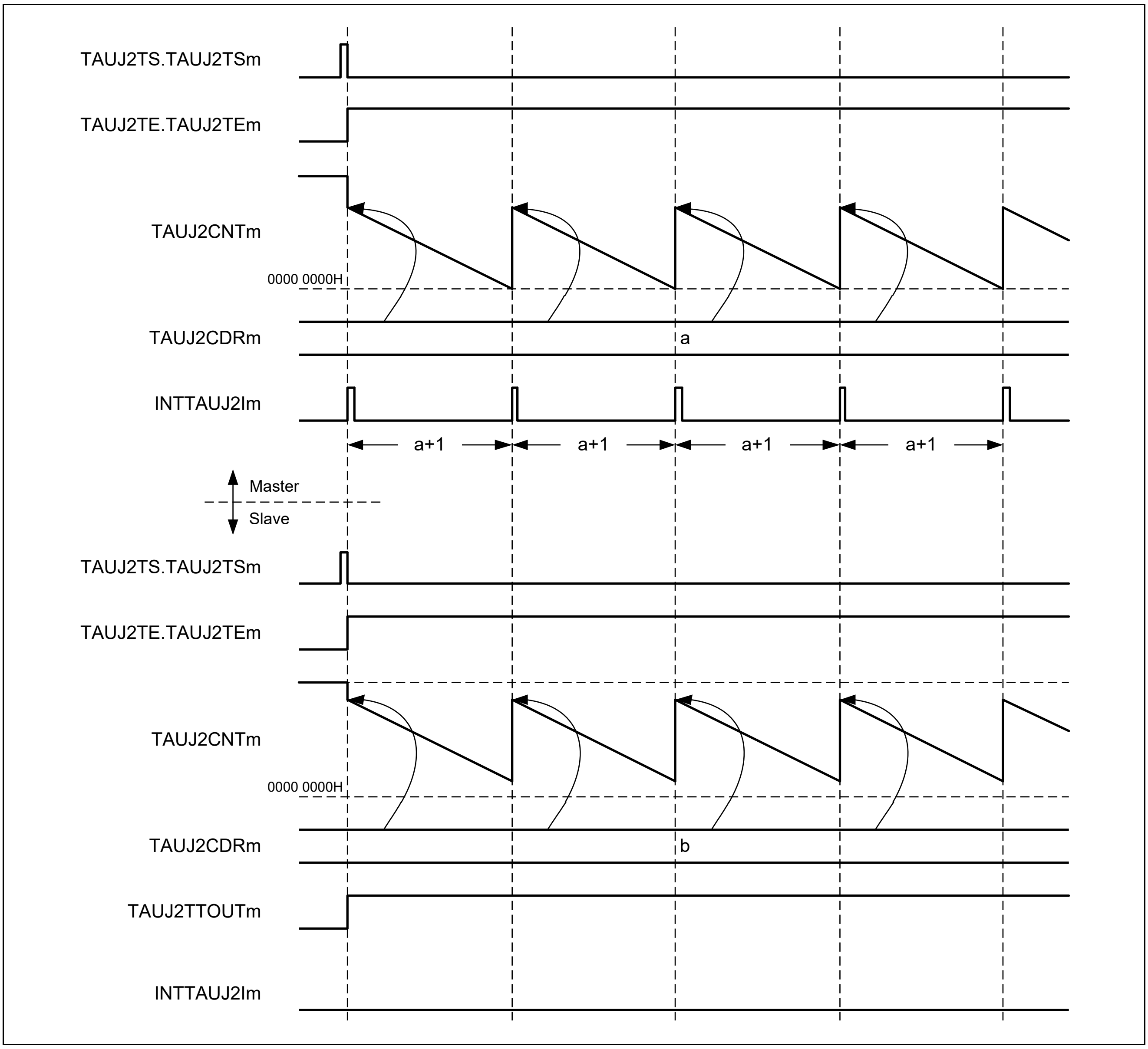


Figure 19.39 TAUJ2CDRm (Slave) ≥ TAUJ2CDRm (Master) + 1, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0)

- If the TAUJ2CDRm (slave) value is higher than the TAUJ2CDRm (master) value, the counter of the slave channel does not become 0000 0000H and is not reset, so TAUJ2TTOUTm remains at the high level.

(c) Stop of operation and restart

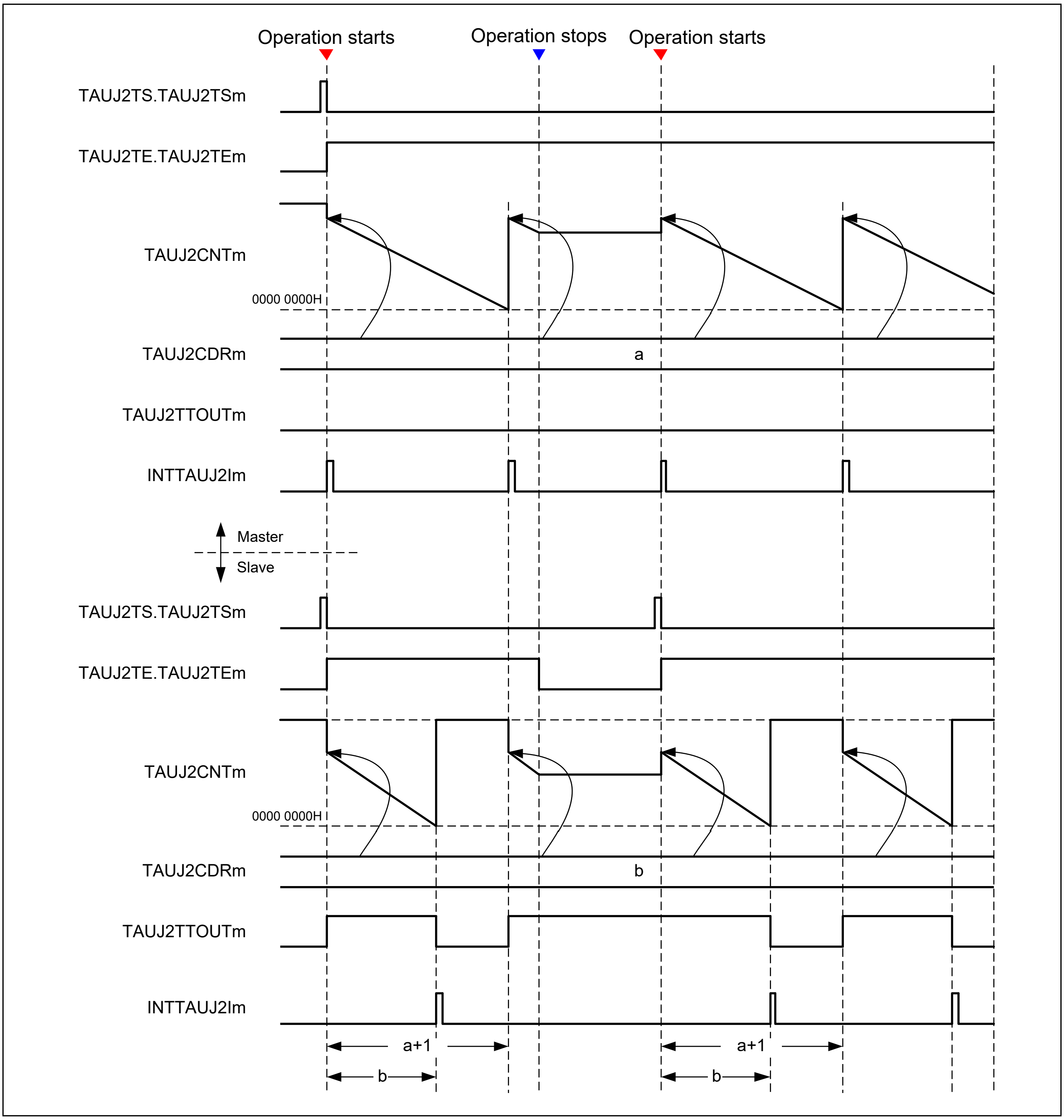


Figure 19.40 Stopping and Restarting Operation, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0)

- The counter can be stopped by setting TAUJ2TT.TAUJ2TTm for the master and slave channel(s) to 1, which in turn sets TAUJ2TE.TAUJ2TEm to 0.
- TAUJ2CNTm and TAUJ2TTOUTm of all channels stop and the current values are retained.
- The counter can be restarted by setting TAUJ2TS.TAUJ2TSm for master and slave channel(s) to 1.
- The TAUJ2CDRm value of master and slave channels is updated in TAUJ2CNTm, and it starts counting down.

(9) Simultaneous Reloading

(a) Functional description

Simultaneous reloading of the data register values for multiple channels (master/slave) (TAUJ2CDRm) and of the output values (TAUJ2TOL.TAUJ2TOLm) is possible.

Simultaneous reloading proceeds when the master channel starts counting in PWM output.

(b) General timing diagram and operation

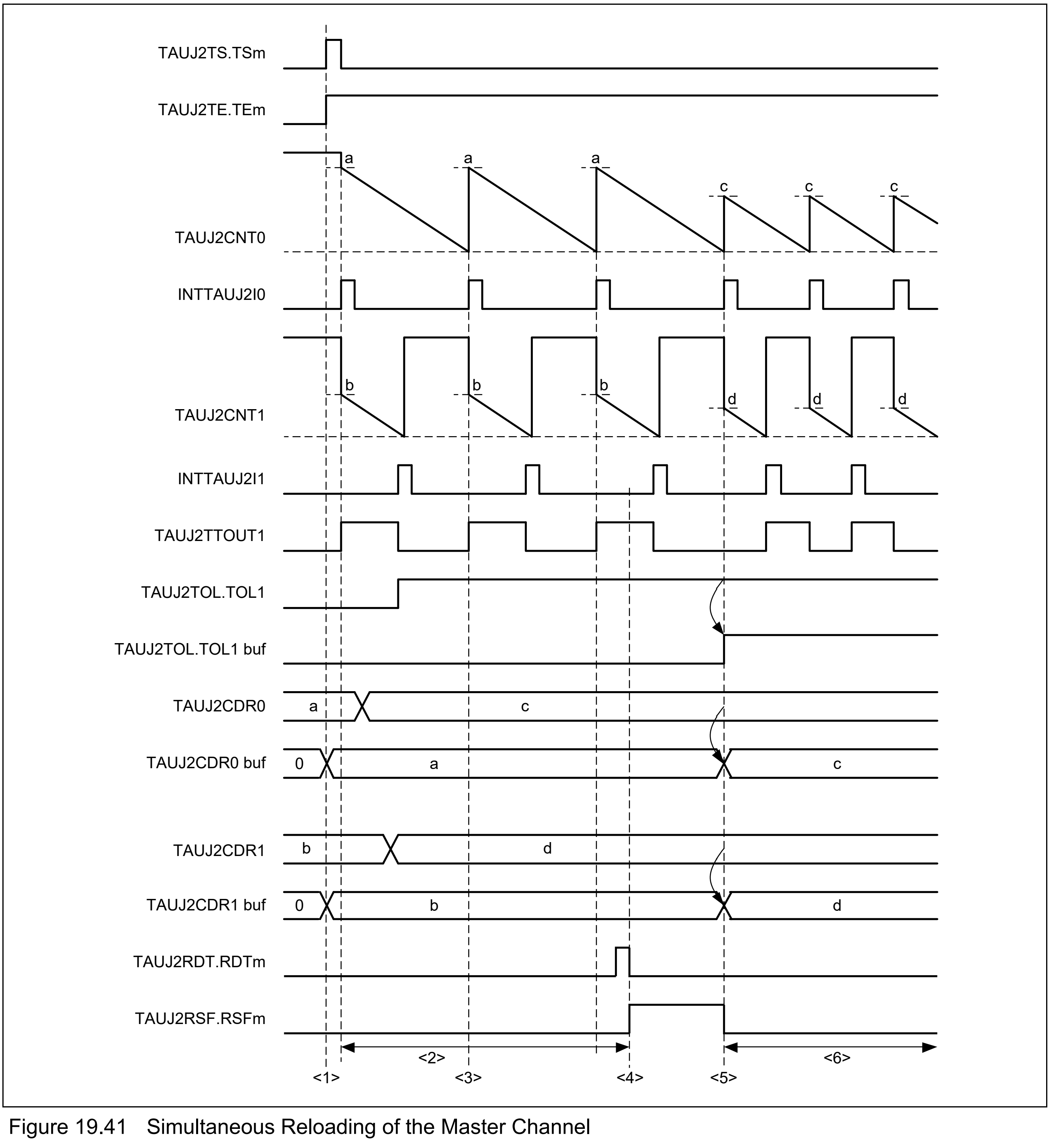


Figure 19.41 Simultaneous Reloading of the Master Channel

- Description:

1. When TAUJ2TS.TAUJ2TSm is set to 1, the values of TAUJ2CDRm and TAUJ2TOL.TAUJ2TOLm are updated in the TAUJ2CDRm and TAUJ2TOL.TAUJ2TOLm buffers respectively.
2. The TAUJ2CDRm and TAUJ2TOL.TAUJ2TOLm registers can be written at any time.
3. The TAUJ2CDRm and TAUJ2TOL.TAUJ2TOLm buffers are not updated because simultaneous reloading is not enabled (TAUJ2RSF.TAUJ2RSFm = 0).
4. Setting the reload data trigger bit (TAUJ2RDT.TAUJ2RDTm) to 1 leads to the status flag being set (TAUJ2RSF.TAUJ2RSFm = 1) and simultaneous reloading being enabled.
5. Simultaneous reloading proceeds when the master channel (CH0) restarts counting. The values of TAUJ2CDRm and TAUJ2TOL.TAUJ2TOLm are updated in the TAUJ2CDRm and TAUJ2TOL.TAUJ2TOLm buffers respectively.
6. The channel operates with the TAUJ2CDRm buffer value.

20. 16-Bit Timer Array Unit (TAUD)

This section explains the 16-bit timer array unit (TAUD).

20.1 Features of TAUD

- Number of units: 1 (n = 0)

Remark: In this section, the index "n" attached to TAUD (such as TAUDn) means n=0 or without index. For example, these three are the same: TAUDnTS, TAUD0TS, and TAUDTS.
Note that the index "m" which means the number of channels is different from "n" above (details are explained below).

- Meaning of "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.

Caution: Because TINDx and TOUTDx (x = 4 to 7) of TAUD are assigned as the alternative pins of TINJx and TOUTJx (x = 0 to 3) of TAUJ2 for the same ports, they cannot be used simultaneously. Select the pins to be used by using the register described in section 28.17, Timer Interface Select Register (TMISEL).
If an external pin is not used such as interval timer with internal clock, both TAUJ2 and TAUD channels can be used simultaneously.

- Input/Output Signals:Input/output signals of TAUD are listed below.

Table 20.1 TAUD Input/Output Signals

TAUD Signal	Function	Pin name
TAUDTTIN0	Channel 0–15 input ports	TIND0 (multiplexed with EXTP0)
TAUDTTIN1		TIND1 (multiplexed with EXTP1)
TAUDTTIN2		TIND2 (multiplexed with EXTP2)
TAUDTTIN3		TIND3 (multiplexed with EXTP3)
TAUDTTIN4		TIND4 (multiplexed with P27)
TAUDTTIN5		TIND5 (multiplexed with P26)
TAUDTTIN6		TIND6 (multiplexed with P57)
TAUDTTIN7		TIND7 (multiplexed with P52)
TAUDTTIN8		TIND8 (multiplexed with RP30)
TAUDTTIN9		TIND9 (multiplexed with RP31)
TAUDTTIN10		TIND10 (multiplexed with RP32)
TAUDTTIN11		TIND11 (multiplexed with RP33)
TAUDTTIN12		TIND12 (multiplexed with RP34)
TAUDTTIN13		TIND13 (multiplexed with RP35)
TAUDTTIN14		TIND14 (multiplexed with RP36)
TAUDTTIN15		TIND15 (multiplexed with RP37)
TAUDTTOUT0	Channel 0–15 output ports	TOUTD0 (multiplexed with EXTP0)
TAUDTTOUT1		TOUTD1 (multiplexed with EXTP1)
TAUDTTOUT2		TOUTD2 (multiplexed with EXTP2)
TAUDTTOUT3		TOUTD3 (multiplexed with EXTP3)
TAUDTTOUT4		TOUTD4 (multiplexed with P27)
TAUDTTOUT5		TOUTD5 (multiplexed with P26)
TAUDTTOUT6		TOUTD6 (multiplexed with P57)
TAUDTTOUT7		TOUTD7 (multiplexed with P52)
TAUDTTOUT8		TOUTD8 (multiplexed with RP30)
TAUDTTOUT9		TOUTD9 (multiplexed with RP31)
TAUDTTOUT10		TOUTD10 (multiplexed with RP32)
TAUDTTOUT11		TOUTD11 (multiplexed with RP33)
TAUDTTOUT12		TOUTD12 (multiplexed with RP34)
TAUDTTOUT13		TOUTD13 (multiplexed with RP35)
TAUDTTOUT14		TOUTD14 (multiplexed with RP36)
TAUDTTOUT15		TOUTD15 (multiplexed with RP37)

Caution: TINDm and TOUTDm are used in the same ports. If TINDm and TOUTDm are used at the same time, input signals to TINDm need to be set to pins except port pins (m = 0 to 7). For details, see Section 28.9, Timer Input Function Select Register (SELCNT, SELCNTD).

- Interrupts and peripheral modules:
  The following interrupt requests from TAUD can be used for interrupt processing and hardware ISR as well as DMA transfer (by the general-purpose DMAC or real-time port DMAC) and for triggering capture by a timer (TAUJ2 or TAUD) and updating the real-time ports (RP00–RP37).

Table 20.2 TAUD Interrupts and Requests for Peripheral Modules

(1/2)

TAUD Interrupt Signal	Function	Connected to
INTTAUDI0	Channel 0 interrupt	• Interrupt controller INTTAUD0I0 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI1	Channel 1 interrupt	• Interrupt controller INTTAUD0I1 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI2	Channel 2 interrupt	• Interrupt controller INTTAUD0I2 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI3	Channel 3 interrupt	• Interrupt controller INTTAUD0I3 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI4	Channel 4 interrupt	• Interrupt controller INTTAUD0I4 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI5	Channel 5 interrupt	• Interrupt controller INTTAUD0I5 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI6	Channel 6 interrupt	• Interrupt controller INTTAUD0I6 • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)

(2/2)

TAUD Interrupt Signal	Function	Connected to
INTTAUDI7	Channel 7 interrupt	• Interrupt controller INTTAUD0I7 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI8	Channel 8 interrupt	• Interrupt controller INTTAUD0I8 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI9	Channel 9 interrupt	• Interrupt controller INTTAUD0I9 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI10	Channel 10 interrupt	• Interrupt controller INTTAUD0I10 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI11	Channel 11 interrupt	• Interrupt controller INTTAUD0I11 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI12	Channel 12 interrupt	• Interrupt controller INTTAUD0I12 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI13	Channel 13 interrupt	• Interrupt controller INTTAUD0I13 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI14	Channel 14 interrupt	• Interrupt controller INTTAUD0I14 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTTAUDI15	Channel 15 interrupt	• Interrupt controller INTTAUD0I15 • HW-RTOS(Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)

20.1.1 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Caution: TAUD supports only functions listed in Table 20.3.
Register settings for functions other than those described in Table 20.3 are prohibited.

Table 20.3 Functional List of TAUD Operations

Operation Function	Example
Independent Channel Operations	Section 20.12
Interval Timer	Section 20.12.1
TAUDTTINm Input Interval Timer	Section 20.12.2
Clock Frequency Division	Section 20.12.3
External Event Counting	Section 20.12.4
Delay Counting	Section 20.12.5
One-Pulse Output	Section 20.12.6
TAUDTTINm Input Pulse Interval Measurement	Section 20.12.7
TAUDTTINm Input Signal Width Measurement	Section 20.12.8
TAUDTTINm Input Position Detection	Section 20.12.9
TAUDTTINm Input Period Count Detection	Section 20.12.10
TAUDTTINm Input Pulse Interval Judgment	Section 20.12.11
TAUDTTINm Input Signal Width Judgment	Section 20.12.12
Overflow Interrupt Output (during TAUDTTINm Width Measurement)	Section 20.12.13
Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)	Section 20.12.14
One-Phase PWM Output	Section 20.12.15
Independent Channel Real-Time Functions	Section 20.13
Real-Time Output Type 1	Section 20.13.1
Real-Time Output Type 2	Section 20.13.2
Independent Channel Simultaneous Reloading	Section 20.14
Simultaneous Reload Trigger Generation Type 1	Section 20.14.1
Simultaneous Reload Trigger Generation Type 2	Section 20.14.2
Synchronous Channel Operation Functions	Section 20.15
PWM Output	Section 20.15.1
One-Shot Pulse Output	Section 20.15.2
Trigger Start PWM Output	Section 20.15.3
Delay Pulse Output	Section 20.15.4
Offset Trigger Output	Section 20.15.5
Triangle PWM Output	Section 20.15.6
Triangle PWM Output with Dead Time	Section 20.15.7
Skipping Interrupt Request Signals	Section 20.15.8
Synchronous Non-Complementary and Complementary Modulation Output	Section 20.16
Synchronous Non-Complementary Modulation Output Type 1	Section 20.16.1
Synchronous Non-Complementary Modulation Output Type 2	Section 20.16.2
Complementary Modulation Output	Section 20.16.3

20.2 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

20.2.1 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

The channel operation defines all functions and 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.

Figure 20.1 shows the main components of the TAUD.

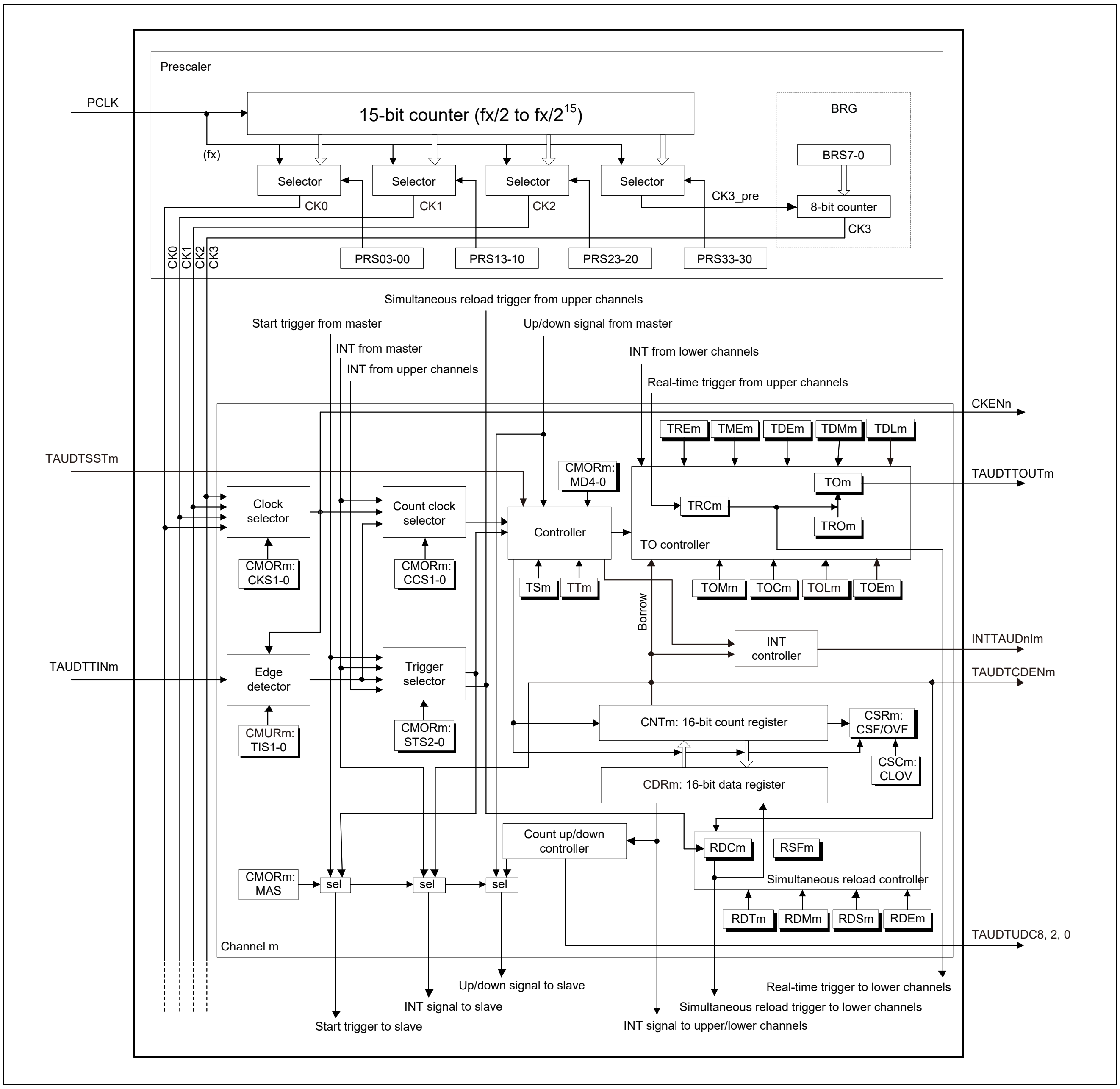


Figure 20.1 Block Diagram of the TAUD

20.2.2 Description of Blocks

This section describes the each operation of TAUD control 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 20 to 215. 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)
- INTTAUDIm from master channel
- Effective edge of the TAUDTTINm input signal

• Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUDCMORm.TAUDMD[4:0])
- Counter start enable (TAUDTS.TAUDTSm) and counter stop (TAUDTT.TAUDTTm) when counter start is enabled, status flag TAUDTE.TAUDTEm 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 (TAUDTE.TAUDTEm = 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 TAUDTSSTm
- Effective edge of the TAUDTTINm input signal
- INTTAUDIm 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 reload controller

Simultaneous reload control is used in synchronous operating modes. The data registers (TAUDCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous reload controller ensures that new data register values of all channels become effective at the same time.

• TAUDTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

20.3 Registers

This section contains the description of all 16 Bit TAUD registers.

Caution: TAUD supports only functions listed in Table 20.4.
Register settings for functions other than those described in Table 20.4 are prohibited.

20.3.1 List of Registers

TAUD controls and operates by the registers listed below in Table 20.4.

Table 20.4 List of Registers

(1/2)

Register	Symbol	Address
TAUD prescaler registers		
TAUD prescaler clock select register	TAUDTPS	4000 0A40H
TAUD prescaler baud rate setting register	TAUDBRS	4000 0A44H
TAUD control registers		
TAUD channel data register m	TAUDCDRm	4000 0800H + m × 4H
TAUD channel counter register m	TAUDCNTm	4000 0880H + m × 4H
TAUD channel mode OS register m	TAUDCMORm	4000 0A00H + m × 4H
TAUD channel mode user register m	TAUDCMURm	4000 08C0H + m × 4H
TAUD channel status register m	TAUDCSRm	4000 0940H + m × 4H
TAUD channel status clear trigger register m	TAUDCSCm	4000 0980H + m × 4H
TAUD channel start trigger register	TAUDTS	4000 09C4H
TAUD channel enable status register	TAUDTE	4000 09C0H
TAUD channel stop trigger register	TAUDTT	4000 09C8H
TAUD reload data registers		
TAUD channel reload data enable register	TAUDRDE	4000 0A60H
TAUD channel reload data mode register	TAUDRDM	4000 0A64H
TAUD channel reload data control CH select register	TAUDRDS	4000 0A68H
TAUD channel reload data control register	TAUDRDC	4000 0A6CH
TAUD channel reload data trigger register	TAUDRDT	4000 0844H
TAUD channel reload status register	TAUDRSF	4000 0848H
TAUD output registers		
TAUD channel output enable register	TAUDTOE	4000 085CH
TAUD channel output register	TAUDTO	4000 0858H
TAUD channel output mode register	TAUDTOM	4000 0A48H
TAUD channel output configuration register	TAUDTOC	4000 0A4CH
TAUD channel output active level register	TAUDTOL	4000 0840H
TAUD dead time output register		
TAUD channel dead time output enable register	TAUDTDE	4000 0A50H
TAUD channel dead time output mode register	TAUDTDM	4000 0A54H
TAUD channel dead time output level register	TAUDTDL	4000 0854H

(2/2)

Register	Symbol	Address
TAUD real-time/modulation output register		
TAUD channel real-time output register	TAUDTRO	4000 084CH
TAUD channel real-time output enable register	TAUDTRE	4000 0A58H
TAUD channel real-time output control register	TAUDTRC	4000 0A5CH
TAUD channel modulation output enable register	TAUDTME	4000 0850H
TAUD Emulation Register		
TAUD emulation register	TAUDEMU	4000 0A90H

20.3.2 Details of TAUD Prescaler Registers

(1) TAUD Prescaler Clock Select Register (TAUDTPS)

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 TAUDBRS.

- Access Readable/writable in 16-bit units.

(1/4)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial Value
TAUDTPS	TAUDPRS3[3:0]				TAUDPRS2[3:0]				TAUDPRS1[3:0]				TAUDPRS0[3:0]				4000 0A40H	FFFFH
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Function
15 to 12	TAUDPRS3[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. (see table below) The above bits are rewritable only when all the counters using CK3 are stopped (TAUDTE.TAUDTEm = 0).

TAUDPRS3[3:0]	CK3_PRE Clock
0000B	$PCLK/2^{0}$
0001B	$PCLK/2^{1}$
0010B	$PCLK/2^{2}$
0011B	$PCLK/2^{3}$
0100B	$PCLK/2^{4}$
0101B	$PCLK/2^{5}$
0110B	$PCLK/2^{6}$
0111B	$PCLK/2^{7}$
1000B	$PCLK/2^{8}$
1001B	$PCLK/2^{9}$
1010B	$PCLK/2^{10}$
1011B	$PCLK/2^{11}$
1100B	$PCLK/2^{12}$
1101B	$PCLK/2^{13}$
1110B	$PCLK/2^{14}$
1111B	$PCLK/2^{15}$

(2/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUDPRS2[3:0]</td><td>Specifies the CK2 clock.
<table><tr><th>TAUDPRS2[3:0]</th><th>CK2 Clock</th></tr><tr><td>0000B</td><td>PCLK/2^0</td></tr><tr><td>0001B</td><td>PCLK/2^1</td></tr><tr><td>0010B</td><td>PCLK/2^2</td></tr><tr><td>0011B</td><td>PCLK/2^3</td></tr><tr><td>0100B</td><td>PCLK/2^4</td></tr><tr><td>0101B</td><td>PCLK/2^5</td></tr><tr><td>0110B</td><td>PCLK/2^6</td></tr><tr><td>0111B</td><td>PCLK/2^7</td></tr><tr><td>1000B</td><td>PCLK/2^8</td></tr><tr><td>1001B</td><td>PCLK/2^9</td></tr><tr><td>1010B</td><td>PCLK/2^{10}</td></tr><tr><td>1011B</td><td>PCLK/2^{11}</td></tr><tr><td>1100B</td><td>PCLK/2^{12}</td></tr><tr><td>1101B</td><td>PCLK/2^{13}</td></tr><tr><td>1110B</td><td>PCLK/2^{14}</td></tr><tr><td>1111B</td><td>PCLK/2^{15}</td></tr></table>
The above bits are rewritable only when all the counters using CK2 are stopped (TAUDTE.TAUDTEm = 0).</td></tr>
</table>

(3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 4</td><td>TAUDPRS1[3:0]</td><td>Specifies the CK1 clock.
<table><tr><th>TAUDPRS1[3:0]</th><th>CK1 Clock</th></tr><tr><td>0000B</td><td>PCLK/2^0</td></tr><tr><td>0001B</td><td>PCLK/2^1</td></tr><tr><td>0010B</td><td>PCLK/2^2</td></tr><tr><td>0011B</td><td>PCLK/2^3</td></tr><tr><td>0100B</td><td>PCLK/2^4</td></tr><tr><td>0101B</td><td>PCLK/2^5</td></tr><tr><td>0110B</td><td>PCLK/2^6</td></tr><tr><td>0111B</td><td>PCLK/2^7</td></tr><tr><td>1000B</td><td>PCLK/2^8</td></tr><tr><td>1001B</td><td>PCLK/2^9</td></tr><tr><td>1010B</td><td>PCLK/2^{10}</td></tr><tr><td>1011B</td><td>PCLK/2^{11}</td></tr><tr><td>1100B</td><td>PCLK/2^{12}</td></tr><tr><td>1101B</td><td>PCLK/2^{13}</td></tr><tr><td>1110B</td><td>PCLK/2^{14}</td></tr><tr><td>1111B</td><td>PCLK/2^{15}</td></tr></table>
The above bits are rewritable only when all the counters using CK1 are stopped (TAUDTE.TAUDTEm = 0).</td></tr>
</table>

(4/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>TAUDPRS0[3:0]</td><td>Specifies the CK0 clock.
<table><tr><th>TAUDPRS0[3:0]</th><th>CK0 Clock</th></tr><tr><td>0000B</td><td>PCLK/2^0</td></tr><tr><td>0001B</td><td>PCLK/2^1</td></tr><tr><td>0010B</td><td>PCLK/2^2</td></tr><tr><td>0011B</td><td>PCLK/2^3</td></tr><tr><td>0100B</td><td>PCLK/2^4</td></tr><tr><td>0101B</td><td>PCLK/2^5</td></tr><tr><td>0110B</td><td>PCLK/2^6</td></tr><tr><td>0111B</td><td>PCLK/2^7</td></tr><tr><td>1000B</td><td>PCLK/2^8</td></tr><tr><td>1001B</td><td>PCLK/2^9</td></tr><tr><td>1010B</td><td>PCLK/2^{10}</td></tr><tr><td>1011B</td><td>PCLK/2^{11}</td></tr><tr><td>1100B</td><td>PCLK/2^{12}</td></tr><tr><td>1101B</td><td>PCLK/2^{13}</td></tr><tr><td>1110B</td><td>PCLK/2^{14}</td></tr><tr><td>1111B</td><td>PCLK/2^{15}</td></tr></table>
The above bits are rewritable only when all the counters using CK0 are stopped (TAUDTE.TAUDTEm = 0).</td></tr>
</table>

(2) TAUD Prescaler Baud Rate Setting Register (TAUDBRS)

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 TAUDTPS.TAUDPRS3[3:0].

- Access Readable/writable in 8-bit units.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUDBRS	TAUDBRS[7:0]								4000 0A44H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 0	TAUDBRS[7:0]	Specifies a CK3_PRE clock division factor for generating CK3.

TAUDBRS[7:0]	CK3 Clock
0000 0000B	CK3_PRE / 1
0000 0001B	CK3_PRE / 2
0000 0010B	CK3_PRE / 3
0000 0011B	CK3_PRE / 4
…	…
1111 1110B	CK3_PRE / 255
1111 1111B	CK3_PRE / 256

20.3.3 Details of TAUD Control Registers

(1) TAUD Channel Data Register (TAUDCDRm)

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUDCMORm.TAUDMD[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.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TAUDCDRm	TAUDCDR[15:0]																4000 0800H + m × 4H Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDCDR[15:0]	Data register for capture/compare values

(2) TAUD Channel Counter Register (TAUDCNTm)

This is a channel m counter register.

- Access Readable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TAUDCNTm	TAUDCNT[15:0]																4000 0880H + m × 4H Initial Value FFFFH Note
R/W	R																

Bit Position	Bit Name	Function
15 to 0	TAUDCNT[15:0]	16-bit counter value

Note: An initial value changes according to the operation mode set by the TAUD channel mode OS register. The initial value is 0000H in capture mode, capture and one-count mode, count capture mode, and capture and gate count mode. The initial value is FFFFH in other modes.
For details of setting operation mode, see Section 20.3.3(3), TAUD Channel Mode OS Register (TAUDCMORm).

(3) TAUD Channel Mode OS Register (TAUDCMORm)

This register controls channel m operation.

- Access Readable/writable in 16-bit units.
  Writable only when the counter is stopped (TAUDTE.TAUDTEm = 0).

(1/4)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TAUD CMORm	TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:0]					4000 0A00H + m × 4H Initial Value 0000H
R/W	R/W		R/W		R/W	R/W			R/W		R	R/W					

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Selects an operation clock. An operation clock is used for the TAUDTTINm input edge detection circuit. Setting of TAUDCMORm.TAUDCCS[1:0] bits also allow the operation clock to serve as the TAUDCNTm counter clock. TAUDCKS1 / TAUDCKS0 / Selection of Operation Clock: 0 / 0 / CK0 0 / 1 / CK1 1 / 0 / CK2 1 / 1 / CK3
13, 12	TAUDCCS[1:0]	Selects a count clock for TAUDCNTm counter. TAUDCCS1 / TAUDCCS0 / Selection of Count Clock: 0 / 0 / Operation clock specified by TAUDCMORm.TAUDCKS[1:0] 0 / 1 / Effective edge of TAUDTTINm input signal 1 / 0 / Setting prohibited 1 / 1 / INTTAUDIm signal of master channel
11	TAUDMAS	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-numbered channels (CHm_even). Odd-numbered channels (CHm_odd) are fixed to 0.

(2/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUDSTS[2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUDSTS2</th><th>TAUDSTS1</th><th>TAUDSTS0</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>Effective edge of TAUDTTINm input signal, which is specified by TAUDCMURm.TAUDTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Effective 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 reloading.</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>
</table>

(3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>TAUDCOS[1:0]</td><td>Specifies the timing for updating capture register TAUDCDRm and overflow flag TAUDCSRm.TAUDOVF of channel m.
These bits are valid only when channel m is in capture mode or capture one count mode
<table>
<tr><th>TAUD COS1</th><th>TAUD COS0</th><th>TAUDCDRm</th><th>TAUDCSRm.TAUDOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of an effective edge of TAUDTTINm input.</td><td>Updated (cleared or set) by detecting an effective edge of TAUDTTINm input:
• If a counter overflow has occurred since the last detection of an effective edge, set TAUDCSRm.TAUDOVF.
• If no counter overflow has occurred since the last detection of an effective edge, clear TAUDCSRm.TAUDOVF.</td></tr>
<tr><td>0</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUDCSCm.TAUDCLOV is set to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of effective edge of TAUDTTINm input and at the occurrence of counter overflow:
• Detection of effective edge of TAUDTTINm input signal:
Counter value is written into TAUDCDRm.
• Occurrence of overflow:
FFFFH is loaded into TAUDCDRm. Detection of the next effective edge of the TAUDTTINm input signal is ignored.</td><td>Updated upon detection of effective edge of TAUDTTINm input and at the occurrence of counter overflow:</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUDCSCm.TAUDCLOV is set to 1.</td></tr>
</table>
</td></tr>
<tr><td>5</td><td>—</td><td>Reserved. This bit is read as 0.</td></tr>
</table>

(4/4)

Bit Position	Bit Name	Function
4 to 0	TAUDMD[4:0]	Specifies an operating mode.

TAUD MD4	TAUD MD3	TAUD MD2	TAUD MD1	TAUD 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	Mode Role of TAUDMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUDIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDIm is not generated. 1: INTTAUDIm is generated.
Event count mode Count-up/-down mode	This bit should be set to 0. (INTTAUDIm is not generated at the beginning of count operation.)
One-count mode Note1 Pulse one-count mode Note2	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection.
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. Note3
Judge mode Judge and one-count mode	Specifies INTTAUDIm output timing. 0: When TAUDCNTm ≤ TAUDCDRm 1: When TAUDCNTm > TAUDCDRm

Notes 1. INTTAUDIm signal is not output at the beginning of count operation in one-count mode.

2. INTTAUDIm signal is output at the beginning of count operation in pulse one-count mode.

3. INTTAUDIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.

(4) TAUD Channel Mode User Register m (TAUDCMURm)

This register specifies a type of effective edge detection used for TAUDTTINm input.

- Access Readable/writable in 8-bit units.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUD CMURm	0	0	0	0	0	0	TAUDTIS[1:0]		4000 08C0H + m × 4H	00H
R/W	0	0	0	0	0	0	R/W			

Bit Position	Bit Name	Function
7 to 2	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
1, 0	TAUDTIS[1:0]	Specifies an effective edge of TAUDTTINm input signal. (see table below) Edge detection of TAUDTTINm input signal is based on the operation clock selected by TAUDCMORm.TAUDCKS[1:0].

TAUDTIS1	TAUDTIS0	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

(5) TAUD Channel Status Register m (TAUDCSRm)

This register indicates the count direction and overflow status of channel m counter.

- Access Only readable in 8-bit units.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUD CSRm	0	0	0	0	0	0	TAUD CSF	TAUD OVF	4000 0940H + m × 4H	00H
R/W	0	0	0	0	0	0	R	R		

Bit Position	Bit Name	Function
7 to 2	—	Reserved. These bits are read as 0.
1	TAUDCSF	Indicates a count direction. 0: Count-up 1: Count-down The read value of this bit is valid only in the count-up/-down mode.
0	TAUDOVF	Indicates counter overflow status. 0: No overflow occurs. 1: Overflow occurs. This bit is used only in the capture mode or capture and one-count mode. The function of this bit depends on the setting of control bit TAUDCMORm.TAUDCOS[1:0].

(6) TAUD Channel Status Clear Trigger Register m (TAUDCSCm)

This is a trigger register for clearing the overflow flag TAUDCSRm.TAUDOVF of channel m.

- Access Only writable in 8-bit units.

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUD CSCm	0	0	0	0	0	0	0	TAUD CLOV	4000 0980H + m × 4H	00H
R/W	0	0	0	0	0	0	0	W		

Bit Position	Bit Name	Function
7 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	TAUDCLOV	0: No function 1: Clears overflow flag TAUDCSRm.TAUDOVF.

(7) TAUD Channel Start Trigger Register (TAUDTS)

This register enables the counter operation of each channel.

- Access Only writable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
																	4000 09C4H
TAUDTS	TAUDTSm																Initial Value
																	0000H
R/W	W																

Bit Position	Bit Name	Function
15 to 0	TAUDTSm	Enables the counter operation of channel m. 0: No function 1: Enables the counter operation and sets TAUDTE.TAUDTEm to 1. The counter operation is only enabled when TAUDTE.TAUDTEm is set to 1. Whether counting is started or not depends on a selected operating mode.

(8) TAUD Channel Enable Status Register (TAUDTE)

This register enables/disables a counter operation.

- Access Only readable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
																	4000 09C0H
TAUDTE	TAUDTEm																Initial Value
																	0000H
R/W	R																

Bit Position	Bit Name	Function
15 to 0	TAUDTEm	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 TAUDTSSTm (synchronous channel start trigger signal) is detected or when TAUDTS.TAUDTSm is set to 1. This bit is set to 0 when TAUDTT.TAUDTTm is set to 1.

(9) TAUD Channel Stop Trigger Register (TAUDTT)

This register stops the counter operation of each channel.

- Access Only writable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TAUDTT	TAUDTTm																4000 09C8H Initial Value 0000H
R/W	W																

Bit Position	Bit Name	Function
15 to 0	TAUDTTm	Stops the counter operation of channel m. 0: No function 1: Stops the counter operation and resets TAUDTE.TAUDTEm. TAUDCNTm, TAUDTO.TAUDTOm, and TAUDTTOUTm retain the values provided before the counter is stopped.

20.3.4 Details of TAUD Simultaneous Reload Registers

(1) TAUD Channel Reload Data Enable Register (TAUDRDE)

This register enables/disables simultaneous reloading of TAUDCDRm and TAUDTOLm data registers.

- Access　Readable/writable in 16-bit units.
  Writable only while TAUDTE.TAUDTEm = 0.

	15–0	
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Address 4000 0A60H
TAUDRDE	TAUDRDEm	Initial Value 0000H
R/W	R/W	

Bit Position	Bit Name	Function
15 to 0	TAUDRDEm	Enables/disables simultaneous reloading of the data registers of channel m. 0: Disables simultaneous reloading 1: Enables simultaneous reloading

(2) TAUD Channel Reload Data Mode Register (TAUDRDM)

This register selects the timing for generating a simultaneous reload control signal.

- Access　Readable/writable in 16-bit units.
  Writable only while TAUDTE.TAUDTEm = 0.

	15–0	
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Address 4000 0A64H
TAUDRDM	TAUDRDMm	Initial Value 0000H
R/W	R/W	

Bit Position	Bit Name	Function
15 to 0	TAUDRDMm	Selects the timing for generating a simultaneous reload 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 TAUDRDE.TAUDRDEm = 1 and TAUDRDS.TAUDRDSm = 0.

(3) TAUD Channel Reload Data Control Channel Select Register (TAUDRDS)

This register selects a channel that controls simultaneous reloading.

- Access Readable/writable in 16-bit units.
  Writable only while TAUDTE.TAUDTEm = 0.

	15–0	
TAUDRDS	TAUDRDSm	Address 4000 0A68H, Initial Value 0000H
R/W	R/W	

Bit Position	Bit Name	Function
15 to 0	TAUDRDSm	Selects a channel that controls a simultaneous reload trigger. 0: Master channel 1: Another upper channel

(4) TAUD Channel Reload Data Control Register (TAUDRDC)

This register specifies a channel which generates an INTTAUDIm signal to trigger simultaneous reloading.

- Access Readable/writable in 16-bit units.
  Writable only while TAUDTE.TAUDTEm = 0.

	15–0	
TAUDRDC	TAUDRDCm	Address 4000 0A6CH, Initial Value 0000H
R/W	R/W	

Bit Position	Bit Name	Function
15 to 0	TAUDRDCm	Specifies whether the channel generates a simultaneous reload trigger signal or not. 0: Does not operate as a simultaneous reload trigger channel. 1: Operates as a simultaneous reload trigger channel. These bit settings are applied only when TAUDRDS.TAUDRDSm = 1.

(5) TAUD Channel Reload Data Trigger Register (TAUDRDT)

This register triggers a simultaneous reload enabled state.

- Access Only writable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TAUDRDT	TAUDRDTm																Address 4000 0844H Initial Value 0000H
R/W	W																

Bit Position	Bit Name	Function
15 to 0	TAUDRDTm	Triggers a simultaneous reload enabled state. 0: No function 1: The simultaneous reload enable flag (TAUDRSFm) is set to 1. The system waits for a trigger for simultaneous reloading. These bits only apply when TAUDRDE.TAUDRDEm = 1.

(6) TAUD Channel Reload Status Register (TAUDRSF)

This flag register indicates simultaneous reload status.

- Access Only readable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TAUDRSF	TAUDRSFm																Address 4000 0848H Initial Value 0000H
R/W	R																

Bit Position	Bit Name	Function
15 to 0	TAUDRSFm	Indicates simultaneous reload status. 0: Indicates that simultaneous reloading has been completed due to the generation of a trigger for simultaneous reloading. 1: Indicates that the system waits for a trigger for simultaneous reloading in the simultaneous reload enabled state (TAUDRDTm = 1).

20.3.5 Details of TAUD Output Registers

(1) TAUD Channel Output Enable Register (TAUDTOE)

This register enables/disables the independent channel output mode controlled by software.

- Access Readable/writable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TAUDTOE	TAUDTOEm																Address 4000 085CH Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTOEm	Enables/disables the independent channel output. 0: Disables the independent timer output (controlled by software). 1: Enables the independent timer output.

(2) TAUD Channel Output Register (TAUDTO)

This register specifies and reads a TAUDTTOUTm level.

- Access Readable/writable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TAUDTO	TAUDTOm																Address 4000 0858H Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTOm	Specifies and reads a TAUDTTOUTm level. 0: Low level 1: High level Only TAUDTOm bits for which Independent Channel Output is disabled (TAUDTOEm = 0) can be written.

(3) TAUD Channel Output Mode Register (TAUDTOM)

This register specifies the output mode of each channel.

- Access Readable/writable in 16-bit units.
  Writable only while the counter is stopped (TAUDTE.TAUDTEm = 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4000 0A48H
TAUDTOM	TAUDTOMm																Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTOMm	Specifies an output mode. 0: Independent channel operation 1: Synchronous channel operation

(4) TAUD Channel Output Configuration Register (TAUDTOC)

This register specifies the output mode of each channel in combination with TAUDTOMm.

- Access Readable/writable in 16-bit units.
  Writable only while the counter is stopped (TAUDTE.TAUDTEm = 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4000 0A4CH
TAUDTOC	TAUDTOCm																Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTOCm	Specifies an output mode. 0: Operating mode 1 1: Operating mode 2 As listed below, the output mode depends on the setting of TAUDTOM.TAUDTOMm.

TAUDTOMm	TAUDTOCm	Functional Description
0	0	Toggle mode: Toggle operation is conducted when INTTAUDIm occurs.
0	1	Set/reset mode: Set when INTTAUDIm occurs at the beginning of count operation, and reset when INTTAUDIm is caused by detection of a match between TAUDCNTm and TAUDCDRm.
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 INTTAUDIm occurs in count-down status, and reset when INTTAUDIm occurs in count-up status.

(5) TAUD Channel Output Active Level Register (TAUDTOL)

This register specifies the output logic of channel output bit (TAUDTO.TAUDTOm).

- Access Readable/writable in 16-bit units.
  Writable only while the counter is stopped (TAUDTE.TAUDTEm = 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4000 0840H
TAUDTOL	TAUDTOLm																Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTOLm	Specifies the output logic of channel m output bit (TAUDTO.TAUDTOm). 0: Positive logic (active high) 1: Negative logic (active low)

20.3.6 Details of TAUD Dead Time Output Registers

(1) TAUD Channel Dead Time Output Enable Register (TAUDTDE)

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 (TAUDTE.TAUDTEm = 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TAUDTDE	TAUDTDEm																4000 0A50H Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTDEm	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 TAUDTOE.TAUDTOEm, TAUDTOM.TAUDTOMm, TAUDTOC.TAUDTOCm = 1

(2) TAUD Channel Dead Time Output Mode Register (TAUDTDM)

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 (TAUDTE.TAUDTEm = 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4000 0A54H
TAUDTDM	TAUDTDMm																Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTDMm	Specifies the timing to add dead time during dead time output. 0: When detecting the duty cycle of an upper even-numbered channel (duty dead time output). 1: When detecting the TIN input edge of a lower odd-numbered 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 TAUDTOE.TAUDTOEm, TAUDTOM.TAUDTOMm, TAUDTOC.TAUDTOCm, TAUDTDE.TAUDTDEm = 1.

(3) TAUD Channel Dead Time Output Level Register (TAUDTDL)

This register selects a phase in which dead time is added.

- Access Readable/writable in 16-bit units.
  Writable only while the counter is stopped (TAUDTE.TAUDTEm = 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4000 0854H
TAUDTDL	TAUDTDLm																Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTDLm	Selects a phase in which dead time is added. 0: Normal phase 1: Reverse phase These bit settings are applied when TAUDTOE.TAUDTOEm, TAUDTOM.TAUDTOMm, TAUDTOC.TAUDTOCm, TAUDTDE.TAUDTDEm = 1.

20.3.7 Details of TAUD Real-Time/Modulation Output Registers

(1) TAUD Channel Real-Time Output Register (TAUDTRO)

This register sets a value which is output to TAUDTTOUTm.

- Access Readable/writable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TAUDTRO	TAUDTROm																4000 084CH Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTROm	Sets a value which is output to TAUDTTOUTm. 0: Low 1: High TAUDTROm value is not output to TAUDTTOUTm when TAUDTRE.TAUDTREm = 0, even if a real-time output trigger occurs.

(2) TAUD Channel Real-Time Output Enable Register (TAUDTRE)

This register enables/disables real-time output.

- Access Readable/writable in 16-bit units.
  Writable only while the counter is stopped (TAUDTE.TAUDTEm = 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TAUDTRE	TAUDTREm																4000 0A58H Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTREm	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 TAUDTOE.TAUDTOEm = 1.These bit settings are applied only when TAUDTOE.TAUDTOEm = 1. When TAUDTRE.TAUDTREm = 0, TAUDTTOUTm is not affected by real-time output. When TAUDTRE.TAUDTREm = 1, TAUDTTOUTm outputs the value of real-time output bit TAUDTRO.TAUDTROm in response to a timer operation.

(3) TAUD Channel Real-Time Output Control Register (TAUDTRC)

This register controls the real-time output trigger of each channel.

- Access Readable/writable in 16-bit units.
  Writable only while the counter is stopped (TAUDTE.TAUDTEm = 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4000 0A5CH
TAUDTRC	TAUDTRCm																Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTRCm	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 TAUDTRE.TAUDTREm = 1.

(4) TAUD Channel Modulation Output Enable Register (TAUDTME)

This register enables/disables modulation output for timer output and real-time output.

- Access Readable/writable in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 4000 0850H
TAUDTME	TAUDTMEm																Initial Value 0000H
R/W	R/W																

Bit Position	Bit Name	Function
15 to 0	TAUDTMEm	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 TAUDTOE.TAUDTOEm and TAUDTRE.TAUDTREm = 1.

20.3.8 Details of TAUD Emulation Register

(1) TAUD Emulation Register (TAUDEMU)

This register controls SVSTOP operations.

- Access Readable/writable in 8-bit units.
  Perform write operations when the counter is being stopped (TAUDTE.TAUDTEm = 0).

	7	6	5	4	3	2	1	0	Address	Initial Value
TAUD EMU	TAUD SVSDIS	0	0	0	0	0	0	0	4000 0A90H	00H
R/W	R/W	0	0	0	0	0	0	0		

Bit Position	Bit Name	Function
7	TAUDSVSDIS	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. These bits are read as 0.

20.4 Operating Procedure

The following lists the general operation procedure for the TAUD. 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 TAUDTPS and TAUDBRS registers to specify the clock frequency of CK0 to CK3.

(2) Configure the desired TAUD function:

– Set the operation mode
– Set the channel output mode
– Set any other control bits

(3) Enable the counter by setting the TAUDTS.TAUDTSm 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 TAUDTT.TAUDTTm bit to 1. The counter can be forcibly restarted by setting the TAUDTS.TAUDTSm bit to 1.

(5) Stop the function by setting the TAUDTT.TAUDTTm bit to 1.

Remark: For operations and register settings of each function, see the detailed description of each function.

20.5 Concepts of Synchronous Channel Operation

Channel groups (consisting of master and slave channels) are combined to realize synchronous channel operation. Several rules apply to the settings of channels. These rules are detailed in Section 20.5.1, Rules of Synchronous Channel Operation. Two special features for synchronous channel operation are detailed in the following:

- Section 20.5.2, Simultaneous Start and Stop of Synchronous Channel Counters.
- Section 20.6, Simultaneous Reloading.

20.5.1 Rules of Synchronous Channel Operation

(1) Number of Masters and Slaves

- Only even-numbered 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.

(2) Operation Clock

- The same operation clock must be set for the slave channel and the master channel. This is achieved using the TAUDCMORm.TAUDCKS[1:0] bits of the slave and master channel.

The basic concepts of master/slave usage and operation clocks are illustrated in Figure 20.2, Grouping of Channels and Assignment of Count Clocks.

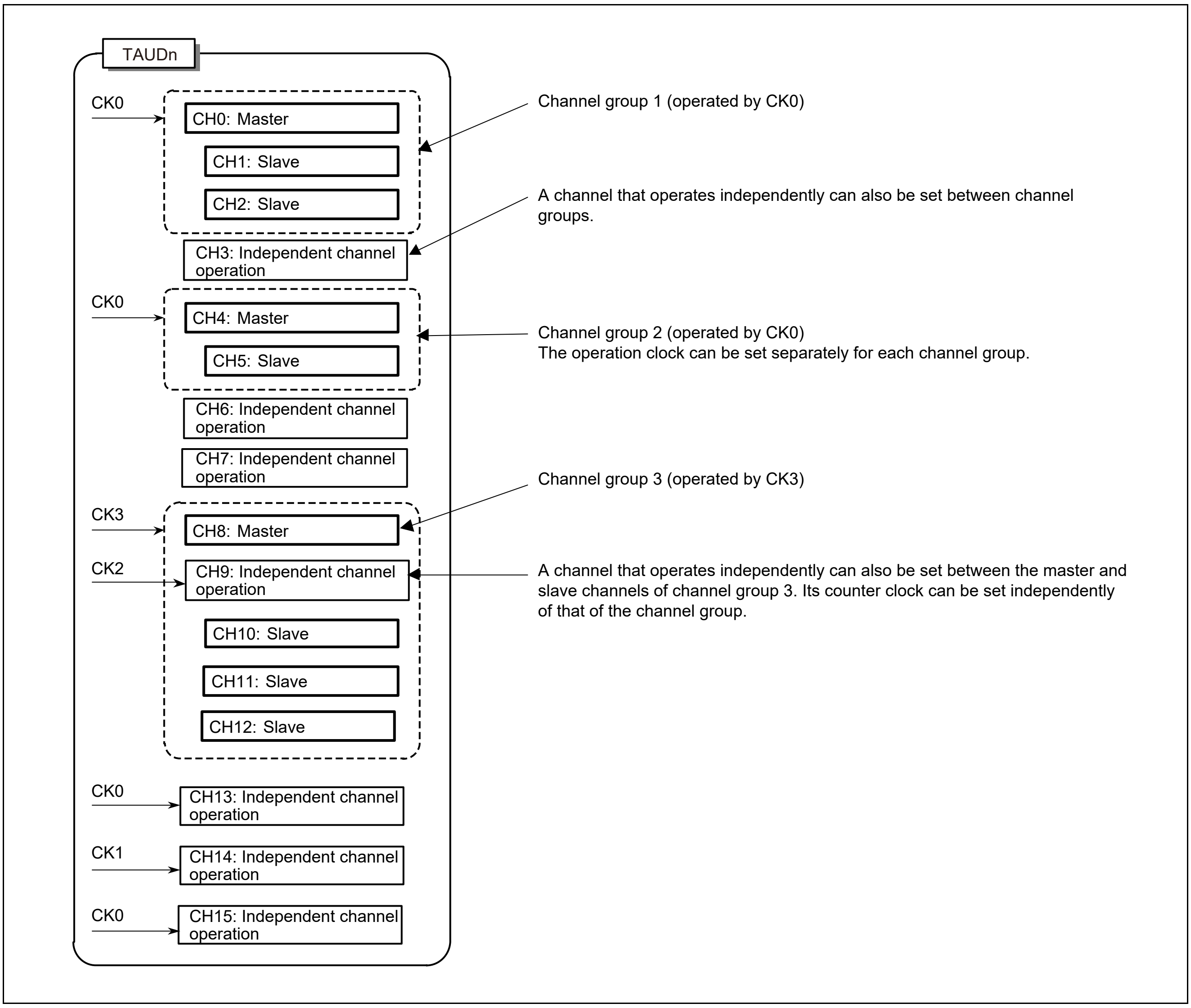


Figure 20.2 Grouping of Channels and Assignment of Count Clocks

(3) 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.

20.5.2 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.

(1) Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUDTS.TAUDTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUDTT.TAUDTTm bits of the channels should be set at the same time.

Setting to the TAUDTS.TAUDTSm bits to 1 also sets the corresponding TAUDTE.TAUDTEm bits to 1, enabling counting. The count start timing depends on operating mode.

20.6 Simultaneous Reloading

20.6.1 Overview of Operations

Simultaneous reloading refers to the ability to change the compare/start value and the output logic of multiple channels at the same time. The corresponding data and control registers (TAUDCDRm and TAUDTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous reloading is triggered.

Simultaneous reloading can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUDIm being issued on the upper channel specified by TAUDRDC.TAUDRDCm

There are four methods for simultaneous reloading. These are listed in Table 20.5, along with how to specify them and when they cause simultaneous reloading to be triggered.

Table 20.5 Simultaneous Reloading Methods and when They are Triggered

Method	Simultaneous Reloading Triggered when	TAUDRDE. TAUDRDEm	TAUDRDS. TAUDRDSm	TAUDRDM. TAUDRDMm
—	No simultaneous reloading	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	INTTAUDIm is generated on an upper channel specified by TAUDRDC.TAUDRDCm	1	1	0/1
C2	INTTAUDIm is generated on an upper channel specified by TAUDRDC.TAUDRDCm that in turn is triggered by an external signal	1	1	0/1

Table 20.6 lists which of these four methods is available for each channel operation. For more information about the individual channel operations, see the corresponding sections in 20.12, Independent Channel Operation and 20.15, Synchronous Channel Operation Functions.

Table 20.6 Channel Operations and Available Methods

Function	A	B	C1	C2	TAUDTOL. TAUDTOLm
Simultaneous Reload Trigger Output Type 1			✓		
PWM Output	✓		✓		✓
One-Shot Pulse Output	✓				
Trigger Start PWM Output	✓			✓	
Delay Pulse Output	✓				
Triangle PWM Output		✓	✓		✓
Triangle PWM Output with Dead Time		✓	✓		
Interrupt Request Signals Culling	✓	✓	✓		
Non-Complementary Modulation Output Type 1	✓		✓		
Non-Complementary Modulation Output Type 2		✓	✓		
Complementary Modulation Output		✓	✓		

Remark: ✓: Available; (Blank): Unavailable

20.6.2 How to Control Simultaneous Reloading

Figure 20.3 shows the general procedure for simultaneous reloading. The three main blocks (initial settings, start and counter count operation, and simultaneous reloading) are explained afterwards.

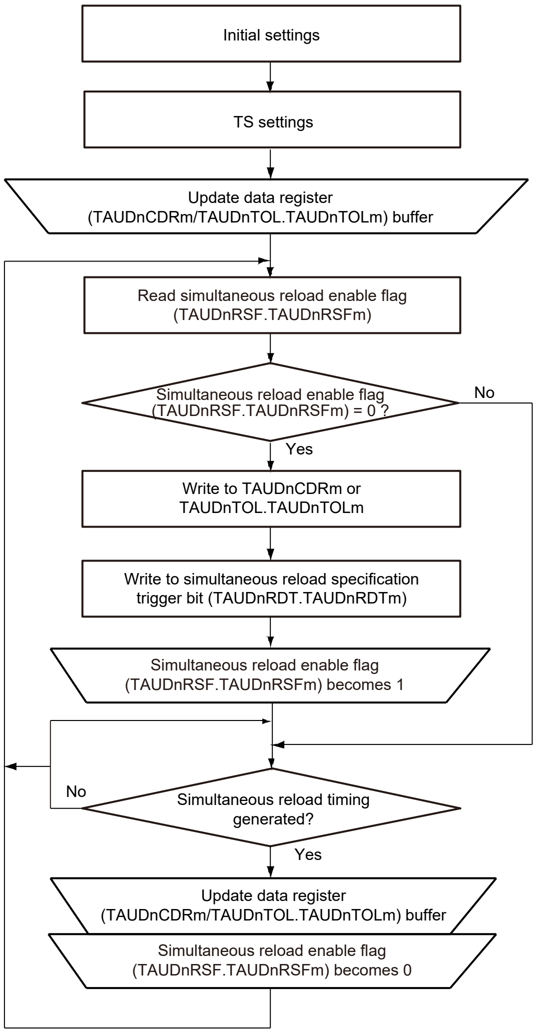


Figure 20.3 General Procedure for Simultaneous Reloading

(1) Initial Settings

- To enable simultaneous reloading of channel m, set TAUDRDE.TAUDRDEm = 1
- To select the type of simultaneous reloading, set TAUDRDM.TAUDRDMm and TAUDRDS.TAUDRDSm according to the values listed in Table 20.5, Simultaneous Reloading Methods and when They are Triggered.
- To select the channel for generation of a trigger for simultaneous reloading, use TAUDRDC.TAUDRDCm (prerequisite: TAUDRDS.TAUDRDSm is set in upper channel).

(2) Start Counter and Count Operation

- To start all the TAUDCNTm counters of the channel group, set the corresponding TAUDTS.TAUDTSm bits to1. The values of TAUDTOL.TAUDTOLm and the data registers (TAUDCDRm) are loaded into the corresponding TAUDTOL.TAUDTOLm buffer (TAUDTOL.TAUDTOLm buf) and data buffer registers (TAUDCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUDRDT.TAUDRDTm) to 1 sets the reload flag TAUDRSF.TAUDRSFm) to 1, enabling simultaneous reloading. TAUDRSF.TAUDRSFm remains set to 1 until simultaneous reloading is completed.
- When the specified trigger for simultaneous reloading is detected, the TAUDRSF.TAUDRSFm bit is checked to see if simultaneous reloading is enabled (TAUDRSF.TAUDRSFm = 1). If it is, simultaneous reloading is carried out. Otherwise simultaneous reloading is not carried out and waits for the next trigger detection.

(3) Simultaneous Reloading

- When simultaneous reloading is enabled (TAUDRSF.TAUDRSFm = 1) and a trigger for simultaneous reloading 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 TAUDRSF.TAUDRSFm bit is set to 0, and the system awaits the next trigger for simultaneous reloading.

20.6.3 Other General Rules of Simultaneous Reloading

The following rules also apply:

- TAUDRDE.TAUDRDEm, TAUDRDS.TAUDRDSm, TAUDRDM.TAUDRDMm, and TAUDRDC.TAUDRDCm cannot be changed while the counter is in operation (TAUDTE.TAUDTEm = 1).
- TAUDTOL.TAUDTOLm can only be rewritten during operation with PWM output or triangle PWM output. For all other outputs, TAUDTOL.TAUDTOLm 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 trigger for simultaneous reloading (TAUDRDS.TAUDRDSm = 1), the TAUDRDC.TAUDRDCm bit controls all the lower channels. This means that if the TAUDRDC.TAUDRDCm bits of CH2 and CH7 are set to 1 and the TAUDRDC.TAUDRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous reload trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous reloading is enabled and an upper channel is selected for generation of a trigger for simultaneous reloading (TAUDRDE.TAUDRDEm and TAUDRDS.TAUDRDSm = 1) but no upper channel is set (TAUDRDC.TAUDRDC[15:0] = 0), simultaneous reloading cannot take place.

20.6.4 Types of Simultaneous Reloading

In the following section, four methods of simultaneous reloading are explained using timing diagrams.

(1) Simultaneous Reloading when the Master Channel (Re)starts Counting (Method A)

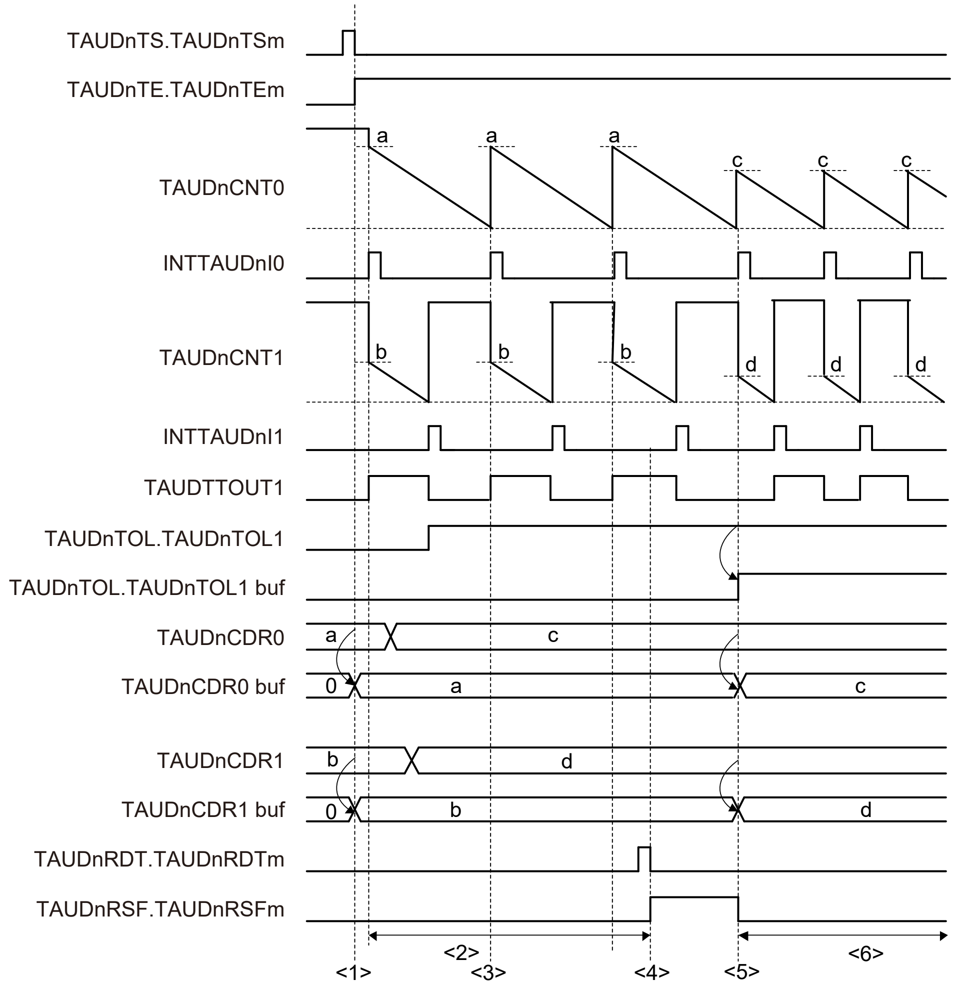


Figure 20.4 Simultaneous Reloading when the Master Channel (Re)starts Counting

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous reloading method A is applied.

Description:

(1) When TAUDTS.TAUDTSm is set to 1, TAUDCDRm value is copied to the TAUDCDRm buffer and TAUDTOL.TAUDTOLm value is copied to the TAUDTOL.TAUDTOLm buffer.

(2) The TAUDCDRm and TAUDTOL.TAUDTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous reloading does not occur because it is disabled (TAUDRSF.TAUDRSFm = 0)

(4) The reload data trigger bit (TAUDRDT.TAUDRDTm) is set to 1 which sets the status flag (TAUDRSF.TAUDRSFm = 1), enabling simultaneous reloading.

(5) Because simultaneous reloading is enabled, it is triggered when CH0 restarts counting. The TAUDCDRm value is loaded into the TAUDCDRm buffer and the TAUDTOL.TAUDTOLm value is loaded into the TAUDTOL.TAUDTOLm buffer.

(6) The counters count down and await the next trigger for simultaneous reloading. The values of TAUDCDRm and TAUDTOL.TAUDTOLm can be changed again.

(2) Simultaneous Reloading at the Peak of a Triangular Wave of Slave Channel (Method B)

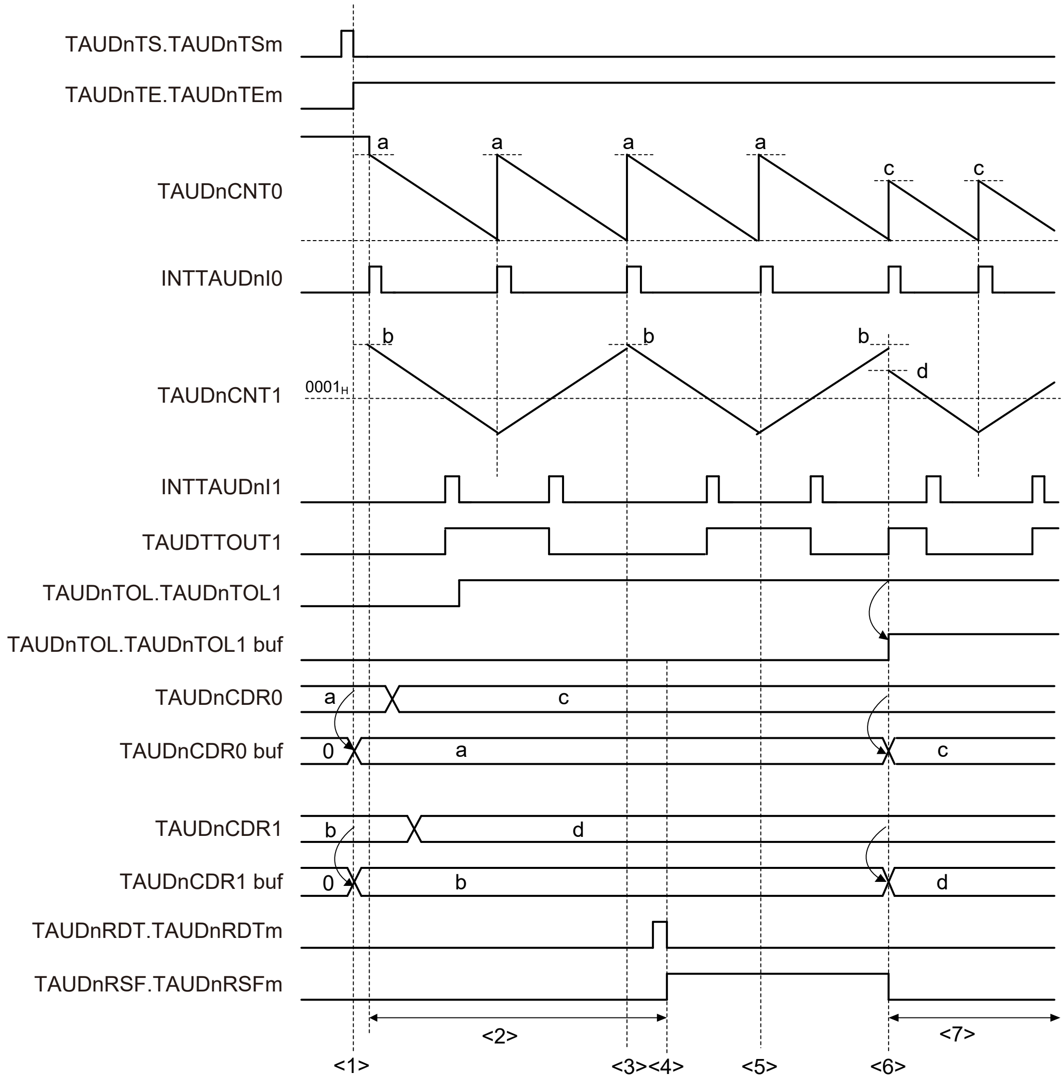


Figure 20.5 Simultaneous Reloading at the Peak of a Triangular Wave of Slave Channel

Setting:

CH0 is the master channel which performs counting down, and CH1 represents an arbitrary slave channel. The simultaneous reloading method B is applied.

Description:

(1) When TAUDTS.TAUDTSm is set to 1, TAUDCDRm value is copied to the TAUDCDRm buffer.

(2) The TAUDCDRm and TAUDTOL registers can be written at any time.

(3) Simultaneous reloading does not occur because it is disabled (TAUDRSF.TAUDRSFm = 0).

(4) The reload data trigger bit (TAUDRDT.TAUDRDTm) is set to 1 which sets the status flag (TAUDRSF.TAUDRSFm = 1), enabling simultaneous reloading.

(5) Simultaneous reloading does not take place at the bottom of the triangular cycle.

(6) Simultaneous reloading takes place at the top of the triangular cycle. The TAUDCDRm value is loaded into the TAUDCDRm buffer, the TAUDTOL.TAUDTOLm value is loaded into the TAUDTOL.TAUDTOLm buffer.

(7) The counters count down and await the next trigger for simultaneous reloading. The values of TAUDCDRm and TAUDTOL.TAUDTOLm can be changed again.

(3) Simultaneous Reloading when INTTAUDIm is Generated on an Upper Channel Specified by TAUDRDC.TAUDRDCm (Method C1)

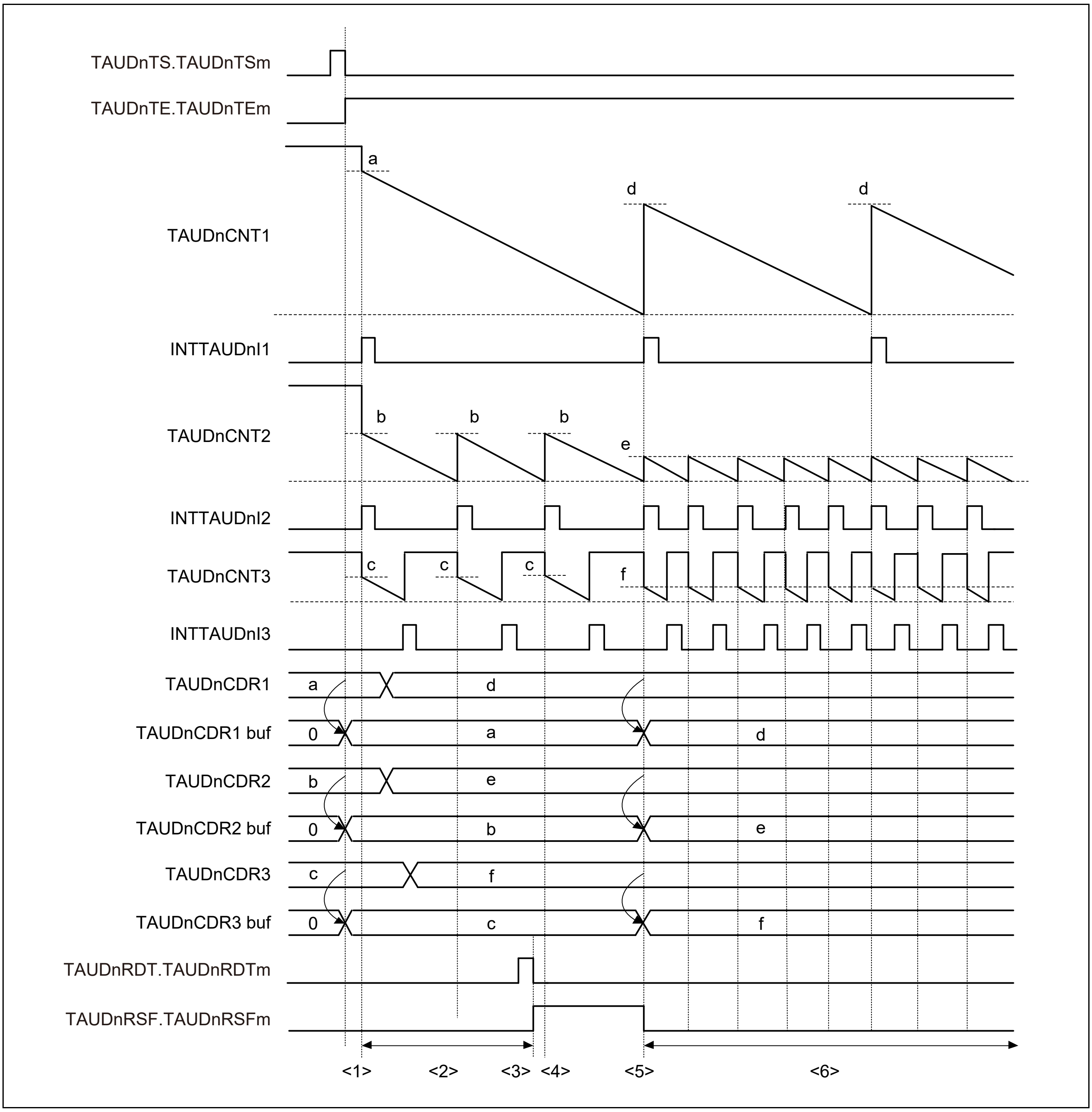


Figure 20.6 Simultaneous Reloading when INTTAUDIm is Generated on an Upper Channel Specified by TAUDRDC.TAUDRDCm

Setting:

CH1 is an upper channel which performs counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous reloading method C1 is applied. The TAUDRDC register specifies a channel which generates a trigger for simultaneous reloading.

Description:

(1) When TAUDTS.TAUDTSm is set to 1, TAUDCDRm value is copied to the TAUDCDRm buffer.

(2) The TAUDCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUDRDT.TAUDRDTm) to 1, the status flag is set (TAUDRSF.TAUDRSFm = 1) to enable simultaneous reloading.

(4) Simultaneous reloading is triggered only by a CH1 interrupt. Therefore, simultaneous reloading is not conducted even if enabled.

(5) Simultaneous reloading is triggered by INT1 which is generated when counter1 reaches 0000H. The TAUDCDRm values are loaded into the corresponding TAUDCDRm buffers.

(6) The counter counts down and awaits the next trigger for simultaneous reloading. The values of the TAUDCDRm registers can be changed again.

(4) Simultaneous Reloading when INTTAUDIm is Generated on an Upper Channel Specified by TAUDRDC.TAUDRDCm that in Turn is Triggered by an External Signal (Method C2)

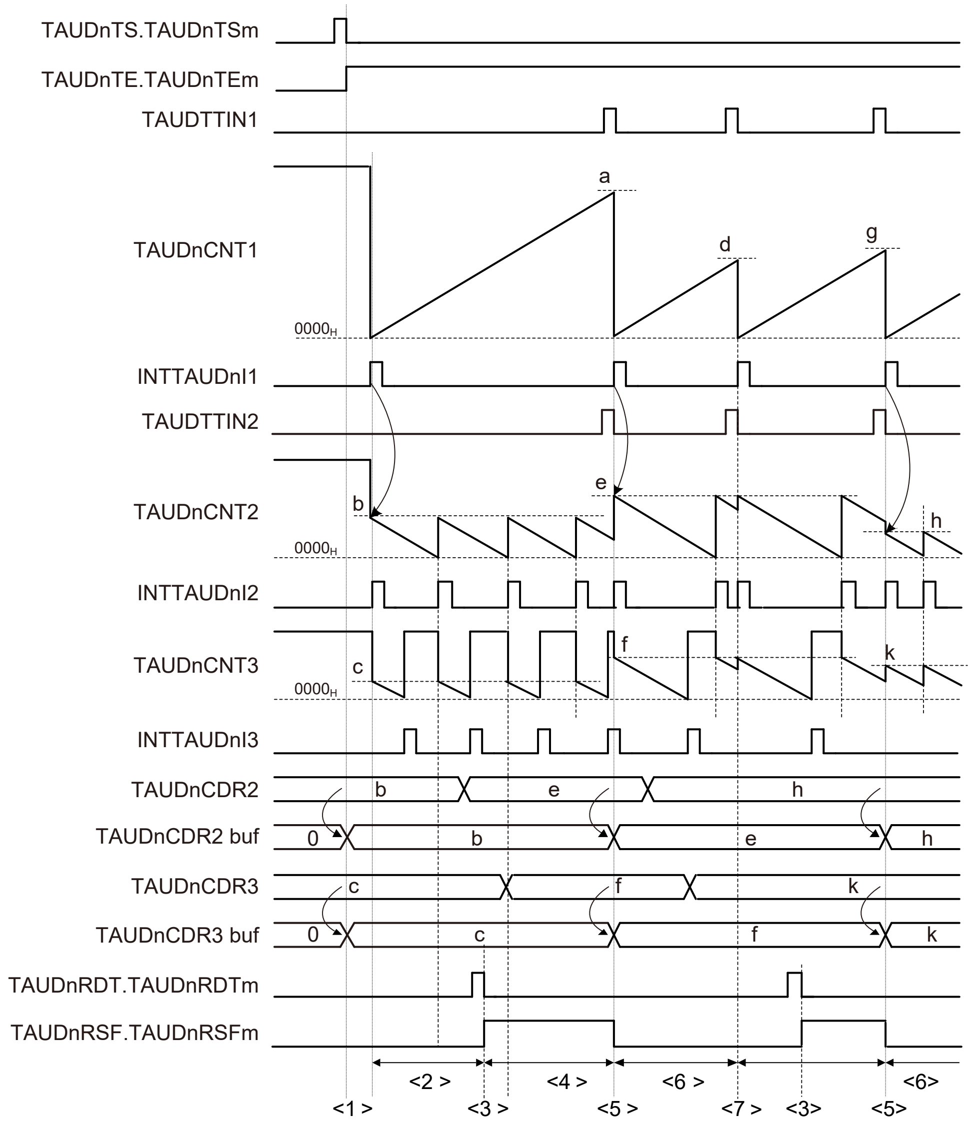


Figure 20.7 Simultaneous Reloading when INTTAUDIm is Generated on an Upper Channel Specified by TAUDRDC.TAUDRDCm that in Turn is Triggered by an External Signal

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 TAUDRDC register specifies which upper channel is monitored for an INTTAUDIm trigger.

Description:

(1) When TAUDTS.TAUDTSm is set to 1, TAUDCDRm value is copied to the TAUDCDRm buffer. However, as TAUDCDR1 operates in capture mode, TAUDCDR1 value is not copied to the TAUDCDR1 buffer.

(2) The TAUDCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUDRDT.TAUDRDTm) to 1, the status flag is set (TAUDRSF.TAUDRSFm = 1) to enable simultaneous reloading.

(4) Simultaneous reloading is triggered only by a CH1 interrupt. Therefore, simultaneous reloading is not conducted even if enabled.

(5) Simultaneous reloading is triggered by INT1 which is caused by external signal TIN1. The TAUDCDRm values are written to the corresponding TAUDCDRm buffers.

(6) The counters count down and await the next trigger for simultaneous reloading. The values of the TAUDCDRm registers can be changed again.

(7) An external signal occurs at TIN2 but simultaneous reloading does not take place because it is disabled (TAUDRSF.TAUDRSFm = 0).

20.7 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 (TAUDTOE.TAUDTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUDTO.TAUDTOm) is sent to the output pin (TAUDTTOUTm).
- By TAUD signals (TAUDTOE.TAUDTOEm = 1)
  When controlled by TAUD signals, the output level of TAUDTTOUTm is set or reset or toggled by internal signals. The value of TAUDTO.TAUDTOm is updated accordingly to reflect the value of TAUDTTOUTm.
  - Independently (TAUDTOM.TAUDTOMm = 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 (TAUDTOM.TAUDTOMm = 0).
  - Synchronously (TAUDTOM.TAUDTOMm = 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 (TAUDTOM.TAUDTOMm = 1).

The TAUDTO.TAUDTOm 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 20.7, Channel Output Modes.

The channel output modes are described in details below.

- Section 20.7.2 Channel Output Modes Controlled Independently by TAUD Signals.
- Section 20.7.3 Channel Output Modes Controlled Synchronously by TAUD Signals.

• Batch operation of TAUDTOm bit

Whether a set value is reflected to the TAUDTOm bit or not is controlled by the TAUDTOE.TAUDTOEm bit.

The TAUDTOm setting is written only to the bit (channel) set with TAUDTOE.TAUDTOEm bit = 0 when a write to the TAUDTO register is attempted. No TAUDTOm setting is reflected to the bit (channel) set with TAUDTOE.TAUDTOEm bit = 1.

• Output logic

Positive logic or negative logic of the output is specified by control bit TAUDTOL.TAUDTOLm. The value of TAUDTOL.TAUDTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output or triangle PWM output. Otherwise, changes to TAUDTOL.TAUDTOLm result in an invalid TAUDTTOUTm signal output. See Section 20.6, Simultaneous Reloading.

The various channel output modes and the channel output control bits are listed in Table 20.7.

Table 20.7 Channel Output Modes

<table>
<tr><th>Channel Output Mode</th><th>TAUDTOE.
TAUDTOEm</th><th>TAUDTOM.
TAUDTOMm</th><th>TAUDTOC.
TAUDTOCm</th><th>TAUDTDE.
TAUDTDEm</th><th>TAUDTRE.
TAUDTREm</th><th>TAUDTME.
TAUDTMEm</th><th>TAUDTDM.
TAUDTDMm</th></tr>
<tr><td colspan="8">By software</td></tr>
<tr><td>Independent channel output mode controlled by software</td><td>0</td><td colspan="6">X</td></tr>
<tr><td colspan="8">By TAUD signals, independently</td></tr>
<tr><td>Independent channel output mode 1</td><td rowspan="3">1</td><td rowspan="3">0</td><td rowspan="2">0</td><td rowspan="3">0</td><td rowspan="3">0</td><td>0</td><td rowspan="3">0</td></tr>
<tr><td>with real-time output</td><td>1</td></tr>
<tr><td>Independent channel output mode 2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">By TAUD signals, synchronously</td></tr>
<tr><td>Synchronous channel output mode 1</td><td rowspan="7">1</td><td rowspan="7">1</td><td rowspan="2">0</td><td rowspan="2">0</td><td>0</td><td>0</td><td rowspan="2">0</td></tr>
<tr><td>with non-complementary modulation output</td><td>1</td><td>X</td></tr>
<tr><td>Synchronous channel output mode 2</td><td rowspan="5">1</td><td>0</td><td rowspan="3">0</td><td rowspan="3">0</td><td rowspan="2">0</td></tr>
<tr><td>with dead time output</td><td rowspan="3">1</td></tr>
<tr><td>with one-phase PWM output</td><td>1</td></tr>
<tr><td>with complementary modulation output</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">0</td></tr>
<tr><td>with non-complementary modulation output</td><td>0</td></tr>
</table>

Cautions 1. All combinations not listed in this table are forbidden.

2. Bits marked with an x can be set to any value.

Remarks 1. The following bits cannot be changed during count operation (TAUDTE.TAUDTE = 1):
- **TAUDTOE.TAUDTOEm**
- **TAUDTOM.TAUDTOMm**
- **TAUDTOC.TAUDTOCm**
- **TAUDTDE.TAUDTDEm**
- **TAUDTRE.TAUDTREm**
- **TAUDTDM.TAUDTDMm**

2. The following bits cannot be changed during count operation (TAUDTE.TAUDTEm = 1) except in channel output modes with modulation output:
- **TAUDTME.TAUDTMEm**
- **TAUDTDL.TAUDTDLm**

20.7.1 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 (TAUDTOE.TAUDTOEm = 0).

(1) Set TAUDTO.TAUDTOm to specify the initial level of the TAUDTTOUTm output.

(2) Set channel output mode according to Table 20.7, Channel Output Modes, and the output logic using the TAUDTOL.TAUDTOLm bit.

(3) Start the counter (TAUDTS.TAUDTSm = 1).

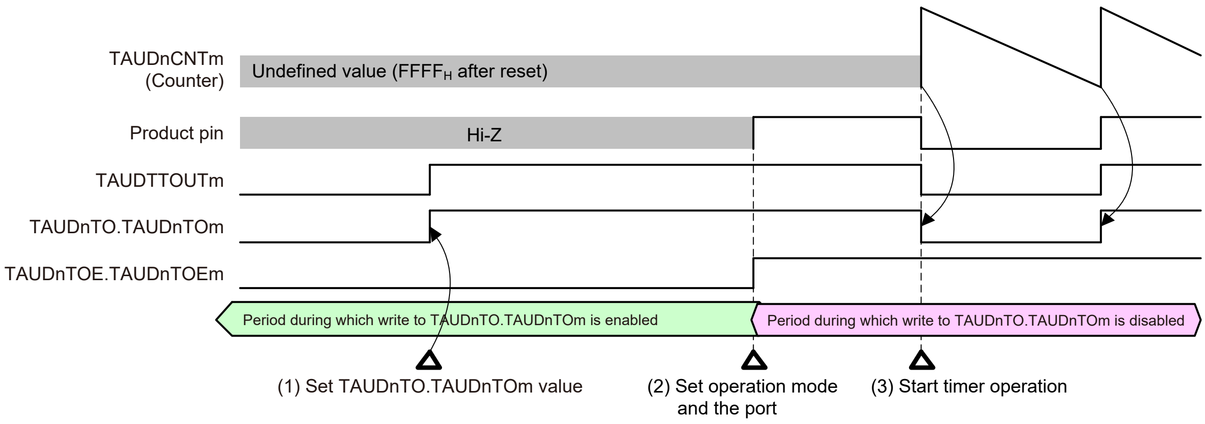


Figure 20.8 General Procedure for Specifying a TAUDTTOUTm Channel Output Mode

20.7.2 Channel Output Modes Controlled Independently by TAUD Signals

This section lists the channel output modes that are controlled independently by TAUD signals. The control bits used to specify a mode are listed in Table 20.7, Channel Output Modes.

(1) Independent Channel Output Mode 1

• Set/reset conditions

In this output mode, TAUDTTOUTm toggles when INTTAUDIm is detected. The value of TAUDTOL.TAUDTOLm is ignored.

• Prerequisites

There are no prerequisites other than those shown in Table 20.7, Channel Output Modes.

(2) Independent Channel Output Mode 1 with Real-Time Output

In this output mode, the value of TAUDTRO.TAUDTROm bit of the trigger channel is output to TAUDTTOUTm. The trigger channel is specified by setting the corresponding TAUDTRC.TAUDTRCm bit to 1. It controls all lower channels for which TAUDTRC.TAUDTRCm = 0.

• Set/reset conditions

The value of TAUDTRO.TAUDTROm bit is sent to TAUDTTOUTm only when an INTTAUDIm interrupt occurs on the trigger channel. The interrupt is generated either:

- at certain specified intervals or
- on detection of an effective edge of TAUDTTINm or the start of counting

The type of trigger is set using the TAUDCMORm.TAUDMD[4:1] bits.

• Prerequisites

Both the master and slave channels can be set as a trigger generation channel. A channel for which TAUDTRC.TAUDTRCm is set to 1 serves as a trigger generation channel even if TAUDTRE.TAUDTREm is set to 0.

If there is no channel for which TAUDTRC.TAUDTRCm is set to 1 or if TAUDTRC.TAUDTRC0 = 0, real-time output cannot take place.

This can be seen in Figure 20.9.

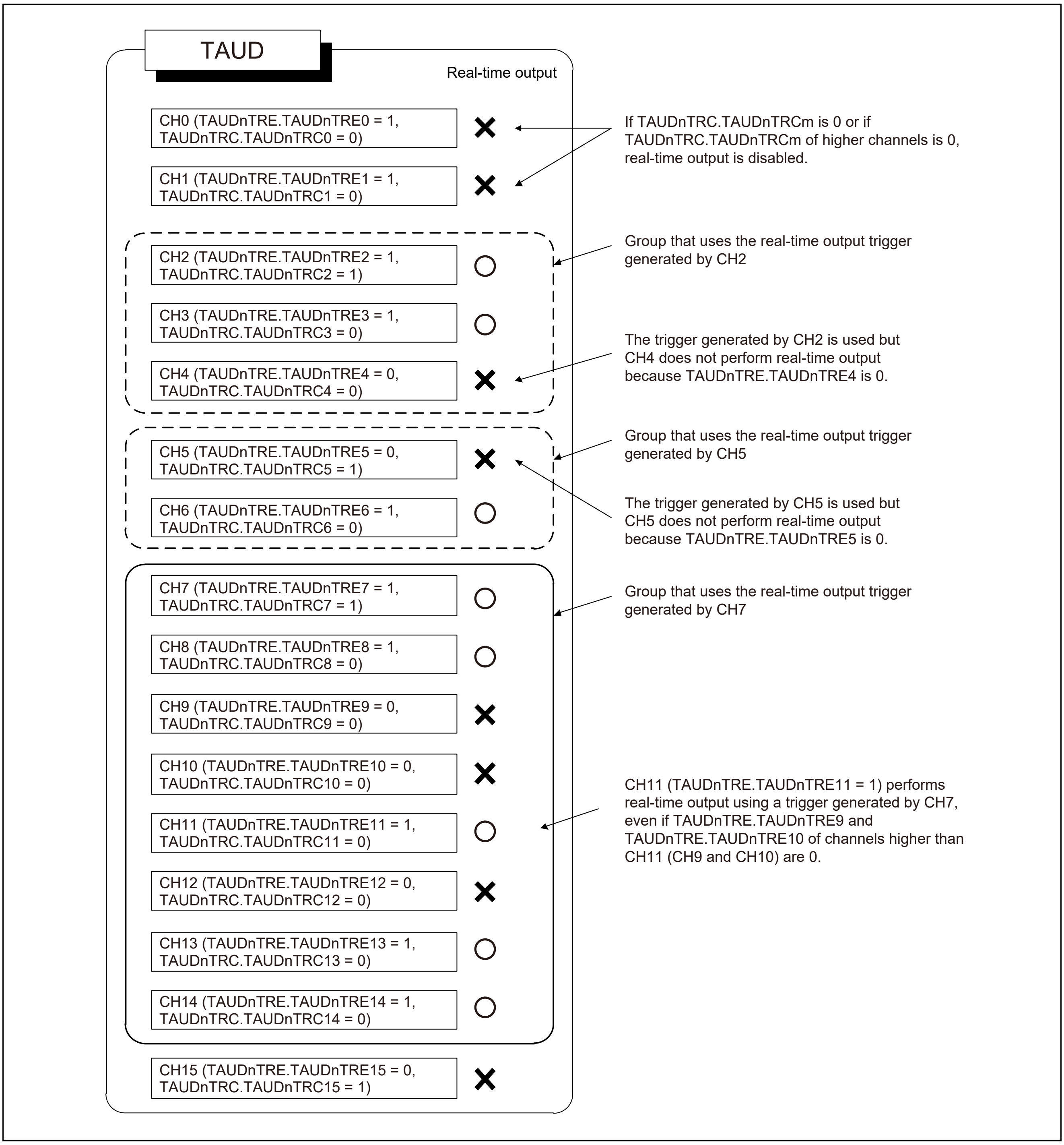


Figure 20.9 Real-Time Output

(3) Independent Channel Output Mode 2

- Set/reset conditions

  In this output mode, TAUDTTOUTm is set when INTTAUDIm occurs at the start of counting, and reset when INTTAUDIm occurs due to a match between TAUDCNTm and TAUDCDRm.

- Prerequisites

  There are no prerequisites other than those shown in Table 20.7, Channel Output Modes.

20.7.3 Channel Output Modes Controlled Synchronously by TAUD Signals

This section lists the channel output modes that are controlled synchronously by TAUD signals. The control bits used to specify a mode are listed in Table 20.7, Channel Output Modes.

(1) Synchronous Channel Output Mode 1

• Set/reset conditions

In this output mode, INTTAUDIm of master channel serves as a set signal and INTTAUDIm of the slave channel as a reset signal. If INTTAUDIm of master channel and INTTAUDIm of the slave channel are generated at the same time, INTTAUDIm of the slave channel (reset signal) has priority over INTTAUDIm (set signal) of master channel, i.e., the master channel is ignored.

• Prerequisites

There are no prerequisites other than those shown in Table 20.7, Channel Output Modes.

(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 (TAUDTRO.TAUDTROm) of a channel. The phase period to which the dead time is added is specified using the TAUDTDL.TDLm bit; for positive phase set TAUDTDL.TAUDTDLm = 0 and for negative phase set TAUDTDL.TAUDTDLm = 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.

TAUDTRO.TAUDTROm, TAUDTME.TAUDTMEm, and TAUDTDL.TAUDTDLm can only be changed during count operation.

- If TAUDTME.TAUDTMEm is changed, its new value is applied upon detection of INTTAUDIm on the specified channel.
- If TAUDTME.TAUDTMEm and TAUDTDL.TAUDTDLm are changed, their new values are applied upon detection of INTTAUDIm on the master channel.

(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 20.15.6, Triangle PWM Output.

• Set/reset conditions

TAUDCNTm 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.

(4) Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUDTTOUTm. The set/reset conditions are shown in Figure 20.10.

• Set/reset conditions

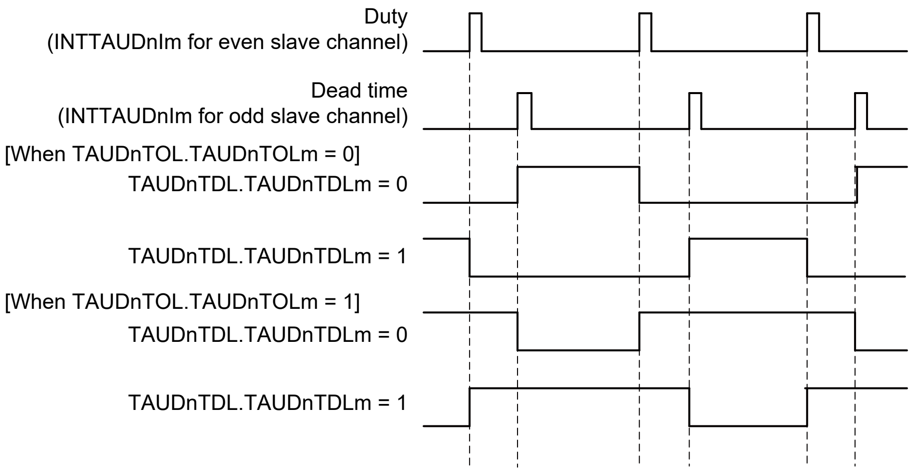


Figure 20.10 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set TAUDTDL.TAUDTDLm = 0 for rising edges and TAUDTDL.TAUDTDLm = 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-numbered 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-numbered channel and the even-numbered channel:

- TAUDTOE.TAUDTOEm
- TAUDTME.TAUDTMEm
- TAUDTRE.TAUDTREm
- TAUDTOM.TAUDTOMm
- TAUDTOC.TAUDTOCm
- TAUDTDE.TAUDTDEm
- TAUDTDM.TAUDTDMm

(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 20.11.

- Set/reset conditions

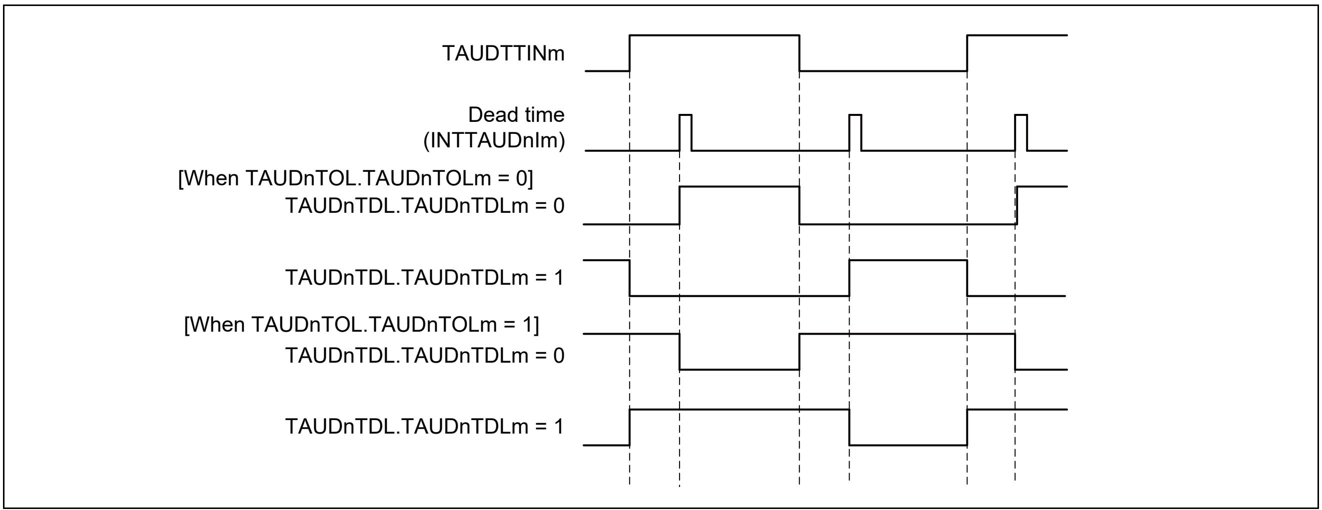


Figure 20.11 Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output

With regard to the edge to which dead time is added, set TAUDTDL.TAUDTDLm = 0 for rising edges and TAUDTDL.TAUDTDLm = 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-numbered 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-numbered channel and the even-numbered channel:
- TAUDTOE.TAUDTOEm
- TAUDTME.TAUDTMEm
- TAUDTRE.TAUDTREm
- TAUDTOM.TAUDTOMm
- TAUDTOC.TAUDTOCm
- TAUDTDE.TAUDTDEm
- TAUDTDM.TAUDTDMm

(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 (TAUDTRO.TAUDTROm), the modulation output bit (TAUDTME.TAUDTMEm), and the output level bit (TAUDTOL.TAUDTOLm) of a pair of slave channels.

For details, see Section 20.16.3, Complementary Modulation Output.

- 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.

TAUDTRO.TAUDTROm, TAUDTME.TAUDTMEm, and TAUDTDL.TAUDTDLm can only be changed during count operation.

  – If TAUDTME.TAUDTMEm is changed during operation, its new value is applied upon detection of INTTAUDIm at the specified channel.
  – If TAUDTME.TAUDTMEm and TAUDTDL.TAUDTDLm are changed, their new values are applied upon detection of INTTAUDIm on an even slave channel.

(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.

20.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUDTS.TAUDTSm 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.

20.8.1 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 TAUDTS.TAUDTSm is set to 1. The value of data register is also loaded when the counter starts.

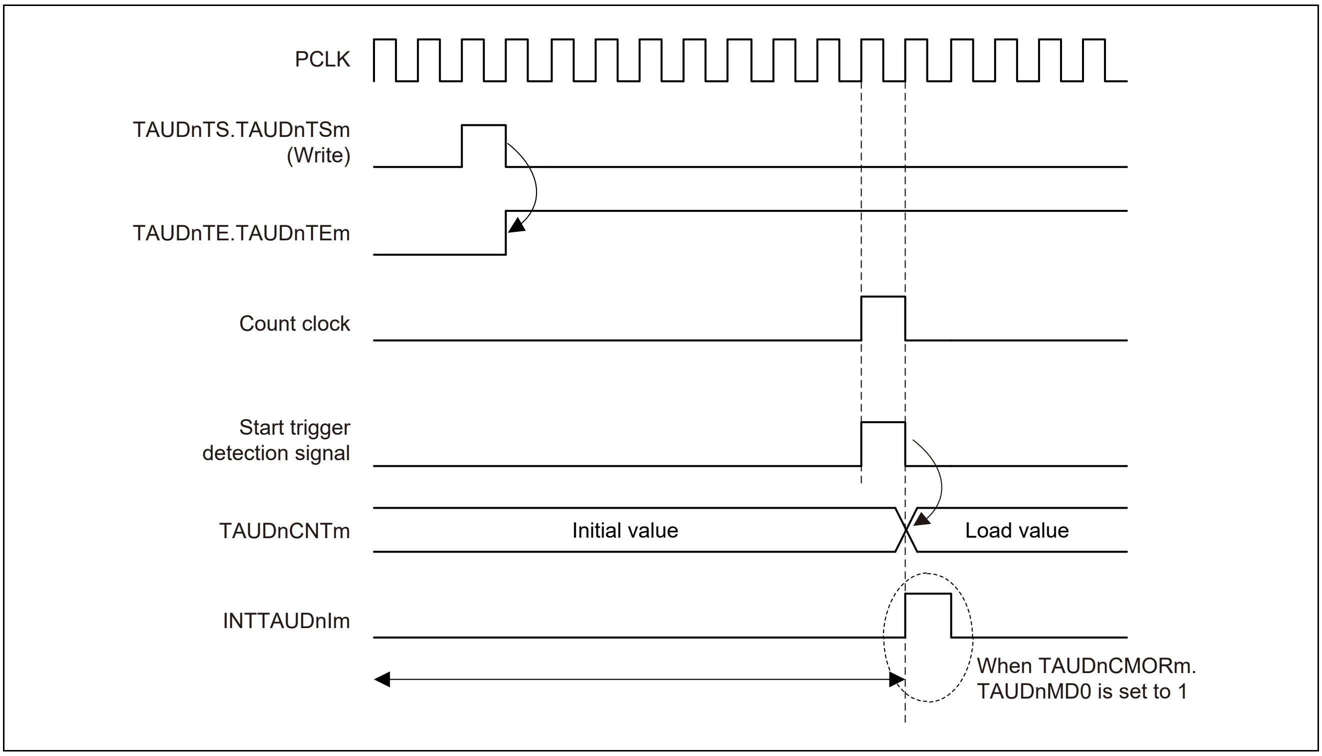


Figure 20.12 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

Remark: Make sure to set TAUDCMORm.TAUDMD0 to 0 when using the count-up/-down mode.

20.8.2 Event Count Mode

The value of data register is loaded as soon as TAUDTS.TAUDTSm is set to 1. The counter also starts immediately. The value of data register increments with subsequent count clocks.

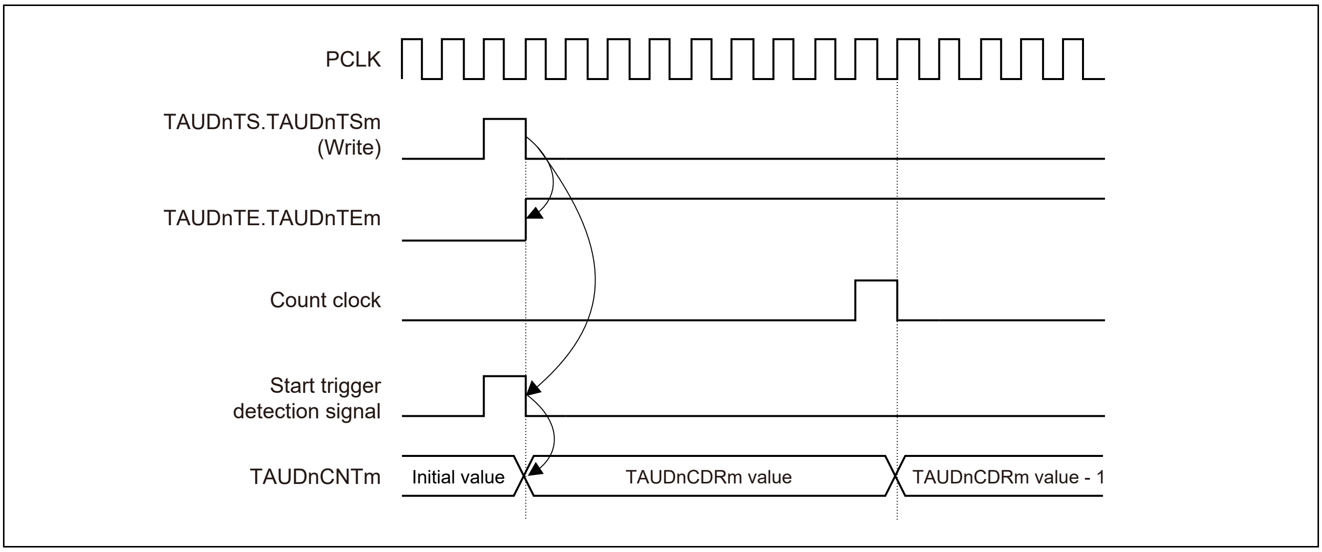


Figure 20.13 Start Timing in Event Count Mode

20.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of an effective 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.

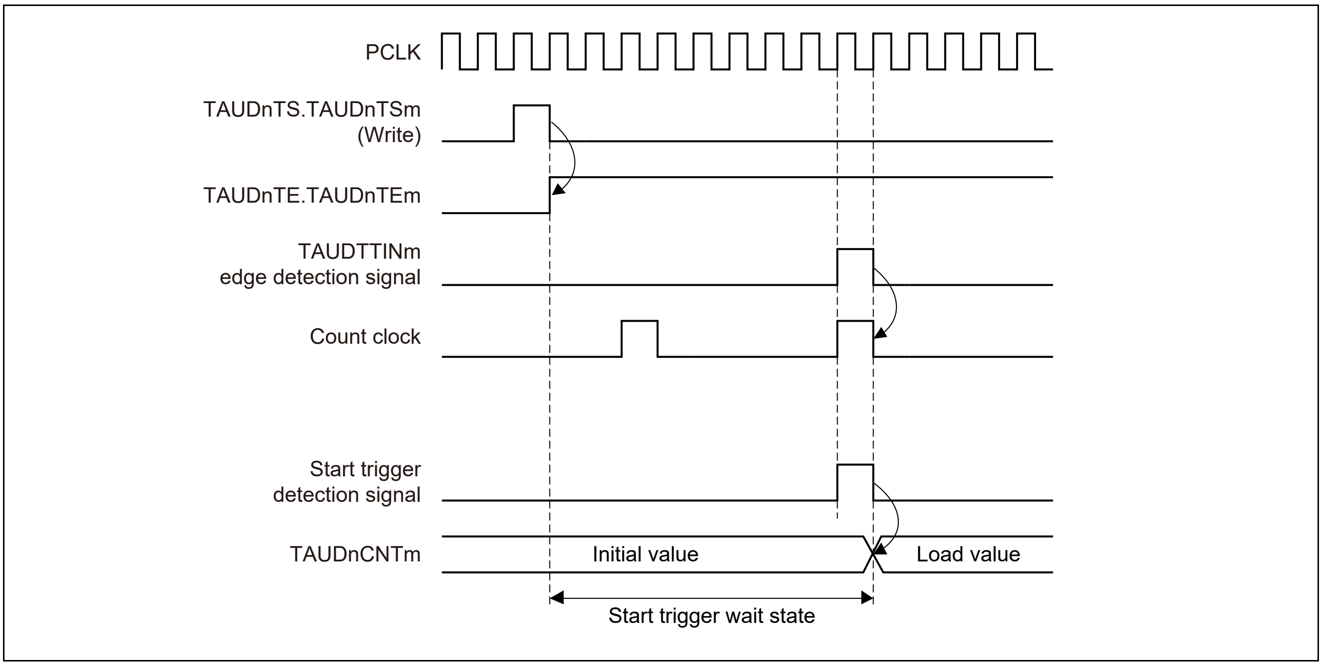


Figure 20.14 Start Timing in Other Operating Modes

20.9 TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUDIm is generated using the TAUDCMORm.TAUDMD0 bit.

The generation of INTTAUDIm when the TAUDCMORm.TAUDMD0 bit starts counting and the effect to TAUDTTOUTm depend on the selected function. For details, refer to the description of TAUDCMORm.TAUDMD0 of each function.

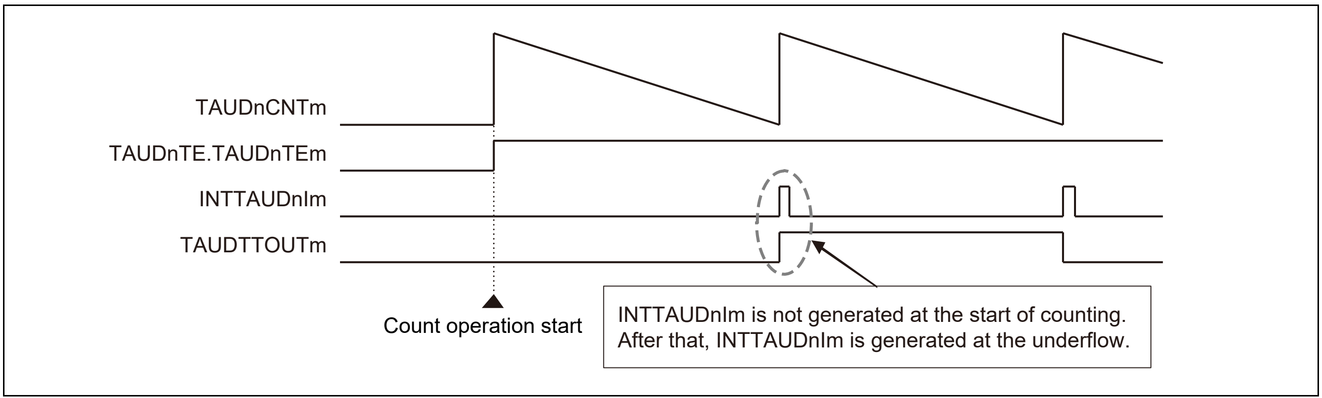


Figure 20.15 INTTAUDIm Generation Timing (when TAUDCMORm.TAUDMD0 = 0)

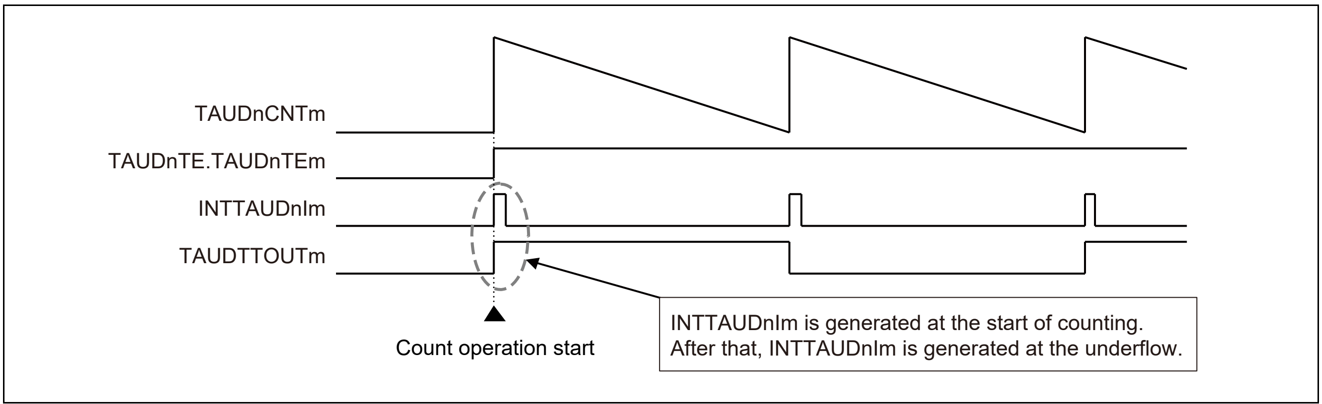


Figure 20.16 INTTAUDIm Generation Timing (when TAUDCMORm.TAUDMD0 = 1)

20.10 Interrupt Generation upon Overflow

Certain independent functions that count up, overflow without generating an interrupt when they reach FFFFH. 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 0000H at the same time as the first channel overflows (TAUDCNTm = FFFFH).
- Set TAUDCDRm of the second channel to FFFFH.
- 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 (TAUDCMORm.TAUDSTS[2:0] and TAUDCMURm.TAUDTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000H at exactly the same time as the up-counter of the first channel overflows (TAUDCNTm = FFFFH). 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.

20.10.1 Combination of the TAUDTTINm Input Pulse Interval Measurement and the TAUDTTINm Input Interval Timer

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDIm of the TAUDTTINm input interval timer can detect the overflow when TAUDCNTm of the TAUDTTINm input pulse interval measurement exceeds FFFFH.

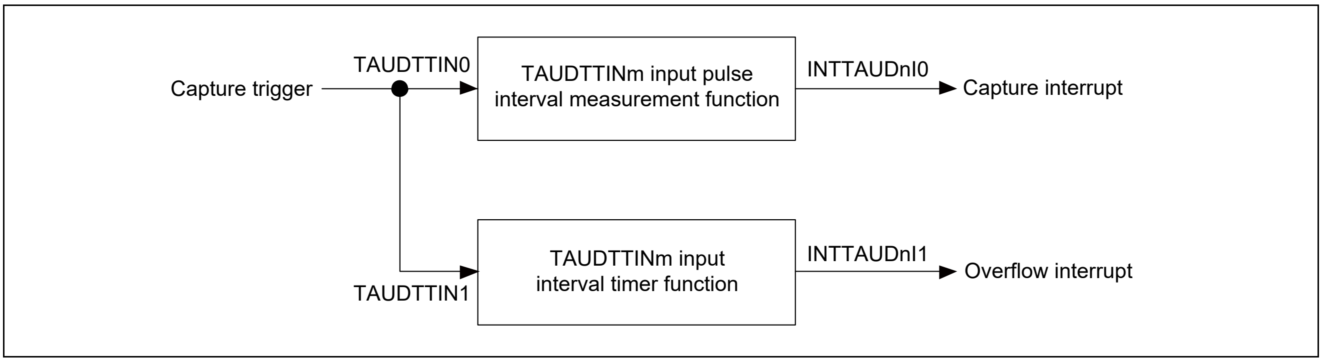


Figure 20.17 Combination of the TAUDTTINm Input Pulse Interval Measurement and the TAUDTTINm Input Interval Timer

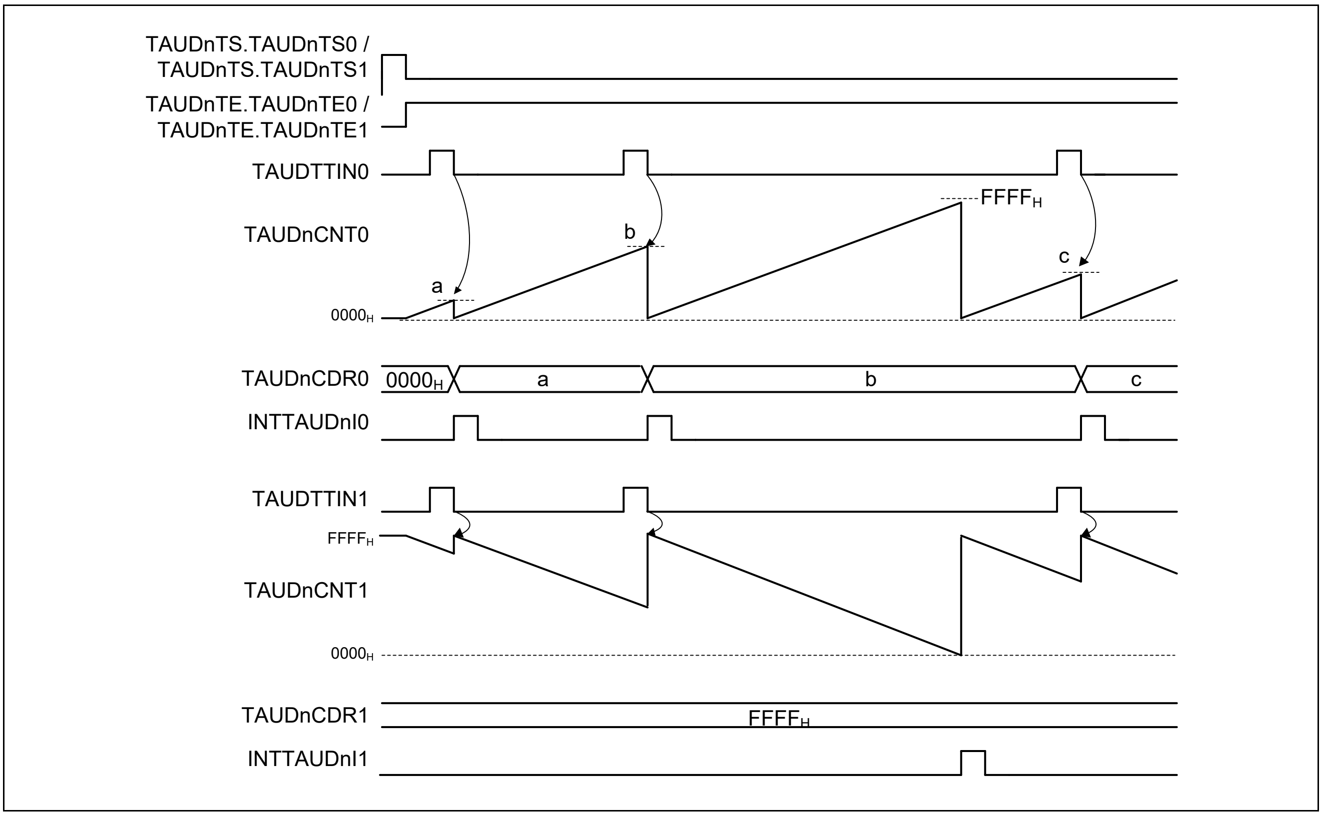


Figure 20.18 Interrupt Generation via Combination of the TAUDTTINm Input Pulse Interval Measurement and the TAUDTTINm Input Interval Timer

20.10.2 Combination of the TAUDTTINm Input Signal Width Measurement and the Overflow Interrupt Output (at Measuring the TAUDTTINm Width)

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDIm of the overflow interrupt output (at measuring the TAUDTTINm width) can detect the overflow when TAUDCNTm of the TAUDTTINm input signal width measurement exceeds FFFFH.

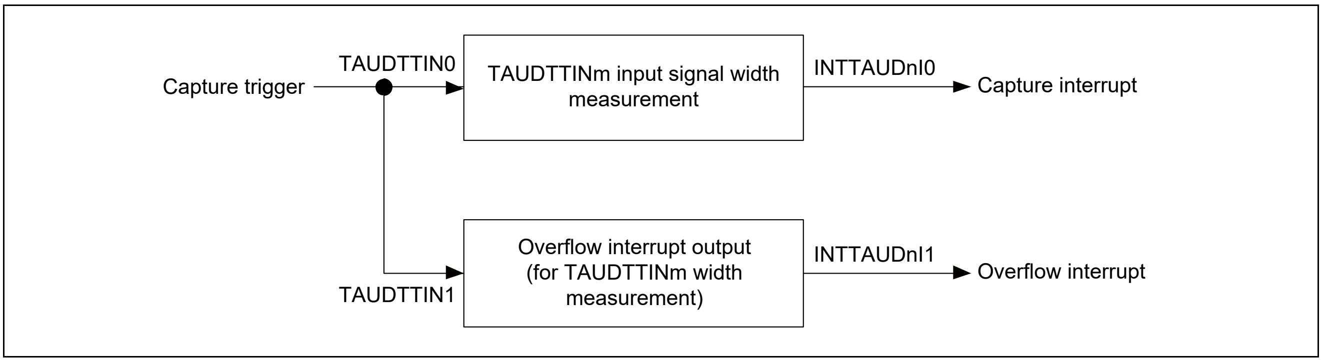


Figure 20.19 Combination of the TAUDTTINm Input Signal Width Measurement and the Overflow Interrupt Output (at Measuring the TAUDTTINm Width)

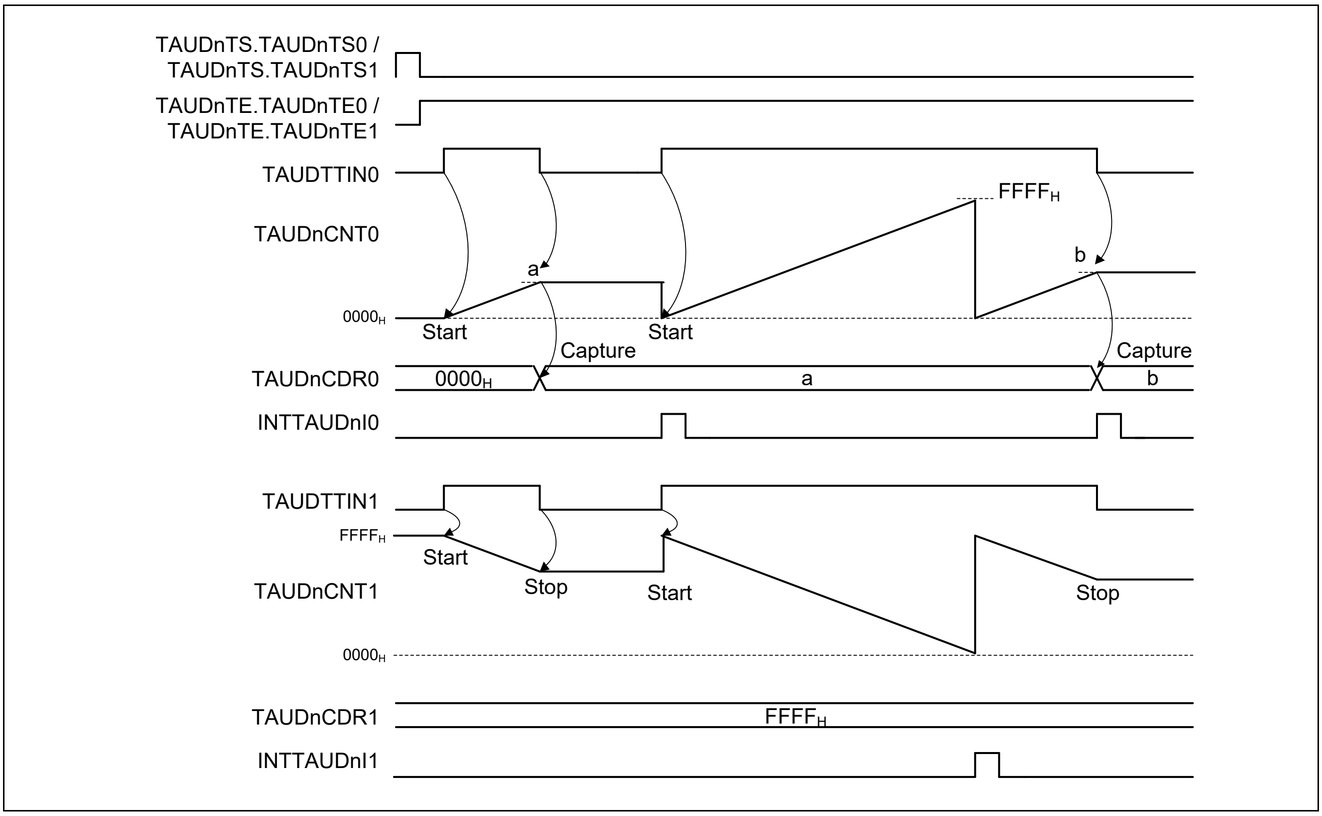


Figure 20.20 Interrupt Generation via Combination of the TAUDTTINm Input Signal Width Measurement and the Overflow Interrupt Output (at Measuring the TAUDTTINm Width)

20.10.3 Combination of the TAUDTTINm Input Position Detection and the Interval Timer

When the counters of both channels are started simultaneously, INTTAUDIm of the interval timer can detect the overflow when TAUDCNTm of the TAUDTTINm input position detection exceeds FFFFH.

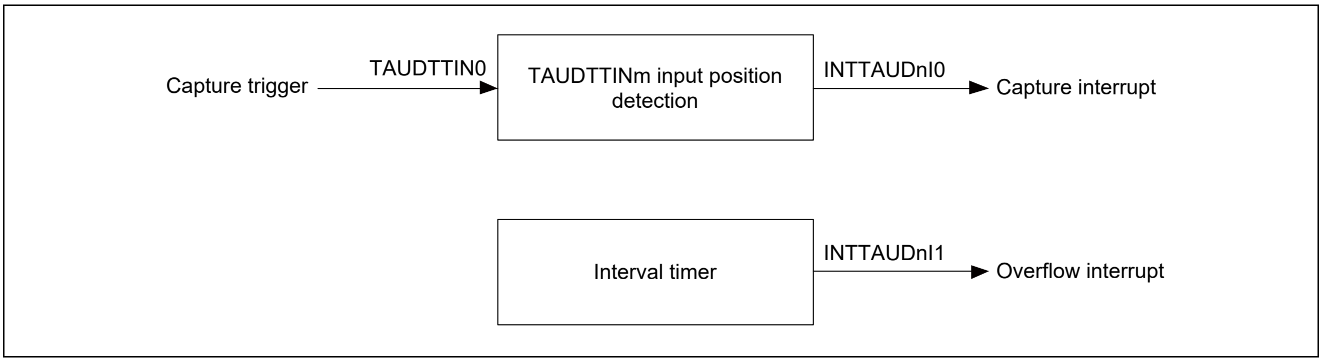


Figure 20.21 Combination of the TAUDTTINm Input Position Detection and the Interval Timer

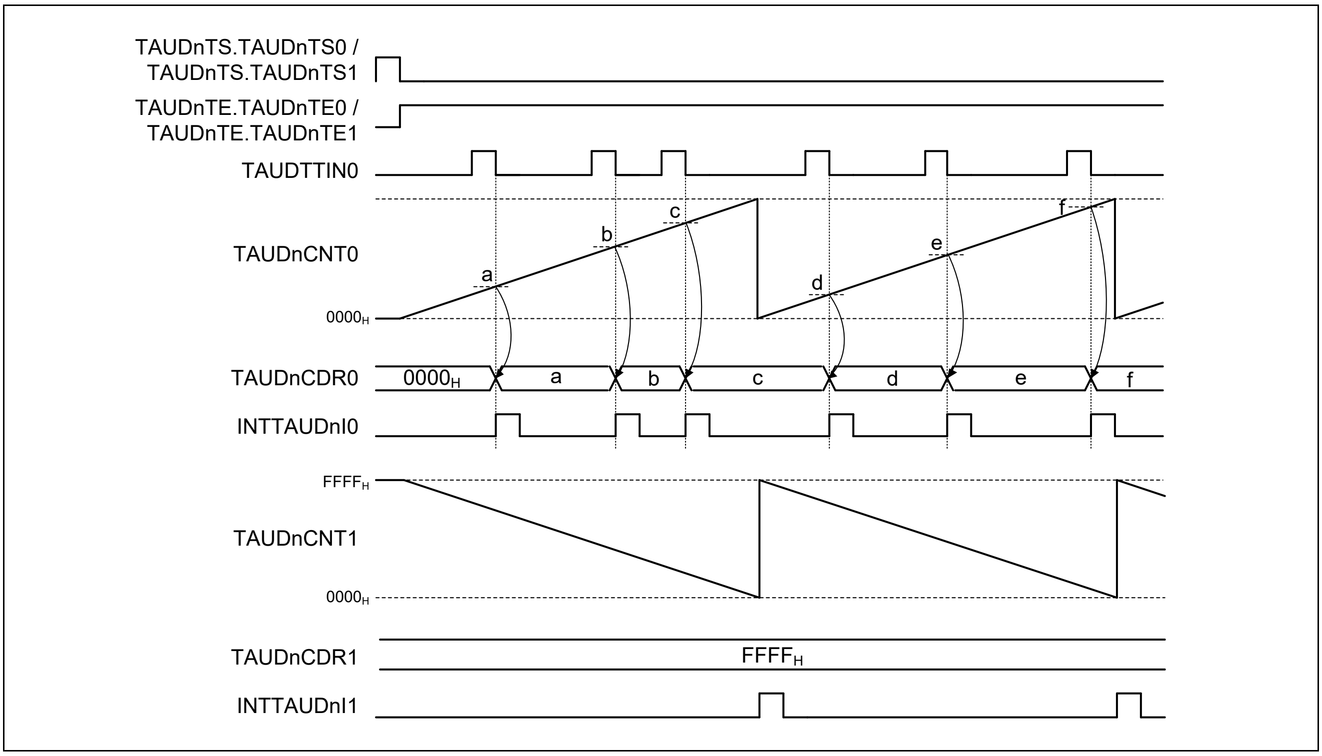


Figure 20.22 Interrupt Generation via Combination of the TAUDTTINm Input Position Detection and the Interval Timer

20.10.4 Combination of the TAUDTTINm Input Period Count Detection and the Overflow Interrupt Output (at Detecting the TAUDTTINm Input Period Count)

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDIm of the overflow interrupt output (at detecting the TAUDTTINm input period count) can detect the overflow when TAUDCNTm of the TAUDTTINm input period count detection exceeds FFFFH.

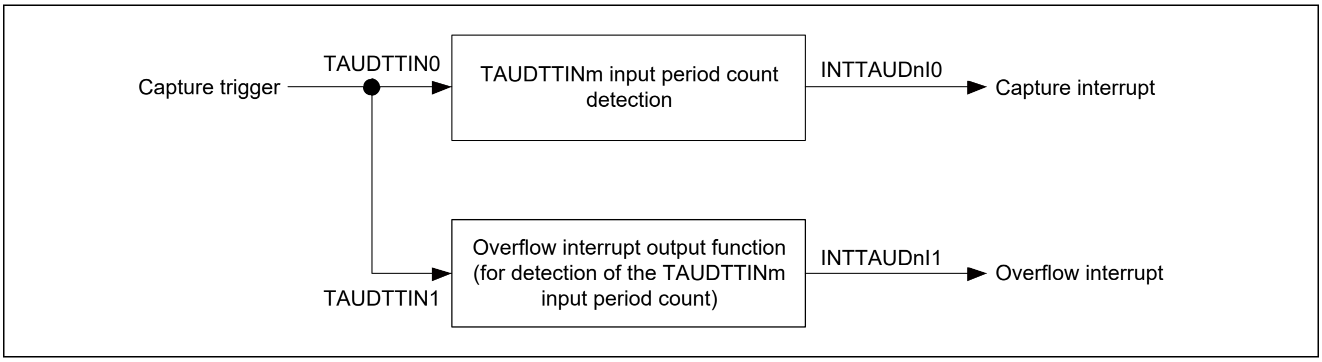


Figure 20.23 Combination of the TAUDTTINm Input Period Count Detection and the Overflow Interrupt Output (at Detecting the TAUDTTINm Input Period Count)

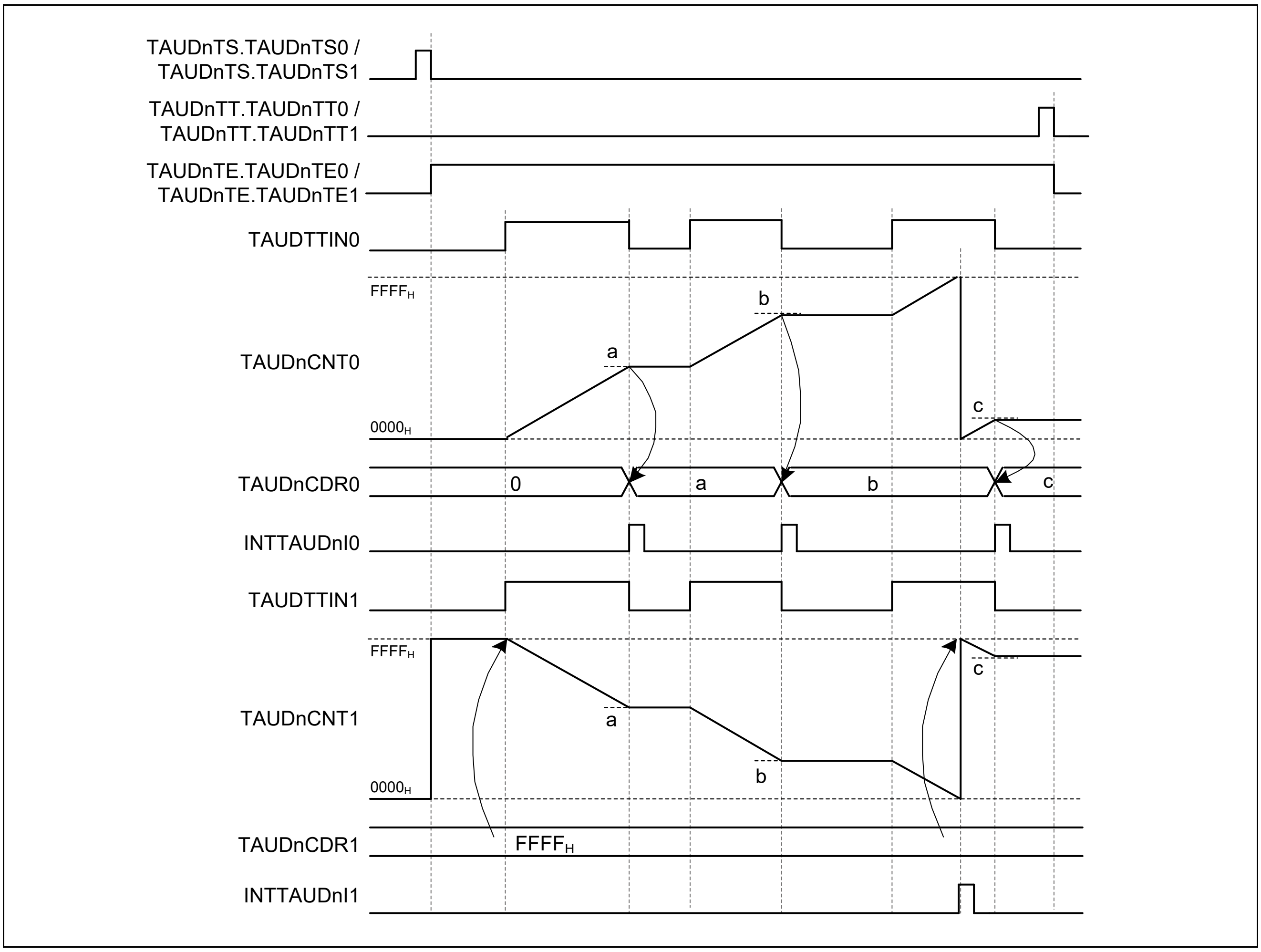


Figure 20.24 Interrupt Generation via Combination of the TAUDTTINm Input Period Count Detection and the Overflow Interrupt Output (at Detecting the TAUDTTINm Input Period Count)

20.11 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 20.25 shows when edge detection takes place.

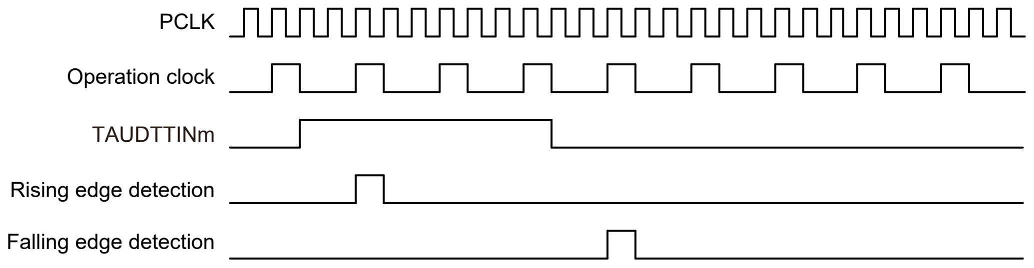


Figure 20.25 Basic Edge Detection Timing

Figure 20.25 shows an operation timing image. Actually, a noise filter or synchronization circuit which is located between the TAUDIm pin and TAUD causes a delay time.

20.12 Independent Channel Operations

The following sections list various independent channel operations of the Timer Array Unit D. For a general overview of independent channel operation, see Section 20.2, Functional Overview.
This section describes functions that generate interrupts at regular intervals or with a specified delay.

20.12.1 Interval Timer

(1) Overview

(a) Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDIm) at regular intervals. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

(b) Prerequisites

- The operation mode must be set to Interval Timer Mode, see Table 20.8, Contents of the TAUDCMORm Register for Interval Timer.
- The channel output mode must be set to Independent Channel Output Mode 1, see Section 20.7, Channel Output Modes.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation. The current value of TAUDCDRm is written to TAUDCNTm and the counter starts to count down from this value.
When the counter reaches 0000H, INTTAUDIm is generated and the TAUDTTOUTm signal toggles. TAUDCNTm then reloads the TAUDCDRm value and subsequently continues operation. The value of TAUDCDRm can be rewritten at any time, and the changed value of TAUDCDRm is applied the next time the counter starts to count down.
The counter can be stopped by setting TAUDTT.TAUDTTm to 1, which in turn sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm stop but retain their values. The counter can be restarted by setting TAUDTS.TAUDTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUDTS.TAUDTSm to 1 during operation.

(d) Conditions

If the TAUDCMORm.TAUDMD0 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 TAUDCMORm.MD0 is set to 1. For details see Section 20.9, TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts.

(2) Equations

INTTAUDIm cycle = count clock cycle × (TAUDCDRm + 1)

TAUDTTOUTm square wave cycle = count clock cycle × (TAUDCDRm + 1) × 2

(3) Block Diagram and General Timing Diagram

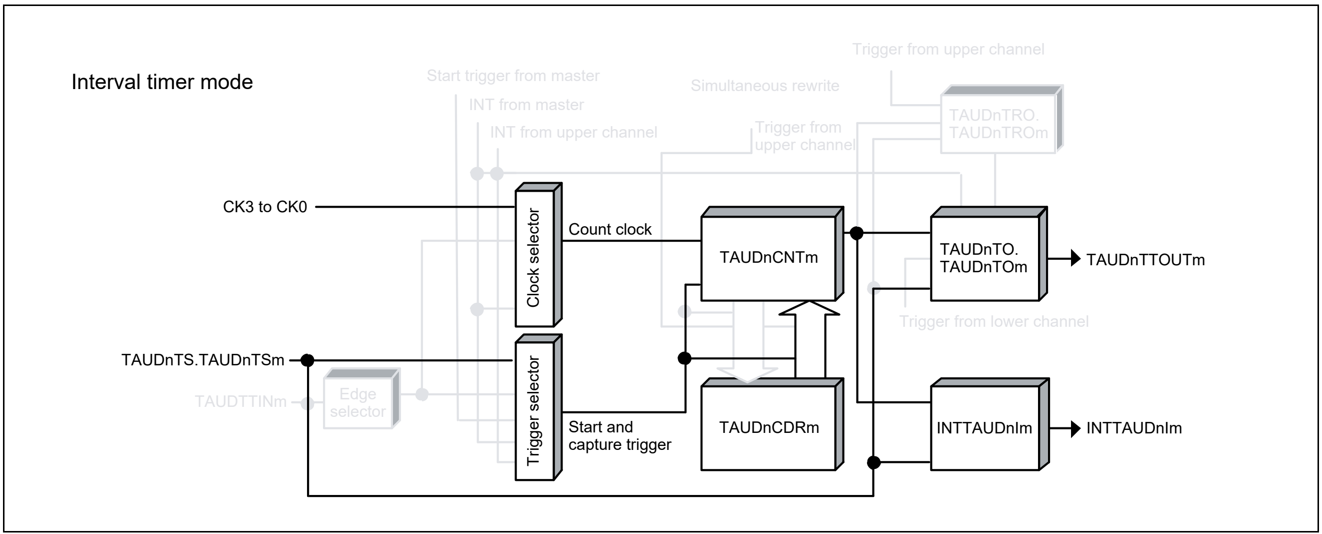


Figure 20.26 Block Diagram of Interval Timer

The following settings apply to the general timing diagram.

- INTTAUDIm is generated at the beginning of operation (TAUDCMORm.TAUDMD0 = 1).

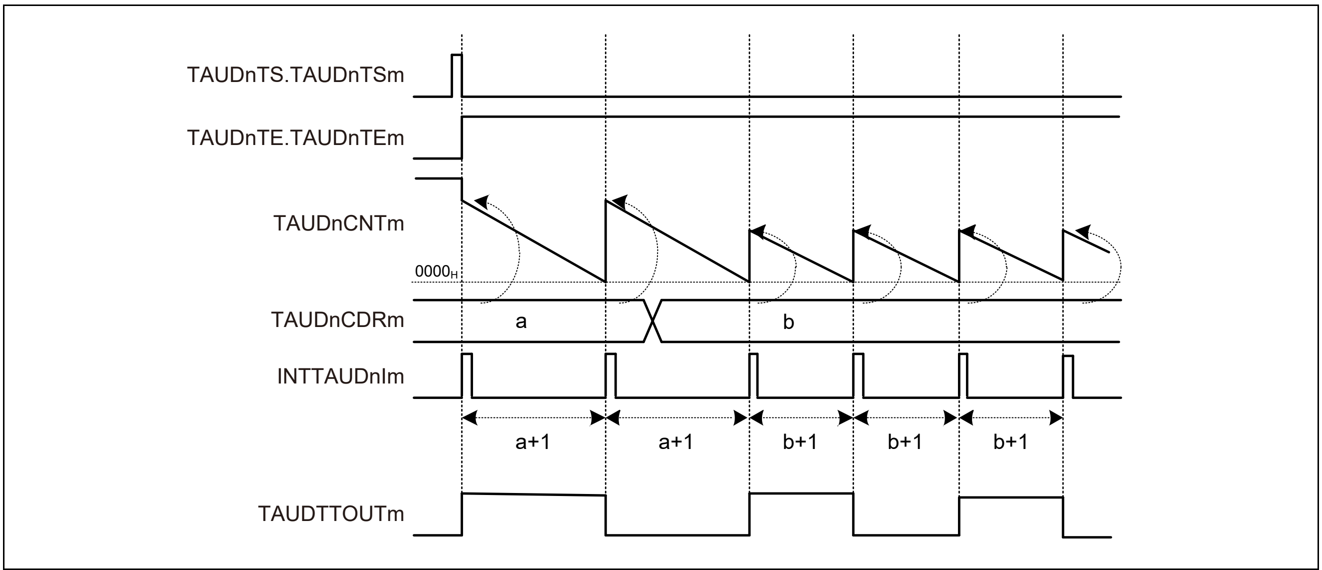


Figure 20.27 General Timing Diagram of Interval Timer

(4) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS[1:0]		TAUDCCS[1:0]		TAUDMAS	TAUDSTS[2:0]			TAUDCOS[1:0]		0	TAUDMD[4:1]				TAUDMD0

Table 20.8 Contents of the TAUDCMORm Register for Interval Timer

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	000: Triggers the counter by software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm is not generated to and TAUDTTOUTm is not toggled at the beginning of operation. 1: INTTAUDIm is generated and TAUDTTOUTm is toggled at the beginning and restarting of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.9 Contents of the TAUDCMURm Register for Interval Timer

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.10 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set these bits to 0
TAUDTRC.TAUDTRCm	
TAUDTME.TAUDTMEm	Disables modulation

Remark: The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDTOE.TAUDTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see Section 20.7, Channel Output Modes.

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the interval timer. Therefore, these registers should be set to 0.

Table 20.11 Simultaneous Reload Settings for Interval Timer

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Operating Procedure for Interval Timer

Table 20.12 Operating Procedure for Interval Timer

		Operation	TAUD Status
	Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.8, Contents of the TAUDCMORm Register for Interval Timer, Table 20.10, Control Bit Settings in Independent Channel Output Mode 1, and Table 20.9, Contents of the TAUDCMURm Register for Interval Timer. Set channel output mode by setting the control bits as described in Table 20.10, Control Bit Settings in Independent Channel Output Mode 1.	Channel operation is stopped.
Restart	Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. The TAUDCDRm value is loaded in TAUDCNTm. When TAUDCMORm.MD0 = 1, INTTAUDIm is generated and TAUDTTOUTm toggles.
	During Operation	The TAUDCDRm register value can be changed at any time. The TAUDCNTm register can be read at all times.	TAUDCNTm counts down. When the counter reaches 0000H: • The TAUDCDRm value is loaded in TAUDCNTm again and count operation continues. • INTTAUDIm is generated and TAUDTTOUTm toggles.
	Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

(6) Specific Timing Diagrams

(a) TAUDCDRm = 0000H, count clock = PCLK/2

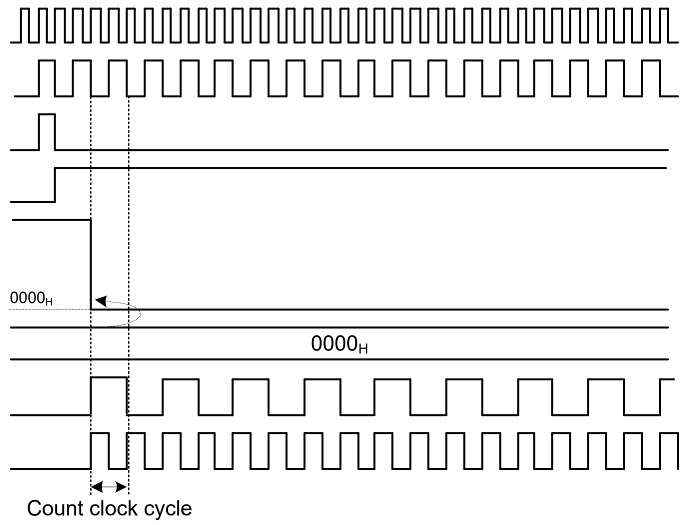


Figure 20.28 TAUDCDRm = 0000H, Count Clock = PCLK/2

- If TAUDCDRm = 0000H and the count clock = PCLK/2, the TAUDCDRm value is loaded into TAUDCNTm every count clock, meaning that TAUDCNTm is always 0000H.
- INTTAUDIm is generated every count clock, resulting in TAUDTTOUTm toggling every count clock.

(b) TAUDCDRm = 0000H, count clock = PCLK

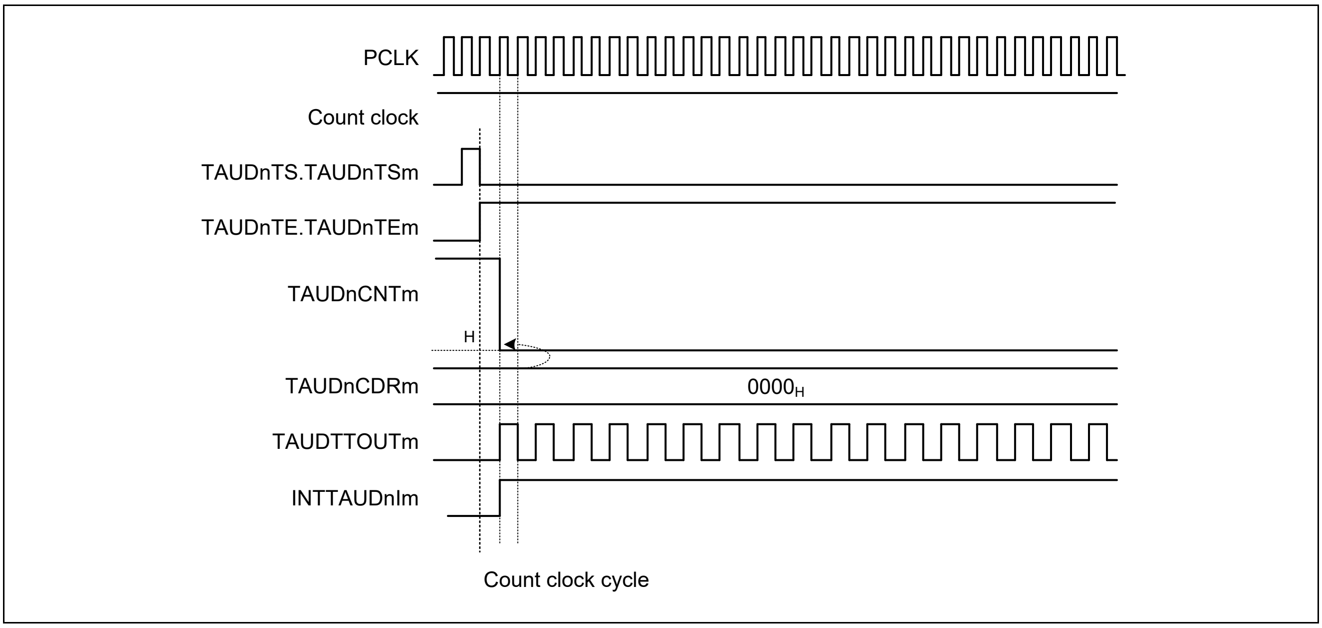


Figure 20.29 TAUDCDRm = 0000H, Count Clock = PCLK

- If TAUDCDRm = 0000H and the count clock = PCLK, the TAUDCDRm value is loaded into TAUDCNTm every PCLK clock, meaning that TAUDCNTm is always 0000H.
- INTTAUDIm is generated continuously, resulting in TAUDTTOUTm toggling every PCLK clock.

(c) Operation stop and restart

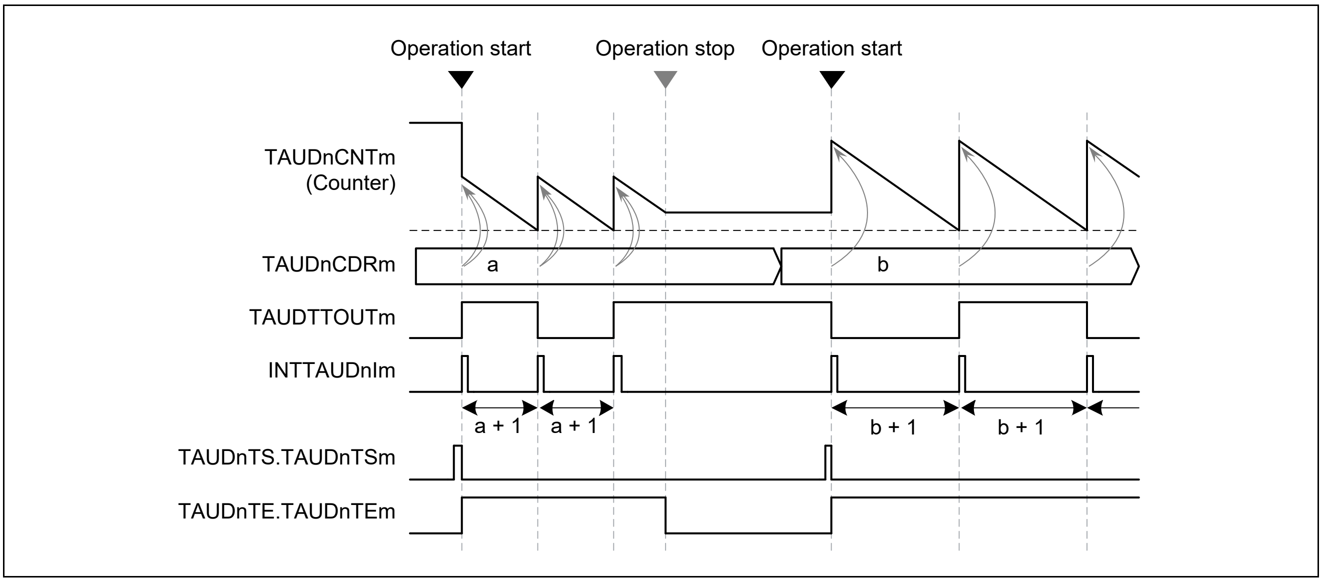


Figure 20.30 Operation Stop and Restart (TAUDCMORm.TAUDMD0 = 1)

- The counter can be stopped by setting TAUDTT.TAUDTTm to 1. This sets TAUDTE.TAUDTEm to 0.
- TAUDCNTm and TAUDTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUDTS.TAUDTSm to 1.

(d) Forced restart (TAUDCMORm.TAUDMD0 = 1)

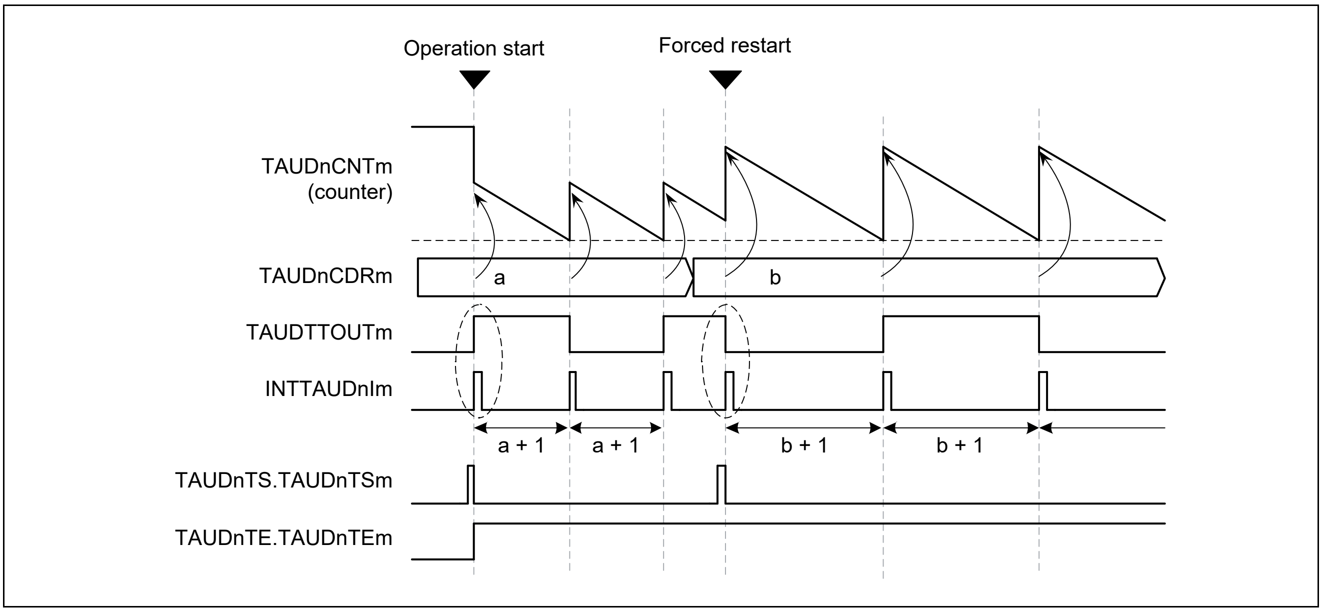


Figure 20.31 Forced Restart Operation (TAUDCMORm.TAUDMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDTS.TAUDTSm to 1 during operation.
- If the TAUDCMORm.TAUDMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When the counter is forcibly restarted, the TAUDCDRm value is reflected to TAUDCNTm and counting starts. Execute a forced restart to reflect the changed TAUDCDRm value immediately.
- When the counter is forcibly restarted, an interrupt (INTTAUDIm) is generated and TAUDTTOUTm is inverted.

(e) Forced restart (TAUDCMORm.TAUDMD0 = 0)

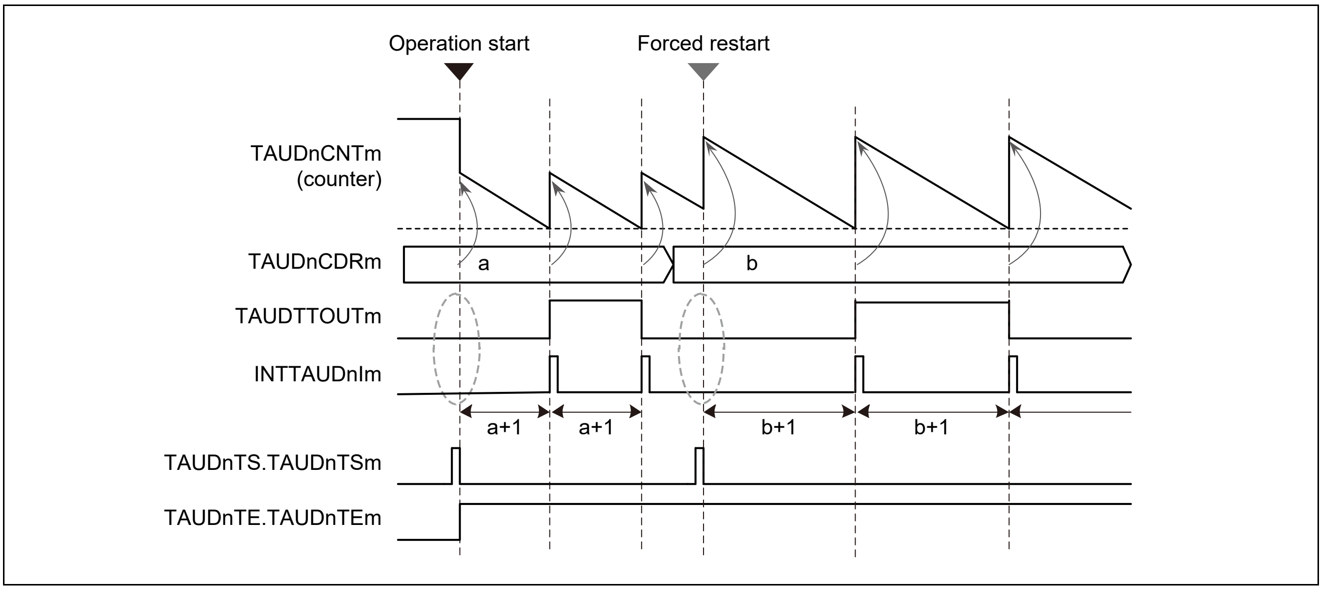


Figure 20.32 Forced Restart Operation (TAUDCMORm.TAUDMD0 = 0)

- When the counter is forcibly restarted, an interrupt (INTTAUDIm) is not generated and TAUDTTOUTm is not inverted.

20.12.2 TAUDTTINm Input Interval Timer

(1) Overview

(a) Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDIm) at regular intervals or when an effective edge of the TAUDTTINm signal is detected. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

(b) Prerequisites

- The operating mode should be set to interval timer mode. See Table 20.13, Contents of the TAUDCMORm Register for TAUDTTINm Input Interval Timer.
- The channel output mode should be set to independent channel output mode 1. See Section 20.7, Channel Output Modes.

(c) Functional description

This function operates in an identical manner to the interval timer (see Section 20.12.1, Interval Timer) except that this function is restarted by an effective edge of the TAUDTTINm input signal.
The type of edge used as a trigger is specified using the TAUDCMURm.TAUDTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

(2) Equations

INTTAUDIm cycle = count clock cycle × (TAUDCDRm + 1)
TAUDTTOUTm square wave cycle = count clock cycle × (TAUDCDRm + 1) × 2

(3) Block Diagram and General Timing Diagram

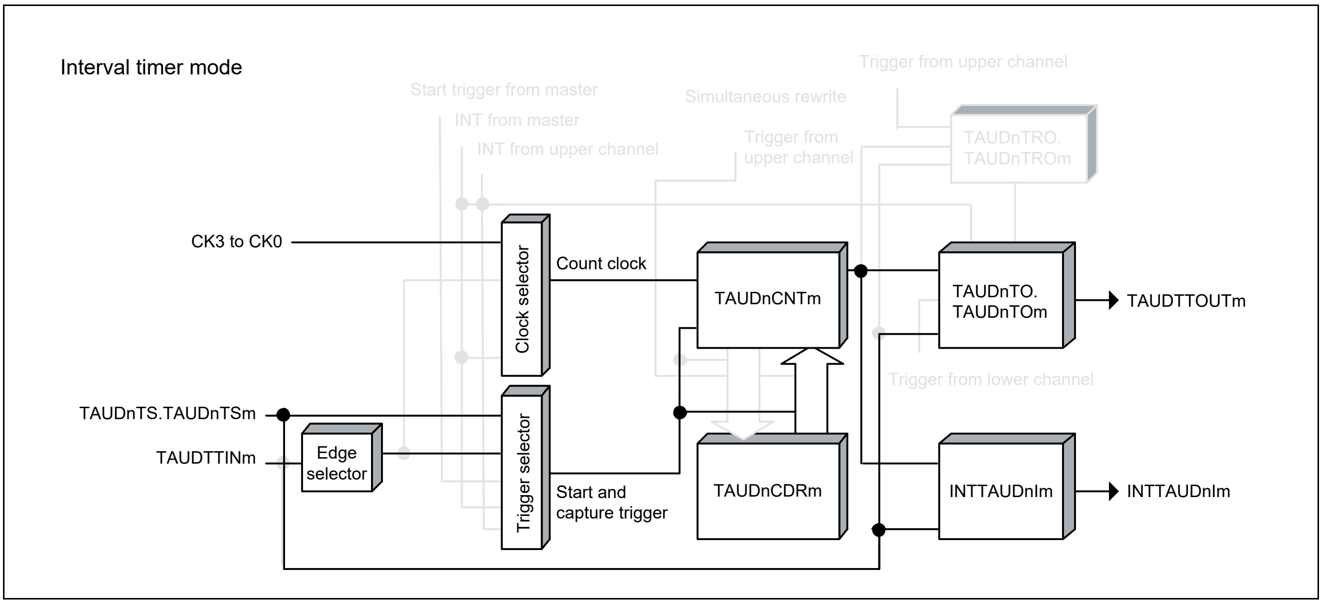


Figure 20.33 Block Diagram of TAUDTTINm Input Interval Timer

The following settings apply to the general timing diagram.

- INTTAUDIm is generated at the beginning of operation (TAUDCMORm.TAUDMD0 = 1)
- Rising edge detection (TAUDCMURm.TAUDTIS[1:0] = 01B)

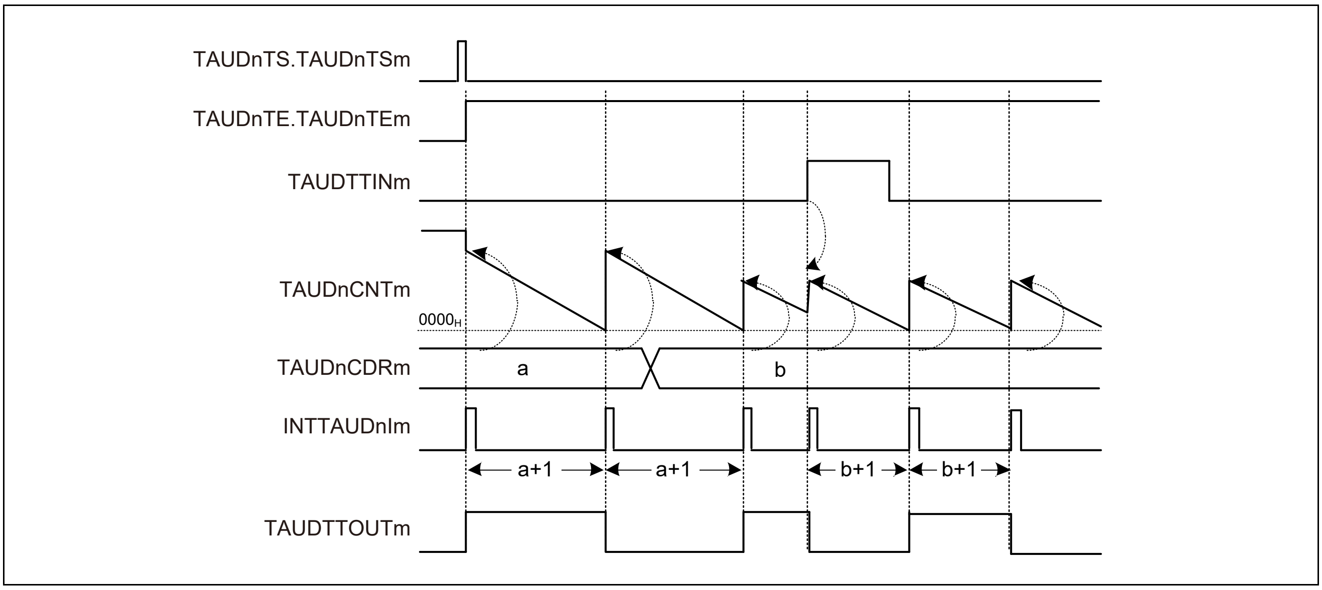


Figure 20.34 General Timing Diagram of TAUDTTINm Input Interval Timer

(4) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.13 Contents of the TAUDCMORm Register for TAUDTTINm Input Interval Timer

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation. 1: INTTAUDIm is generated and TAUDTTOUTm is toggled at the beginning and restarting of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.14 Contents of the TAUDCMURm Register for TAUDTTINm Input Interval Timer

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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges

(c) Channel output mode

Table 20.15 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled(TAUDTDE.TAUDTDEm = 0)
TAUDTDL.TAUDTDLm	(TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled
TAUDTRC.TAUDTRCm	(TAUDTRE.TAUDTREm = 0), set these bits to 0
TAUDTME.TAUDTMEm	0: Disables modulation

Remark: The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDTOE.TAUDTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see Section 20.7, Channel Output Modes.

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the TAUDTTINm Input Interval Timer. Therefore, these registers should be set to 0.

Table 20.16 Simultaneous Reload Settings for TAUDTTINm Input Interval Timer

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled
TAUDRDM.TAUDRDMm	(TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDC.TAUDRDCm	

(5) Operating Procedure for TAUDTTINm Input Interval Timer

Table 20.17 Operating Procedure for TAUDTTINm Input Interval Timer

		Operation	TAUD Status
	Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.13, Contents of the TAUDCMORm Register for TAUDTTINm Input Interval Timer, and Table 20.14, Contents of the TAUDCMURm Register for TAUDTTINm Input Interval Timer. Set the value of TAUDCDRm register. Set channel output mode by setting the control bits as described in Table 20.15, Control Bit Settings in Independent Channel Output Mode 1.	Channel operation is stopped.
Restart	Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. The TAUDCDRm value is loaded in TAUDCNTm. When TAUDCMORm.TAUDMD0 = 1, INTTAUDIm is generated and TAUDTTOUTm toggles.
	During Operation	The values of the TAUDCMURm.TAUDTIS[1:0] and the TAUDCDRm register are changeable at any time. The TAUDCNTm register can be read at all times. Detection of TAUDTTINm edge	TAUDCNTm counts down. When the counter reaches 0000H: • The TAUDCDRm value is loaded in TAUDCNTm again and count operation continues. • INTTAUDIm is generated and TAUDTTOUTm toggles. When an effective edge of the TAUDTTINm input signal is detected during count operation, the TAUDCDRm value is loaded in TAUDCNTm and count operation continues. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

(6) Specific Timing Diagrams

The timing diagrams in Section 20.12.1, Interval Timer, are applied, but the counter can also be restarted by an effective TAUDTTINm input edge excepting the interval timer.

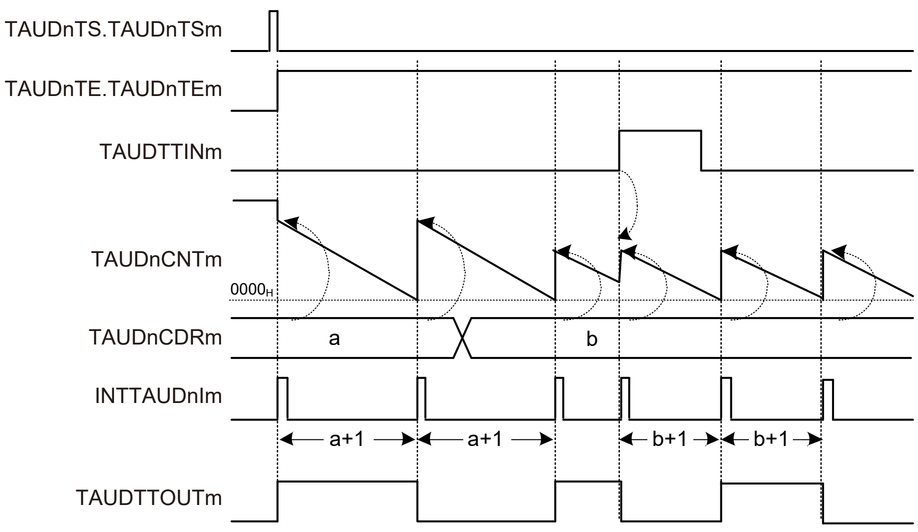


Figure 20.35 Counter Triggered by Rising TAUDTTINm Input Edge (TAUDCMURm.TAUDTIS[1:0] = 01B), TAUDCMORm.TAUDMD0 = 1

- If an effective TAUDTTINm input edge is detected, an interrupt is generated which causes TAUDTTOUTm to toggle. In this example, the effective edge is a rising edge (TAUDCMURm.TAUDTIS[1:0] = 01B)

20.12.3 Clock Frequency Division

(1) Overview

(a) Summary

This function is used as a frequency divider. The frequency of the input signal TAUDTTINm is divided by a factor related to TAUDCDRm, and the resulting signal is output to TAUDTTOUTm.

(b) Prerequisites

- TAUDTTINm should have a fixed frequency.
- The operating mode should be set to interval timer mode. (See Table 20.18, Contents of the TAUDCMORm Register for Clock Frequency Division).
- The channel output mode should be set to independent channel output mode 1. (See Section 20.7, Channel Output Modes).

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation. The current value of TAUDCDRm is loaded into TAUDCNTm and the counter starts to count down from this value, using TAUDTTINm as a count clock.

When the counter value reaches 0000H, INTTAUDIm occurs and TAUDTTOUTm signal is toggled.
Then, TAUDCDRm value is loaded into TAUDCNTm to continue operation subsequently.

The value of TAUDCDRm can be rewritten at any time. The changed value of TAUDCDRm is applied when the counter starts to count down next time.

The counter can be stopped by setting TAUDTT.TAUDTTm = 1. This sets TAUDTE.TAUDTEm = 0. TAUDCNTm and TAUDTTOUTm stop but retain their values. The function can be restarted by setting TAUDTS.TAUDTSm = 1. The counter can also be forcibly restarted without making a stop by setting TAUDTS.TAUDTSm = 1 during operation (forced restart).

(d) Conditions

If the TAUDCMORm.TAUDMD0 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 TAUDCMORm.TAUDMD0 is set to 1. For details, see Section 20.9, TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts.

Remark: TAUDTTINm input signals are sampled at the frequency of the operation clock set by TAUDCMORm.TAUDCKS[1:0] bits. Therefore, the TAUDTTOUTm output clock cycle has an error of ± 1 operation clock cycle.

(2) Equations

- When rising edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/[(TAUDCDRm + 1) × 2]
- When falling edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/[(TAUDCDRm + 1) × 2]
- When falling and rising edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/(TAUDCDRm + 1)

(3) Block Diagram and General Timing Diagram

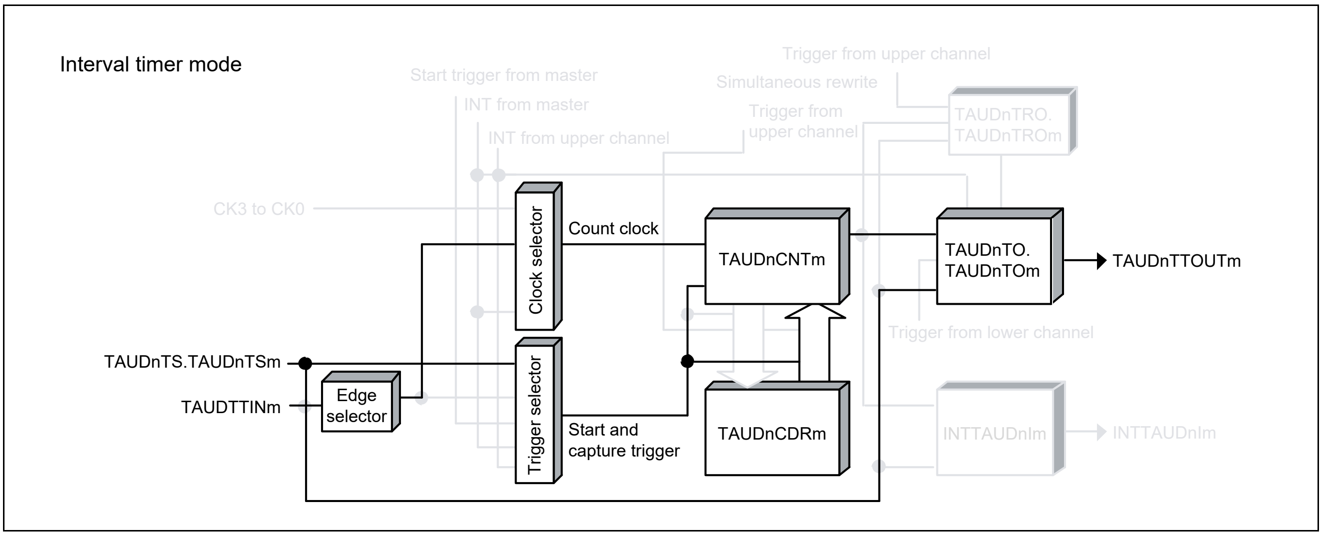


Figure 20.36 Block Diagram of Clock Frequency Division

The following settings apply to the general timing diagram.

- INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 1)
- Detection of rising edges (TAUDCMURm.TAUDTIS[1:0] = 01B)

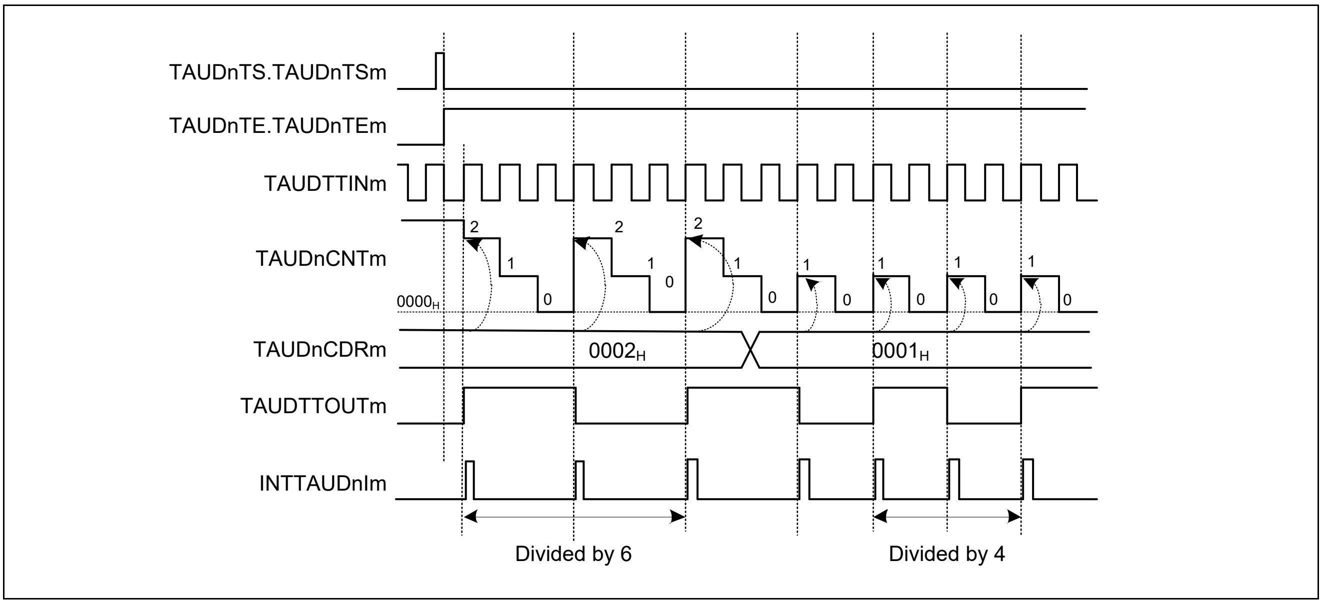


Figure 20.37 General Timing Diagram of Clock Frequency Division

(4) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.18 Contents of the TAUDCMORm Register for Clock Frequency Division

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	01: Effective edge of the TAUDTTINm input signal is used as a count clock.
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation. 1: INTTAUDIm is generated and TAUDTTOUTm is toggled at the beginning and restarting of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.19 Contents of the TAUDCMURm Register for Clock Frequency Division

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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges

(c) Channel output mode

Table 20.20 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set these bits to 0
TAUDTRC.TAUDTRCm	
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 20.21 Simultaneous Reload Settings for Clock Frequency Division

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Operating Procedure for TAUDTTINm Input Interval Timer

Table 20.22 Operating Procedure for Clock Frequency Division

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.18, Contents of the TAUDCMORm Register for Clock Frequency Division, and Table 20.19, Contents of the TAUDCMURm Register for Clock Frequency Division. Set the value of TAUDCDRm register. Set channel output mode by setting the control bits as described in Table 20.20, Control Bit Settings in Independent Channel Output Mode 1.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. TAUDCNTm loads TAUDCDRm value. If TAUDCMORm.TAUDMD0 is set to 1, INTTAUDIm is generated and TAUDTTOUTm is toggled.
During Operation	The value of TAUDCDRm is changeable at any time. The TAUDCNTm register can be read at all times.	TAUDCNTm counts down each time TAUDTTINm input edge is detected. When the counter reaches 0000H: • TAUDCDRm value is loaded in TAUDCNTm and count operation continues. • INTTAUDIm is generated. • TAUDTTOUTm is toggled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops. TAUDCNTm and TAUDTTOUTm retain their current values.

Restart (from Stop Operation back to During Operation)

(6) Specific Timing Diagrams

(a) TAUDCDRm = 0000H

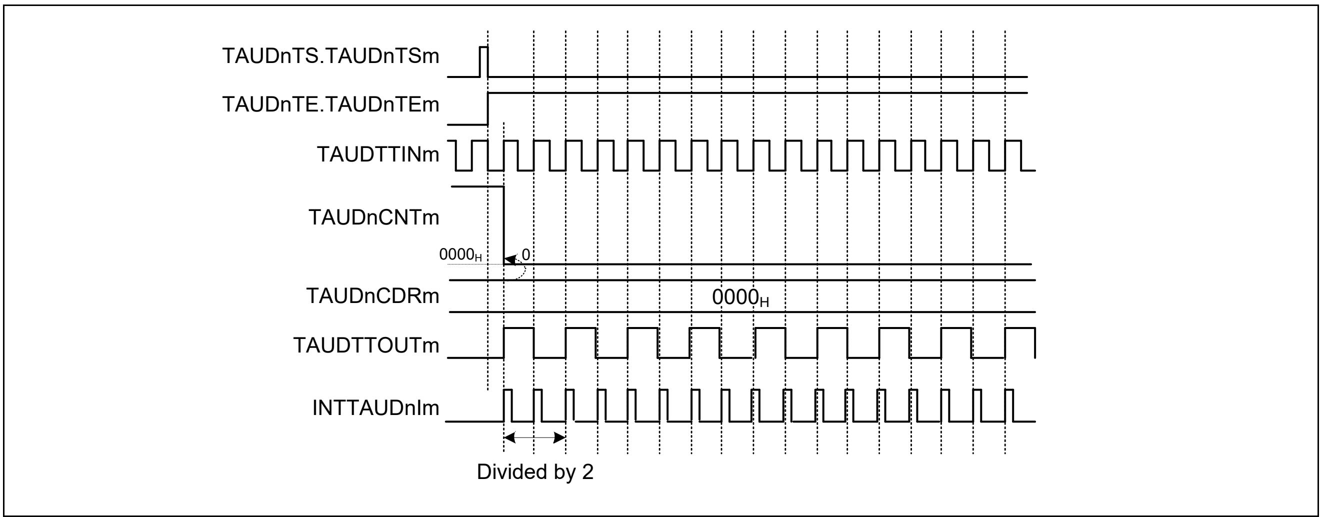


Figure 20.38 TAUDCDRm = 0000H, TAUDCMORm.TAUDMD0 = 1, TAUDCMURm.TAUDTIS[1:0] = 01B

- If TAUDCDRm is 0000H, TAUDCNTm is always 0000H.
- INTTAUDIm is generated every count clock, resulting in TAUDTTOUTm toggling every count clock.

Figure 20.38 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 TAUDIm pin and TAUD.

(b) Restart

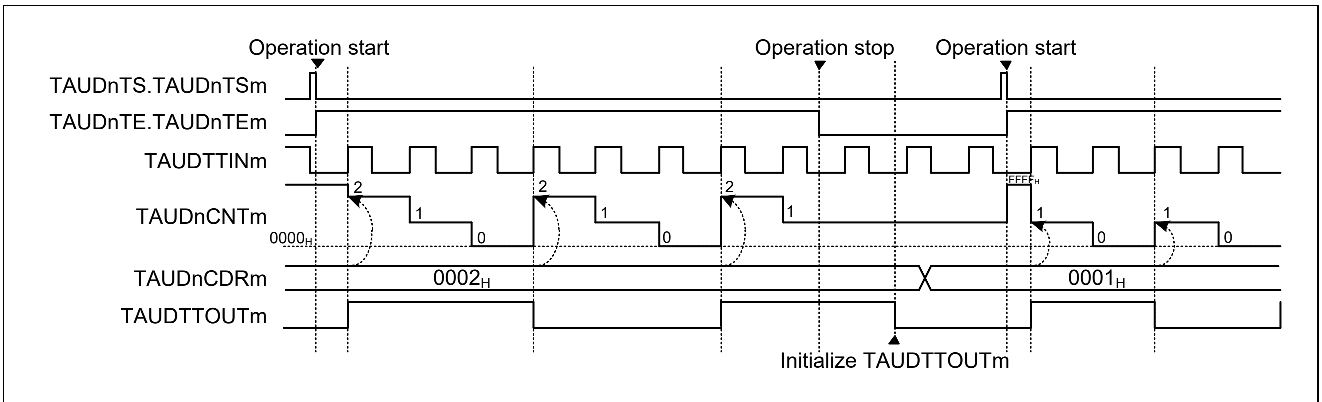


Figure 20.39 Restart (TAUDCMORm.TAUDMD0 = 1, TAUDCMURm.TAUDTIS[1:0] = 01B)

To reset the value of TAUDTTOUTm:

- Set TAUDTOE.TAUDTOEm = 0 when the counter is stopped (TAUDTE.TAUDTEm = 0).
- Then, write either 0 or 1 to TAUDTO.TAUDTOm to set the new start value of TAUDTTOUTm.

(c) Forced restart

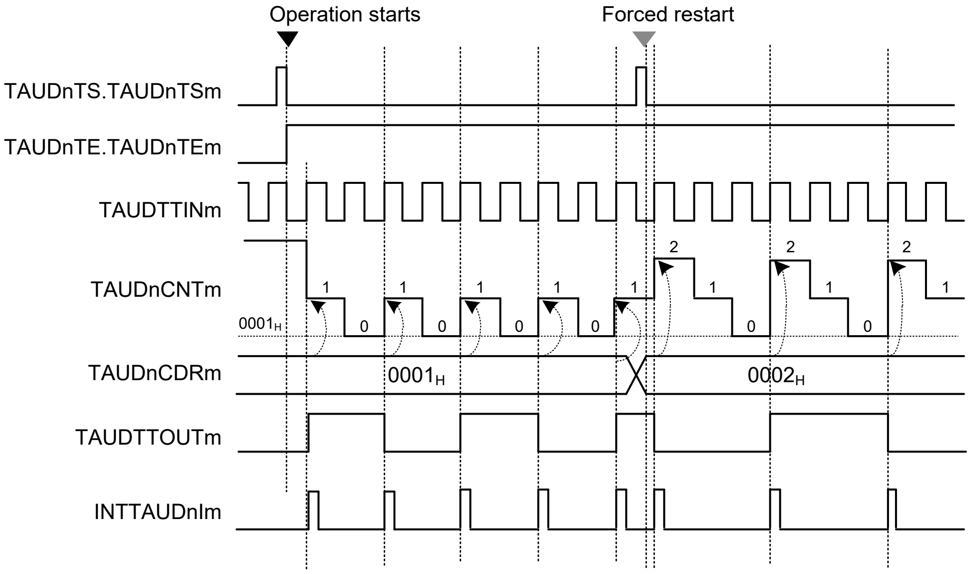


Figure 20.40 Forced Restart Operation (TAUDCMORm.TAUDMD0 = 1, TAUDCMURm.TAUDTIS[1:0] = 01B)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDTS.TAUDTSm = 1 during operation.
- The value of TAUDCDRm is written to TAUDCNTm and the count operation restarts.
- TAUDTTOUTm restarts at the same level as before the forced restart.

20.12.4 External Event Counting

(1) Overview

(a) Summary

This function is used as an event timer, which generates an interrupt (INTTAUDIm) when a specific number of TAUDTTINm input pulses has occurred.

(b) Prerequisites

- The operating mode should be set to the event count mode. (See Table 20.23, Contents of the TAUDCMORm Register for External Event Counting.
- TAUDTTOUTm is not used with this function.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation. When the counter starts, the current value of TAUDCDRm is loaded into TAUDCNTm.
When an effective TAUDTTINm input edge is detected, the value of TAUDCNTm decrements by 1. TAUDCNTm retains this value until an effective TAUDTTINm input edge is detected or the counter is restarted.
When the effective edge is detected for the (TAUDCDRm + 1) times, INTTAUDIm is generated. Then, TAUDCDRm value is loaded into TAUDCNTm to continue operation subsequently.
The counter can be stopped by setting TAUDTT.TAUDTTm to 1. This sets TAUDTE.TAUDTEm to 0. The counter can be restarted by setting TAUDTS.TAUDTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUDTS.TAUDTSm to 1 during operation.
The value of TAUDCDRm can be rewritten at any time, and the changed value of TAUDCDRm is applied the next time the counter starts to count down.

(d) Conditions

An edge type used as a trigger is specified by TAUDCMURm.TAUDTIS[1:0] bits.

- When TAUDCMURm.TAUDTIS[1:0] = 00B, falling edges are counted.
- When TAUDCMURm.TAUDTIS[1:0] = 01B, rising edges are counted.
- When TAUDCMURm.TAUDTIS[1:0] = 10B, both edges are counted.

(2) Equations

Number of effective edges detected before INTTAUDIm generation = TAUDCDRm + 1

(3) Block Diagram and General Timing Diagram

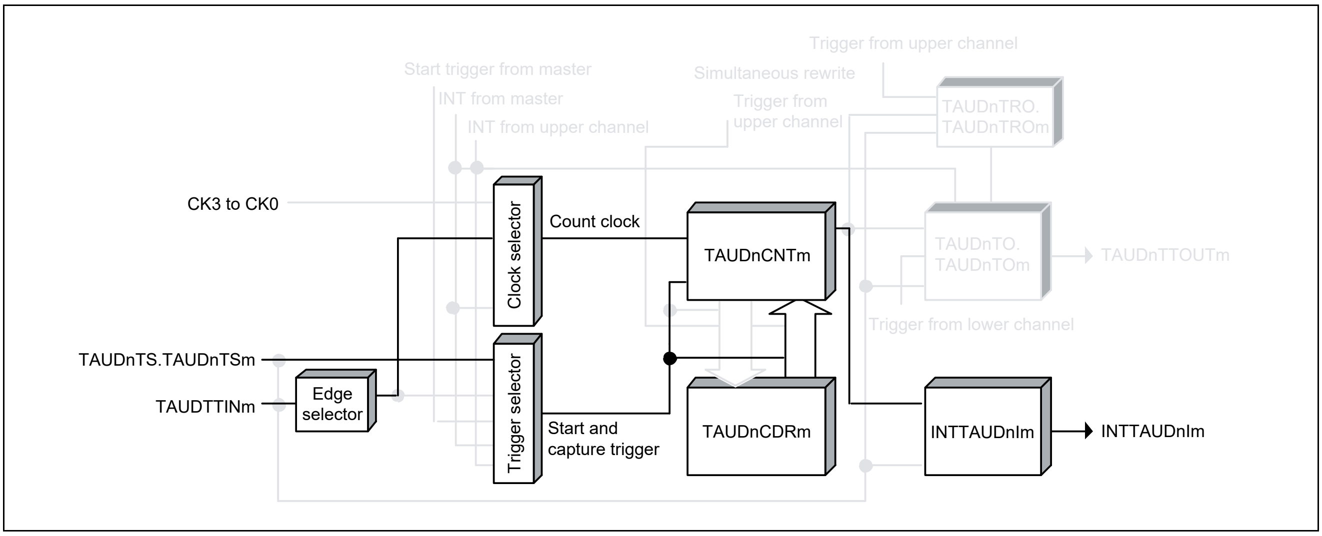


Figure 20.41 Block Diagram of External Event Counting

The following settings apply to the general timing diagram.

- Detection of rising edges (TAUDCMURm.TAUDTIS[1:0] = 01B)

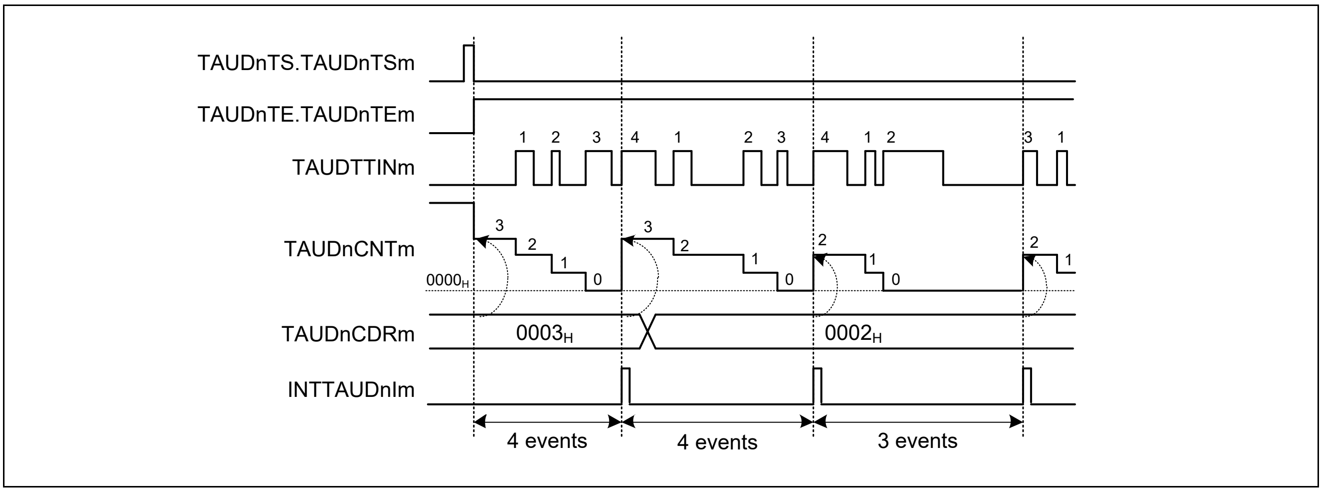


Figure 20.42 General Timing Diagram of External Event Counting

(4) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.23 Contents of the TAUDCMORm Register for External Event Counting

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	01: Effective edge of the TAUDTTINm input signal is used as a count clock.
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0011: Event count mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.24 Contents of the TAUDCMURm Register for External Event Counting

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	TAUDTIS[1:0]	00: Falling edge is detected. 01: Rising edge is detected. 10: Both edges are detected.

(c) Channel output mode

The channel output mode is not used by this function.

(d) Simultaneous reloading

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 20.25 Simultaneous Reload Settings for External Event Counting

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Operating Procedure for External Event Counting

Table 20.26 Operating Procedure for External Event Counting

		Operation	TAUD Status
	Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.23, Contents of the TAUDCMORm Register for External Event Counting, and Table 20.24, Contents of the TAUDCMURm Register for External Event Counting. Set the value of TAUDCDRm register.	Channel operation is stopped.
	Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. TAUDCNTm loads TAUDCDRm value and waits for TAUDTTINm input edge detection.
Restart	During Operation	Detection of TAUDTTINm edge The value of TAUDCDRm is changeable at any time. The TAUDCNTm register can be read at any time.	TAUDCNTm counts down each time TAUDTTINm input edge is detected. When the counter reaches 0000H: • TAUDCDRm value is loaded in TAUDCNTm and count operation continues. • INTTAUDIm is generated. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

(6) Specific Timing Diagrams

(a) TAUDCDRm = 0000H

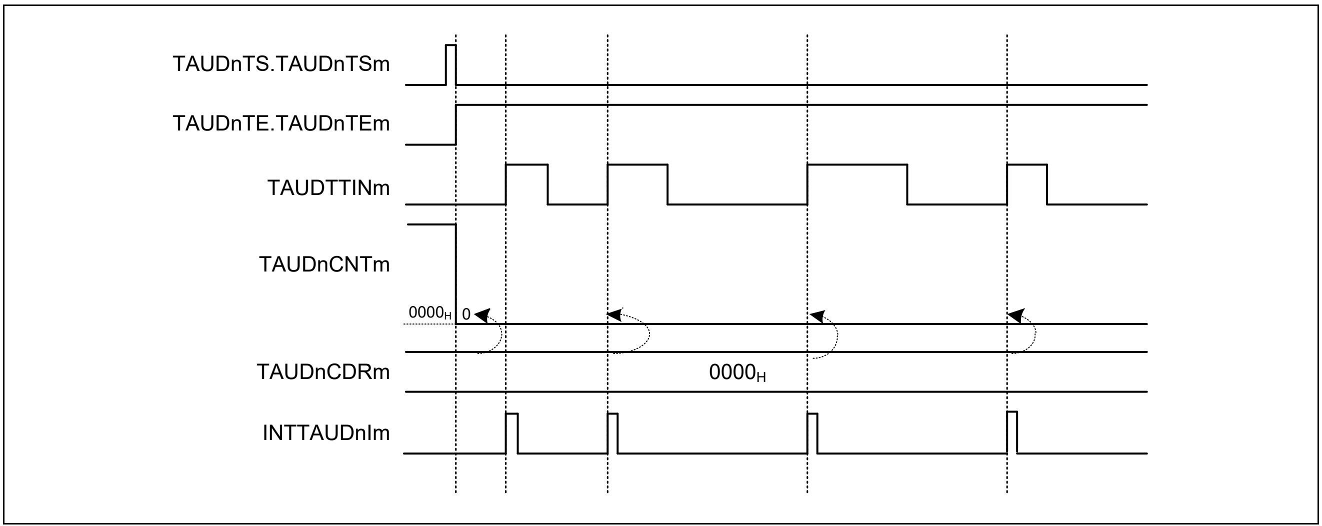


Figure 20.43 TAUDCDRm = 0000H, TAUDCMURm.TAUDTIS[1:0] = 01B

- If 0000H = TAUDCDRm, 0000H is loaded into TAUDCNTm each time an effective TAUDTTINm input edge is detected.
  In other words, INTTAUDIm is generated each time an effective TAUDTTINm input edge is detected.

(b) Operation stop and restart

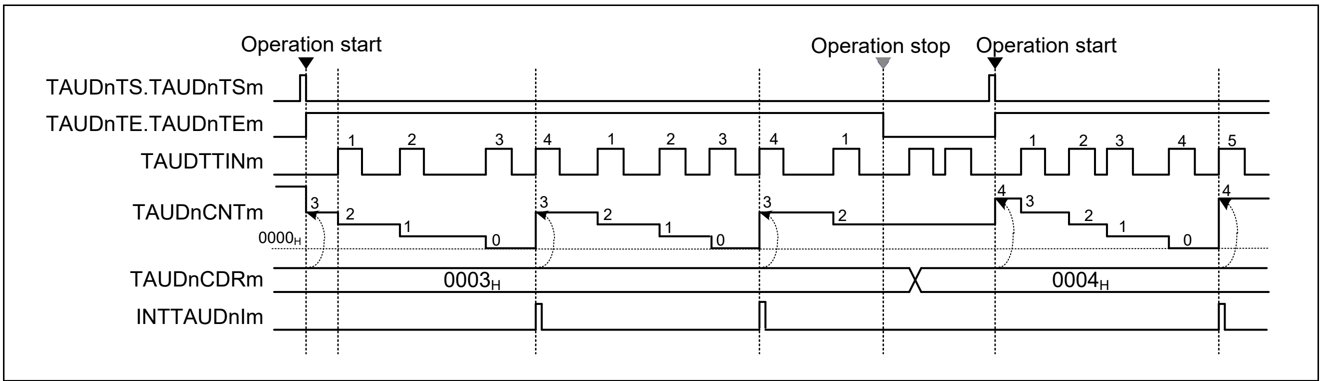


Figure 20.44 Operation Stop and Restart (TAUDCMURm.TAUDTIS[1:0] = 01B)

- The counter can be stopped by setting TAUDTT.TAUDTTm to 1. This sets TAUDTE.TAUDTEm to 0.
- TAUDCNTm stops and retains its current value. TAUDTTINm continues and TAUDCNTm ignores the effective edge.
- The counter can be restarted by setting TAUDTS.TAUDTSm to 1. TAUDCNTm loads the TAUDCDRm value and restarts count operation.

(c) Forced restart

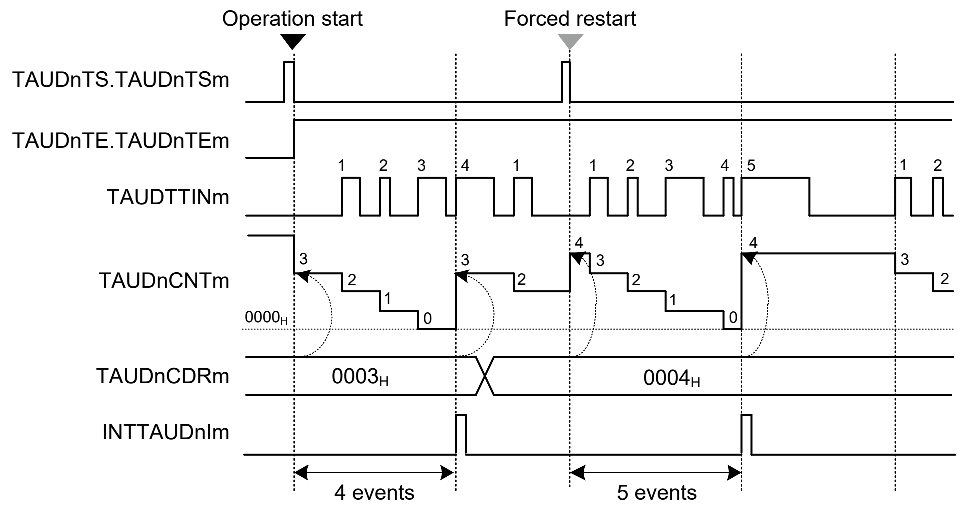


Figure 20.45 Forced Restart Operation (TAUDCMURm.TAUDTIS[1:0] = 01B)

When the counter is forcibly restarted, the changed TAUDCDRm value is applied to TAUDCNTm.

- The counter can be restarted without making a stop by setting TAUDTS.TAUDTSm to 1 during operation.
- The value of TAUDCDRm is loaded into TAUDCNTm and the counter awaits the next effective TAUDTTINm input edge.

20.12.5 Delay Counting

(1) Overview

(a) Summary

This function generates interrupts (INTTAUDIm), which have a defined delay to the TAUDTTINm input signal.
TAUDTTINm input signal pulses that occur within the delay period are ignored.

(b) Prerequisites

- The operating mode should be set to one-count mode. See Table 20.27, Contents of the TAUDCMORm Register for Delay Counting.
- TAUDTTOUTm is not used with this function.
- Trigger detection should be disabled during counting (TAUDCMORm.TAUDMD0 = 0).

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation.
The counter starts when an effective TAUDTTINm input start edge is detected. The value of TAUDCDRm is loaded into TAUDCNTm and the counter starts to count down from the TAUDCDRm value.
When the counter reaches 0000_H, an interrupt is generated. The counter returns to $FFFF_H$ and awaits the next effective TAUDTTINm input edge.
When the counter is counting down, further TAUDTTINm input signals are ignored, i.e., the counter is not reset.
The value of TAUDCDRm can be rewritten at any time, and the changed value of TAUDCDRm is applied the next time the counter starts to count down.

(d) Conditions

The type of edge used as a trigger is specified by the TAUDCMURm.TAUDTIS[1:0] bits.

- If TAUDCMURm.TAUDTIS[1:0] = 00B, the falling edge is counted.
- If TAUDCMURm.TAUDTIS[1:0] = 01B, the rising edge is counted.
- If TAUDCMURm.TAUDTIS[1:0] = 10B, both edges are counted.

(2) Equations

Delay between TAUDTTINm and INTTAUDIm = count clock cycle × (TAUDCDRm + 1)

(3) Block Diagram and General Timing Diagram

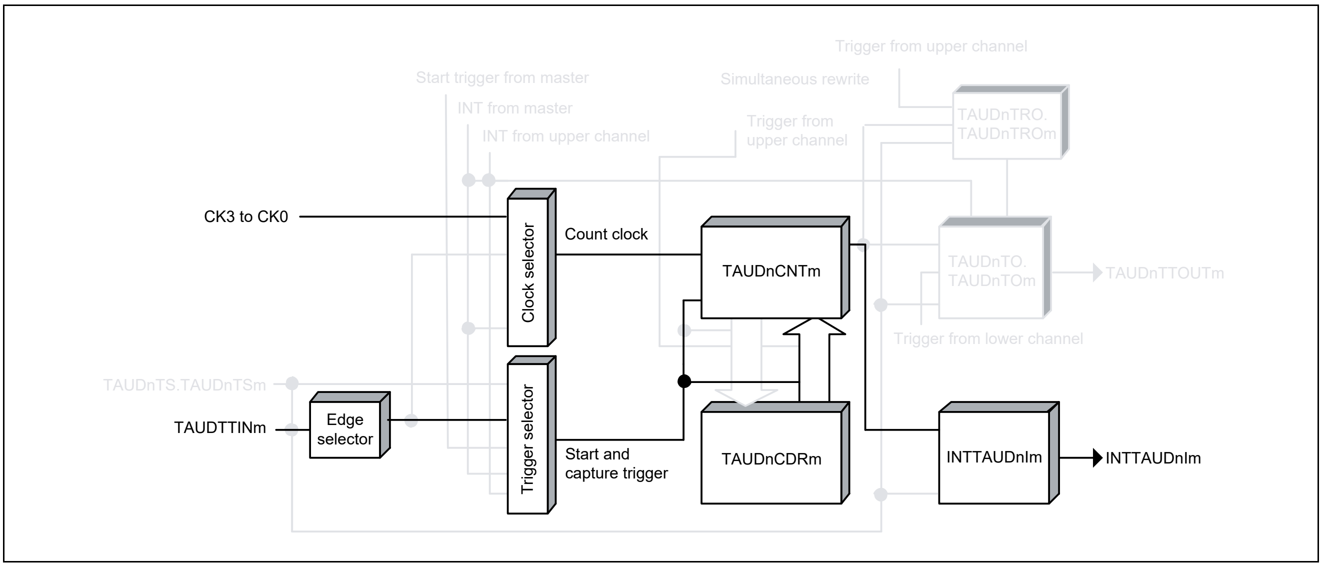


Figure 20.46 Block Diagram of Delay Counting

The following settings apply to the general timing diagram.

- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

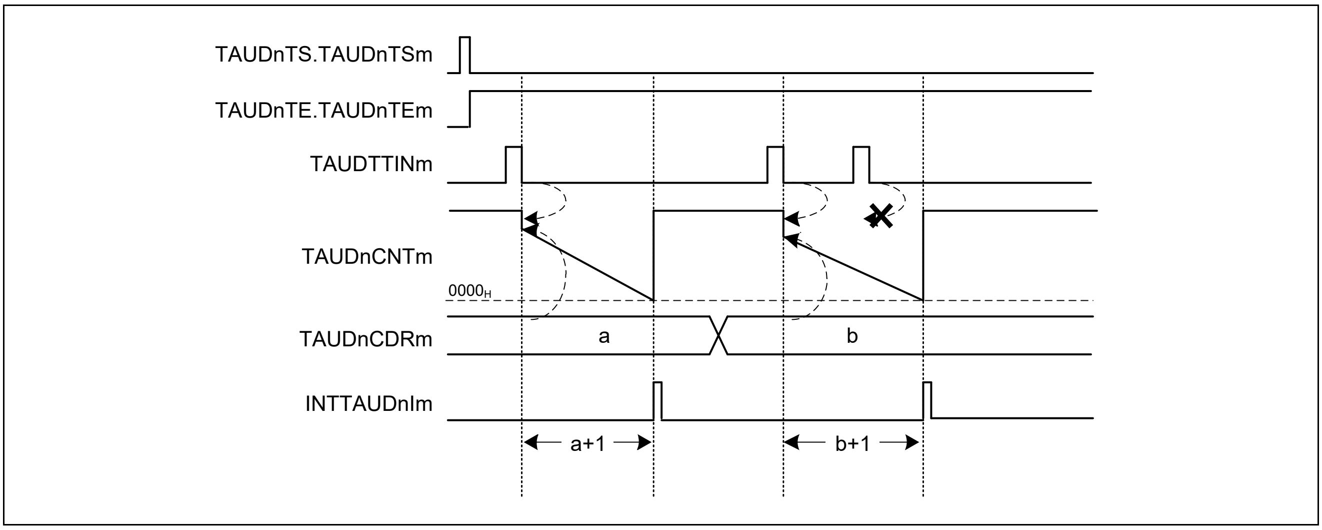


Figure 20.47 General Timing Diagram of Delay Counting

(4) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.27 Contents of the TAUDCMORm Register for Delay Counting

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	0: Disables a start trigger during operation

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.28 Contents of the TAUDCMURm Register for Delay Counting

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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode

TAUDTOE.TAUDTOEm 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.

(d) Simultaneous reloading

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 20.29 Simultaneous Reload Settings for Delay Counting

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Operating Procedure for Delay Counting

Table 20.30 Operating Procedure for Delay Counting

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.27, Contents of the TAUDCMORm Register for Delay Counting, and Table 20.28, Contents of the TAUDCMURm Register for Delay Counting. Set the value of TAUDCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDTE.TAUDTEm is set to 1 and TAUDCNTm waits for detection of the TAUDTTINm start edge. When a start edge is detected, the TAUDCDRm value is loaded in TAUDCNTm
During Operation	The TAUDCDRm register value can be changed at any time. The TAUDCNTm register can be read at all times.	TAUDCNTm counts down. When the counter reaches 0000H. INTTAUDIm is generated. TAUDCNTm stops counting, returns FFFFH, and waits for a trigger. If a trigger occurs while TAUDCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its value.

Restart (from Stop Operation back to During Operation)

20.12.6 One-Pulse Output

(1) Overview

(a) Summary

This function generates an interrupt (INTTAUDIm) when an effective 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.

(b) Prerequisites

- The channel output mode should be set to independent channel output mode 2. (See Table 20.31, Contents of the TAUDCMORm Register for One-Pulse Output.)
- The channel output mode should be set to independent channel output mode 1. (See Section 20.7, Channel Output Modes.)
- Trigger detection should be disabled during counting (TAUDCMORm.TAUDMD0 = 0).

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation.
The counter starts when an effective TAUDTTINm input edge is detected. The value of TAUDCDRm is loaded into TAUDCNTm and the counter starts to count down from the TAUDCDRm value. An interrupt is generated and TAUDTTOUTm toggles.
When the counter reaches 0001H, an interrupt is generated and TAUDTTOUTm is set to the inactive level. The counter stops at 0000H and awaits the next effective TAUDTTINm input edge.
When the counter is counting down, further TAUDTTINm input signals are ignored, i.e., the counter is not reset.
The value of TAUDCDRm can be rewritten at any time, and the changed value of TAUDCDRm is applied the next time the counter starts to count down.

(d) Conditions

The type of edge used as a trigger is specified by the TAUDCMURm.TAUDTIS[1:0] bits.

- If TAUDCMURm.TAUDTIS[1:0] = 00B, falling edges trigger the counter.
- If TAUDCMURm.TAUDTIS[1:0] = 01B, rising edges trigger the counter.
- If TAUDCMURm.TAUDTIS[1:0] = 10B, rising and falling edges trigger the counter.

(2) Equations

Interval between TAUDTTINm and INTTAUDIm
= TAUDTTOUTm (timer output) width = count clock cycle × TAUDCDRm

(3) Block Diagram and General Timing Diagram

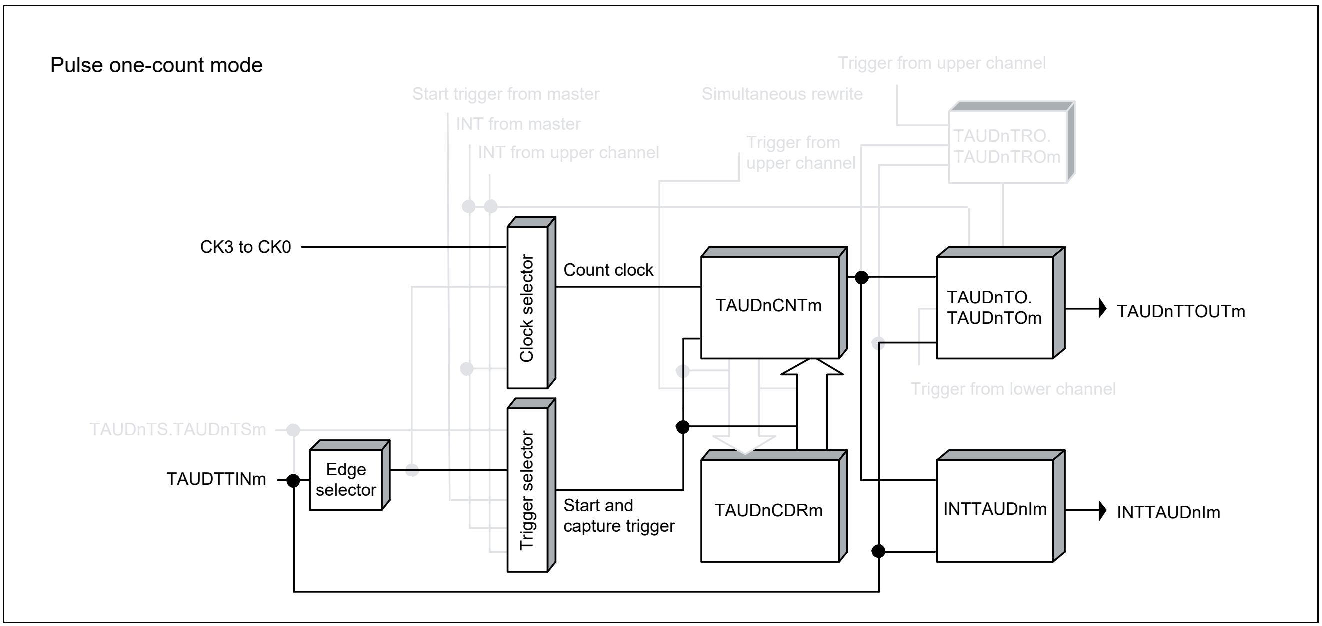


Figure 20.48 Block Diagram of One-Pulse Output

The following settings apply to the general timing diagram.

- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

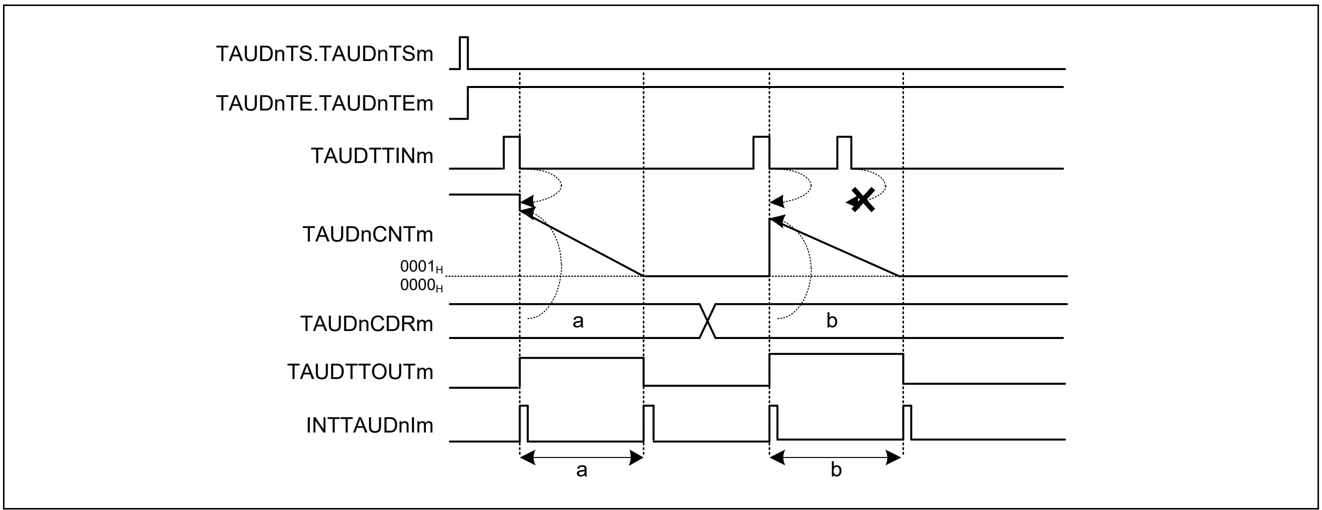


Figure 20.49 General Timing Diagram of One-Pulse Output

(4) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.31 Contents of the TAUDCMORm Register for One-Pulse Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	1010: Pulse one-count mode
0	TAUDMD0	0: Disables a start trigger during operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.32 Contents of the TAUDCMURm Register for One-Pulse Output

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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode

Table 20.33 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode controlled by software.
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	1: Set/reset mode
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set these bits to 0
TAUDTRC.TAUDTRCm	
TAUDTME.TAUDTMEm	0: Disables modulation

Remark: The channel output mode can also be set to channel output mode controlled by software by setting TAUDTOE.TAUDTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts.
For details, see Table 20.7, Channel Output Modes.

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the One-Pulse Output. Therefore, these registers should be set to 0.

Table 20.34 Simultaneous Reload Settings for One-Pulse Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Operating Procedure for One-Pulse Output

Table 20.35 Operating Procedure for One-Pulse Output

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.31, Contents of the TAUDCMORm Register for One-Pulse Output, and Table 20.32, Contents of the TAUDCMURm Register for One-Pulse Output. Set the value of TAUDCDRm register. Set channel output mode by setting the control bits as described in Table 20.33, Control Bit Settings in Independent Channel Output Mode 2.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDTE.TAUDTEm is set to 1 and TAUDCNTm waits for detection of the TAUDTTINm start edge. When a start edge is detected, TAUDCNTm loads the TAUDCDRm value.
During Operation	The value of TAUDCDRm register is changeable at any time. The TAUDCNTm register can be read at all times.	INTTAUDIm is generated when TAUDCNTm starts and TAUDTTOUTm is set to its active level. TAUDCNTm counts down. When the counter reaches 0001H: • INTTAUDIm is generated. • TAUDTTOUTm is set to its inactive level. TAUDCNTm stops counting and waits for a trigger. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

Restart (from Stop Operation back to Start Operation)

20.12.7 TAUDTTINm Input Pulse Interval Measurement

(1) Overview

(a) Summary

This function captures the count value and uses this value and the overflow bit TAUDCSRm.TAUDOVF to measure the interval of the TAUDTTINm input signal.

(b) Prerequisites

- The operating mode should be set to capture mode. See Table 20.37, Contents of the TAUDCMORm Register for TAUDTTINm Input Pulse Interval Measurement.
- TAUDTTOUTm is not used with this function.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation. The counter TAUDCNTm starts to count up from 0000H. When an effective TAUDTTINm edge is detected, the value of TAUDCNTm is captured, transferred to TAUDCDRm, and an interrupt INTTAUDIm is generated. The counter resets to 0000H and subsequently continues operation.
If the counter reaches FFFFH before an effective TAUDTTINm edge is detected, it overflow. The counter is reset to 0000H and subsequently continues operation. The values transferred to TAUDCDRm and TAUDCSRm.TAUDOVF respectively depend on the values of bits TAUDCMORm.TAUDCOS[1:0].

Table 20.36 Effects of Overflow

TAUDCMORm. TAUDCOS[1:0]	When Overflow Occurs		When an Effective TAUDTTINm Input is Detected	
	TAUDCDRm	TAUDCSRm. TAUDOVF	TAUDCDRm, TAUDCNTm	TAUDCSRm. TAUDOVF
00	Unchanged	0	TAUDCNTm loaded into TAUDCDRm	1
01		1		
10	Set to FFFFH	0	TAUDCNTm set to 0, TAUDCDRm unchanged	Unchanged
11		1		

When TAUDCMORm.TAUDCOS[0] = 1, the overflow bit (TAUDCSRm.TAUDOVF) can be cleared only by setting TAUDCSCm.TAUDCLOV = 1.
The combination of the value of TAUDCDRm and TAUDCSRm.TAUDOVF can be used to deduce the interval of the TAUDTTINm signal. However, if an overflow occurs multiple times before an effective TAUDTTINm input is detected, the overflow bit TAUDCSRm.TAUDOVF cannot indicate the occurrence of multiple overflows.
The function can be stopped by setting TAUDTT.TAUDTTm = 1. This sets TAUDTE.TAUDTEm = 0. TAUDCNTm stops but retains its value. While the function is stopped, effective TAUDTTINm input edge detection and TAUDCNTm capture are not performed.
The counter is reset to 0000H and subsequently continues operation.

(d) Conditions

If the TAUDCMORm.TAUDMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see Section 20.9, TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts.

Remark: When TAUDCMORm.TAUDCOS[1:0] = 10B or 11B, the value of TAUDCNTm is not loaded into TAUDCDRm when the first effective TAUDTTINm input edge occurs after an overflow. However, an interrupt is generated.

(e) Equations

TAUDTTINm input pulse interval
= count clock cycle × [(TAUDCSRm.TAUDOV F × ($FFFF_H$ + 1)) + TAUDCDRm capture value + 1]

(2) Block Diagram and General Timing Diagram

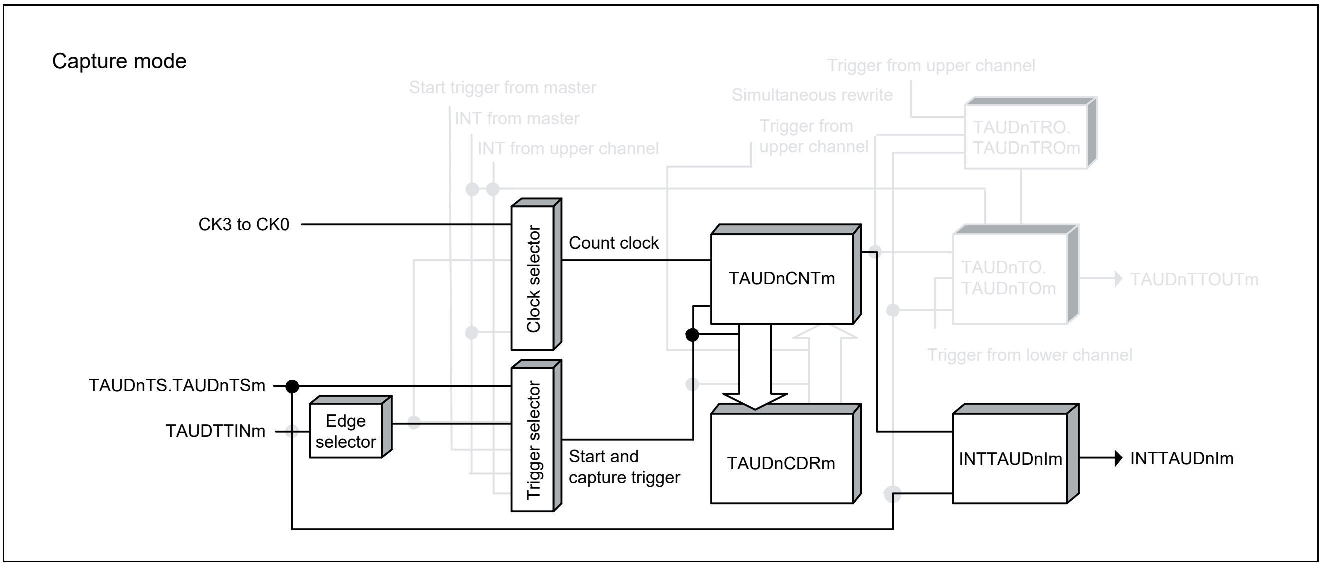


Figure 20.50 Block Diagram of TAUDTTINm Input Pulse Interval Measurement

The following settings apply to the general timing diagram.

- INTTAUDIm is not generated at the beginning of operation (TAUDCMORm.TAUDMD0 = 0).
- Falling edge detection (TAUDCMURm.TAUDTIS[1:0] = 00B)
- When an effective TAUDTTINm input is detected after an overflow, TAUDCDRm is changed and TAUDCSRm.TAUDOVF is set to 1 (TAUDCMORm.TAUDCOS[1:0] = 00B).

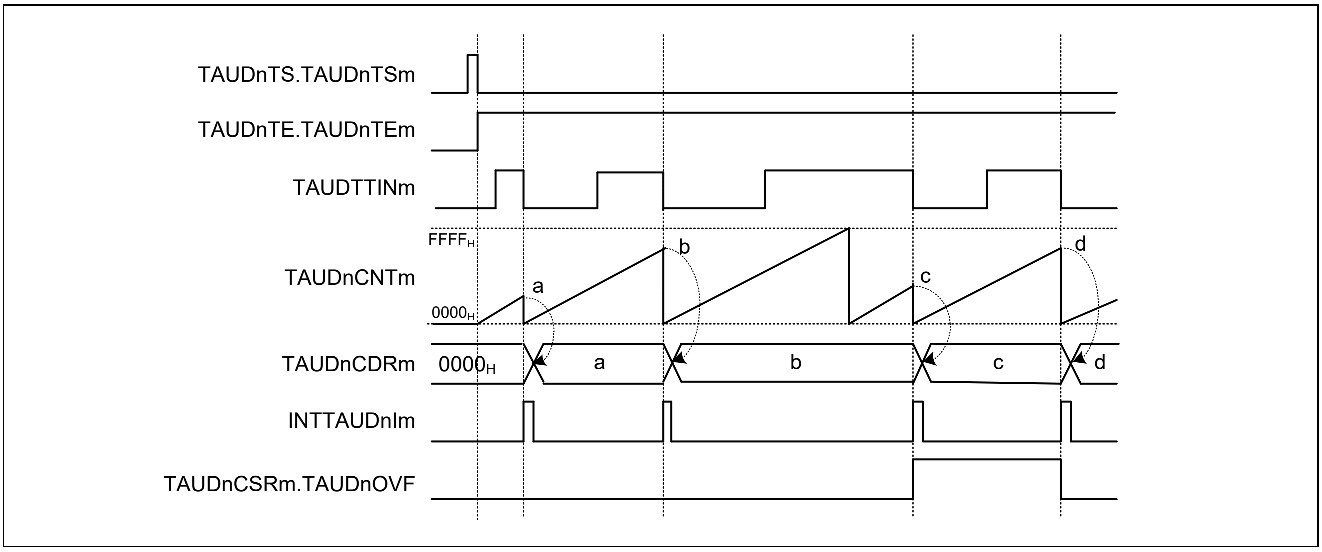


Figure 20.51 General Timing Diagram of TAUDTTINm Input Pulse Interval Measurement

(3) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.37 Contents of the TAUDCMORm Register for TAUDTTINm Input Pulse Interval Measurement

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is the external capture trigger.
7, 6	TAUDCOS[1:0]	See Table 20.36, 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	TAUDMD[4:1]	0010: Capture mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation. 1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.38 Contents of the TAUDCMURm Register for TAUDTTINm Input Pulse Interval 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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the TAUDTTINm input pulse interval measurement. Therefore, these registers should be set to 0.

Table 20.39 Simultaneous Reload Settings for TAUDTTINm Input Pulse Interval Measurement

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(4) Operating Procedure for TAUDTTINm Input Pulse Interval Measurement

Table 20.40 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.37, Contents of the TAUDCMORm Register for TAUDTTINm Input Pulse Interval Measurement, and Table 20.38, Contents of the TAUDCMURm Register for TAUDTTINm Input Pulse Interval Measurement. The TAUDCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. TAUDCNTm is cleared to 0000H. INTTAUDIm is generated when TAUDCMORm.TAUDMD0 is set to 1.
During Operation	Detection of TAUDTTINm edge The values of TAUDCMURm.TAUDTIS[1:0] bits can be changed at any time. The TAUDCDRm and TAUDCSRm registers can be read at any time. TAUDCSCm.TAUDCLOV can be written to 1. (TAUDCSRm.TAUDOVF bit is cleared to 0.)	TAUDCNTm starts to count up from 0000H. When an effective edge of TAUDTTINm is detected: • TAUDCNTm transfers (captures) its value to TAUDCDRm, and returns to 0000H. • INTTAUDIm is then generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and both it and TAUDCSRm.TAUDOVF retain their current values.

Restart

(5) Specific Timing Diagrams: Overflow Operation

(a) TAUDCMORm.TAUDCOS[1:0] = 00B

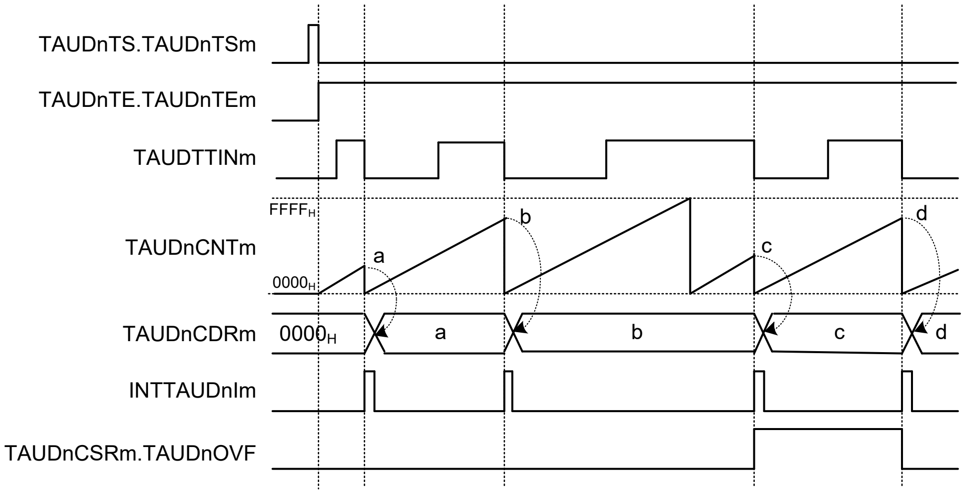


Figure 20.52 TAUDCMORm.TAUDCOS[1:0] = 00B, TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 00B

- When an overflow occurs, the value of TAUDCDRm remains unchanged and TAUDCSRm.TAUDOVF remains 0.
- Upon detection of the next effective TAUDTTINm input edge, the value of TAUDCNTm is loaded into TAUDCDRm and TAUDCSRm.TAUDOVF is set to1.
- Upon detection of the next effective TAUDTTINm input edge with no overflow occurring, TAUDCSRm.TAUDOVF is cleared to 0.

(b) TAUDCMORm.TAUDCOS[1:0] = 01B

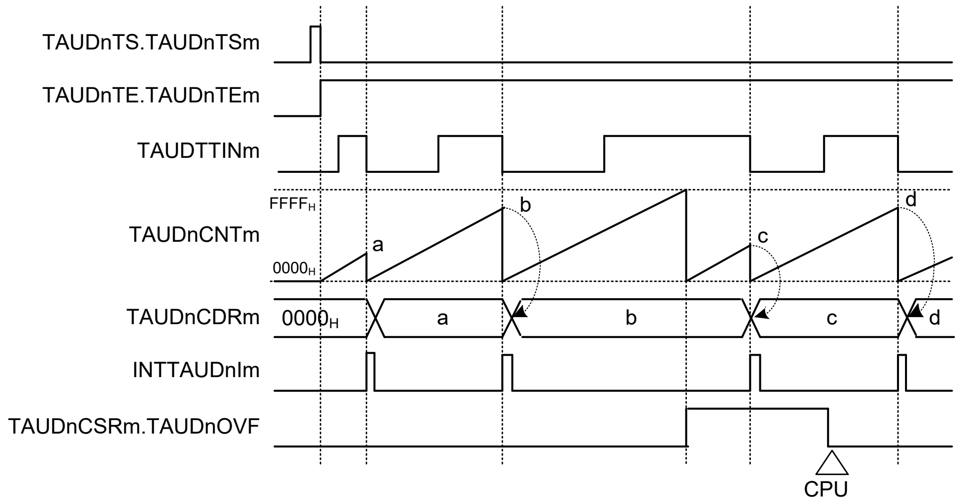


Figure 20.53 TAUDCMORm.TAUDCOS[1:0] = 01B, TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 00B

- When an overflow occurs, the value of TAUDCDRm remains unchanged and TAUDCSRm.TAUDOVF is set to1.
- Upon detection of the next effective TAUDTTINm input edge, the value of TAUDCNTm is loaded into TAUDCDRm.
- TAUDCSRm.TAUDOVF is only cleared by a CPU command (by setting TAUDCSCm.TAUDCLOV bit to 1).

(c) TAUDCMORm.TAUDCOS[1:0] = 10B

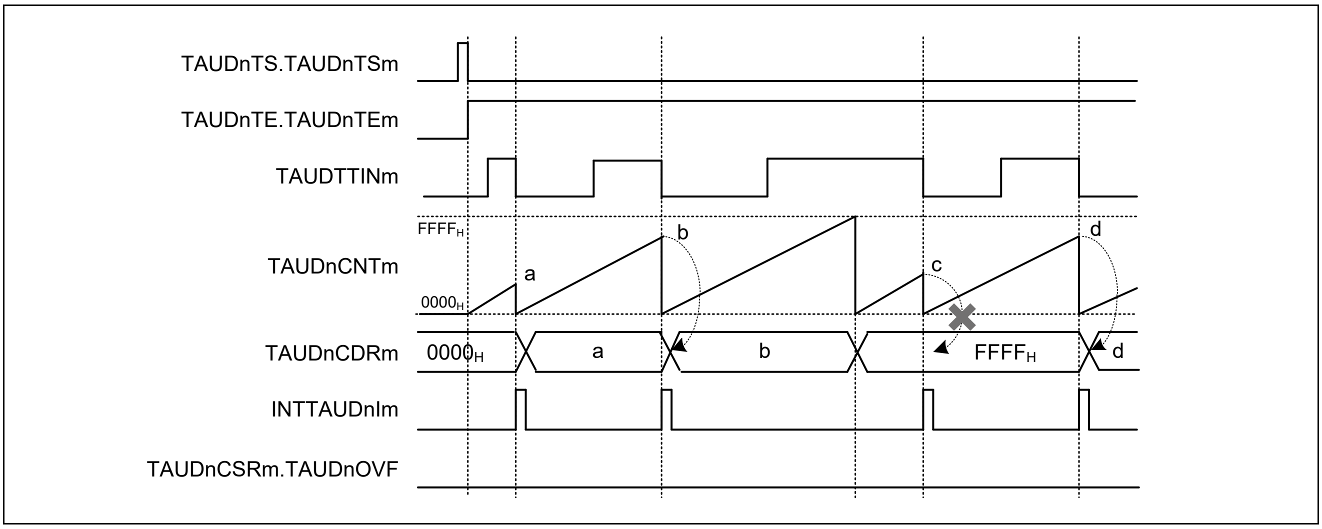


Figure 20.54 TAUDCMORm.TAUDCOS[1:0] = 10B, TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 00B

- When an overflow occurs, TAUDCDRm is set to FFFFH and TAUDCSRm.TAUDOVF remains 0.
- Upon detection of the next effective TAUDTTINm input edge, TAUDCNTm is reset to 0, but TAUDCDRm and TAUDCSRm.TAUDOVF remain unchanged.
- Thus, the next effective TAUDTTINm input edge after the overflow is ignored.

TAUDCMORm.TAUDCOS[1:0] = 11B

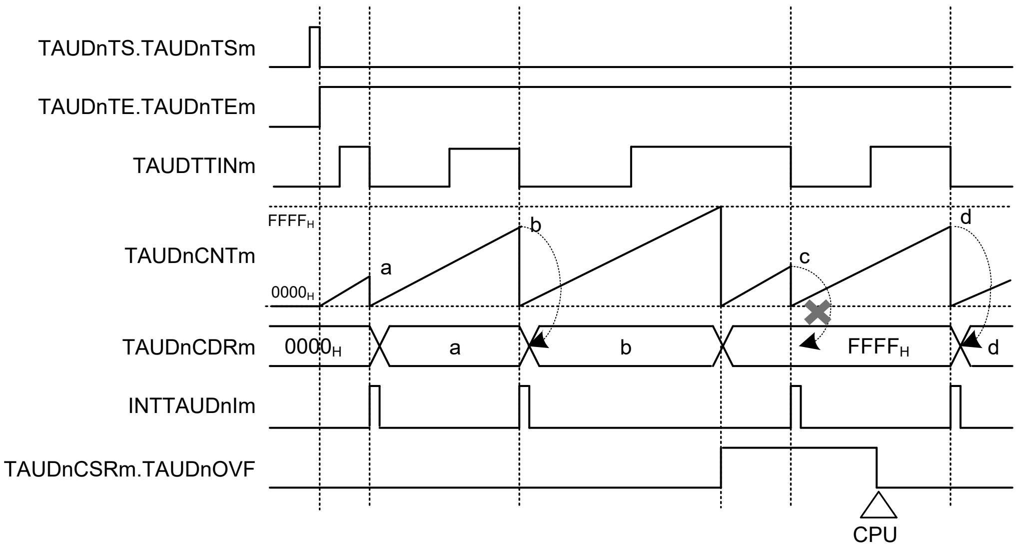


Figure 20.55 TAUDCMORm.TAUDCOS[1:0] = 11B, TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 00B

- When an overflow occurs, TAUDCDRm is set to FFFFH and TAUDCSRm.TAUDOVF is set to 1.
- Upon detection of the next effective TAUDTTINm input edge, TAUDCNTm is reset to 0, but TAUDCDRm and TAUDCSRm.TAUDOVF remain unchanged.
- Thus, the next effective TAUDTTINm input edge after the overflow is ignored.
- TAUDCSRm.TAUDOVF is cleared by setting TAUDCSCm.TAUDCLOV to 1.

20.12.8 TAUDTTINm Input Signal Width Measurement

(1) Overview

(a) 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.

(b) Prerequisites

- The operating mode should be set to capture and one-count mode. See Table 20.42, Contents of the TAUDCMORm Register for TAUDTTINm Input Signal Width Measurement.
- TAUDTTOUTm is not used with this function.
- TAUDCMORm.TAUDMD0 should be set to 0.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation. When an effective TAUDTTINm start edge is detected, the counter TAUDCNTm starts to count up from 0000H. When an effective TAUDTTINm stop edge is detected, the value of TAUDCNTm is captured, transferred to TAUDCDRm, and an interrupt INTTAUDIm is generated. The counter retains its value (TAUDCDRm +1) and awaits the next effective TAUDTTINm input start edge.
If the counter reaches FFFFH before an effective TAUDTTINm stop edge is detected, it overflows. The counter is reset to 0000H and subsequently continues operation. The values transferred to TAUDCDRm and TAUDCSRm.TAUDOVF respectively depend on the values of bits TAUDCMORm.TAUDCOS[1:0].

Table 20.41 Effects of Overflow

TAUDCMORm. TAUDCOS[1:0]	When Overflow Occurs		When an Effective TAUDTTINm Input is Detected	
	TAUDCDRm	TAUDCSRm. TAUDOVF	TAUDCDRm, TAUDCNTm	TAUDCSRm. TAUDOVF
00	Unchanged	0	TAUDCNTm loaded into TAUDCDRm	1
01		1		
10	Set to FFFFH	0	TAUDCNTm stops counting TAUDCDRm unchanged	Unchanged
11		1		

When TAUDCMORm.TAUDCOS[0] = 1, overflow bit TAUDCSRm.TAUDOVF can be cleared only by setting TAUDCSCm.TAUDCLOV to 1.
The combination of the value of TAUDCDRm and TAUDCSRm.TAUDOVF can be used to deduce the width of the TAUDTTINm signal. However, if an overflow occurs multiple times before an effective TAUDTTINm input is detected, overflow bit TAUDCSRm.TAUDOVF cannot indicate the occurrence of multiple overflows.
This function cannot be forcibly restarted.

Remark: When TAUDCMORm.COS[1] = 1, the value of TAUDCNTm is not loaded to TAUDCDRm when the first effective TAUDTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUDTTINm input signal width

= count clock cycle × [(TAUDCSRm.TAUDOVF × (FFFF$_H$ + 1)) + TAUDCDRm capture value + 1]

(3) Block Diagram and General Timing Diagram

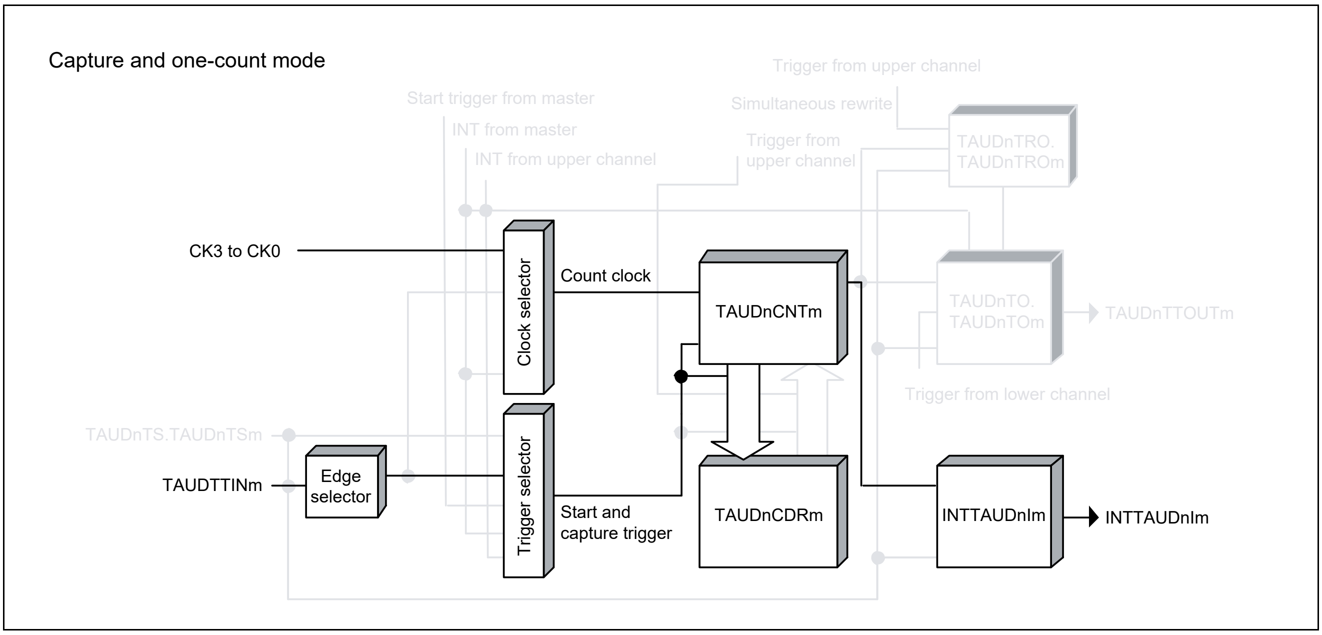


Figure 20.56 Block Diagram of TAUDTTINm Input Signal Width Measurement

- Detection of rising and falling edges = high width measurement (TAUDCMURm.TAUDTIS[1:0] = 11B)
- When an effective TAUDTTINm input is detected after an overflow, TAUDCDRm is changed and TAUDCSRm.TAUDOVF is set to 1. (TAUDCMORm.TAUDCOS[1:0] = 00B)

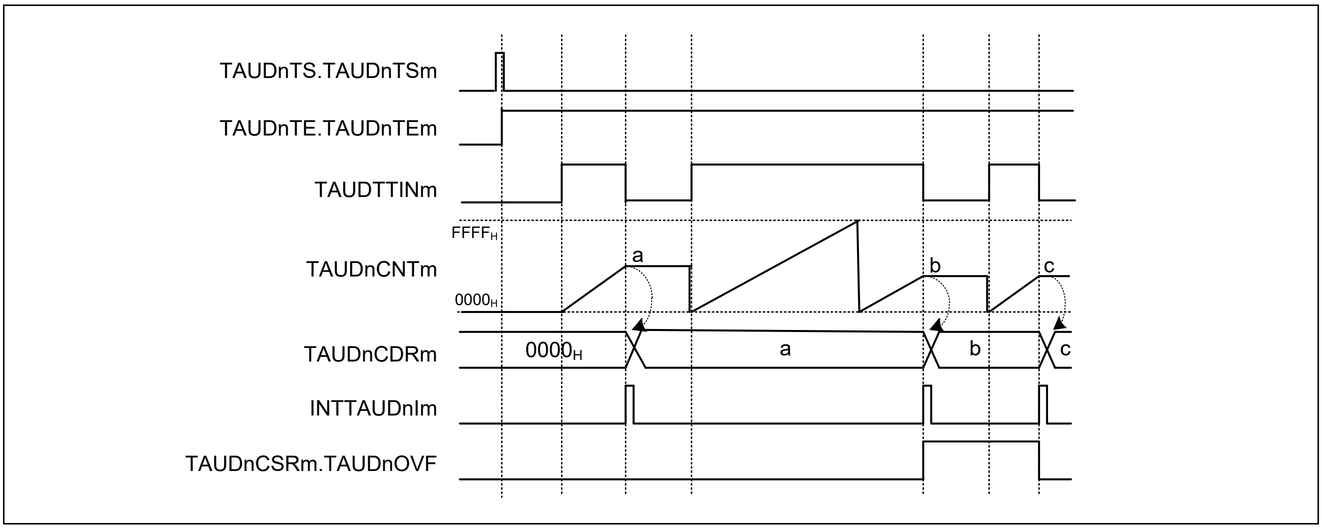


Figure 20.57 General Timing Diagram of TAUDTTINm Input Signal Width Measurement

(4) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.42 Contents of the TAUDCMORm Register for TAUDTTINm Input Signal Width Measurement

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	00: Uses an operation clock as a count clock
10 to 8	TAUDSTS[2:0]	010: Effective edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDCOS[1:0]	See Table 20.41, 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	TAUDMD[4:1]	0110: Capture and one-count mode
0	TAUDMD0	0: Disables the start trigger during operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.43 Contents of the TAUDCMURm Register for TAUDTTINm Input Signal 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	TAUDTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the TAUDTTINm input signal width measurement. Therefore, these registers should be set to 0.

Table 20.44 Simultaneous Reload Settings for TAUDTTINm Input Signal Width Measurement

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Operating Procedure for TAUDTTINm Input Signal Width Measurement

Table 20.45 Operating Procedure for TAUDTTINm Input Signal Width Measurement

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.42, Contents of the TAUDCMORm Register for TAUDTTINm Input Signal Width Measurement, and Table 20.43, Contents of the TAUDCMURm Register for TAUDTTINm Input Signal Width Measurement. The TAUDCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and TAUDCNTm waits for detection of the TAUDTTINm start edge. When a TAUDTTINm start edge is detected, TAUDCNTm starts to count up.
During Operation	TAUDCDRm, TAUDCNTm, and TAUDCSRm registers can be read at any time. TAUDCSCm.TAUDCLOV bit can be set to 1.	TAUDCNTm starts to count up from 0000H. When an effective edge of TAUDTTINm is detected: • TAUDCNTm transfers (captures) its value to TAUDCDRm, and retains its value. INTTAUDIm is then generated. Counting stops at the "value that transferred to TAUDCDRm + 1" and TUACNTm waits for detection of the TAUDTTINm start edge. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and both it and TAUDCSRm.TAUDOVF retain their current values.

Restart (from Stop Operation back to During Operation)

(6) Specific Timing Diagrams: Overflow Operation

(a) TAUDCMORm.TAUDCOS[1:0] = 00B

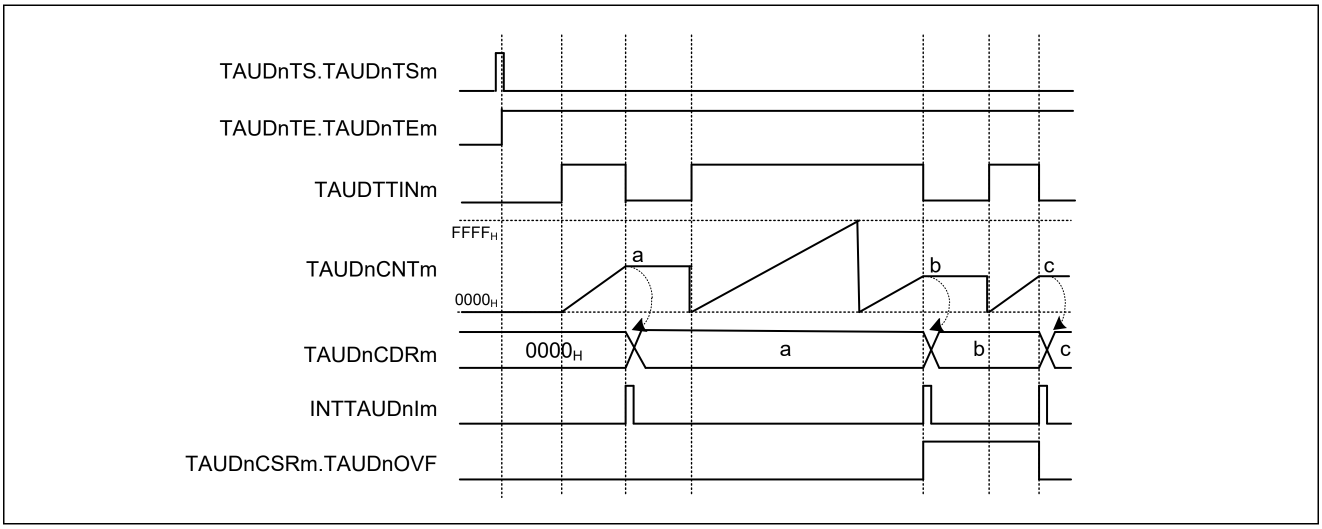


Figure 20.58 TAUDCMORm.TAUDCOS[1:0] = 00B, TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 11B

- When an overflow occurs, the value of TAUDCDRm remains unchanged and TAUDCSRm.TAUDOVF remains 0.
- Upon detection of the next effective TAUDTTINm input edge, the value of TAUDCNTm is loaded into TAUDCDRm and TAUDCSRm.TAUDOVF is set to 1.
- Upon detection of the next effective TAUDTTINm input edge with no overflow occurring, TAUDCSRm.TAUDOVF is cleared to 0.

(b) TAUDCMORm.TAUDCOS[1:0] = 01B

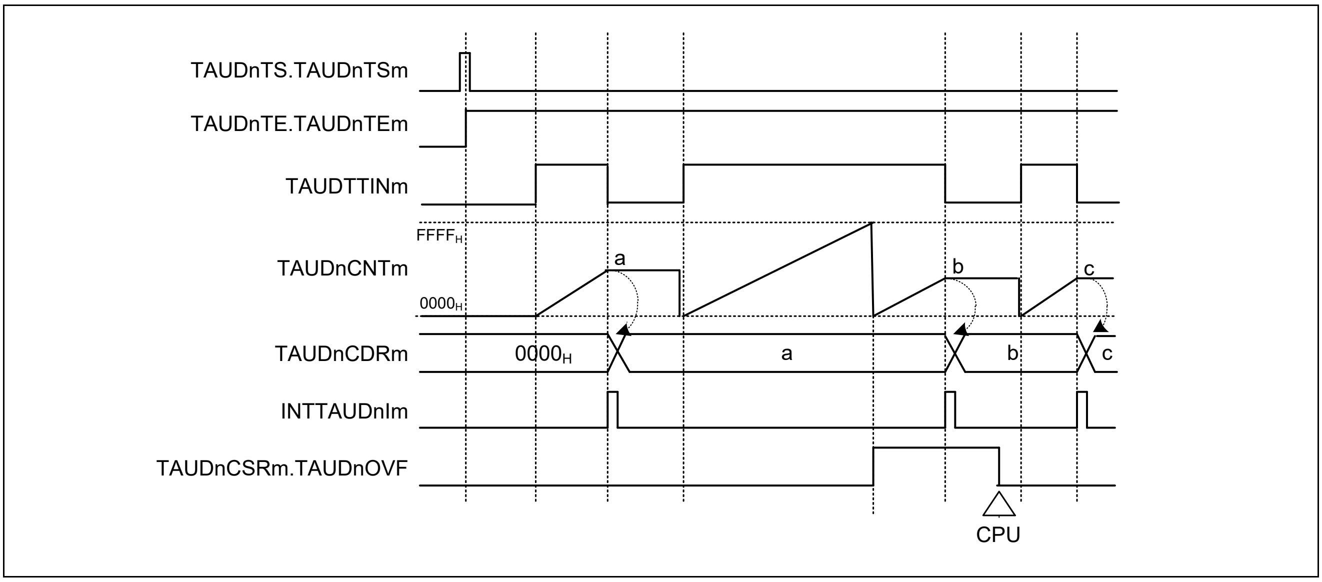


Figure 20.59 TAUDCMORm.TAUDCOS[1:0] = 01B, TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 11B

- When an overflow occurs, the value of TAUDCDRm remains unchanged and TAUDCSRm.TAUDOVF is set to 1.
- Upon detection of the next effective TAUDTTINm input edge, the value of TAUDCNTm is loaded into TAUDCDRm.
- TAUDCSRm.TAUDOVF is only cleared by a CPU command (by setting TAUDCSCm.TAUDCLOV bit to 1).

(c) TAUDCMORm.TAUDCOS[1:0] = 10B

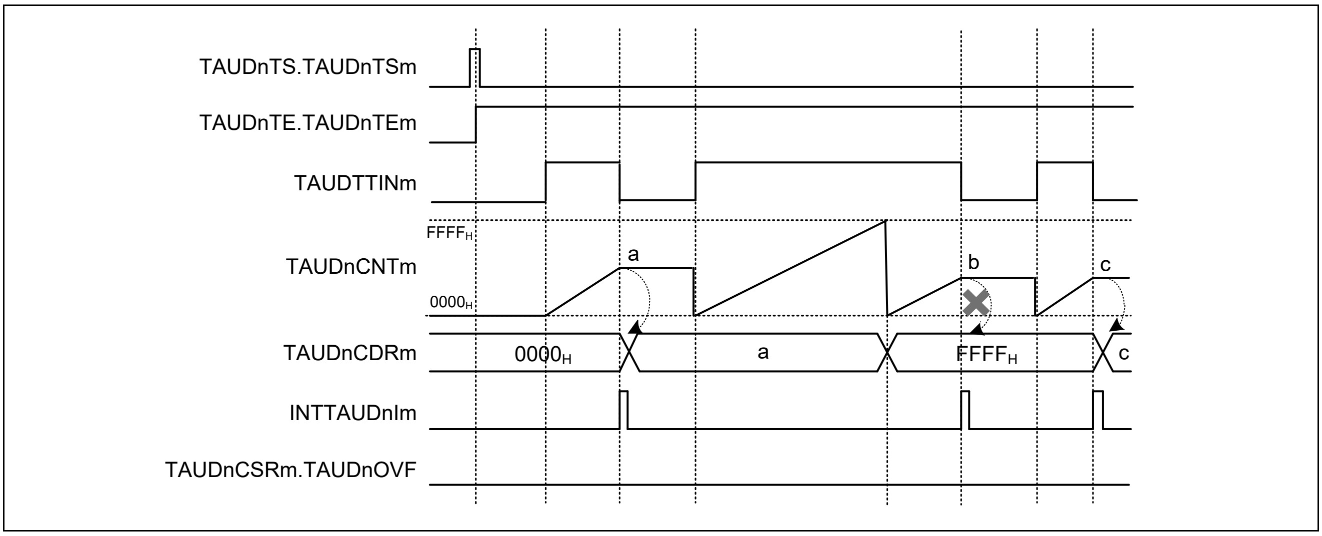


Figure 20.60 TAUDCMORm.TAUDCOS[1:0] = 01B, TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 11B

- When an overflow occurs, the value of TAUDCDRm remains unchanged and TAUDCSRm.TAUDOVF is set to 1.
- Upon detection of the next effective TAUDTTINm input edge, the value of TAUDCNTm is loaded into TAUDCDRm.
- TAUDCSRm.TAUDOVF is only cleared by a CPU command (by setting TAUDCSCm.TAUDCLOV bit to 1).

(d) TAUDCMORm.TAUDCOS[1:0] = 11B

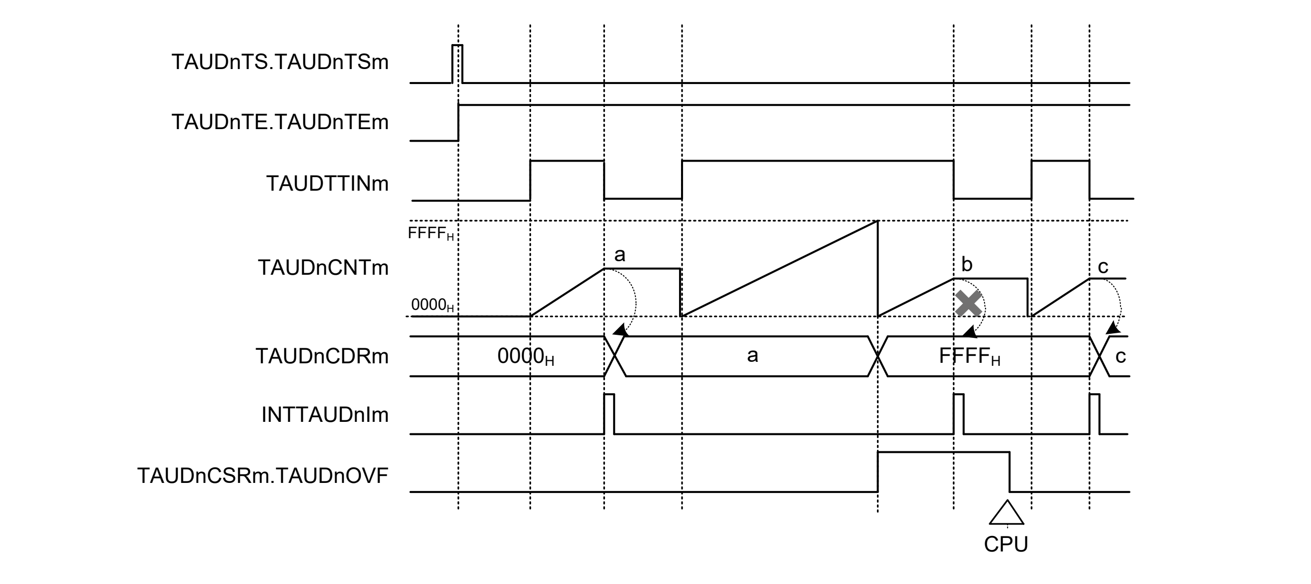


Figure 20.61 TAUDCMORm.TAUDCOS[1:0] = 11B, TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 11B

- When an overflow occurs, TAUDCDRm is set to FFFFH and TAUDCSRm.TAUDOVF is set to 1.
- Upon detection of the next effective TAUDTTINm input edge, TAUDCNTm stops counting, but TAUDCDRm and TAUDCSRm.TAUDOVF remain unchanged.
- Thus, the next effective TAUDTTINm input edge after the overflow is ignored.
- TAUDCSRm.TAUDOVF is cleared by setting TAUDCSCm.TAUDCLOV to 1.

20.12.9 TAUDTTINm Input Position Detection

(1) Overview

(a) Summary

This function measures the input signal duration by capturing the count value at the effective edge of TAUDTTINm.

(b) Prerequisites

- The operating mode should be set to count capture mode. (See Table 20.46, Contents of the TAUDCMORm Register for TAUDTTINm Input Position Detection.)
- TAUDTTOUTm is not used with this function.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation. The counter starts counting from 0000H. When an effective TAUDTTINm input edge is detected, the current value of TAUDCNTm is loaded into TAUDCDRm and an interrupt (INTTAUDIm) is generated. The count operation continues.
When the counter reaches FFFFH, the counter restarts from 0000H.

(d) Conditions

If the TAUDCMORm.TAUDMD0 bit is set to 0, the first interrupt does not occur at the beginning of operation or after restart. For details, see Section 20.9, TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts.

(2) Equations

Functional duration at a TAUDTTINm input pulse = count clock cycle × (TAUDCDRm capture value + 1)

(3) Block Diagram and General Timing Diagram

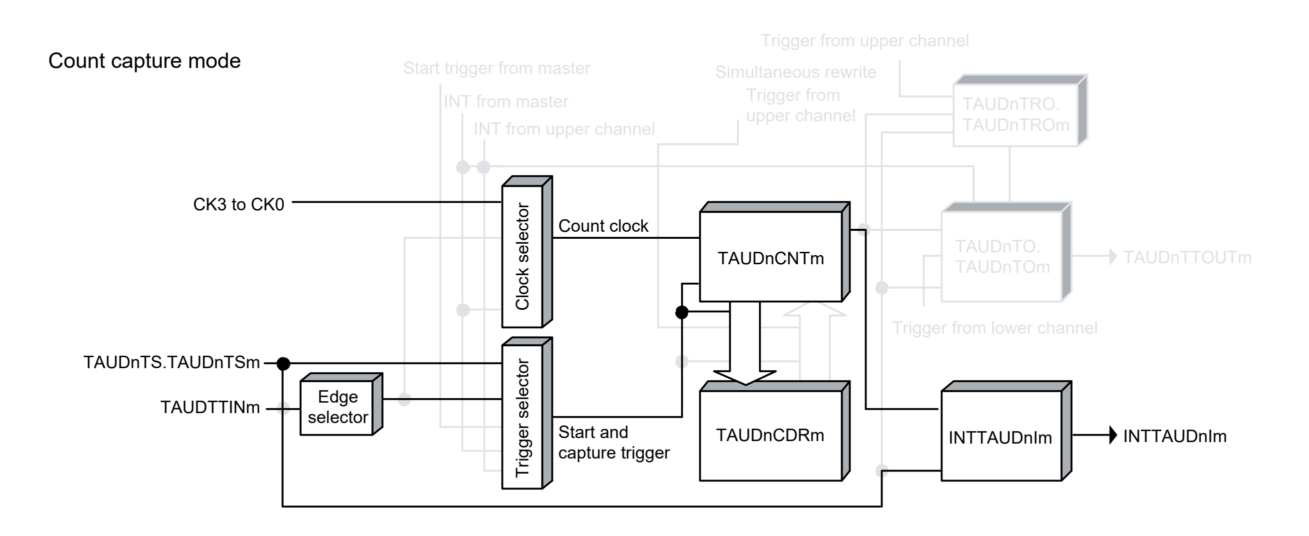


Figure 20.62 Block Diagram of TAUDTTINm Input Position Detection

The following settings apply to the general timing diagram.

- INTTAUDIm is not generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 0)
- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

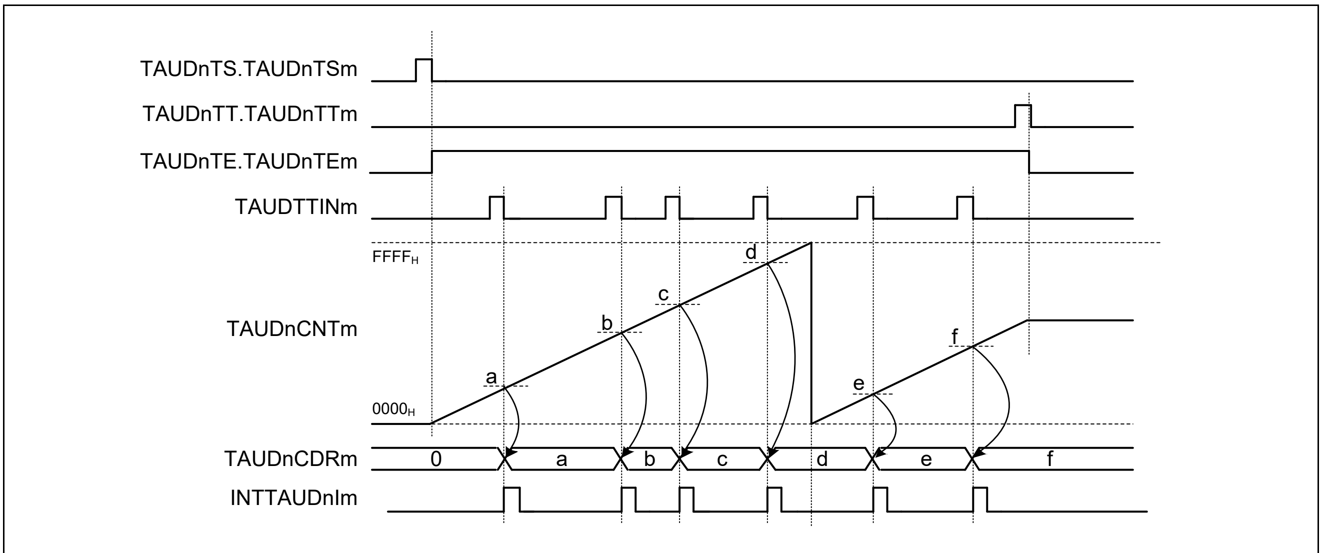


Figure 20.63 General Timing Diagram of TAUDTTINm Input Position Detection

(4) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.46 Contents of the TAUDCMORm Register for TAUDTTINm Input Position Detection

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as an external capture trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	1011: Count capture mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation. 1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.47 Contents of the TAUDCMURm Register for TAUDTTINm Input Position 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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges

(c) Channel output mode

The channel output mode is not used by this function.

(d) Simultaneous reloading

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the TAUDTTINm input position detection. Therefore, these registers should be set to 0.

Table 20.48 Simultaneous Reload Settings for TAUDTTINm Input Position Detection

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Operating Procedure for TAUDTTINm Input Position Detection

Table 20.49 Operating Procedure for TAUDTTINm Input Position Detection

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.46, Contents of the TAUDCMORm Register for TAUDTTINm Input Position Detection, and Table 20.47, Contents of the TAUDCMURm Register for TAUDTTINm Input Position Detection. The TAUDCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. If TAUDCMORm.TAUDMD0 is 1, INTTAUDIm occurs.
During Operation	The values of TAUDCMURm.TIS[1:0] bits can be changed at any time. The TAUDCDRm and TAUDCSRm registers can be read at any time.	TAUDCNTm starts to count up from 0000H. When an effective TAUDTTINm edge is detected: • TAUDCNTm transfers (captures) its own value to TAUDCDRm. • INTTAUDIm occurs. • The counter is not cleared to 0000H and TAUDCNTm continues counting. Afterwards, this procedure is repeated. When TAUDCNTm reaches FFFFH, the counter restarts from 0000H.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

Restart

(6) Specific Timing Diagrams

(a) Operation stop and restart

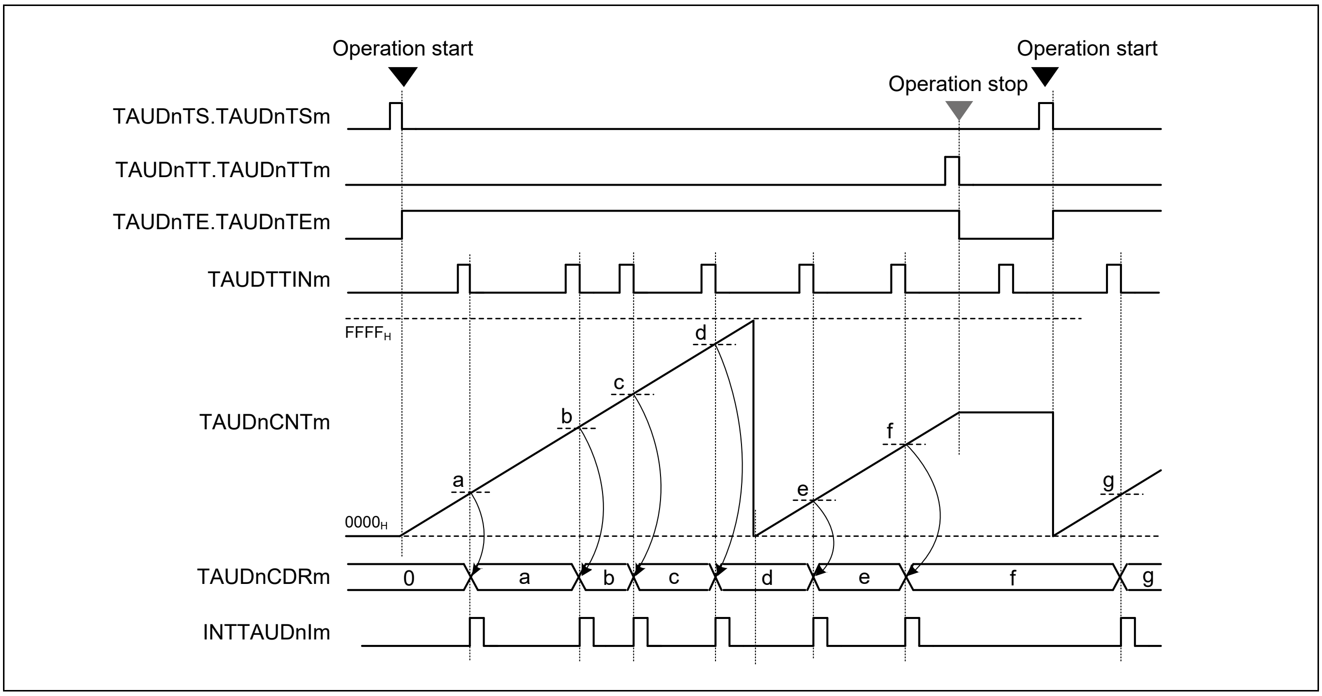


Figure 20.64 Operation Stop and Restart (TAUDCMORm.TAUDMD0 = 0, TAUDCMURm.TAUDTIS[1:0] = 00B)

- The counter can stop operating by setting TAUDTT.TAUDTTm to 1. This sets TAUDTE.TAUDTEm to 0.
- TAUDCNTm stops and retains its current value.
- If the counter stops operating, effective TAUDTTINm input edges are ignored.
- The counter can be restarted by setting TAUDTS.TAUDTSm to 1. TAUDCNTm restarts to count from 0000H.

20.12.10 TAUDTTINm Input Period Count Detection

(1) Overview

(a) Summary

This function measures the cumulative width of a TAUDTTINm input signal.

(b) Prerequisites

- The operating mode should be set to capture and gate count mode. (See Table 20.50, Contents of the TAUDCMORm Register for TAUDTTINm Input Period Count Detection.)
- TAUDTTOUTm is not used with this function.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation. The counter awaits an effective TAUDTTINm input edge.
When an effective TAUDTTINm input stop edge is detected, the current TAUDCNTm value is loaded into TAUDCDRm and an interrupt (INTTAUDIm) is generated. The counter stops and retains its value (TAUDCDRm + 1) until the next effective TAUDTTINm input start edge is detected.
When the next effective TAUDTTINm input start edge is detected, the counter restarts to count from the value retained when stopped.
If the counter reaches FFFFH, the counter restarts from 0000H.

Remarks 1. TAUDTTINm input signal is sampled at the frequency of an operation clock set by the TAUDCMORm.TAUDCKS[1:0] bits.
2. As this function is to measure the TAUDTTINm input signal width, setting TAUDTS.TAUDTSm to 1 is disabled while TAUDTE.TAUDTEm = 1.

(d) Conditions

The effective start and stop edges are specified by the TAUDCMURm.TAUDTIS[1:0] bits.

- If TAUDCMURm.TAUDTIS[1:0] = 10B, the TAUDTTINm input low period is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDCMURm.TAUDTIS[1:0] = 11B, the TAUDTTINm input high period is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

(e) Equations

Cumulative TAUDTTINm input width = count clock cycle × (TAUDCDRm capture value + 1)

(2) Block Diagram and General Timing Diagram

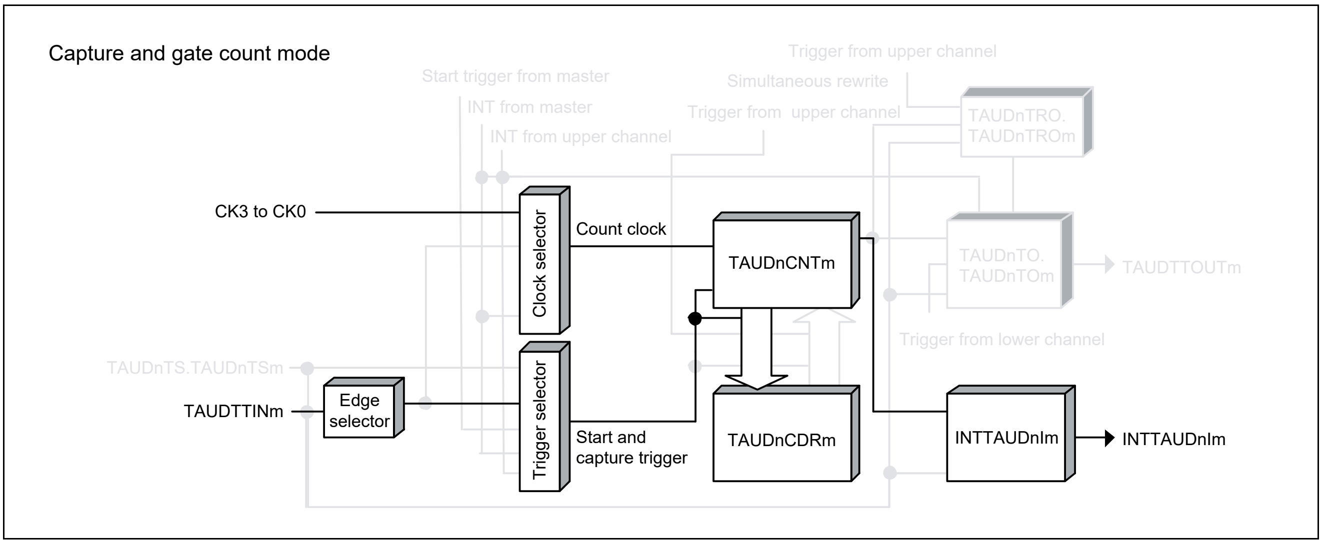


Figure 20.65 Block Diagram of TAUDTTINm Input Period Count Detection

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDCMURm.TAUDTIS[1:0] = 11B)

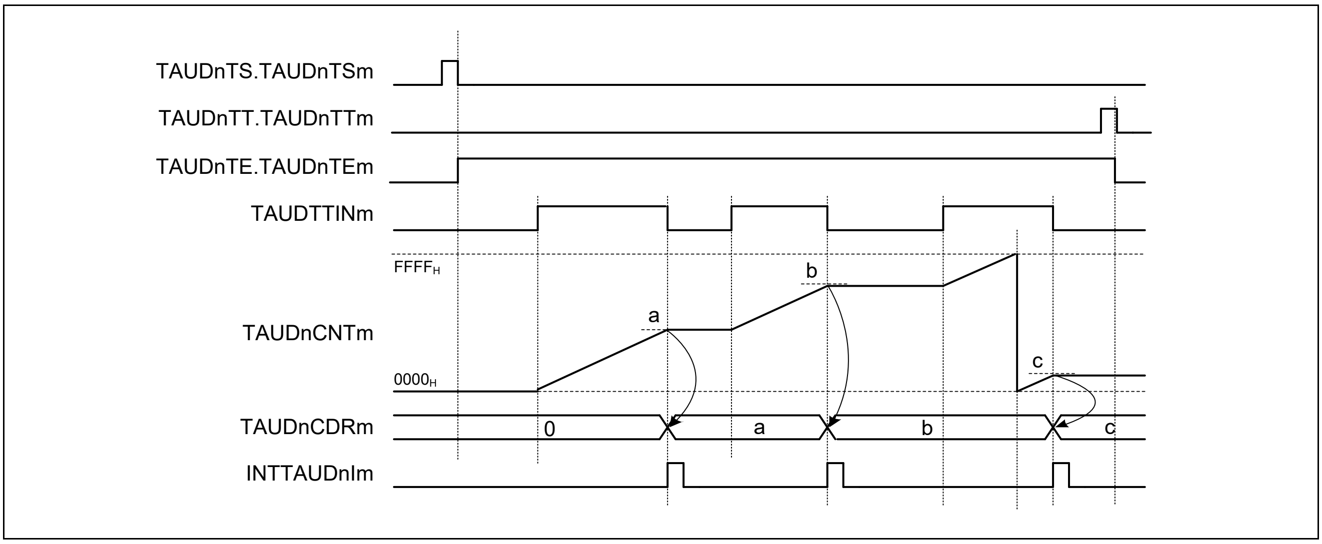


Figure 20.66 General Timing Diagram of TAUDTTINm Input Period Count Detection

(3) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.50 Contents of the TAUDCMORm Register for TAUDTTINm Input Period Count Detection

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	010: Effective edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	1101: Capture and gate count mode
0	TAUDMD0	0: Disables the start trigger during operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.51 Contents of the TAUDCMURm Register for 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	TAUDTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous reloading

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the TAUDTTINm input period count detection. Therefore, these registers should be set to 0.

Table 20.52 Simultaneous Reload Settings for TAUDTTINm Input Period Count Detection

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(4) Operating Procedure for TAUDTTINm Input Period Count Detection

Table 20.53 Operating Procedure for TAUDTTINm Input Period Count Detection

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.50, Contents of the TAUDCMORm Register for TAUDTTINm Input Period Count Detection, and Table 20.51, Contents of the TAUDCMURm Register for TAUDTTINm Input Period Count Detection. The TAUDCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and TAUDCNTm waits for detection of the TAUDTTINm start edge.
During Operation	Detection of TAUDTTINm edge The TAUDCDRm, TAUDCNTm, and TAUDCSRm 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, TAUDCNTm starts counting up from the stop value. When TAUDCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUDCDRm and INTTAUDIm is generated. Counting stops at the "value transferred to TAUDCDRm + 1" and TAUDCNTm waits for detection of the TAUDTTINm start edge. When TAUDCNTm reaches FFFFH, the counter restarts from 0000H. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

(Restart: from Stop Operation back to During Operation)

(5) Specific Timing Diagrams

(a) Operation stop and restart

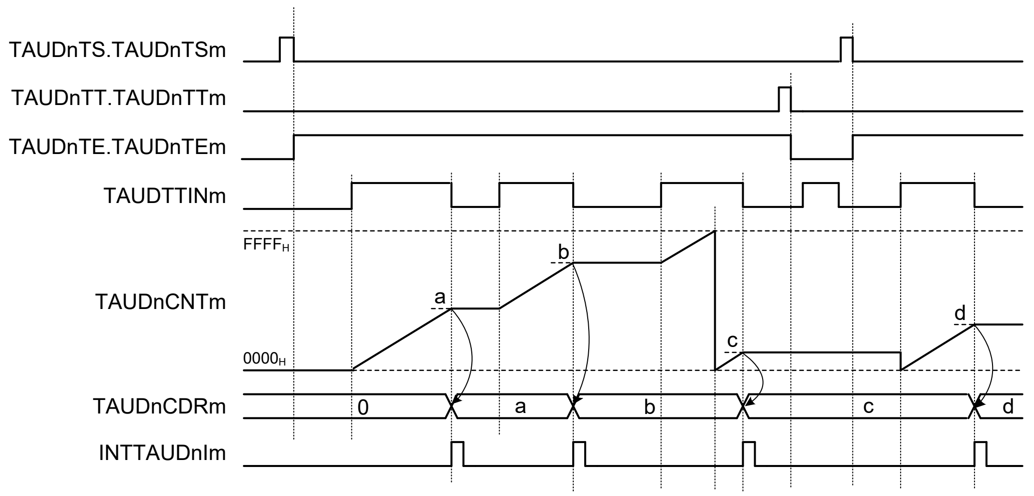


Figure 20.67 Operation Stop and Restart (TAUDCMURm.TAUDTIS[1:0] = 11B)

- The counter can be stopped by setting TAUDTT.TAUDTTm to 1. This sets TAUDTE.TAUDTEm to 0.
- TAUDCNTm stops and retains its current value.
- If the counter is stopped, effective TAUDTTINm input edges are ignored.
- The counter can be restarted by setting TAUDTS.TAUDTSm to 1. TAUDCNTm restarts to count from 0000H.

20.12.11 TAUDTTINm Input Pulse Interval Judgment

(1) Overview

(a) Summary

This function outputs the result of a comparison between the count value (TAUDCNTm) and the value in the channel data register (TAUDCDRm) when a TAUDTTINm input pulse occurs. An interrupt request signal INTTAUDIm is generated if the result of the comparison is true.

(b) Prerequisites

- The operating mode should be set to judge mode. See Table 20.54, Contents of the TAUDCMORm Register for TAUDTTINm Input Pulse Interval Judgment.
- TAUDTTOUTm is not used with this function.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation. The current value of TAUDCDRm is loaded into TAUDCNTm and the counter starts to count down from this value.
When an effective edge of TAUDTTINm is detected or TAUDTS.TAUDTSm is set to 1, the function compares the current values of TAUDCNTm and TAUDCDRm. An interrupt request signal INTTAUDIm is generated if the result of the comparison is true. TAUDCNTm reloads the value of TAUDCDRm and subsequently continues operation, regardless of the result of the comparison.
If the counter reaches 0000H before an effective edge of TAUDTTINm is detected, TAUDCNTm overflows and is set to FFFFH. It then continues to count down.
The value of TAUDCDRm can be rewritten at any time, and the changed value of TAUDCDRm is applied the next time the counter starts to count down.

(d) Conditions

The TAUDCMORm.TAUDMD0 bit specifies the type of comparison:

- If TAUDCMORm.TAUDMD0 = 0, INTTAUDIm is generated when TAUDCNTm ≤ TAUDCDRm.
- If TAUDCMORm.TAUDMD0 = 1, INTTAUDIm is generated when TAUDCNTm > TAUDCDRm.

(2) Block Diagram and General Timing Diagram

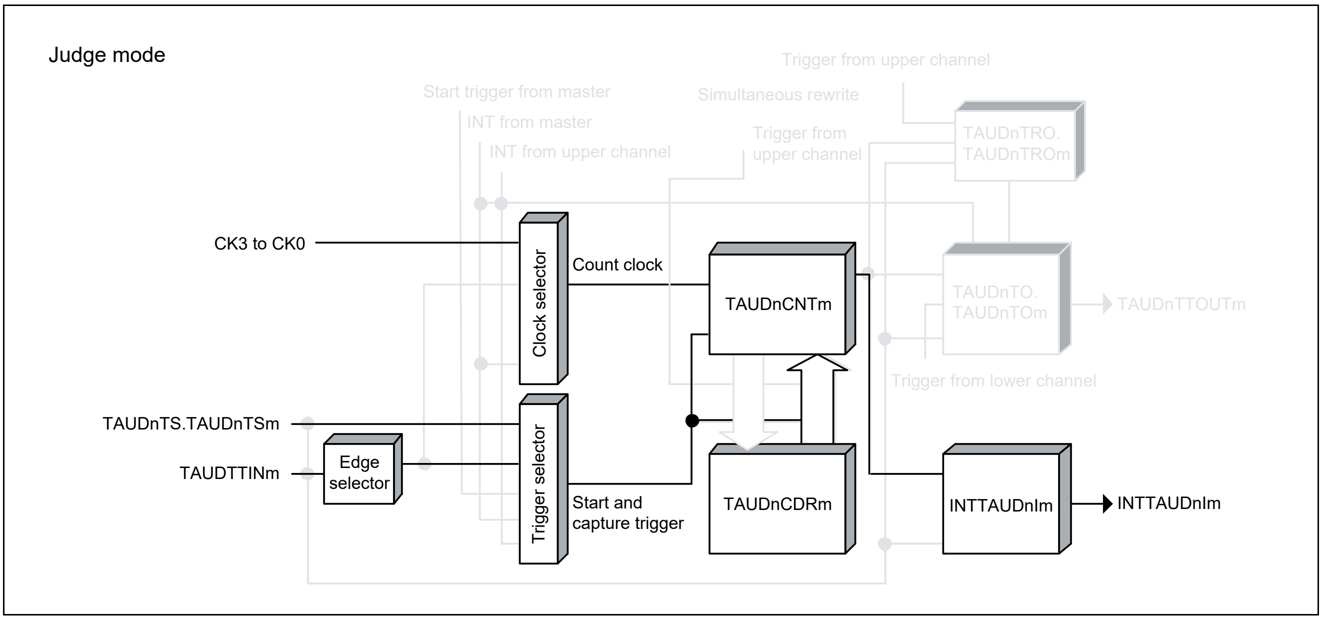


Figure 20.68 Block Diagram of TAUDTTINm Input Pulse Interval Judgment

The following settings apply to the general timing diagram.

- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

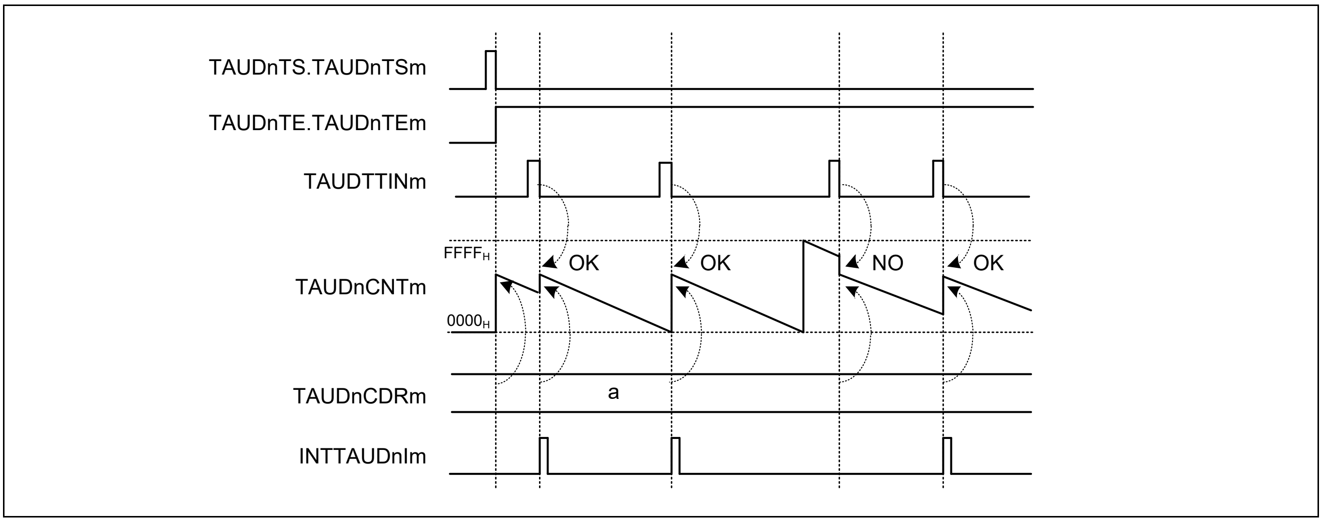


Figure 20.69 General Timing Diagram of TAUDTTINm Input Pulse Interval Judgment

(3) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.54 Contents of the TAUDCMORm Register for TAUDTTINm Input Pulse Interval Judgment

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	Effective edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0001: Judge mode
0	TAUDMD0	0: INTTAUDIm is generated when TAUDCNTm ≤ TAUDCDRm 1: INTTAUDIm is generated when TAUDCNTm > TAUDCDRm

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.55 Contents of the TAUDCMURm Register for TAUDTTINm Input Pulse Interval Judgment

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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous reloading

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the TAUDTTINm input pulse interval Judgment. Therefore, these registers should be set to 0.

Table 20.56 Simultaneous Reload Settings for TAUDTTINm Input Pulse Interval Judgment

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(4) Operating Procedure for TAUDTTINm Input Pulse Interval Judgment

Table 20.57 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.54, Contents of the TAUDCMORm Register for TAUDTTINm Input Pulse Interval Judgment, and Table 20.55, Contents of the TAUDCMURm Register for TAUDTTINm Input Pulse Interval Judgment. Set the value of TAUDCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. TAUDCDRm value is loaded into TAUDCNTm.
During Operation	The following register can be changed at any time: ☐ TAUDCDRm register	When TAUDCMORm.TAUDMD0 = 0 If TAUDCNTm ≤ TAUDCDRm when a TAUDTTINm input edge is detected, INTTAUDIm is generated. When TAUDCMORm.TAUDMD0 = 1 If TAUDCNTm > TAUDCDRm when a TAUDTTINm input edge is detected, INTTAUDIm is generated. If a TAUDTTINm input edge is detected, then TAUDCNTm starts to count down from the value of TAUDCDRm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

Restart

20.12.12 TAUDTTINm Input Signal Width Judgment

(1) Overview

(a) Summary

This function compares the count value (TAUDCNTm) for the high or low level width of a TAUDTTINm input signal and the TAUDCDRm value, and outputs the judgment result from the interrupt request signal INTTAUDIm.

(b) Prerequisites

- The operating mode should be set to judge and one-count mode. (See Table 20.58, Contents of the TAUDCMORm Register for TAUDTTINm Input Signal Width Judgment.)
- TAUDTTOUTm is not used with this function.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation. When an effective TAUDTTINm input start edge is detected, the current value of TAUDCDRm is loaded into TAUDCNTm and the counter starts to count down from this value.
When an effective TAUDTTINm stop edge is detected, the function compares the current values of TAUDCNTm and TAUDCDRm. An interrupt request signal INTTAUDIm is generated if the result of the comparison is true. The counter TAUDCNTm retains its value until the next effective TAUDTTINm start edge is detected, regardless of the result of the comparison.
If the counter reaches 0000H before an effective TAUDTTINm stop edge is detected, TAUDCNTm overflows and is set to FFFFH. The counter then continues to count down.
The value of TAUDCDRm can be rewritten at any time, and the changed value of TAUDCDRm is applied the next time the counter starts to count down.

(d) Conditions

- The TAUDCMORm.TAUDMD0 bit specifies the type of comparison:
  - If TAUDCMORm.TAUDMD0 = 0, INTTAUDIm is generated when TAUDCNTm ≤ TAUDCDRm.
  - If TAUDCMORm.TAUDMD0 = 1, INTTAUDIm is generated when TAUDCNTm > TAUDCDRm.
- The TAUDCMURm.TAUDTIS[1:0] bits specify a type of width measurement:
  - For high width measurement (TAUDCMURm.TAUDTIS[1:0] = 11B), TAUDTTINm rising edge is used as a start edge and TAUDTTINm falling edge as a stop edge.
  - For low width measurement (TAUDCMURm.TAUDTIS[1:0] = 10B), TAUDTTINm falling edge is used as a start edge and TAUDTTINm rising edge as a stop edge.
- The counter cannot be forcibly restarted with this function.

(2) Block Diagram and General Timing Diagram

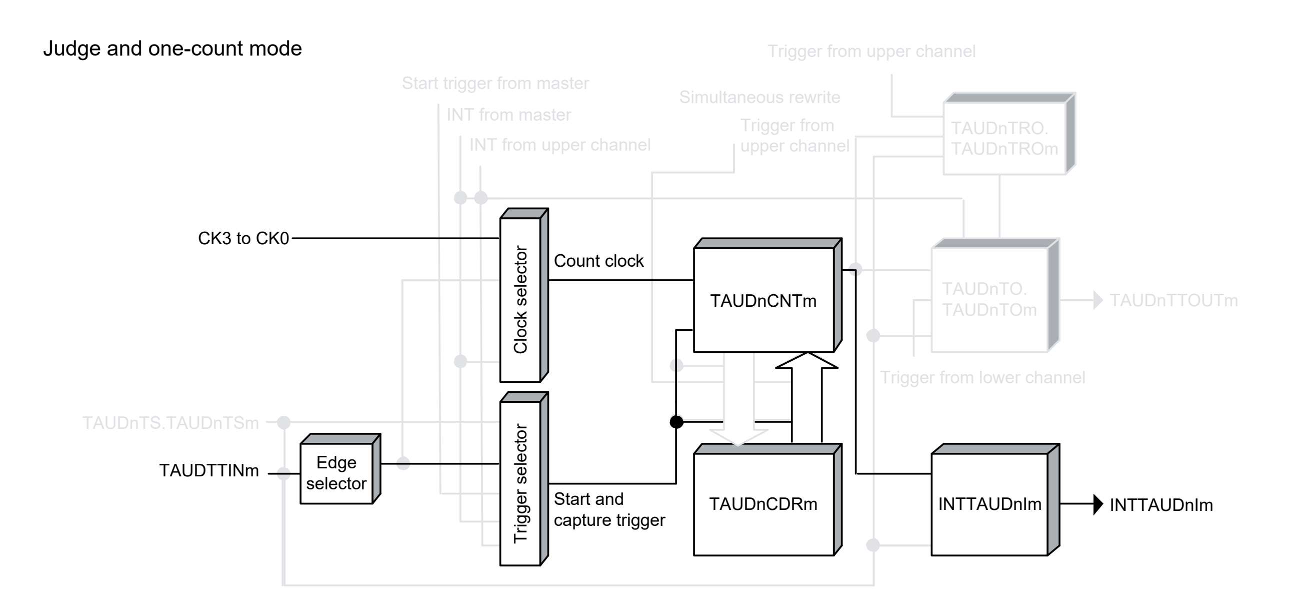


Figure 20.70 Block Diagram of TAUDTTINm Input Signal Width Judgment

The following settings apply to the general timing diagram.

- INTTAUDIm is generated when TAUDCNTm ≤ TAUDCDRm (TAUDCMORm.TAUDMD0 = 0).
  Effective TAUDTTINm start edge = rising edge, effective TAUDTTINm stop edge = falling edge (TAUDCMURm.TAUDTIS[1:0] = 11B)

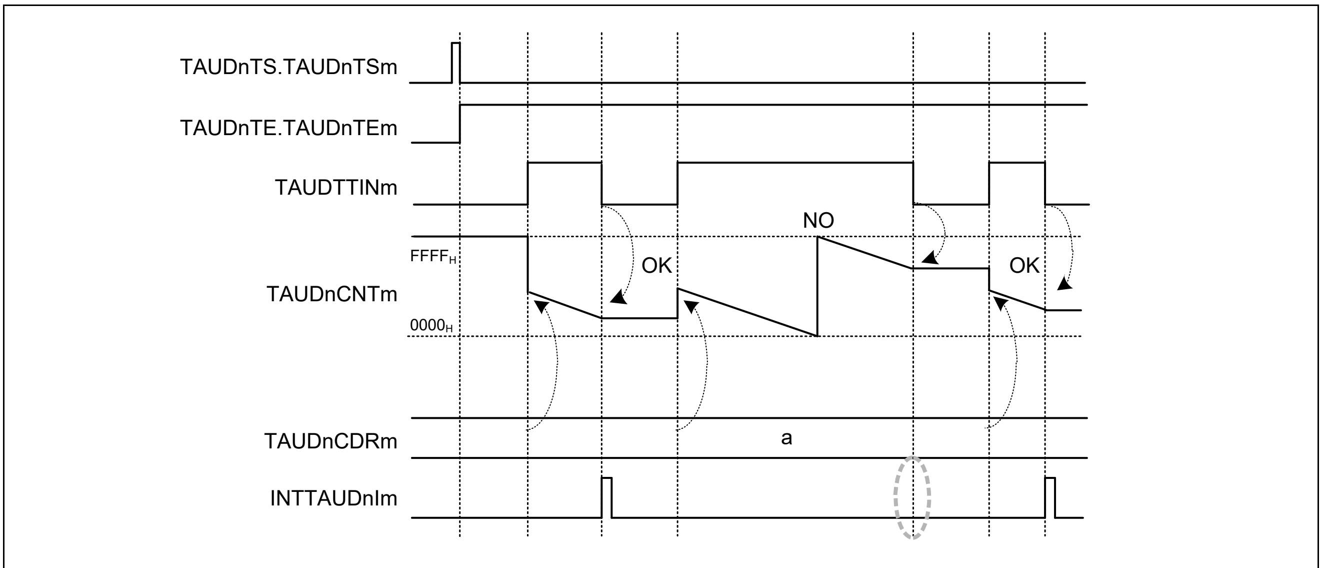


Figure 20.71 General Timing Diagram of TAUDTTINm Input Signal Width Judgment

(3) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.58 Contents of the TAUDCMORm Register for TAUDTTINm Input Signal Width Judgment

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	010: Effective edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0111: Judge and one-count mode
0	TAUDMD0	0: INTTAUDIm is generated when TAUDCNTm ≤ TAUDCDRm 1: INTTAUDIm is generated when TAUDCNTm > TAUDCDRm

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.59 Contents of the TAUDCMURm Register for TAUDTTINm Input Signal Width Judgment

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	TAUDTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous reloading

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the TAUDTTINm input signal width judgment. Therefore, these registers should be set to 0.

Table 20.60 Simultaneous Reload Settings for TAUDTTINm Input Signal Width Judgment

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(4) Operating Procedure for TAUDTTINm Input Signal Width Judgment

Table 20.61 Operating Procedure for TAUDTTINm Input Signal Width Judgment

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.58, Contents of the TAUDCMORm Register for TAUDTTINm Input Signal Width Judgment, and Table 20.59, Contents of the TAUDCMURm Register for TAUDTTINm Input Signal Width Judgment. Set the value of TAUDCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and TAUDCNTm waits for detection of the TAUDTTINm start edge.
During Operation (Restart)	The following register can be changed at any time: ☐ TAUDCDRm register	Upon detection of a TAUDTTINm start edge, TAUDCNTm starts count down from the value of TAUDCDRm. When TAUDCMORm.TAUDMD0 = 0 If TAUDCNTm ≤ TAUDCDRm when a TAUDTTINm input stop edge is detected, INTTAUDIm is generated. When TAUDCMORm.TAUDMD0 = 1 If TAUDCNTm > TAUDCDRm when a TAUDTTINm input stop edge is detected, INTTAUDIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

20.12.13 Overflow Interrupt Output (during TAUDTTINm Width Measurement)

(1) Overview

(a) Summary

This function measures the width of an individual TAUDTTINm input signal. An interrupt is generated if the TAUDTTINm input width is longer than FFFFH + 1.

(b) Prerequisites

- The operation mode must be set to One-Count Mode (see Table 20.62, Contents of the TAUDCMORm Register for Overflow Interrupt Output (during TAUDTTINm Width Measurement)).
- TAUDTTOUTm is not used for this function.
- The value of TAUDCDRm must be set to FFFFH.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation.
The counter starts when an effective TAUDTTINm input start edge is detected. FFFFH is written to TAUDCNTm and the counter starts to count down.
When an effective stop edge is detected, the counter stops and retains the current value.
When the next TAUDTTINm input start edge is detected, TAUDCNTm reloads FFFFH and starts to count down.
If the counter reaches 0000H before a stop edge is detected, an interrupt is generated.

(d) Conditions

The effective start and stop edges are specified by the TAUDCMURm.TAUDTIS[1:0] bits.

- If TAUDCMURm.TAUDTIS[1:0] = 10B, the TAUDTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDCMURm.TAUDTIS[1:0] = 11B, the TAUDTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

Remark: The counter cannot be restarted during operation.

(2) Block Diagram and General Timing Diagram

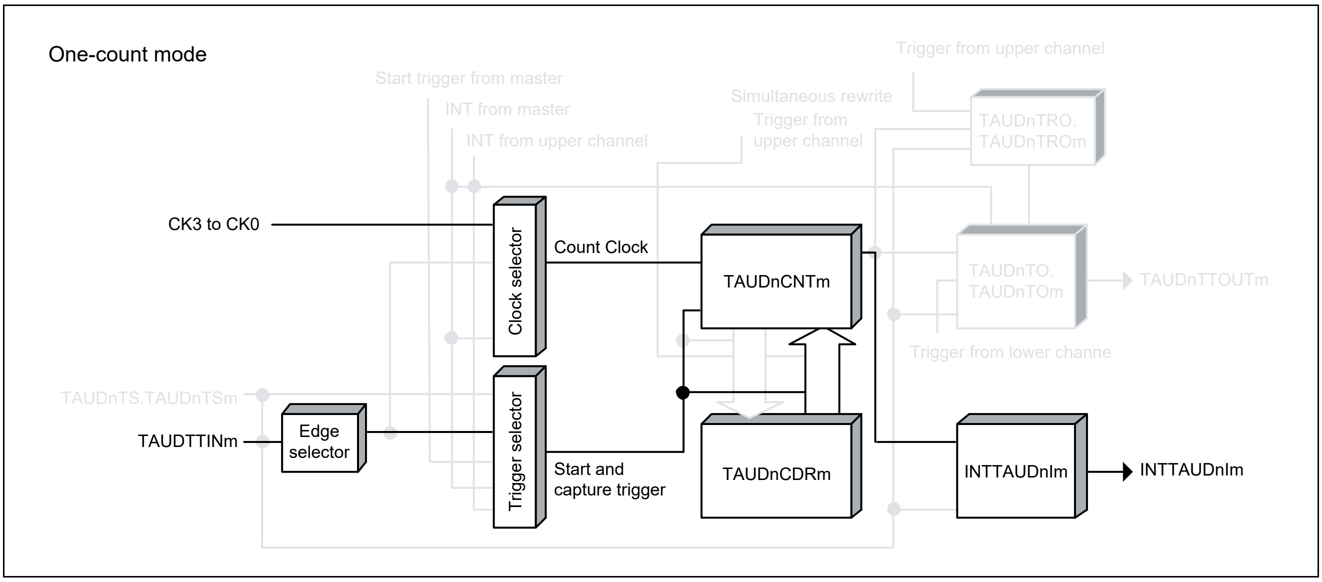


Figure 20.72 Block Diagram for Overflow Interrupt Output (during TAUDTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDCMURm.TAUDTIS[1:0] = 11B)

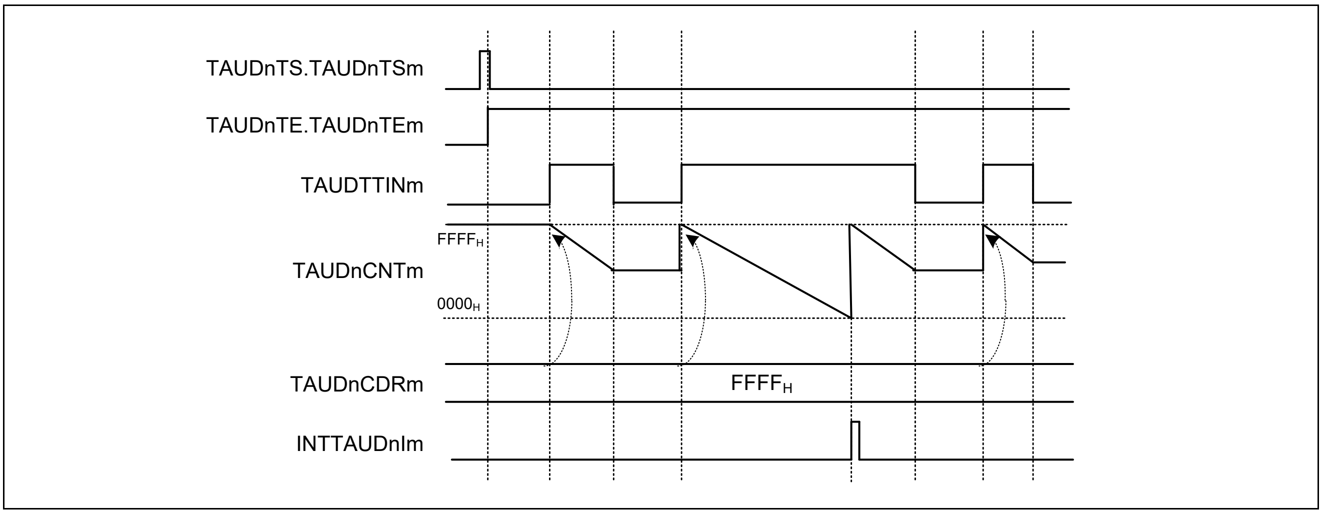


Figure 20.73 General Timing Diagram for Overflow Interrupt Output (during TAUDTTINm Width Measurement)

(3) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.62 Contents of the TAUDCMORm Register for Overflow Interrupt Output (during TAUDTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	010: Effective edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	0: Disables the start trigger during operation

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.63 Contents of the TAUDCMURm Register for Overflow Interrupt Output (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	TAUDTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used by this function.

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the Overflow Interrupt Output (during TAUDTTINm Width Measurement). Therefore, these registers must be set to 0.

Table 20.64 Simultaneous Reload Settings for Overflow Interrupt Output (during TAUDTTINm Width Measurement)

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(4) Operating Procedure for Overflow Interrupt Output (during TAUDTTINm Width Measurement)

Table 20.65 Operating Procedure for Overflow Interrupt Output (during TAUDTTINm Width Measurement)

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.62, Contents of the TAUDCMORm Register for Overflow Interrupt Output (during TAUDTTINm Width Measurement), and Table 20.63, Contents of the TAUDCMURm Register for Overflow Interrupt Output (during TAUDTTINm Width Measurement). Set the value of TAUDCDRm register to FFFFH.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDTE.TAUDTEm is set to 1 and TAUDCNTm waits for detection of the start edge. When a start edge is detected, TAUDCNTm loads the TAUDCDRm value (FFFFH).
During Operation	The TAUDCNTm register can be read at any time.	TAUDCNTm counts down. When the counter reaches 0000H: • INTTAUDIm is generated. When TAUDTTINm input stop edge is detected during count operation: • TAUDCNTm stops and retains its current value. When TAUDTTINm input start edge is detected during count operation: • TAUDCNTm loads the TAUDCDRm value (FFFFH) again, and continues to count down. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

Restart

20.12.14 Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)

(1) Overview

(a) Summary

This function measures the cumulative width of a TAUDTTINm input signal. If the cumulative TAUDTTINm input width is longer than FFFFH, an interrupt is generated and an overflow interrupt can be output.

(b) Prerequisites

- The operation mode must be set to Gate Count Mode, (see Table 20.66, Contents of the TAUDCMORm Register for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)).
- TAUDTTOUTm is not used with this function.
- The value of TAUDCDRm must be set to FFFFH.

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm = 1, enabling count operation.
The counter starts when an effective TAUDTTINm input start edge is detected. FFFFH is written to TAUDCNTm and the counter starts to count down.
When an effective 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 0000H, an interrupt is generated. FFFFH is written to TAUDCNTm and the counter continues to count down until a TAUDTTINm input stop edge is detected.

(d) Conditions

The effective start and stop edges are specified by the TAUDCMURm.TIS[1:0] bits.

- If TAUDCMURm.TAUDTIS[1:0] = 10B, the TAUDTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDCMURm.TAUDTIS[1:0] = 11B, the TAUDTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

Remark: The counter cannot be restarted during operation.

(2) Block Diagram and General Timing Diagram

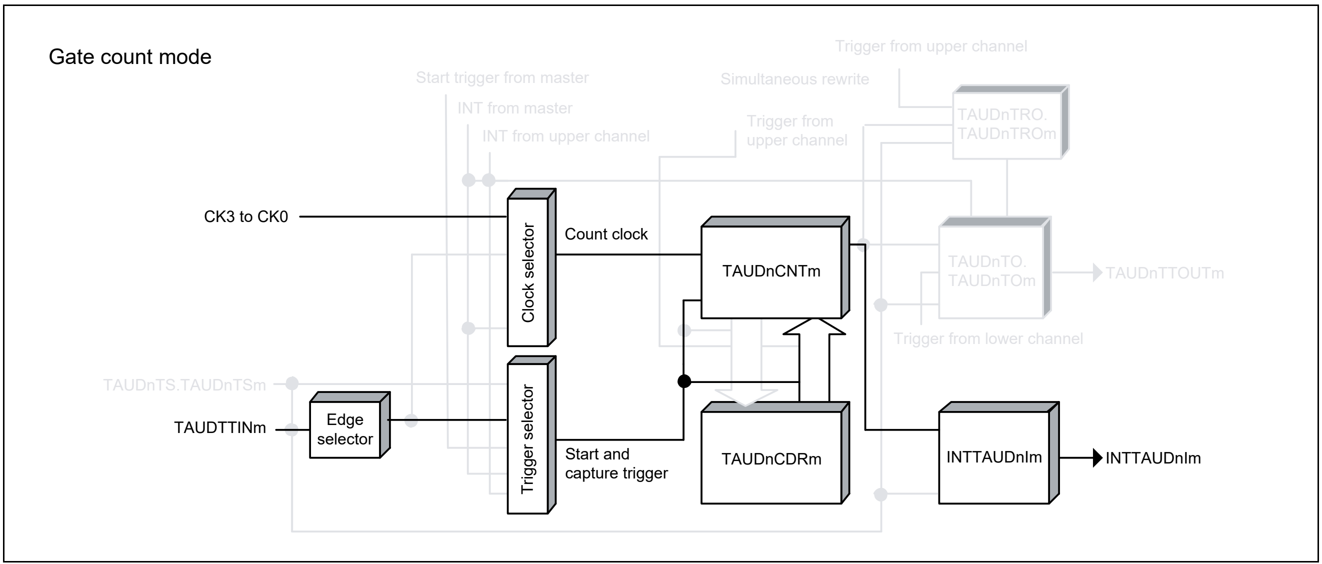


Figure 20.74 Block Diagram for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDCMURm.TAUDTIS[1:0] = 11B)

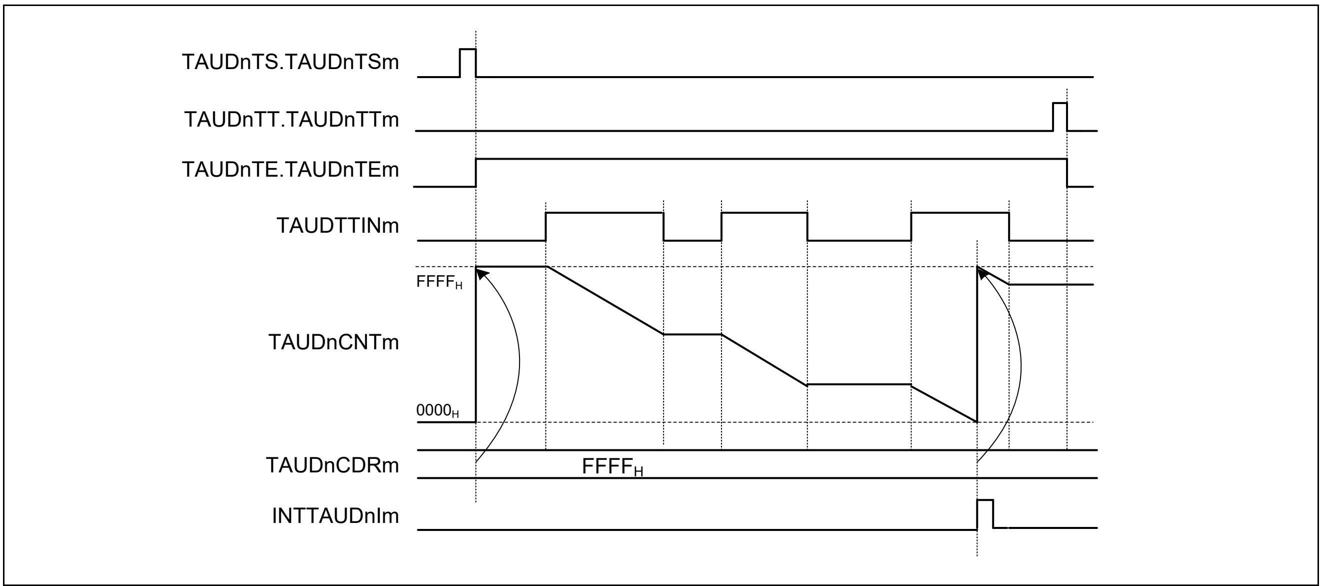


Figure 20.75 General Timing Diagram for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)

(3) Register Settings

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.66 Contents of the TAUDCMORm Register for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	010: Effective edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	1100: Gate count mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.67 Contents of the TAUDCMURm Register for Overflow Interrupt Output (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	TAUDTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 20.68 Simultaneous Reload Settings for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(4) Operating Procedure for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)

Table 20.69 Operating Procedure for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection)

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers as described in Table 20.66, Contents of the TAUDCMORm Register for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection), and Table 20.67, Contents of the TAUDCMURm Register for Overflow Interrupt Output (during TAUDTTINm Input Period Count Detection). Set the value of TAUDCDRm register to FFFFH.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDTE.TAUDTEm is set to 1 and TAUDCNTm waits for detection of the start edge. When a start edge is detected, TAUDCNTm loads the TAUDCDRm value (FFFFH).
During Operation (Restart)	The TAUDCNTm register can be read at all times.	TAUDCNTm counts down. When the counter reaches 0000H: • INTTAUDIm is generated. • TAUDCDRm loads the TAUDCDRm value (FFFFH) and continues to count down. When TAUDTTINm input stop edge is detected during count operation: • TAUDCNTm stops and retains the stop value. When TAUDTTINm input start edge is detected during count operation: • TAUDCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

20.12.15 One-Phase PWM Output

(1) Overview

(a) Summary

This function adds dead time to a TAUDTTINm input signal. The resulting PWM signal is output via TAUDTTOUTm of the channel and TAUDTTOUTm of upper channels.

(b) Prerequisites

- Each of two (or more) channels is enabled for dead time control (TAUDTDE.TAUDTDEm = 1).
- The operating mode for the lower channel should be set to one-count mode. (See Table 20.71, Contents of the TAUDCMORm Register for the Lower Channel of the One-Phase PWM Output.)
- 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 20.7, Channel Output Modes.)

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation.
The counter starts when an effective TAUDTTINm input start edge is detected. The value of TAUDCDRm is loaded into TAUDCNTm and the counter starts to count down from the TAUDCDRm value.
When the counter reaches 0000H, an interrupt occurs. The counter is reset to FFFFH and waits for the next effective TAUDTTINm input start edge.

Table 20.70 TAUDTTOUTm to which Dead Time is Added and State of TAUDTTINm

TAUDCMUR. TAUDTISm	TAUDTOL. TAUDTOLm	TAUDTTOUTm to which Dead Time is Added	TAUDTDL. TAUDTDLm	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

(d) Conditions

- TAUDCMURm.TAUDTIS[1:0] bits specify the type of width measurement:
  - TAUDCMURm.TAUDTIS[1:0] = 10B: Uses both edges as effective edges for detection (Low width measurement).
  - TAUDCMURm.TAUDTIS[1:0] = 11B: Uses both edges as effective edges for detection (High width measurement).
- The TAUDTDL.TAUDTDLm bit specifies the operation of TAUDTTOUTm for each channel when an interrupt or effective edge of TAUDTTINm is detected on the lower channel:
  - If TAUDTDL.TAUDTDLm = 0, an interrupt is used as a TAUDTTOUTm set trigger and an effective TAUDTTINm edge as a TAUDTTOUTm reset trigger.
  - If TAUDTDL.TAUDTDLm = 1, an effective TAUDTTINm edge is used as a TAUDTTOUTm set trigger and an interrupt as a TAUDTTOUTm reset trigger.
- The counter cannot be forcibly restarted with this function.

(2) Block Diagram and General Timing Diagram

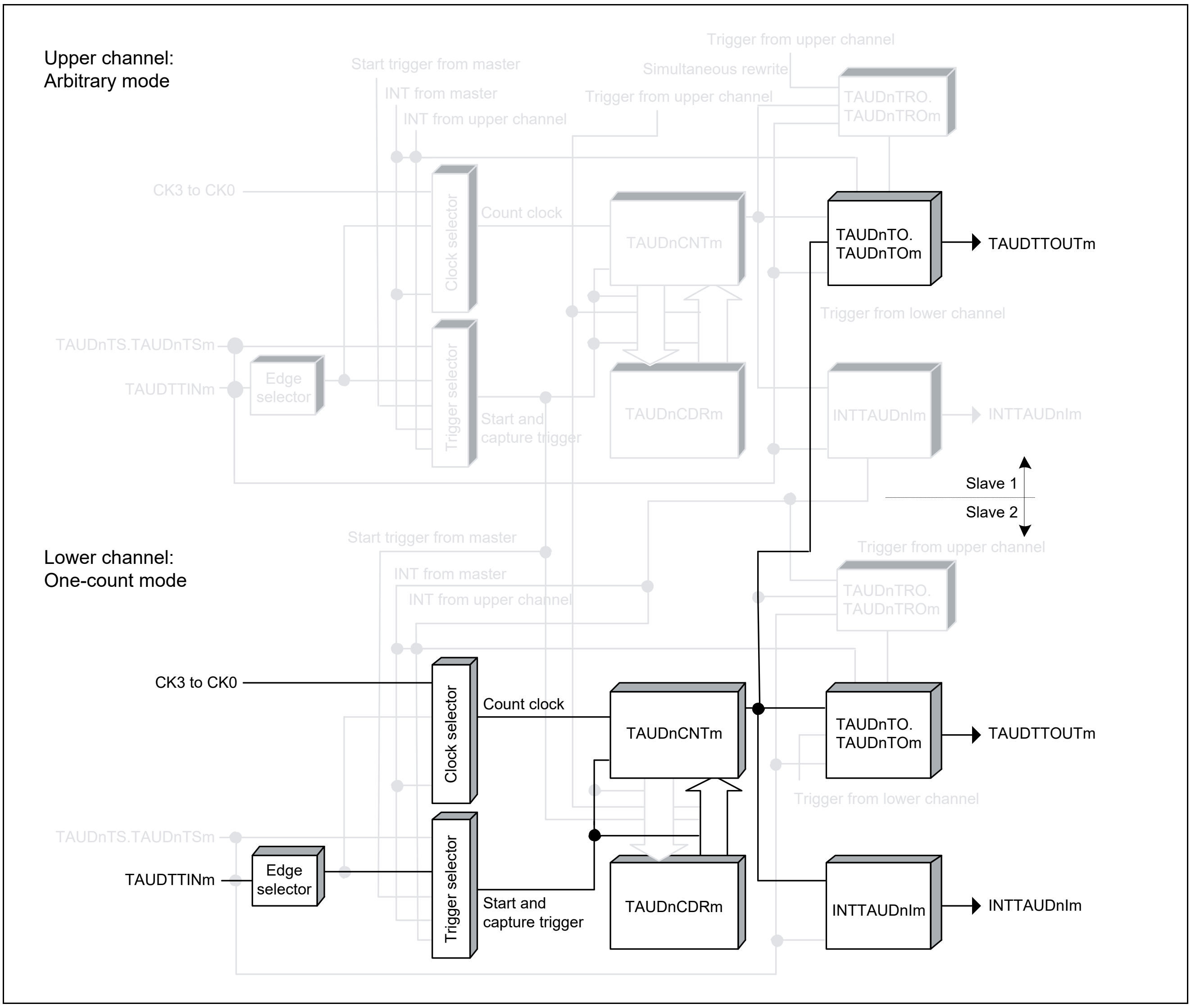


Figure 20.76 Block Diagram of One-Phase PWM Output

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDCMURm.TAUDTIS[1:0] = 11B)

This setting considers a duty as an active high.

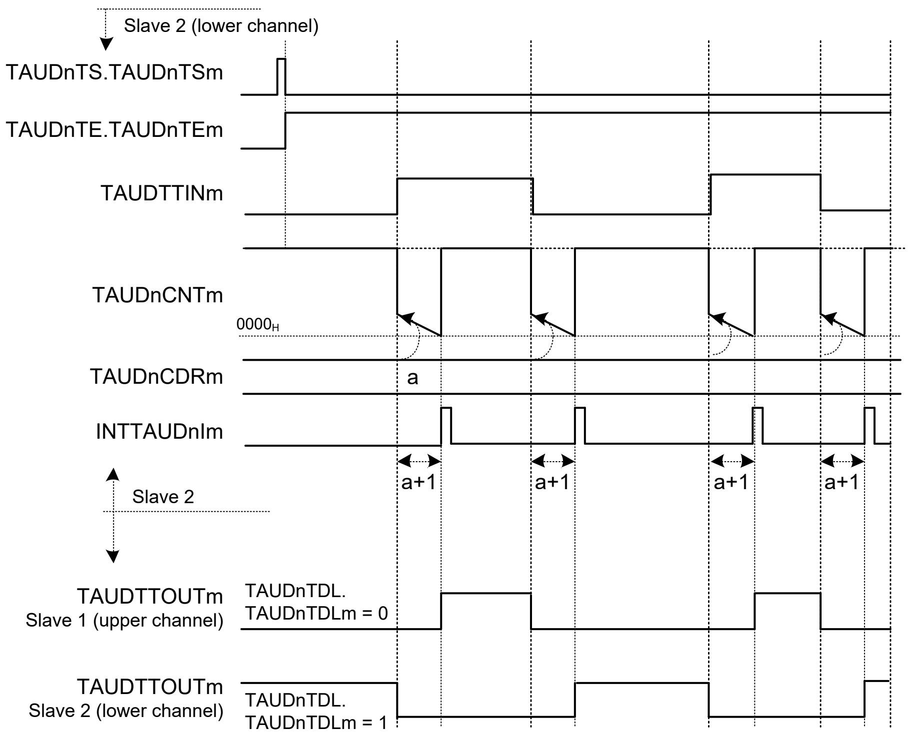


Figure 20.77 General Timing Diagram of One-Phase PWM Output

(3) Register Settings for Lower Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.71 Contents of the TAUDCMORm Register for the Lower Channel of the One-Phase PWM Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Enables start trigger detection while counting.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.72 Contents of the TAUDCMURm Register for the Lower Channel of the One-Phase PWM Output

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	TAUDTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(c) Channel output mode for lower channels

Table 20.73 Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel output
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	1: Enables dead time operation.
TAUDTDM.TAUDTDMm	1: Adds dead time upon detection of a TAUDTTINm input edge on a lower odd channel.
TAUDTDL.TAUDTDLm	0: An interrupt is used as a TAUDTTOUTm set trigger and an effective TAUDTTINm edge as a TAUDTTOUTm reset trigger. 1: an effective TAUDTTINm edge is used as a TAUDTTOUTm set trigger and an interrupt as a TAUDTTOUTm reset trigger.
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

Caution: Set TAUDTDL.TAUDTDLm exclusively from upper channels.

(d) Simultaneous reloading of lower channels

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 20.74 Simultaneous Reload Settings for One-Phase PWM Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(4) Register Settings for Upper Channels

(a) TAUDCMORm for upper channels

TAUDCMORm register for upper channels can be set arbitrarily.

(b) TAUDCMURm for upper channels

TAUDCMURm register for upper channels can be set arbitrarily.

(c) Channel output mode for upper channels

Table 20.75 Control Bit Settings for Upper Channels in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel output
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	1: Enables dead time operation.
TAUDTDM.TAUDTDMm	1: Adds dead time upon detection of a TAUDTTINm input edge on a lower odd channel.
TAUDTDL.TAUDTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

Caution: Set TAUDTDL.TAUDTDLm exclusively from lower channels.

(d) Simultaneous reloading of upper channels

Simultaneous reload register for upper channels can be set arbitrarily.

(5) Operating Procedure for One-Phase PWM Output

Table 20.76 Operating Procedure for One-Phase PWM Output

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers for the lower channel as described in Table 20.71, Contents of the TAUDCMORm Register for the Lower Channel of the One-Phase PWM Output, and Table 20.72, Contents of the TAUDCMURm Register for the Lower Channel of the One-Phase PWM Output. Set TAUDCMORm and TAUDCMURm registers for the upper channel as described in 20.12.15(4), Register Settings for Upper Channels. Set the value of TAUDCDRm register. Set channel output mode by setting the control bits as described in Table 20.73, Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output.	Channel operation is stopped.
Start Operation (Restart)	Set TAUDTOE.TAUDTOEm (slave channels 1 and 2) to 1 (at restart time only). Set TAUDTS.TAUDTSm = 1 for slave channel 2. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDTE.TAUDTEm is set to 1 (slave channel 2) and TAUDCNTm waits for detection of TAUDTTINm start edge. TAUDCNTm loads TAUDCDRm value.
During Operation	The TAUDCDRm register value can be changed at any time. The TAUDCNTm register can be read at any time.	TAUDCNTm of slave channel 2 counts down. When the counter reaches 0000H: • INTTAUDIm is generated. • TAUDCNTm stops counting TAUDTTOUTm is changed by a TAUDTTINm edge detection signal and slave channel 2 INTTAUDIm signal to output one-phase PWM waveform with dead time. Afterwards, this operation is repeated.
Stop Operation	Set TAUDTT.TAUDTTm = 1 for slave channel 2. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops. TAUDCNTm and TAUDTTOUTm retain their current values.

20.13 Independent Channel Real-Time Functions

This section describes functions that output the value of the TAUDTRO.TAUDTROm bit in real time.

20.13.1 Real-Time Output Type 1

(1) Overview

(a) Summary

This function outputs a value of the TAUDTRO.TAUDTROm bit from TAUDTTOUTm when a specified channel generates an interrupt (INTTAUDIm). In this function, the interrupt is generated at certain specified intervals.

The upper channel (TAUDTRC.TAUDTRCm = 1) generates a trigger for real-time output, and real-time output of the lower channel (TAUDTRC.TAUDTRCm = 0) takes place in response to the trigger from the upper channel.

(b) 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 20.77, Contents of the TAUDCMORm Register for the Upper Channel of Real-Time Output 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 20.7, Channel Output Modes.)
- Real-time output should be enabled for the upper channel (TAUDTRE.TAUDTREm = 1).

(c) Functional description

The counter of the upper channel is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm, enabling count operation. The current value of the data register of the upper channel (TAUDCDRm) is loaded into the counter (TAUDCNTm) and the counter starts to count down from this value.

When the counter of the upper channel reaches 0000H, INTTAUDIm is generated and TAUDTTOUTm outputs the current value of the real-time output bit (TAUDTRO.TAUDTROm) of every channel (only channels with TAUDTRE.TAUDTREm = 1). TAUDCNTm then reloads the TAUDCDRm 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 TAUDTRO.TAUDTROm at the moment that the interrupt is generated.

(d) Conditions

- The channel which is monitored for INTTAUDIm occurrence is specified by setting TAUDTRC.TAUDTRCm to 1 for the corresponding channel. The TAUDTRC.TAUDTRCm bit should be 0 for all other channels.
- If real-time output of a lower channel is disabled (TAUDTRE.TAUDTREm = 0) or if the channel itself is used as a rewrite trigger (TAUDTRC.TAUDTRCm = 1), the value of that channel’s TAUDTRO.TAUDTROm bit is output when INTTAUDIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUDTRE.TAUDTREm = 1) and TAUDTRC.TAUDTRCm = 0, the value of that channel's TAUDTRO.TAUDTROm bit is output when INTTAUDIm is generated in the upper channel.
- If the TAUDCMORm.TAUDMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see Section 20.9, TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts.

(2) Equations

INTTAUDIm generation cycle = count clock cycle × (TAUDCDRm value + 1)

(3) Block Diagram and General Timing Diagram

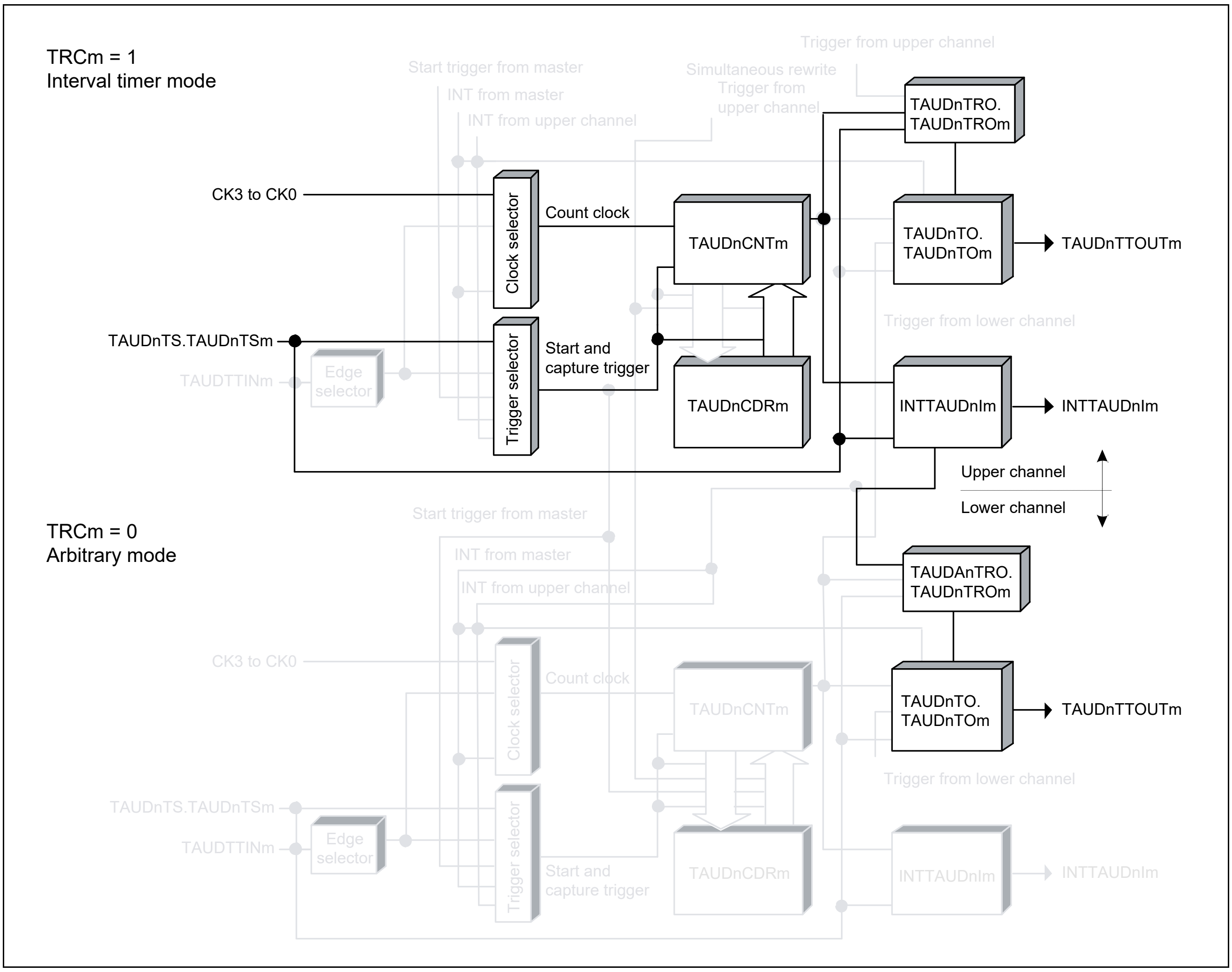


Figure 20.78 Block Diagram of Real-Time Output Type 1

The following settings apply to the general timing diagram.

- INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 1)

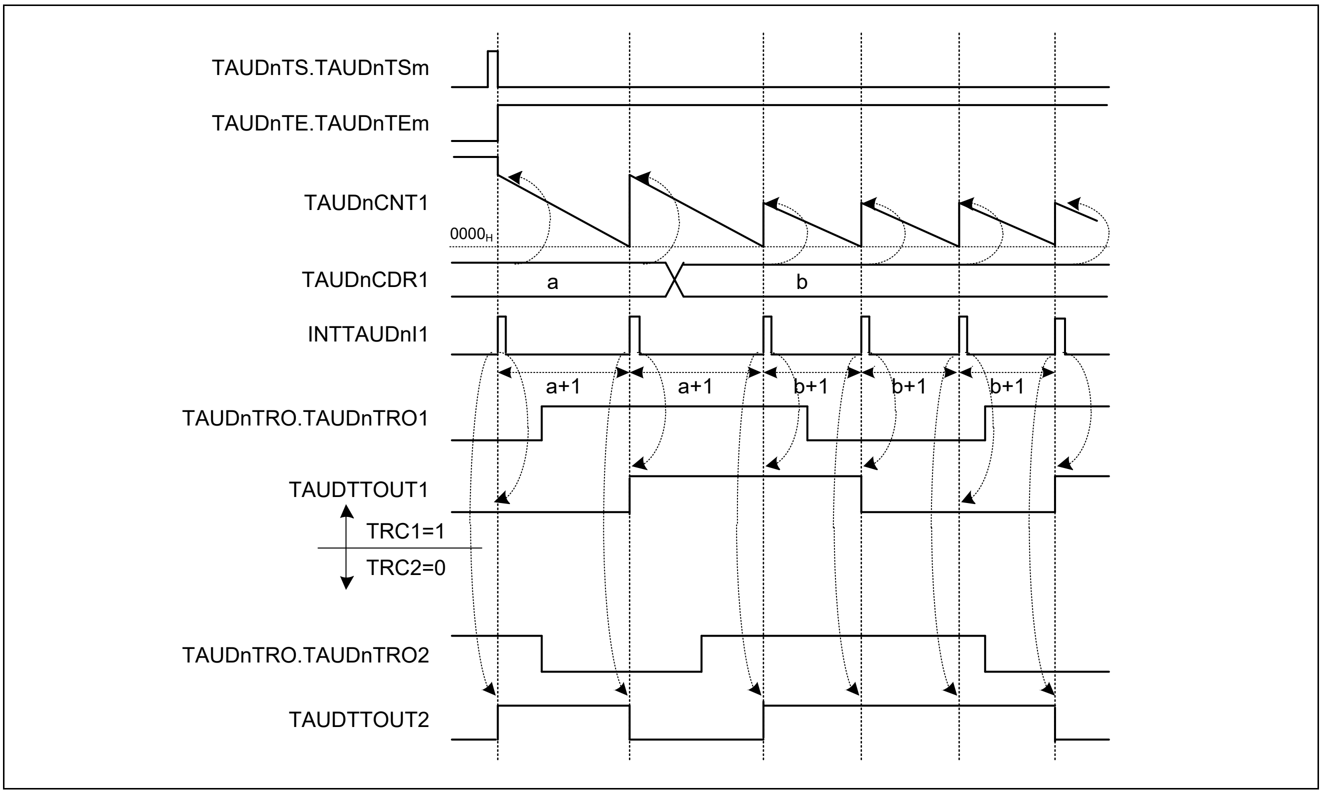


Figure 20.79 General Timing Diagram of Real-Time Output Type 1

(4) Register Settings for Upper Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.77 Contents of the TAUDCMORm Register for the Upper Channel of Real-Time Output Type 1

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation. 1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.78 Contents of the TAUDCMURm Register for the Upper Channel of Real-Time Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.79 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	1: Enables real-time output
TAUDTRO.TAUDTROm	0: Real-time output is low 1: Real-time output is high
TAUDTRC.TAUDTRCm	1: Channel m generates a unique real-time output trigger
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the real-time output type 1. Therefore, these registers should be set to 0.

Table 20.80 Simultaneous Reload Settings for Real-Time Output Type 1

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Register Settings for Lower Channels

(a) TAUDCMORm

The TAUDCMORm register for lower channels is available for any setting.

(b) TAUDCMURm

The TAUDCMURm register for lower channels is available for any setting.

(c) Channel output mode

Table 20.81 Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set this bit to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	1: Enables real-time output
TAUDTRO.TAUDTROm	0: Real-time output is low 1: Real-time output is high
TAUDTRC.TAUDTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading of lower channels

Simultaneous reload registers for lower channels is available for any setting.

(6) Operating Procedure for Real-Time Output Type 1

Table 20.82 Operating Procedure for Real-Time Output Type 1

	Operation	TAUD Status
Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers for upper channels as described in Table 20.77, Contents of the TAUDCMORm Register for the Upper Channel of Real-Time Output Type 1, and Table 20.78, Contents of the TAUDCMURm Register for the Upper Channel of Real-Time Output Type 1. Set TAUDCMORm and TAUDCMURm registers for lower channels as described in 20.13.1(5), Register Settings for Lower Channels. Set the value of TAUDCDRm register (only channels with TAUDTRC.TAUDTRCm = 1) Set channel output mode by setting the control bits as described in Table 20.79, 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 20.81, Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm = 1 on the channel with TAUDTRC.TAUDTRCm set to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUDTRC.TAUDTRCm set to 1] TAUDTE.TAUDTEm is set to 1 and the counter starts. TAUDCDRm value is loaded TAUDCNTm. If TAUDCMORm.TAUDMD0 is 1, INTTAUDIm is generated.
During Operation (Restart)	TAUDCDRm and TAUDTRO.TAUDTROm can be changed at any time. The TAUDCNTm register can be read at any time.	TAUDCNTm counts down. When the counter reaches 0000H: • The TAUDCDRm value is loaded in TAUDCNTm again and count operation continues. • INTTAUDIm is generated. • TAUDTTOUTm outputs the current value of the real-time output bit TAUDTRO.TAUDTROm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops. Both TAUDCNTm and TAUDTTOUTm retain their current values.

(7) Specific Timing Diagrams

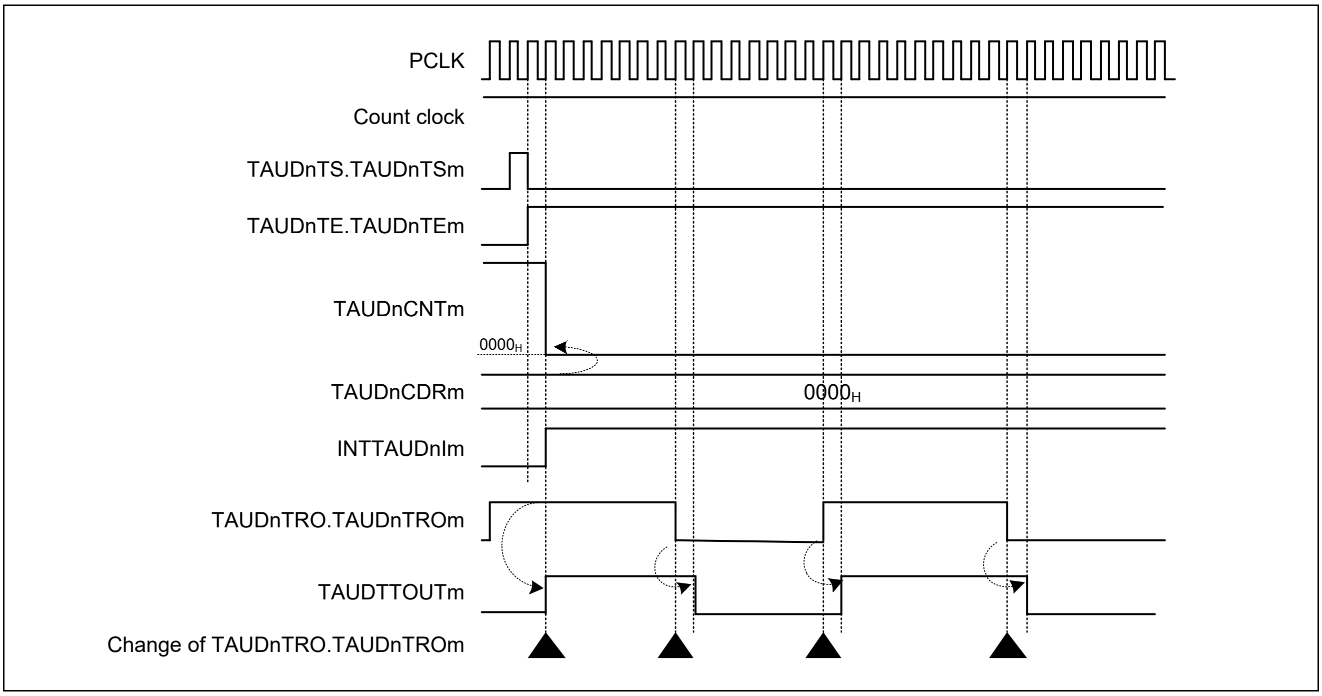


Figure 20.80 TAUDCDRm = 0000H, TAUDCMORm.TAUDMD0 = 1

- The value of TAUDTTOUTm changes according to the setting of TAUDTRO.TAUDTROm with a delay of one PCLK cycle.

20.13.2 Real-Time Output Type 2

(1) Overview

(a) Summary

This function outputs the value of TAUDTRO.TAUDTROm bit from TAUDTTOUTm when a specified channel generates an interrupt (INTTAUDIm). In this function, the interrupt is generated when an effective TAUDTTINm input edge is detected or the function starts.

The upper channel (TAUDTRC.TAUDTRCm = 1) generates a trigger for real-time output, and real-time output of the lower channel (TAUDTRC.TAUDTRCm = 0) takes place in response to the trigger from the upper channel.

(b) 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 20.83, Contents of the TAUDCMORm Register for the Upper Channel of Real-Time Output 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 1with real-time output. (See Section 20.7, Channel Output Modes.)
- Real-time output should be enabled for the upper channel (TAUDTRE.TAUDTREm = 1).

(c) Functional description

The counter for upper channels is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm to 1, enabling count operation. The counter starts to count up.

When an effective 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 (TAUDTRO.TAUDTROm) of every channel (only channels with TAUDTRE.TAUDTREm = 1).

The TAUDTTOUTm signal changes only when an interrupt is generated, and when TAUDTTOUTm value is different from the current value of TAUDTRO.TAUDTROm during the occurrence of the interrupt.

(d) Conditions

- The channel which is monitored for INTTAUDIm occurrence is specified by setting TAUDTRC.TAUDTRCm to 1 for the corresponding channel. The TAUDTRC.TAUDTRCm bit should be 0 for all other channels.
- If real-time output of a lower channel is disabled (TAUDTRE.TAUDTREm = 0) or if the channel itself is used as a rewrite trigger (TAUDTRC.TAUDTRCm = 1), the value of that channel's TAUDTRO.TAUDTROm bit is output when INTTAUDIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUDTRE.TAUDTREm = 1) and TAUDTRC.TAUDTRCm = 0, the value of that channel's TAUDTRO.TAUDTROm bit is output when INTTAUDIm is generated in the upper channel.
- If the TAUDCMORm.TAUDMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see Section 20.9, TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts.

(2) Block Diagram and General Timing Diagram

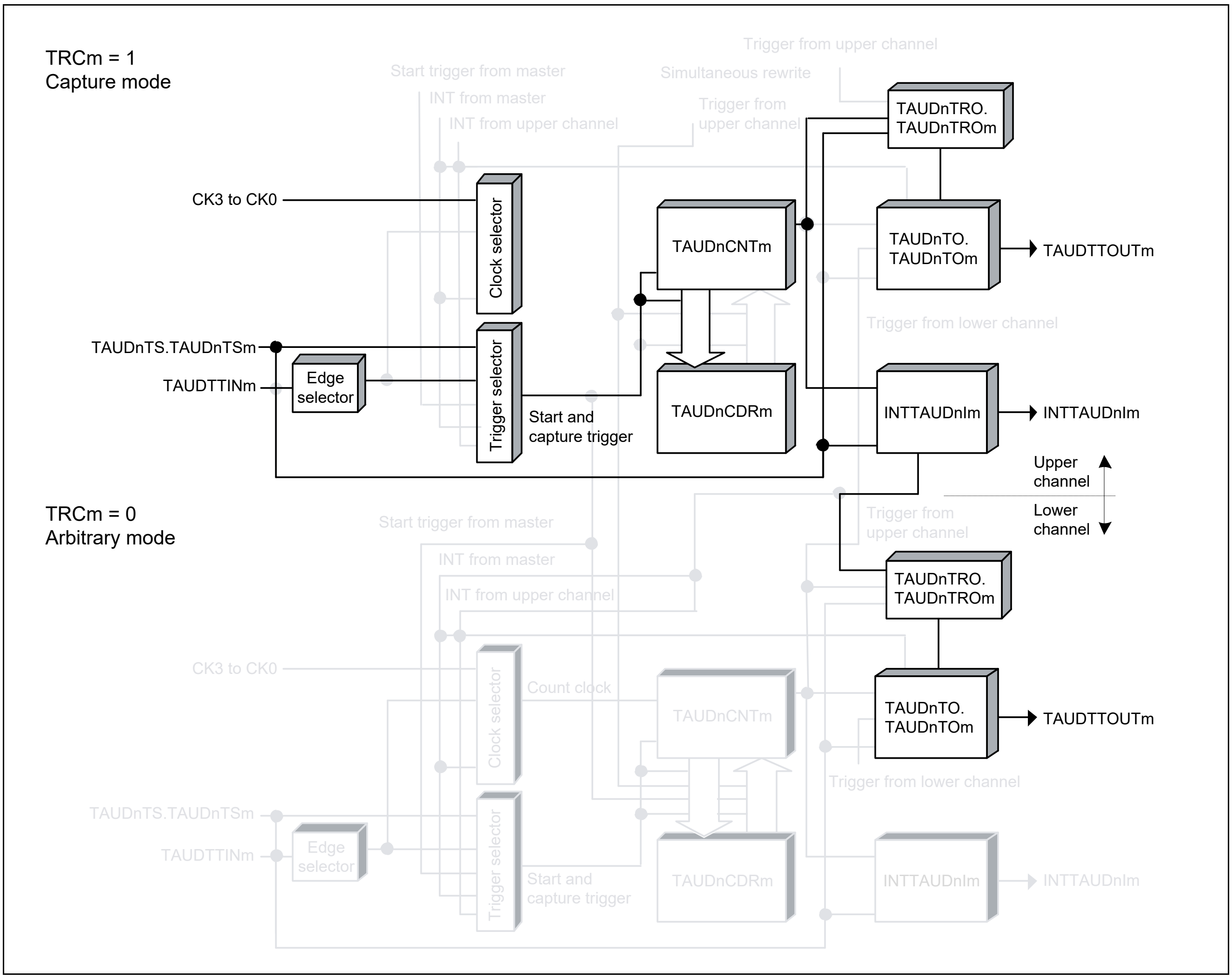


Figure 20.81 Block Diagram of Real-Time Output Type 2

The following settings apply to the general timing diagram.

- INTTAUDIm is not generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 0)

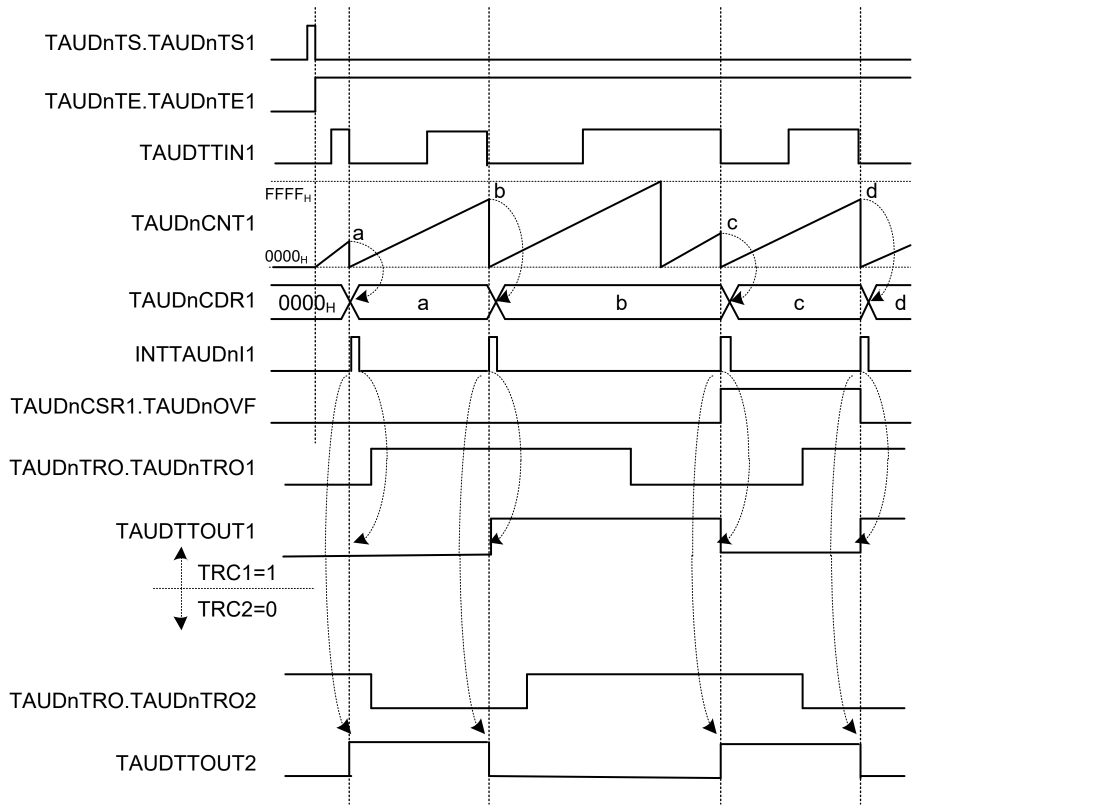


Figure 20.82 General Timing Diagram of Real-Time Output Type 2

(3) Register Settings for Upper Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.83 Contents of the TAUDCMORm Register for the Upper Channel of Real-Time Output Type 2

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0010: Capture mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation. 1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.84 Contents of the TAUDCMURm Register for the Upper Channel of Real-Time Output 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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode

Table 20.85 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	1: Enables real-time output
TAUDTRO.TAUDTROm	0: Real-time output is low 1: Real-time output is high
TAUDTRC.TAUDTRCm	1: Channel m generates a unique real-time output trigger
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

The simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with the real-time output type 2. Therefore, these registers should be set to 0.

Table 20.86 Simultaneous Reload Settings for Real-Time Output Type 2

Bit Name	Setting
TAUDRDE.TAUDRDEm	0: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(4) Register Settings for Lower Channels

(a) TAUDCMORm

The TAUDCMORm register for lower channels is available for any setting.

(b) TAUDCMURm

The TAUDCMURm register for lower channels is available for any setting.

(c) Channel output mode

Table 20.87 Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	1: Enables real-time output.
TAUDTRO.TAUDTROm	0: Real-time output is low 1: Real-time output is high
TAUDTRC.TAUDTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading of lower channels

Simultaneous reload registers for lower channels can be set arbitrarily.

(5) Operating Procedure for Real-Time Output Type 2

Table 20.88 Operating Procedure for Real-Time Output Type 2

		Operation	TAUD Status
Restart	Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers for upper channels as described in Table 20.83, Contents of the TAUDCMORm Register for the Upper Channel of Real-Time Output Type 2, and Table 20.84, Contents of the TAUDCMURm Register for the Upper Channel of Real-Time Output Type 2. Set TAUDCMORm and TAUDCMURm registers for the lower channel as described in 20.13.1(5), Register Settings for Lower Channels. Set TAUDCDRm register (only for channels with TAUDTRC.TAUnTRCm = 1). Set channel output mode by setting the control bits as described in Table 20.85, 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 20.87, Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output.	Channel operation is stopped.
	Start Operation	Set TAUDTS.TAUDTSm = 1 on the channel with TAUDTRC.TAUDTRCm set to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUDTRC.TAUDTRCm set to 1] TAUDTE.TAUDTEm is set to 1 and the counter starts. TAUDCNTm is cleared to 0000H. If TAUDCMORm.TAUDMD0 is 1, INTTAUDIm is generated.
	During Operation	TAUDTRO.TAUDTROm can be changed at any time.	TAUDCNTm starts to count up from 0000H. When an effective TAUDTTINm input edge is detected: • TAUDCNTm captures the TAUDCDRm value, and the counter is cleared to 0000H. • INTTAUDIm is generated. • When the effective edge of the TAUDTTINm input signal is detected immediately after the generation of an overflow, the TAUDCSRm.TAUDOVF bit is set to 1. When detected before the generation of an overflow, the TAUDCSRm.TAUDOVF bit is cleared to 0. TAUDTTOUTm outputs the current value of realtime output bit TAUDTRO.TAUDTROm. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops. TAUDCNTm, TAUDCSRm.TAUDOVF, and TAUDTTOUTm retain their current values.

(6) Specific Timing Diagrams

(a) Operation start and stop

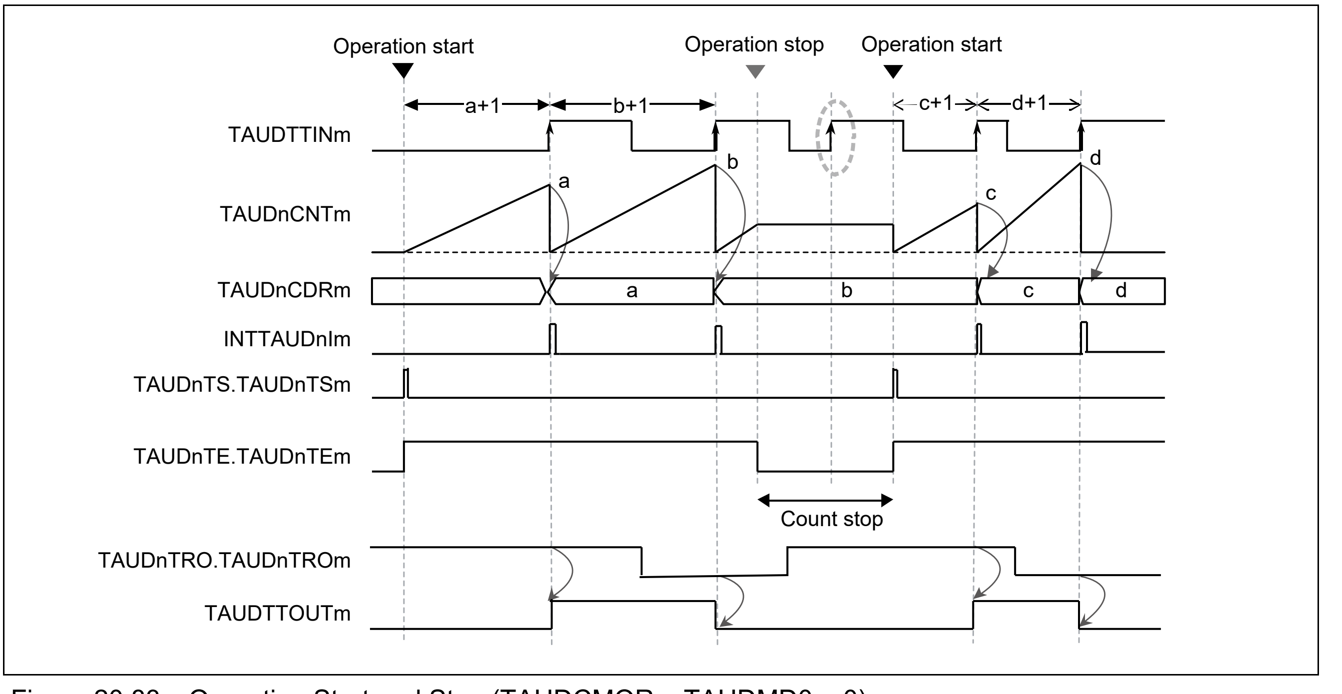


Figure 20.83 Operation Start and Stop (TAUDCMORm.TAUDMD0 = 0)

- When TAUDTS.TAUDTSm is set to 1, the counter starts counting up.
- When an effective input edge is detected, the current value of the counter is written to the data register (TAUDCDRm) and an interrupt is generated.
- TAUDTTOUTm outputs the current value of the real-time output bit (TAUDTRO.TAUDTROm) 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 TAUDTRO.TAUDTROm at the moment that the interrupt is generated.
- If the counter is stopped (TAUDTE.TAUDTEm = 0), effective input edges are ignored and no interrupt is generated.

20.14 Independent Channel Simultaneous Reloading

This section describes functions that carry out simultaneous reloading.

20.14.1 Simultaneous Reload Trigger Generation Type 1

(1) Overview

(a) Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a trigger for simultaneous reloading. The interrupt is generated at regular intervals.

The upper channel (TAUDRDC.TAUDRDCm = 1) generates a trigger for simultaneous reloading, and simultaneous reloading of the lower channel (TAUDRDC.TAUDRDCm = 0) takes place in response to the trigger from the upper channel.

(b) Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous reloading (TAUDRDE.TAUDRDEm = 1).
- The operating mode for the upper channel should be set to interval timer mode. (See Table 20.89, Contents of the TAUDCMORm Register for the Upper Channel of Simultaneous Reload Trigger Generation Type 1.)
- For the operating mode that can be set for lower channels, see Table 20.6, Channel Operations and Available Methods.
- TAUDTTOUTm is not used for any channel in this function.

(c) Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDTS.TAUDTSm) for upper and lower channels to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation. The current value of the data register buffer for upper channels (TAUDCDRm buf) is loaded into the counter (TAUDCNTm) 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 0000H, an interrupt occurs on the channel. The current value of the corresponding TAUDCDRm buffer is loaded into TAUDCNTm to continue operation subsequently.

If the channel where an interrupt occurs is specified as a trigger channel for simultaneous reloading (TAUDRDC.TAUDRDCm = 1) and is an upper channel, simultaneous reloading takes place on all lower channels in which simultaneous reloading is currently possible (TAUDRSF.TAUDRSFm = 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 reloading occurs.

(d) Conditions

- The channel which is monitored for INTTAUDIm occurrence is specified by setting TAUDRDC.TAUDRDCm = 1 for the corresponding channel. The TAUDRDC.TAUDRDCm bit should be 0 for all other channels in which simultaneous reloading should take place.
- If the TAUDCMORm.TAUDMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see Section 20.9, TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts.

(2) Equations

Simultaneous reload trigger generation cycle = count clock cycle × (TAUDCDRm + 1)
To control simultaneous reloading, the following condition should be satisfied:

[PWM]

TAUDCDRm
= [(value of TAUDCDRm of master channel subject to simultaneous reloading + 1) × number of interrupts] – 1

[Triangle PWM]

TAUDCDRm
= [(value of TAUDCDRm of master channel subject to simultaneous reloading + 1) × 2 × number of interrupts] – 1

That is, the ratio of TAUDCDRm + 1 and TAUDCDRm_master + 1 should be an integer. This integer corresponds to the number of interrupts.
For triangle PWM, remember that the cycle doubles.

(3) Block Diagram and General Timing Diagram

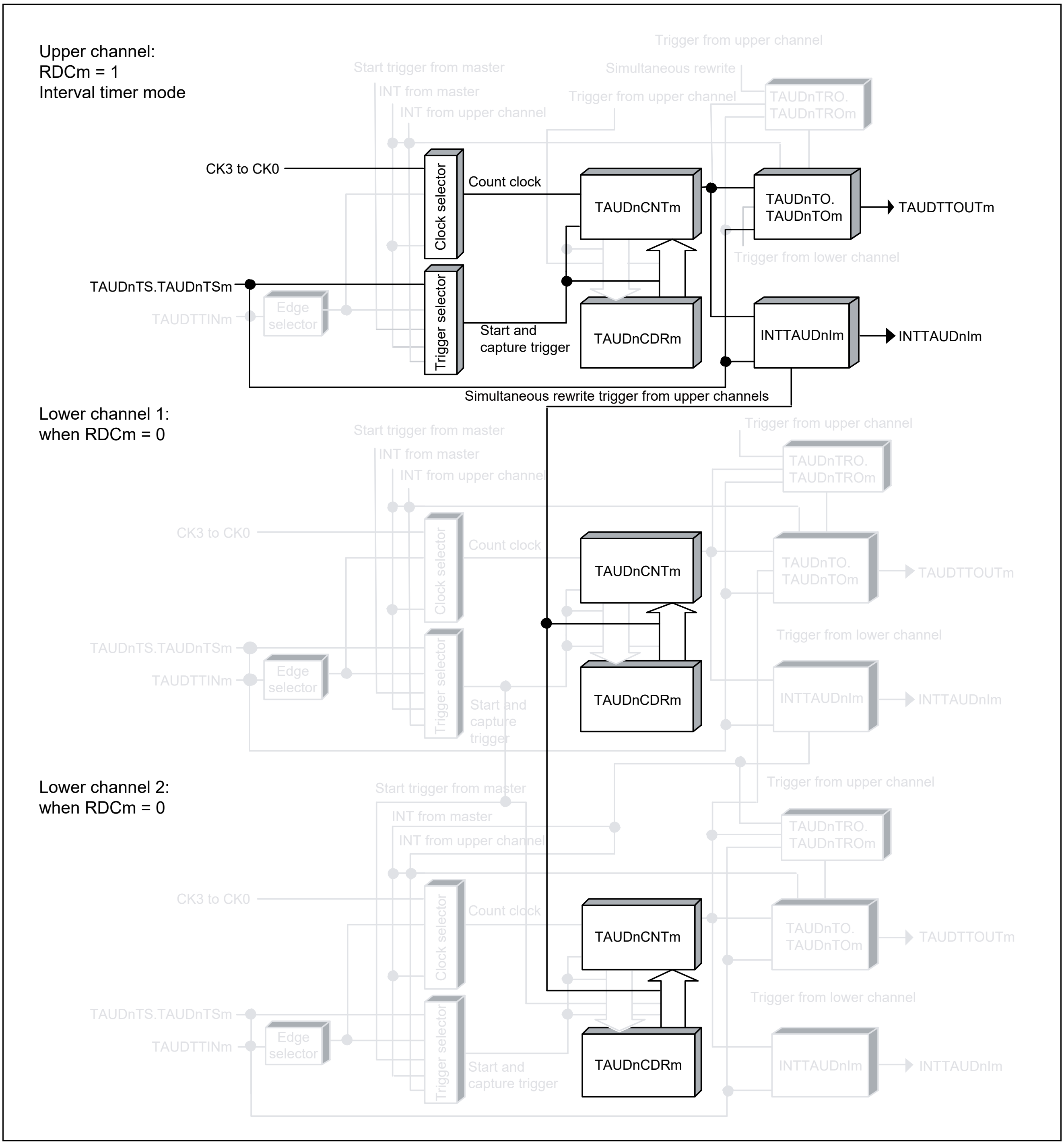


Figure 20.84 Block Diagram of Simultaneous Reload Trigger Generation Type 1

The following settings apply to the general timing diagram.

- INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 1)

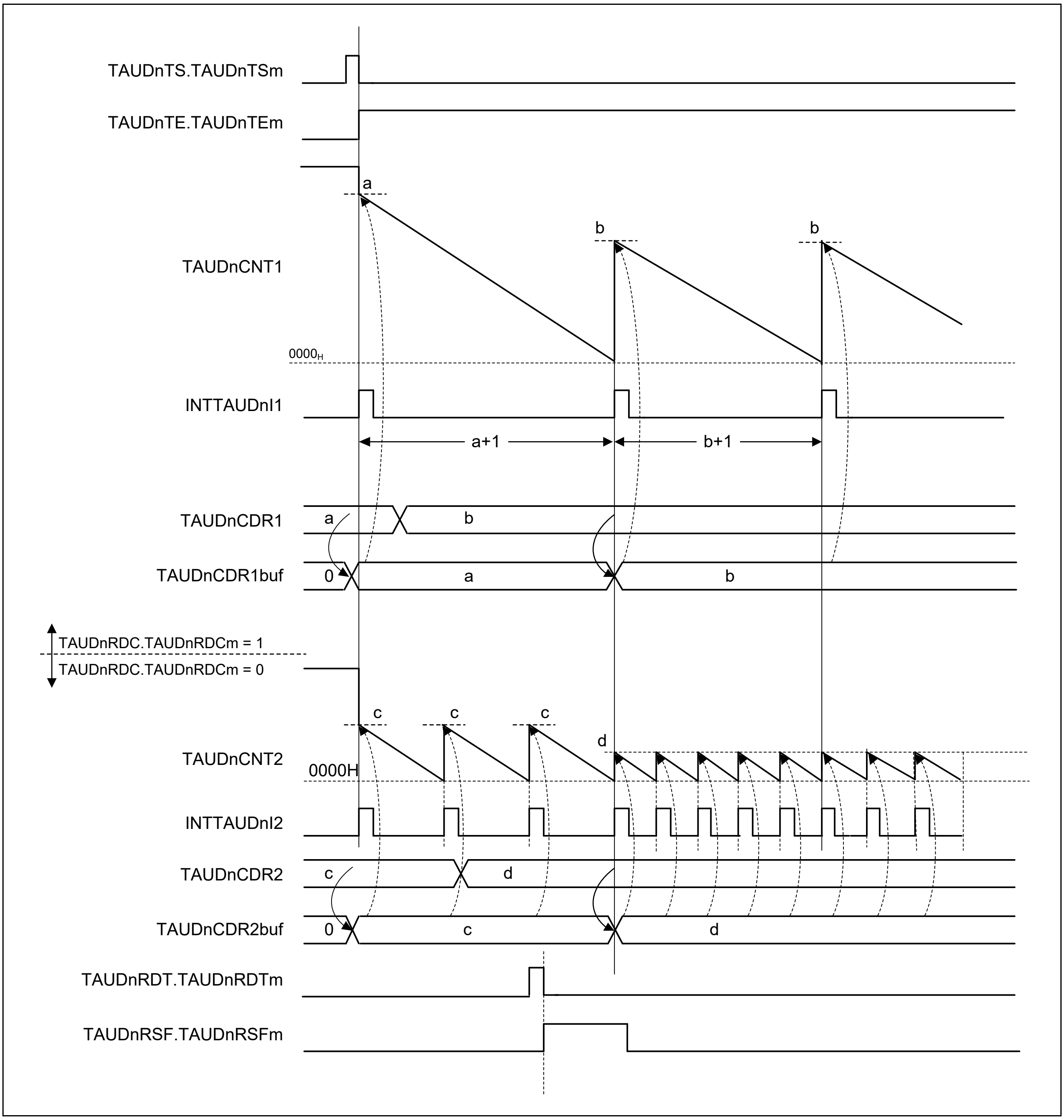


Figure 20.85 General Timing Diagram of Simultaneous Reload Trigger Generation Type 1

(4) Register Settings for Upper Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.89 Contents of the TAUDCMORm Register for the Upper Channel of Simultaneous Reload Trigger Generation Type 1

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation. 1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.90 Contents of the TAUDCMURm Register for the Upper Channel of Simultaneous Reload Trigger Generation 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm 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.

(d) Simultaneous reloading

Table 20.91 Simultaneous Reload Settings for Simultaneous Reload Trigger Generation Type 1

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	1: Selects one of upper channels as simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Set TAUDRDM.TAUDRDMm bit according to the settable operation mode.
TAUDRDC.TAUDRDCm	1: Monitors INTTAUDIm signal which triggers simultaneous rewriting on the channel.

(5) Register Settings for Lower Channels

(a) TAUDCMORm

TAUDCMORm register for lower channels must follow the TAUDCMORm register settings in the operating mode which can be set. (See Table 20.6, Channel Operations and Available Methods.)

(b) TAUDCMURm

TAUDCMURm register for lower channels must follow the TAUDCMURm register settings in the operating mode which can be set. (See Table 20.6, Channel Operations and Available Methods.)

(c) Channel output mode

Output according to the setting for lower channels (master/slave) is possible. As for the available function for simultaneous reload trigger generation type 1, see Table 20.6, Channel Operations and Available Methods. Simultaneous Reload Settings for Lower Channels in Simultaneous Reload Trigger Generation Type 1

(d) Simultaneous reloading

Table 20.92 Simultaneous Reload Settings for Lower Channels in Simultaneous Reload Trigger Generation Type 1

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	1: Selects one of upper channels as simultaneous reload control channel.
TAUDRDM.TAUDRDMm	Set TAUDRDM.TAUDRDMm bit according to the settable operation mode.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(6) Operating Procedure for Simultaneous Reload Trigger Generation Type 1

Table 20.93 Operating Procedure for Simultaneous Reload Trigger Generation Type 1

		Operation	TAUD Status
	Initial Channel Setting	Set TAUDCMORm and TAUDCMURm registers for the upper channel as described in Table 20.89, Contents of the TAUDCMORm Register for the Upper Channel of Simultaneous Reload Trigger Generation Type 1, and Table 20.90, Contents of the TAUDCMURm Register for the Upper Channel of Simultaneous Reload Trigger Generation Type 1. Set TAUDCMORm and TAUDCMURm registers for lower channels as described in 20.14.1(5), Register Settings for Lower Channels. Set the value of TAUDCDRm register.	Channel operation is stopped.
	Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. TAUDCDRm value is loaded into TAUDCNTm. If TAUDCMORm.TAUDMD0 = 1, INTTAUDIm is generated.
Restart	During Operation	TAUDRDT.TAUDRDTm and TAUDCDR.TAUDCDRm is changeable. TAUDRSF.TAUDRSFm can be always read.	TAUDCNTm counts down. When the counter reaches 0000H: • The TAUDCDRm value is loaded in TAUDCNTm again and count operation continues. • INTTAUDIm is generated. If INTAUDIm is generated on the channel where TAUDRDC.TAUDRDCm is set to 1, simultaneous reloading is controlled. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

20.14.2 Simultaneous Reload Trigger Generation Type 2

(1) Overview

(a) Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a trigger for simultaneous reloading. The interrupt is triggered when this function starts or by an effective TAUDTTINm input edge being detected.

The upper channel (TAUDRDC.TAUDRDCm = 1) generates a trigger for simultaneous reloading, and simultaneous reloading of the lower channel (TAUDRDC.TAUDRDCm = 0) takes place in response to the trigger from the upper channel.

(b) Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous reloading (TAUDRDE.TAUDRDEm = 1).
- The operation mode of the upper channel must be set to Capture Mode (see Table 20.94, Contents of the TAUDCMORm Register for the Upper Channel of Simultaneous Reload Trigger Generation Type 2).
- For the operation mode that can be set for a lower channel, see Table 20.6, Channel Operations and Available Methods.

(c) Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDTS.TAUDTSm) for upper and lower channels to 1. This sets TAUDTE.TAUDTEm = 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 TAUDRDC.TAUDRDCm = 1 on the upper channel, simultaneous reloading takes place on all lower channels in which simultaneous reloading is currently possible (TAUDRSF.TAUDRSFm = 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 reloading occurs.

(d) Conditions

- The channel which is monitored for INTTAUDIm is specified by setting TAUDRDC.TAUDRDCm = 1 for the corresponding channel. The TAUDRDC.TAUDRDCm bit must be 0 for all other channels in which simultaneous reloading should take place.
- If the TAUDCMORm.TAUDMD0 bit is set to 1, an interrupt is generated when the function starts. For details, see Section 20.9, TAUDTTOUTm Output and INTTAUDIm Generation when Counter Starts or Restarts.

(2) Block Diagram and General Timing Diagram

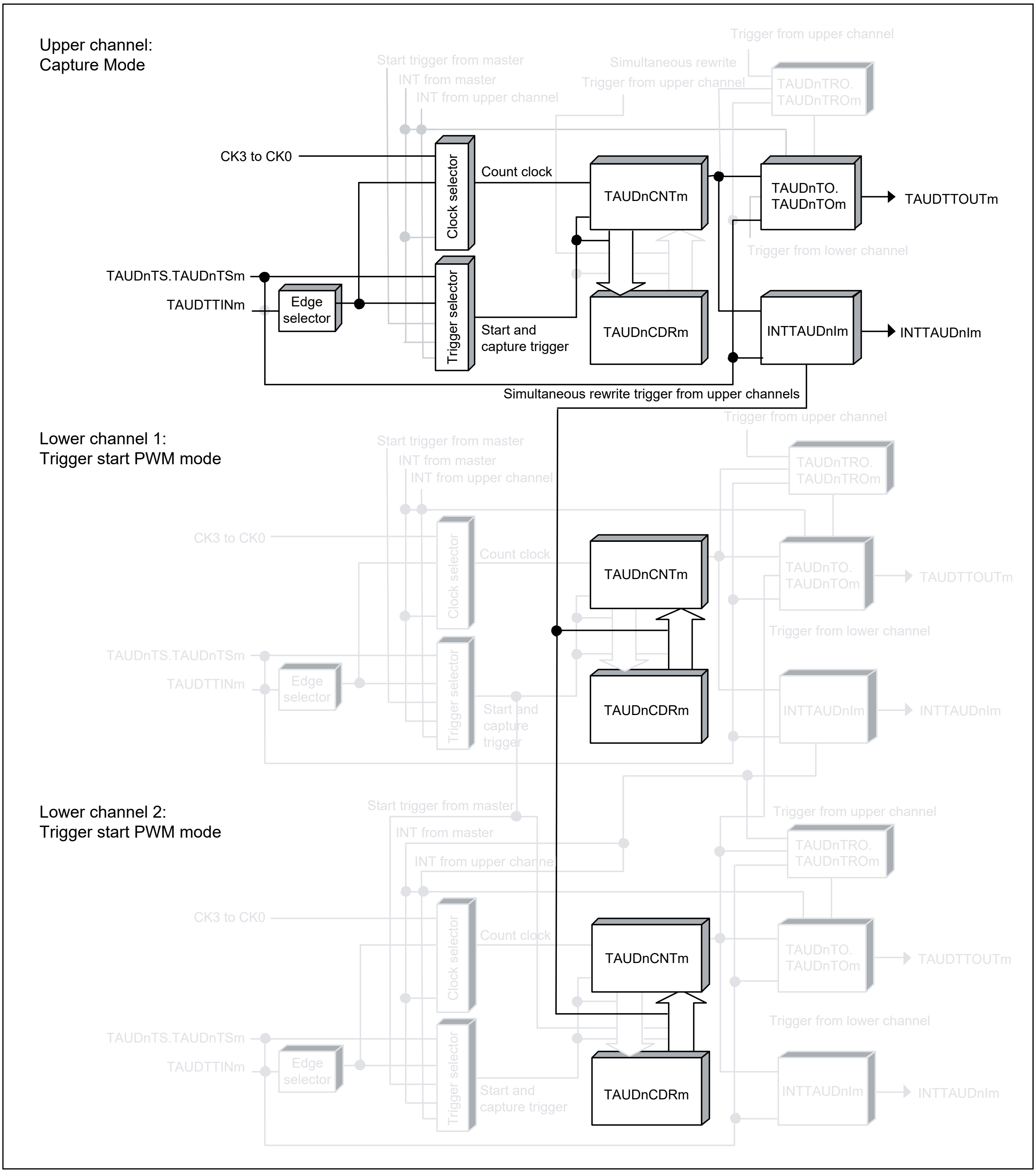


Figure 20.86 Block Diagram for Simultaneous Reload Trigger Generation Type 2

The following settings apply to the general timing diagram.

- INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 1)
- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)
- Upper channel (channel 1) generates simultaneous reload trigger.

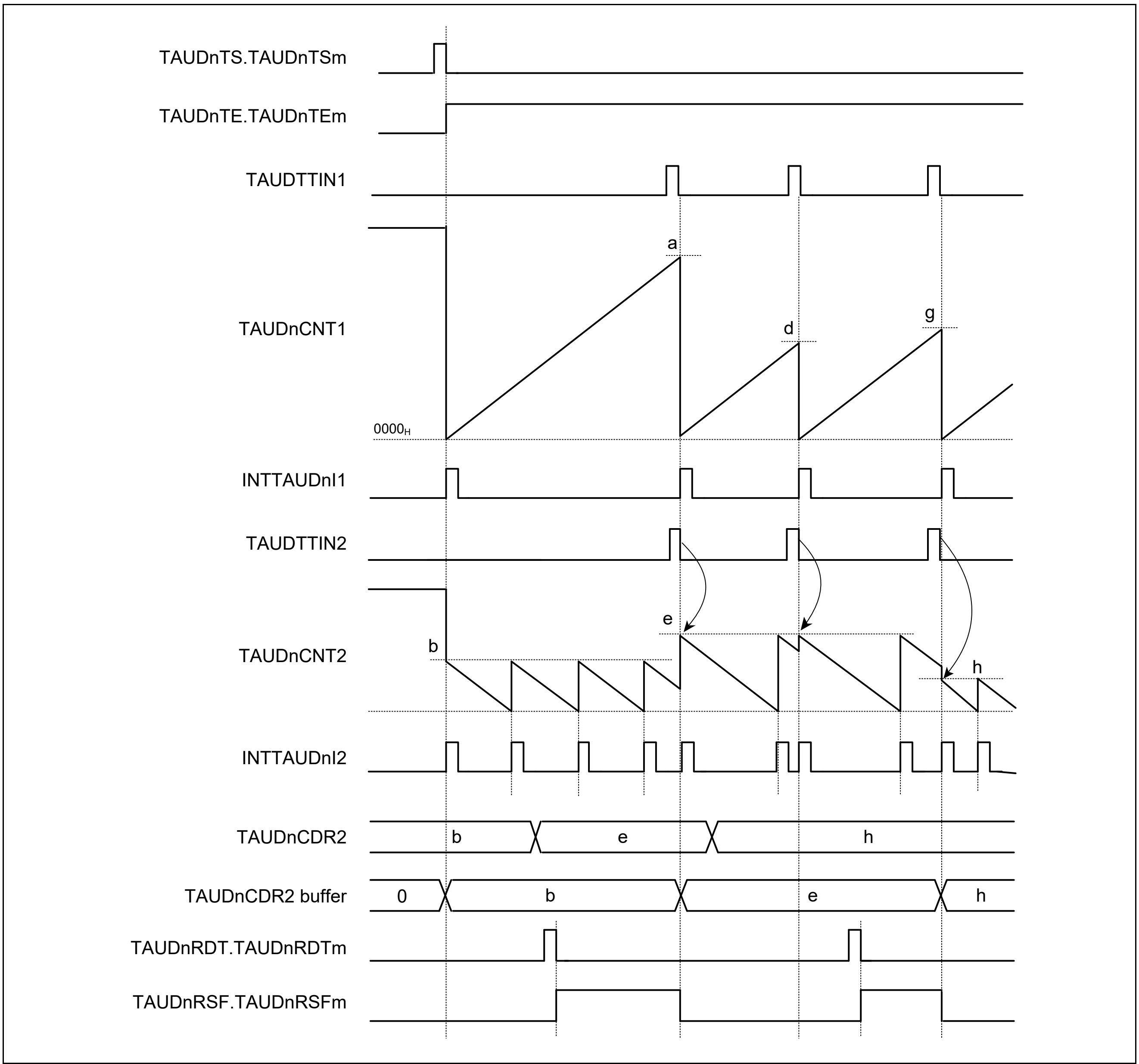


Figure 20.87 General Timing Diagram for Simultaneous Reload Trigger Generation Type 2

(3) Register Settings for Upper Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.94 Contents of the TAUDCMORm Register for the Upper Channel of Simultaneous Reload Trigger Generation Type 2

Bit Position	Bit Name	Function
15, 14	TAUDCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Independent operation. Set to 0.
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is the external capture trigger
7, 6	TAUDCOS[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	TAUDMD[4:1]	0010: Capture mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation. 1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.95 Contents of the TAUDCMURm Register for the Upper Channel of Simultaneous Reload Trigger Generation 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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode for upper channels

The channel output mode is not used by this function.

(d) Simultaneous reloading of upper channels

Table 20.96 Simultaneous Reload Settings for Simultaneous Reload Trigger Generation Type 2

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	1: Selects one of upper channels as simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Set TAUDRDM.TAUDRDMm bit according to the settable operation mode.
TAUDRDC.TAUDRDCm	1: Monitors INTTAUDIm signal which triggers simultaneous reloading on the channel.

(4) Register Settings for Lower Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.97 Contents of the TAUDCMORm Register for the Lower channel of Simultaneous Reload Trigger Generation Type 2

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as the start trigger
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.98 Contents of the TAUDCMURm Register for the Lower Channel of Simultaneous Reload Trigger Generation 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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode

Output according to the trigger start PWM mode setting is possible.

(d) Simultaneous reloading

Table 20.99 Simultaneous Reload Settings for the Lower Channel in Simultaneous Reload Trigger Generation Type 2

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	1: Selects one of upper channels as simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Loads a simultaneous reload control signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(5) Operating Procedure for Simultaneous Reload Trigger Generation Type 2

Table 20.100 Operating Procedure for Simultaneous Reload Trigger Generation Type 2

	Operation	TAUD Status
Initial Channel Setting (Restart)	Set the TAUDCMORm register and TAUDCMURm registers for the upper channel as described in Table 20.94, Contents of the TAUDCMORm Register for the Upper Channel of Simultaneous Reload Trigger Generation Type 2, and Table 20.95, Contents of the TAUDCMURm Register for the Upper Channel of Simultaneous Reload Trigger Generation Type 2. Set the TAUDCMORm register and TAUDCMURm registers for the lower channel as described in Table 20.97, Contents of the TAUDCMORm Register for the Lower channel of Simultaneous Reload Trigger Generation Type 2, and Table 20.98, Contents of the TAUDCMURm Register for the Lower Channel of Simultaneous Reload Trigger Generation Type 2. The TAUDCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm to 1. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is set to 1 and the counter starts. TAUDCNTm is cleared to 0000H. INTTAUDIm is generated when TAUDCMORm.TAUDMD0 is set to 1.
During Operation	TAUDRDT.TAUDRDTm can be set at any time. TAUDRSF.TAUDRSFm can be read at any time.	TAUDCNTm counts up from 0000H. When an effective edge of TAUDTTINm is detected: • TAUDCNTm transfers (captures) its value to TAUDCDRm and returns to 0000H. • INTTAUDIm is generated. Simultaneous reloading is controlled when INTTAUDIm is generated from the channel where TAUDRDC.TAUDRDCm is set to 1. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDTT.TAUDTTm to 1. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and it retains its current value.

20.15 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 20.2, Functional Overview.

This section describes functions that generate PWM signals at regular intervals.

20.15.1 PWM Output

(1) Overview

(a) 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.

(b) Prerequisites

- Two channels
- The operating mode for the master channel should be set to interval timer mode. (See Table 20.101, Contents of the TAUDCMORm Register for the Master Channel of the PWM Output.)
- The operating mode for the slave channels should be set to one-count mode. (See Table 20.104, Contents of the TAUDCMORm Register for the Slave Channel of the PWM Output.)
- 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 20.7, Channel Output Modes.)

(c) Functional description

The counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation. The current value of TAUDCDRm is loaded into TAUDCNTm, and the counter starts counting down from the TAUDCDRm value. If an INTTAUDIm is generated on the master channel and TAUDTTOUTm (slave) is set/reset, PWM output is enabled.

- Master channel:
  When the master channel counter reaches 0000H and the pulse cycle time has passed, INTTAUDIm is generated. The counter loads TAUDCDRm value into TAUDCNTm and counts down.
- Slave channel:
  When INTTAUDIm is generated on the master channel, the counter operation of the slave channel is triggered. The current value of TAUDCDRm (slave) is loaded into TAUDCNTm (slave) and the counter starts counting down from the TAUDCDRm value. TAUDTTOUTm signal is set to the active level.
  When the counter reaches to 0000H (duty time has elapsed), INTTAUDIm is generated and a TAUDTTOUTm signal is set to an inactive level. The counter is reset to FFFFH and waits for the next INTTAUDIm (start of the next pulse cycle) of the master channel.

The counter can stop operating by setting the TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDTS.TAUDTSm to 1.

(d) Conditions

Simultaneous reloading can be used with this function. See Section 20.6, Simultaneous Reloading.

(2) Equations

Pulse cycle = (TAUDCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUDCDRm (slave)/(TAUDCDRm (master) + 1)) × 100

- Duty cycle = 0%
  TAUDCDRm (slave) = 0000H
- Duty cycle = 100%
  TAUDCDRm (slave) ≥ TAUDCDRm (master) + 1

(3) Block Diagram and General Timing Diagram

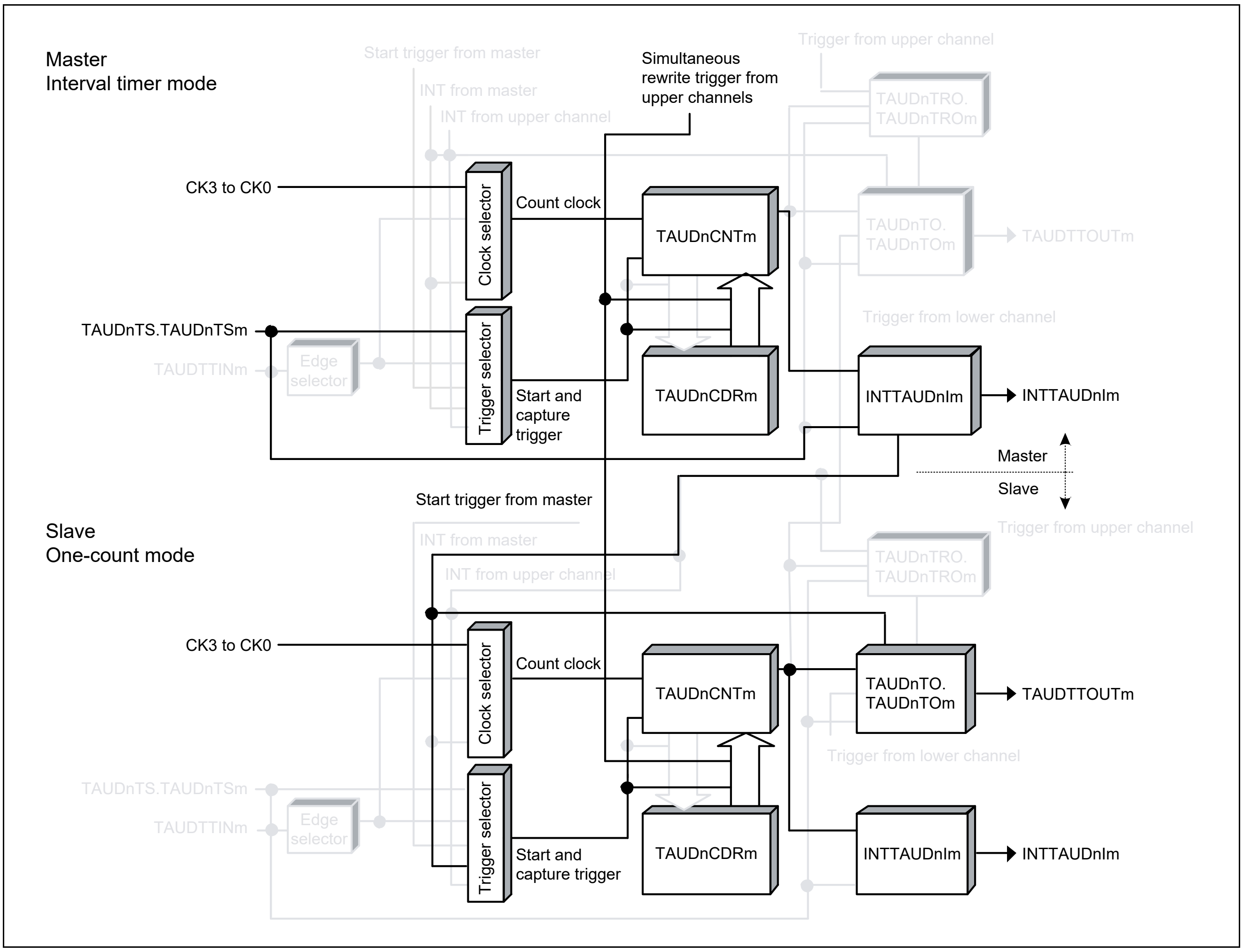


Figure 20.88 Block Diagram of PWM Output

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUDTOL.TAUDTOLm = 0)

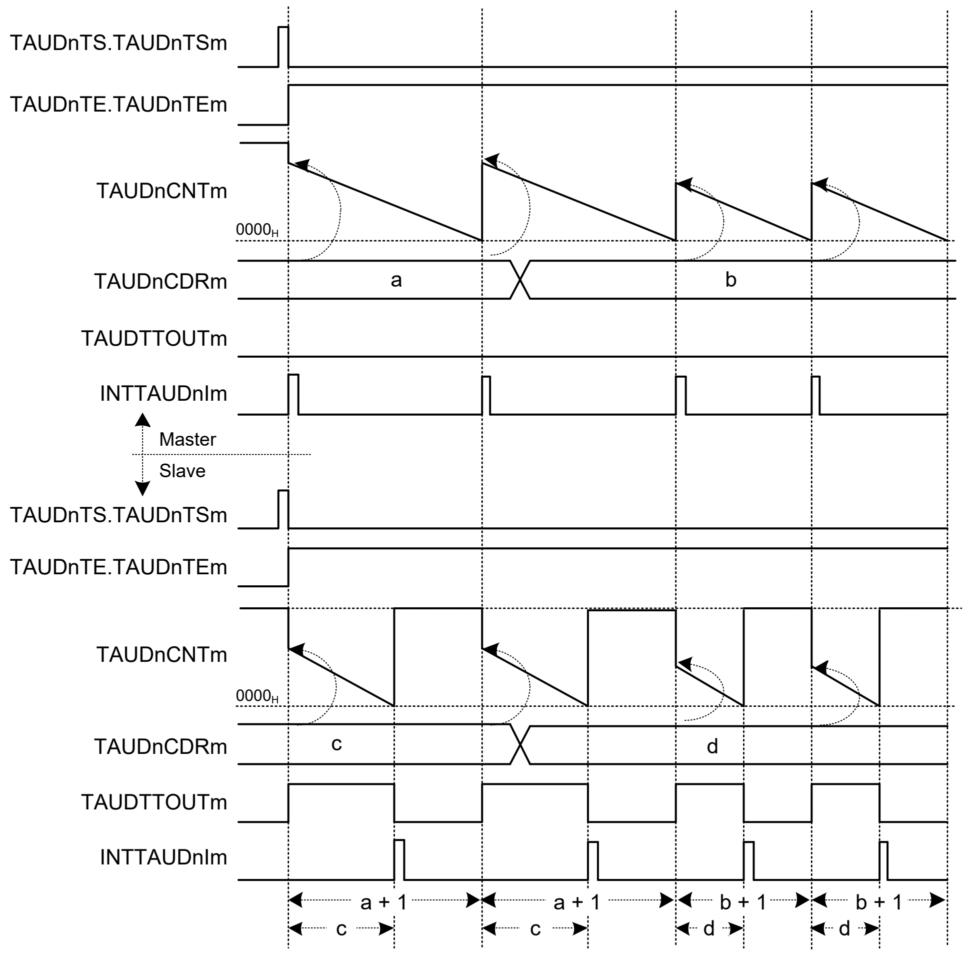


Figure 20.89 General Timing Diagram of PWM Output

Remarks 1. The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUDCDRm, whereas for the master channel the interval is the value of the corresponding TAUDCDRm + 1.

2. TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDIm of the master channel.

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.101 Contents of the TAUDCMORm Register for the Master Channel of the PWM Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.102 Contents of the TAUDCMURm Register for the Master Channel of the PWM Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

The channel output mode is not used with this function.

(d) Simultaneous reloading

Table 20.103 Simultaneous Reload Settings for the Master Channel of the PWM Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

Remark: Use with TAUDRDS.TAUDRDSm bit = 1 requires an upper channel higher than the master channel that operates with Section 20.14.1, Simultaneous Reload Trigger Generation Type 1.

Conduct operation settings under the following conditions:

- Simultaneous reload trigger output type 1 setting channel:
  TAUDRDCm = 1, TAUDRDSm = 1
  TAUDCDRm settings for this channel are as follows:
  = ((TAUDCDR setting for the master channel targeted for simultaneous reloading + 1) × interrupt count) – 1
- Master channel: TAUDRDCm = 0, TAUDRDSm = 1
- Slave channel: TAUDRDCm = 0, TAUDRDSm = 1

If TAUDCDRm (slave) setting > TAUDCDRm (master) setting + 1, the duty value (which exceeds 100%) is aggregated to be 100% output.

(5) Register Settings for Slave Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.104 Contents of the TAUDCMORm Register for the Slave Channel of the PWM Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	100: INTTAUDIm of master channel is a start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Start trigger during operation is valid.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.105 Contents of the TAUDCMURm Register for the Slave Channel of the PWM Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.106 Control Bit Settings in Independent Channel Output Mode 1

Bit Name Setting	Bit Name Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel operation
TAUDTOC.TAUDTOCm	0: Operating mode 1
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm TAUDTRC.TAUDTRCm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set these bits to 0
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.107 Simultaneous Reload Settings for Slave Channels of PWM Output

Bit Name Setting	Bit Name Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(6) Operating Procedure for PWM Output

Table 20.108 Operating Procedure for PWM Output

	Operation	TAUD Status
Initial Channel Setting	Master channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.15.1(4), Register Settings for the Master Channel. Slave channel: Set TAUDCMORm/ TAUDCMURm register and the channel output mode as described in Section 20.15.1(5), Register Settings for Slave Channels. Set the value of TAUDCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDIm is generated on the master channel and TAUDTTOUTm (slave) is set.
During operation	TAUDCDRm can be changed at any time. TAUDTOL.TAUDTOLm can be changed. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	TAUDCNTm of master channel loads TAUDCDRm value and counts down. When the counter reaches 0000H: • INTTAUDIm (master) is generated. • TAUDCDRm value is loaded into TAUDCNTm (master) to continue count operation. • TAUDCDRm value is loaded into TAUDCNTm (slave) to perform counting down. • TAUDTTOUTm (slave) is set to the active level. If TAUDCNTm (slave) reaches 0000H: • INTTAUDIm (slave) is generated. • TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.
Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

Restart (arrow from Stop Operation back to During operation)

(7) Specific Timing Diagrams

(a) Duty cycle = 0%

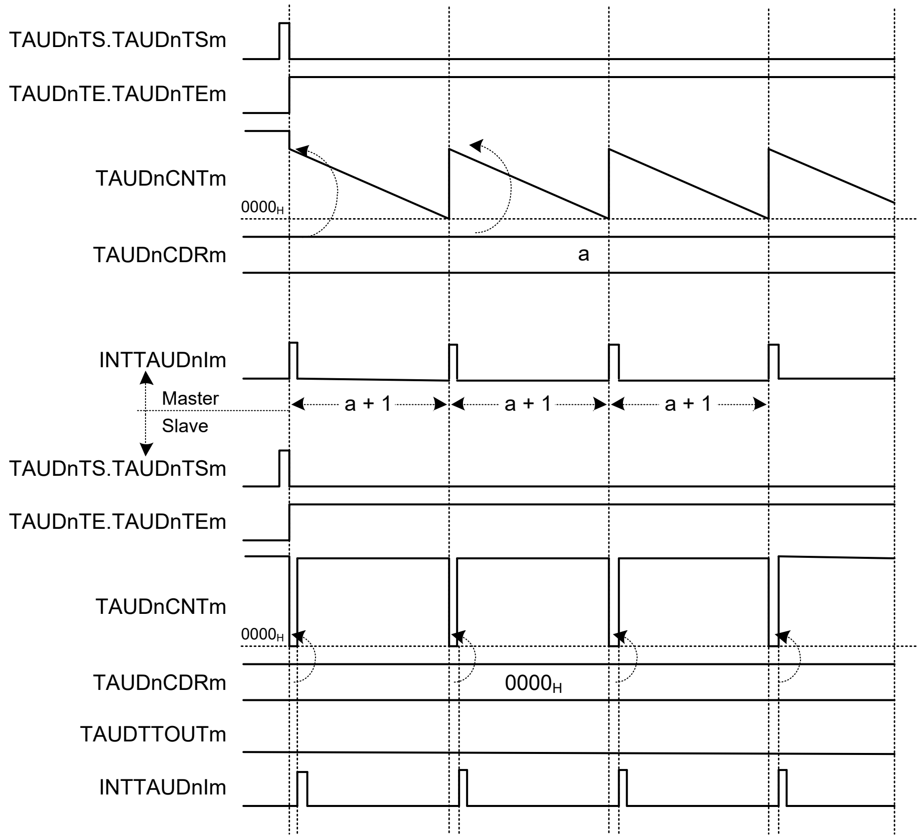


Figure 20.90 TAUDCDRm (Slave) = 0000H, Positive Logic (TAUDTOL.TAUDTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUDIm), 0000H is loaded into TAUDCNTm (slave). As a result, a slave channel interrupt (INTTAUDIm) is generated at the same time and TAUDTTOUTm remains inactive.
- TAUDCDRm value is loaded into TAUDCNTm (slave) to generate an interrupt.

(b) Duty cycle = 100%

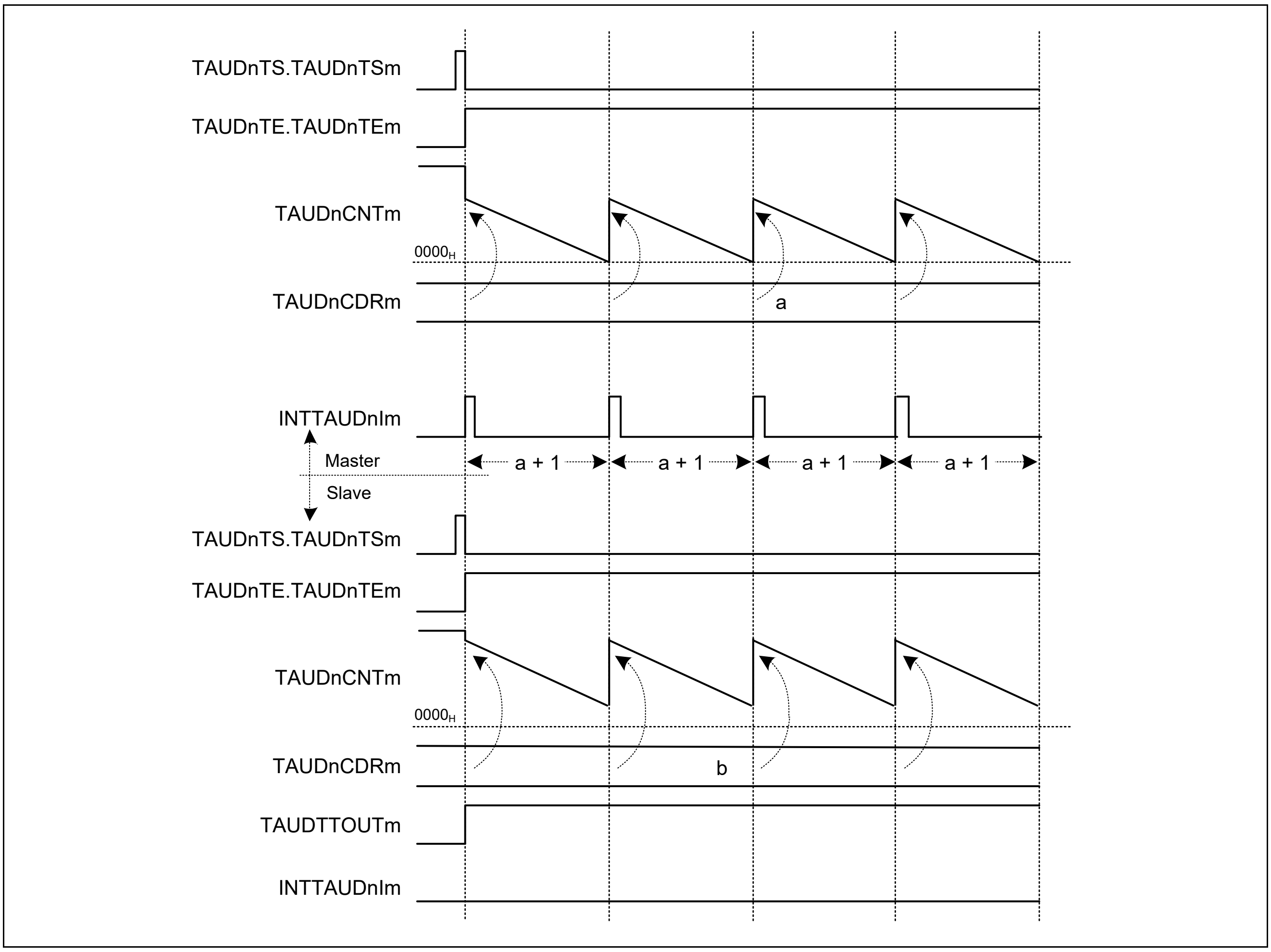


Figure 20.91 TAUDCDRm (Slave) ≥ TAUDCDRm (Master) + 1 Positive Logic (TAUDTOL.TAUDTOLm (Slave) = 0)

- If TAUDCDRm (slave) value is greater than TAUDCDRm (master) value, the slave channel counter does not reach 0000H and consequently, no interrupt occurs. TAUDTTOUTm remains active.

(c) Operation stop and restart

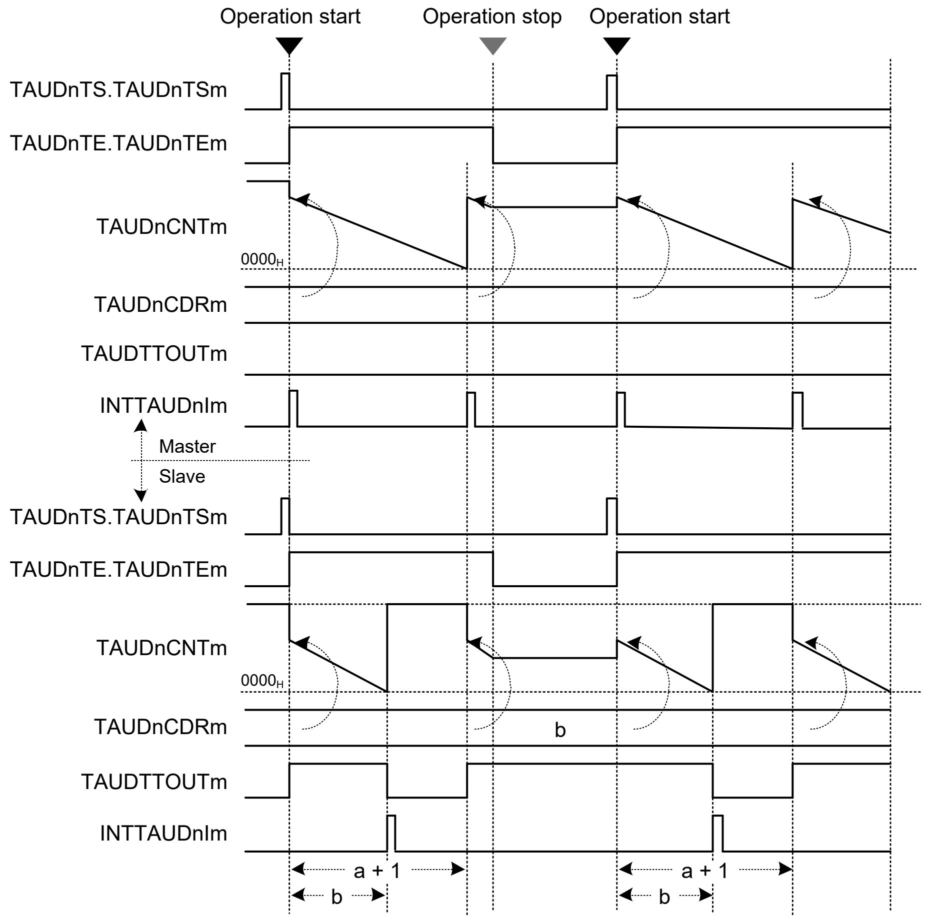


Figure 20.92 Operation Stop and Restart Positive Logic (TAUDTOL.TAUDTOLm (Slave) = 0)

- The counter can be stopped by setting TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0.
- TAUDCNTm and TAUDTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUDTS.TAUDTSm of master and slave channels to 1. TAUDCNTm of master and slave channels reload the current values of TAUDCDRm and start to count down from these values.

20.15.2 One-Shot Pulse Output

(1) Overview

(a) 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.

(b) Prerequisites

- Two channels
- The operating mode for the master channel should be set to one-count mode. (See Table 20.109, Contents of the TAUDCMORm Register for the Master Channel of the One-Shot Pulse Output.)
- The operating mode for slave channels should be set to pulse one-count mode. (See Table 20.112, Contents of the TAUDCMORm Register for the Slave Channel of the One-Shot Pulse Output.)
- 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 20.7, Channel Output Modes.)
- TAUDTTINm (master) has to be detected while TAUDCNTm (master) and TAUDCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUDTTINm (slave).

(c) Functional description

The counters are enabled by setting the channel trigger bits (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm, enabling count operation.

- Master channel:

  When the next effective edge of the TAUDTTINm input signal is detected, the current value of TAUDCDRm is loaded into TAUDCNTm. The counter starts to count down from this value. If TAUDCMORm.TAUDMD0 = 0, a trigger (TAUDTTINm) which is detected within the delay time is ignored.

  When the counter of master channel reaches 0000H, INTTAUDIm is generated. The counter is reset to FFFFH and waits for the next effective edge of the TAUDTTINm input signal.

- Slave channel:

  INTTAUDIm generated on master channel triggers the counter operation of slave channel. The current value of TAUDCDRm (slave) is loaded into TAUDCNTm (slave). The counter starts counting down from this value. An interrupt occurs and the TAUDTTOUTm signal is set.

  When the counter reaches 0001H, INTTAUDIm is generated and TAUDTTOUTm signal is reset. The counter stops at 0000H and waits for the next INTTAUDIm of master channel.

The counter can be stopped by setting TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDTS.TAUDTSm to 1.

Setting TAUDTS.TAUDTSm to 1 while counting allows the counter to restart counting of master channel without making a stop (forced restart).

(d) Conditions

- If TAUDCMORm.TAUDMD0 of master channel is set to 0, TAUDTTINm input edges detected during counting are ignored.
- Simultaneous reloading can be used with this function. See Section 20.6, Simultaneous Reloading.

(2) Equations

Delay from trigger input to pulse output = (TAUDCDRm (master) + 1) × count clock cycle

Pulse width = (TAUDCDRm (slave)) × count clock cycle

(3) Block Diagram and General Timing Diagram

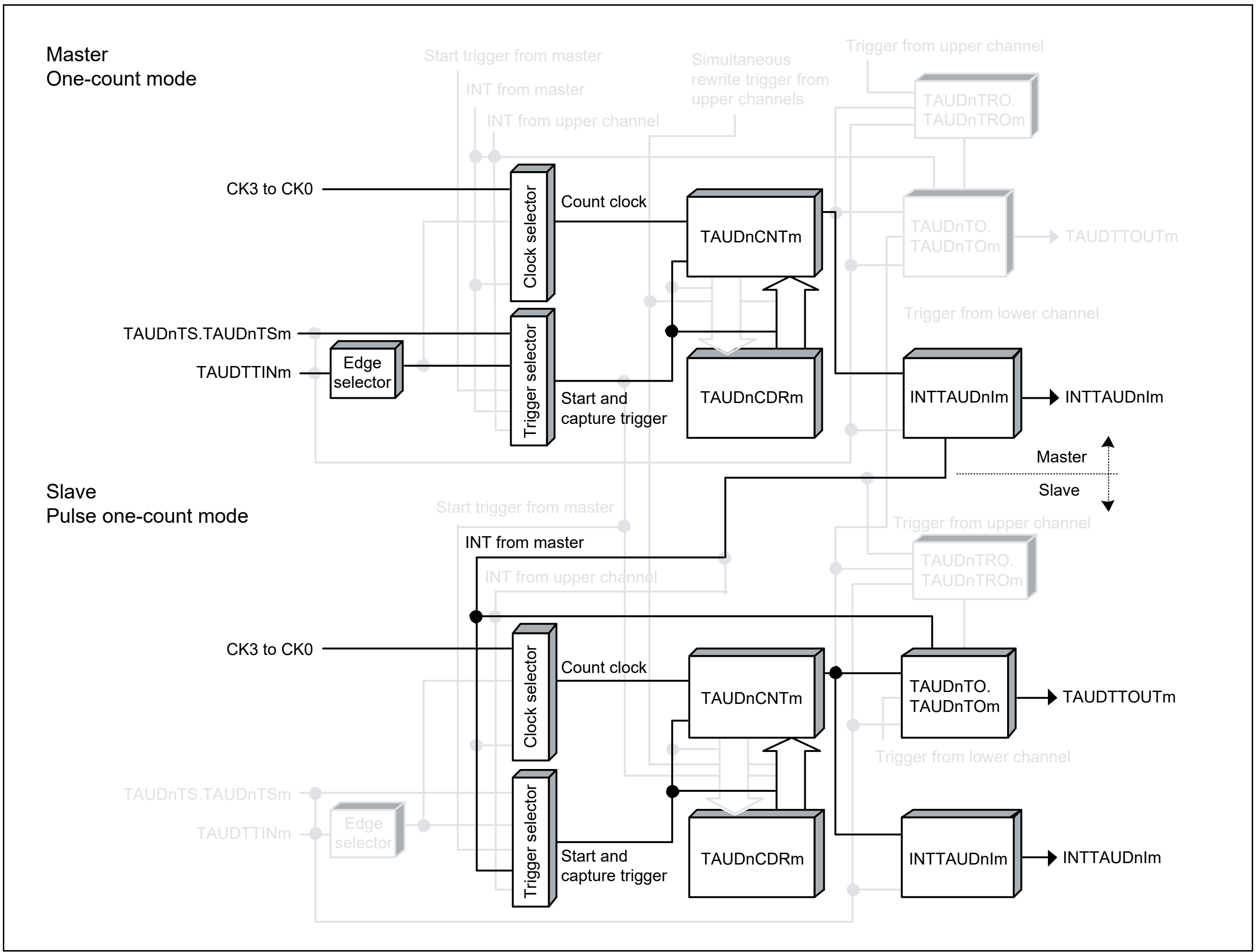


Figure 20.93 Block Diagram of One-Shot Pulse Output

The following settings apply to the general timing diagram.

- Start trigger detection is disabled during counting (TAUDCMORm.TAUDMD0 = 0).
- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

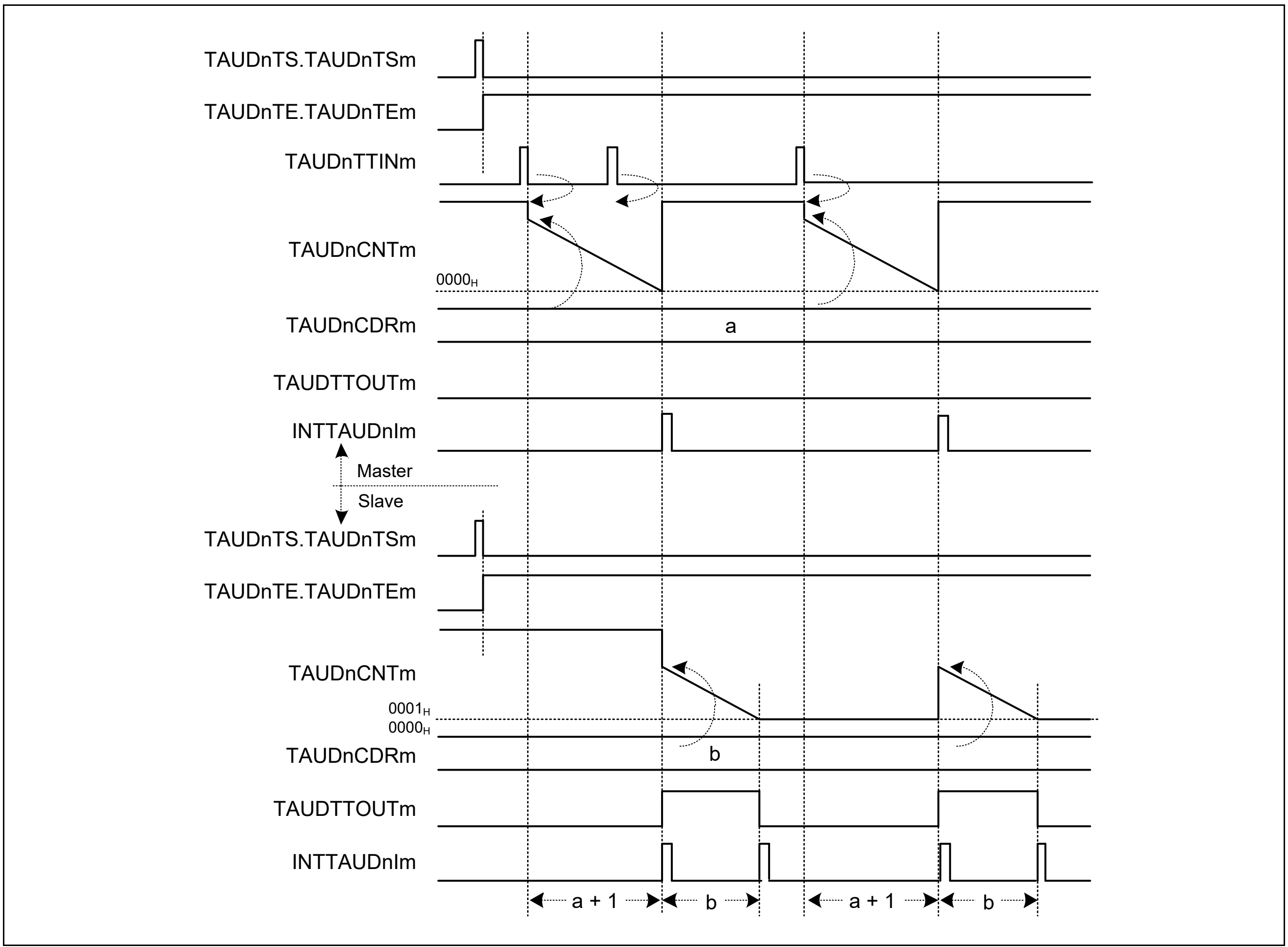


Figure 20.94 General Timing Diagram of One-Shot Pulse Output

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.109 Contents of the TAUDCMORm Register for the Master Channel of the One-Shot Pulse Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as the start trigger
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	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.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.110 Contents of the TAUDCMURm Register for the Master Channel of the One-Shot Pulse Output

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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because channel output mode is not used with this function.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.111 Simultaneous Reload Settings for the Master Channel of One-Shot Pulse Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Master channel is simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(5) Register Settings for Slave Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.112 Contents of the TAUDCMORm Register for the Slave Channel of the One-Shot Pulse Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	100: INTTAUDIm of master channel is a start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	1010: Pulse one-count mode
0	TAUDMD0	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.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.113 Contents of the TAUDCMURm Register for the Slave Channel of the One-Shot Pulse Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.114 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set this bit to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.115 Simultaneous Reload Settings for Slave Channels of One-Shot Pulse Output

Bit Name Setting	Bit Name Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Master channel is simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(6) Operating Procedure for One-Shot Pulse Output

Table 20.116 Operating Procedure for One-Shot Pulse Output

	Operation	TAUD Status
Initial Channel Setting	Master channel: Set TAUDCMORm/ TAUDCMURm register and the channel output mode as described in Section 20.15.1(4), Register Settings for the Master Channel. Slave channel: Set TAUDCMORm/ TAUDCMURm register and channel output mode as described in Section 20.15.1(5) Register Settings for Slave Channels. Set the value of TAUDCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm (master and slave channels) is set to 1 and the master channel awaits a TAUDTTINm input.
During Operation	TAUDCDRm can be changed at any time. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	When an effective edge of the TAUDTTINm input signal is detected, TAUDCDRm value of master channel is loaded into TAUDCNTm to perform counting down. When the counter reaches 0000H: • INTTAUDIm (master) is generated. • TAUDCNTm (master) is reset to FFFFH and waits for the next effective edge of the TAUDTTINm input signal. • TAUDCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down. • INTTAUDIm (slave) is generated. • TAUDTTOUTm (slave) is set to the active level. When TAUDCNTm (slave) reaches 0001H: • INTTAUDIm (slave) is generated. • TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.
Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

Restart (from Stop Operation back to During Operation)

(7) Specific Timing Diagrams

(a) TAUDCDRm (master) = 0000H

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDCMORm.TAUDMD0 = 0)
- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

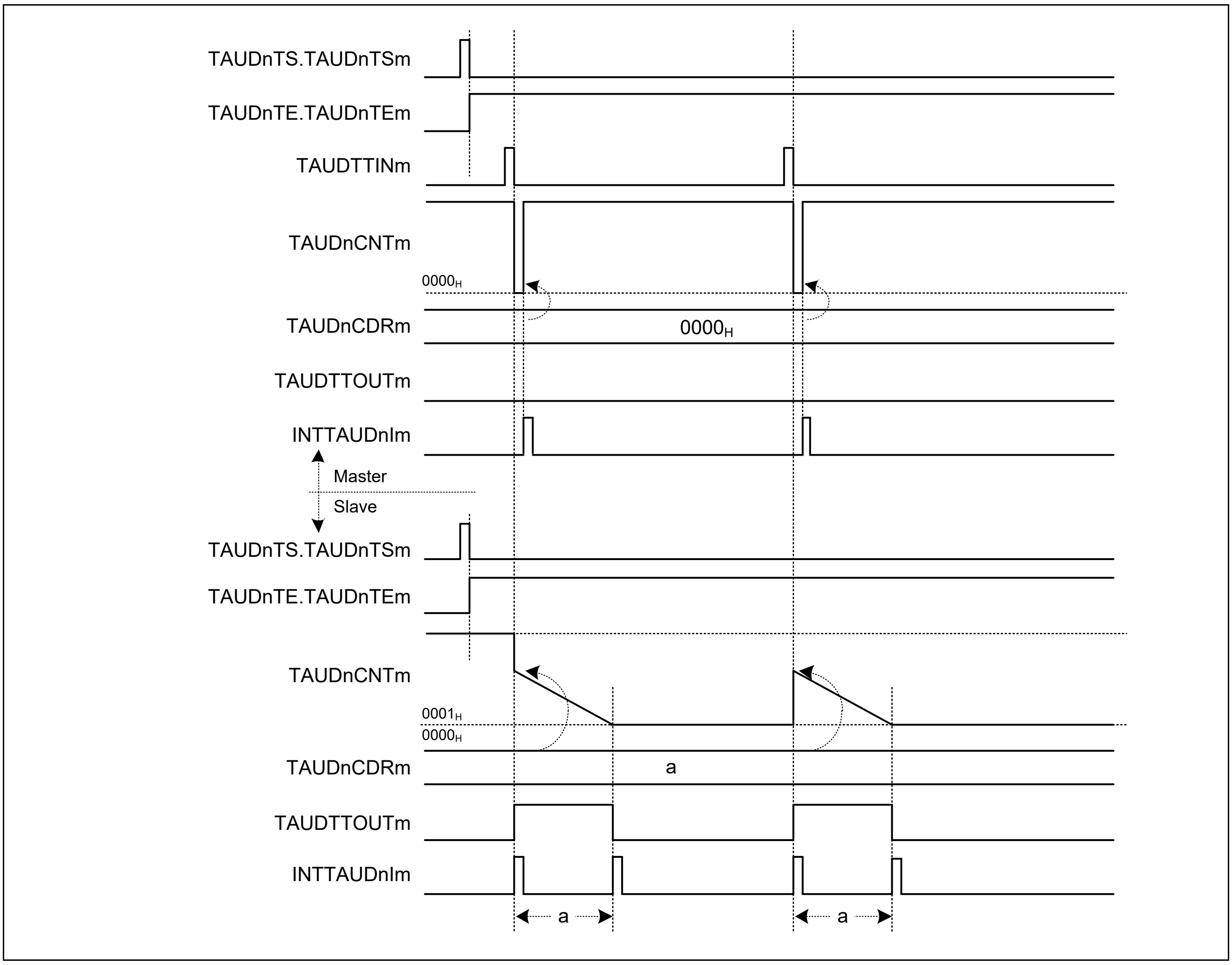


Figure 20.95 TAUDCDRm (Master) = 0000H

- When an effective TAUDTTINm input edge is detected, the value 0000H is written to TAUDCNTm (master). The counter is set to 0000H for one count and returns to FFFFH.
  Thus, the slave channel starts to count down one count clock later than TAUDTTINm (master).

(b) TAUDCDRm (slave) = 0000H

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDCMORm.TAUDMD0 = 0)
- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

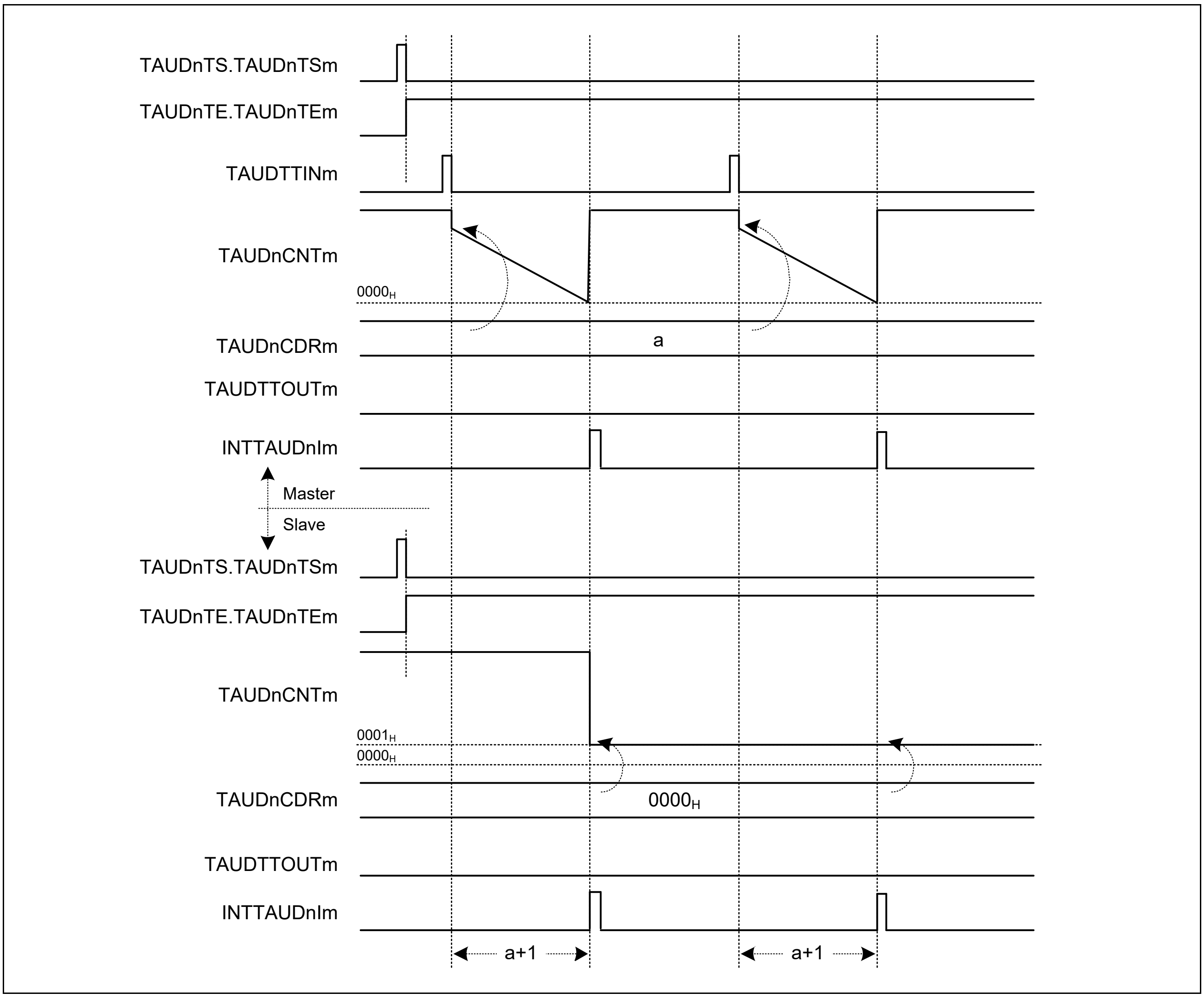


Figure 20.96 TAUDCDRm (Slave) = 0000H

- TAUDTTOUTm remains inactive, because the pulse width is zero.

(c) TAUDCMORm.TAUDMD0 = 1

The following settings apply to this diagram.

- Enables start trigger detection while counting. (TAUDCMORm.TAUDMD0 = 1)
- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

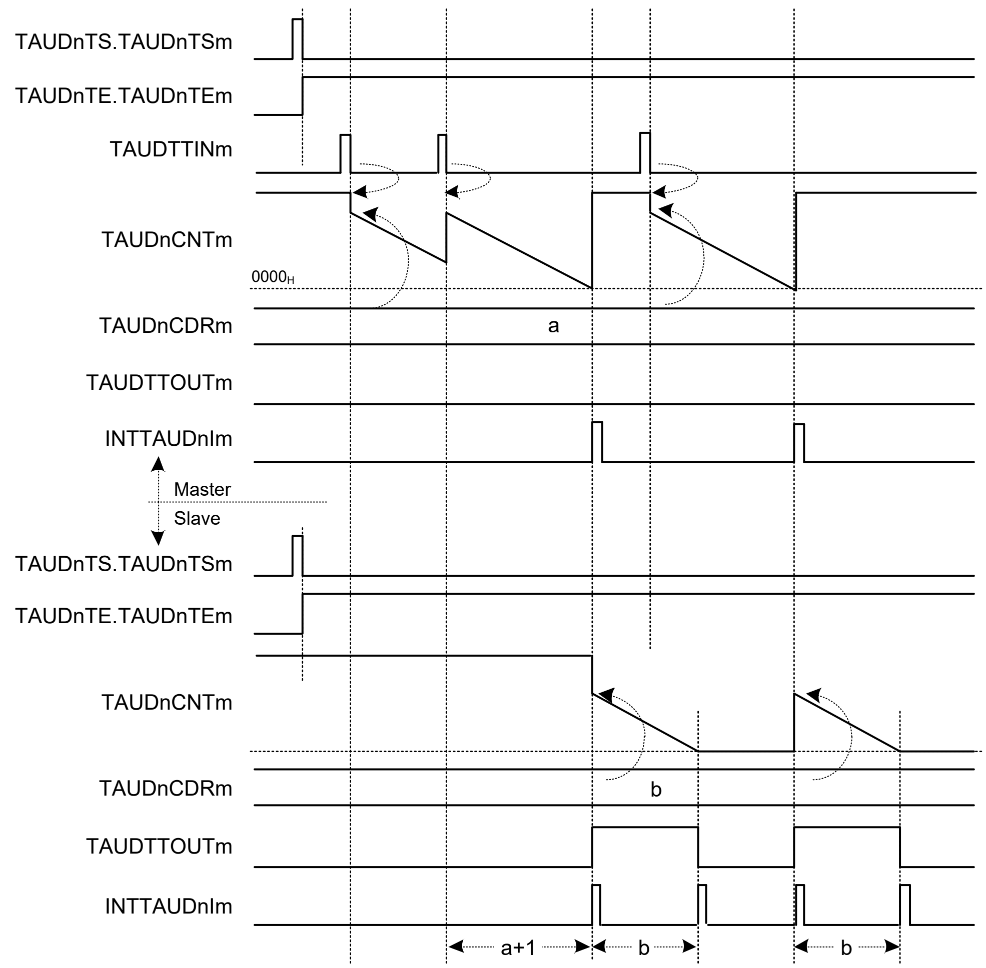


Figure 20.97 TAUDCMORm.TAUDMD0 = 1

- If an effective TAUDTTINm input edge is detected while the counter of the master channel counts down, TAUDCNTm reloads the value of TAUDCDRm. The counter restarts to count down. This means the delay is extended by the value of TAUDCNTm at the time when an effective TAUDTTINm input edge is detected.

(d) 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. (TAUDCMORm.TAUDMD0 = 0)
- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

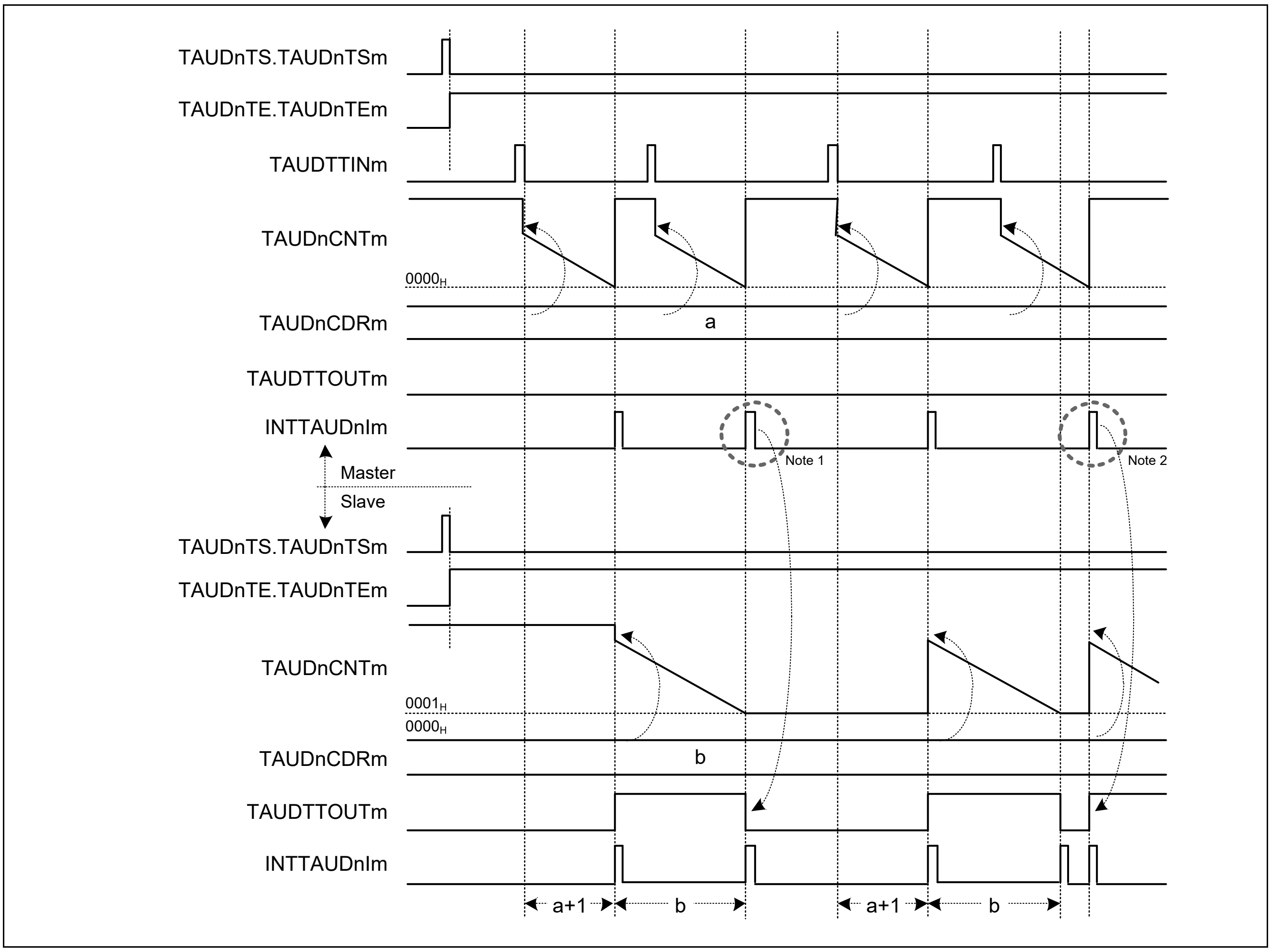


Figure 20.98 TAUDTTINm input interval ≤ Delay Time + Pulse Width +1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001H or exactly when 0001H 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 TAUDCDRm (slave) is reloaded. An interrupt is generated and TAUDTTOUTm toggles. If TAUDCNTm (master) has started to count down while the TAUDCNTm (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.

20.15.3 Trigger Start PWM Output

(1) Overview

(a) 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 is identical to PWM Output except that the master channel of this function can be reset by an effective TAUDTTINm input edge.

(b) Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode (see Table 20.117, Contents of the TAUDCMORm Register for the Master Channel of the Trigger Start PWM Output).
- The operation mode of the slave channel must be set to One-Count Mode (see Table 20.120, Contents of the TAUDCMORm Register for the Slave Channel of the Trigger Start PWM Output).
- The channel output mode of the slave channel must be set to Synchronous Channel Output Mode 1 (see Section 20.7, Channel Output Modes).
- TAUDTTOUTm is not used with the master channel of this function.

(c) Functional description

The counters (master and slave) are enabled by setting the channel trigger bits (TAUDTS.TAUDTSm) to 1. This in turn sets TAUDTE.TAUDTEm, enabling count operation. The current value of TAUDCDRm is loaded to TAUDCNTm, and the counter starts to count down from this value. INTTAUDIm is generated on the master channel, and a PWM output is realized by setting and resetting TAUDTTOUTm (slave).

- Master channel:
  The current value of TAUDCDRm is loaded to the counter (TAUDCNTm), INTTAUDIm is generated and the counter starts to count down from this value.
  When the counter reaches 0000H and the pulse cycle time has elapsed, INTTAUDIm is generated and the counters (master and slave) reload the current TAUDCDRm values.
  If an effective TAUDTTINm input edge is detected, the counter of the master channel reloads the current TAUDCDRm 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 TAUDCDRm. The TAUDTTOUTm signal is set to the active level.
  When the counter reaches 0000H (duty time has elapsed), INTTAUDIm is generated and the TAUDTTOUTm signal is reset. The counter returns to FFFFH and awaits the next INTTAUDIm of the master channel.

The counter can be stopped by setting TAUDTT.TAUDTTm to 1 for the master and slave channel, which in turn sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channel stop but retain their values. The counters can be restarted by setting TAUDTS.TAUDTSm to 1.

(d) Conditions

Simultaneous reloading can be used with this function. See Section 20.6, Simultaneous Reloading.

(2) Equations

Pulse cycle = (TAUDCDRm (master) + 1) × count clock cycle

Duty cycle [%] = [TAUDCDRm (slave) / (TAUDCDRm (master) + 1)] × 100

- Duty cycle = 0%
  TAUDCDRm (slave) = 0000H
- Duty cycle = 100%
  TAUDCDRm (slave) ≥ TAUDCDRm (master) + 1

(3) Block Diagram and General Timing Diagram

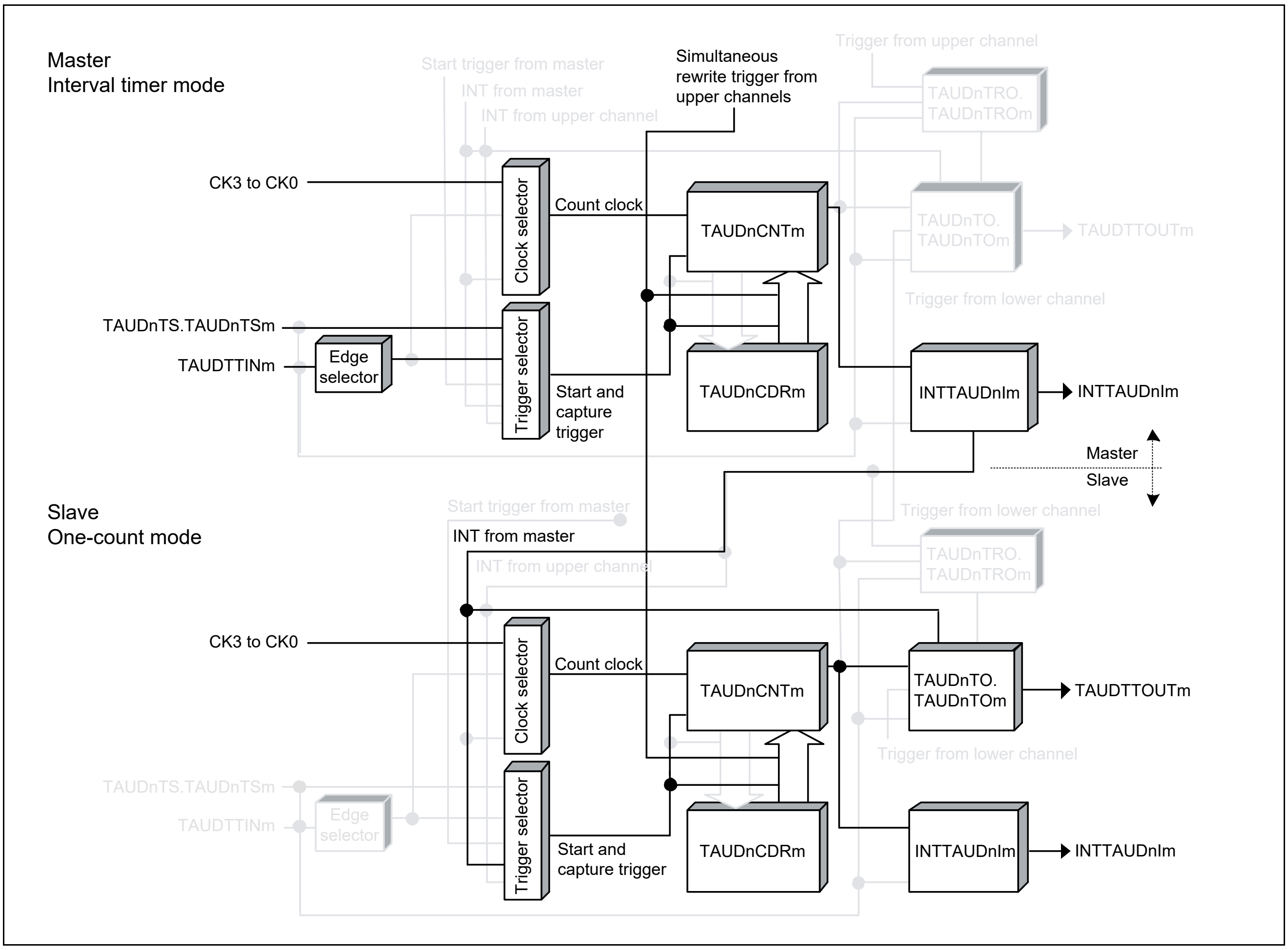


Figure 20.99 Block Diagram for Trigger Start PWM Output

The following settings apply to the general timing diagram.

- Detection of rising edges (TAUDCMURm.TAUDTIS[1:0] = 01B)
- Positive logic (TAUDTOL.TAUDTOLm (slave) = 0)

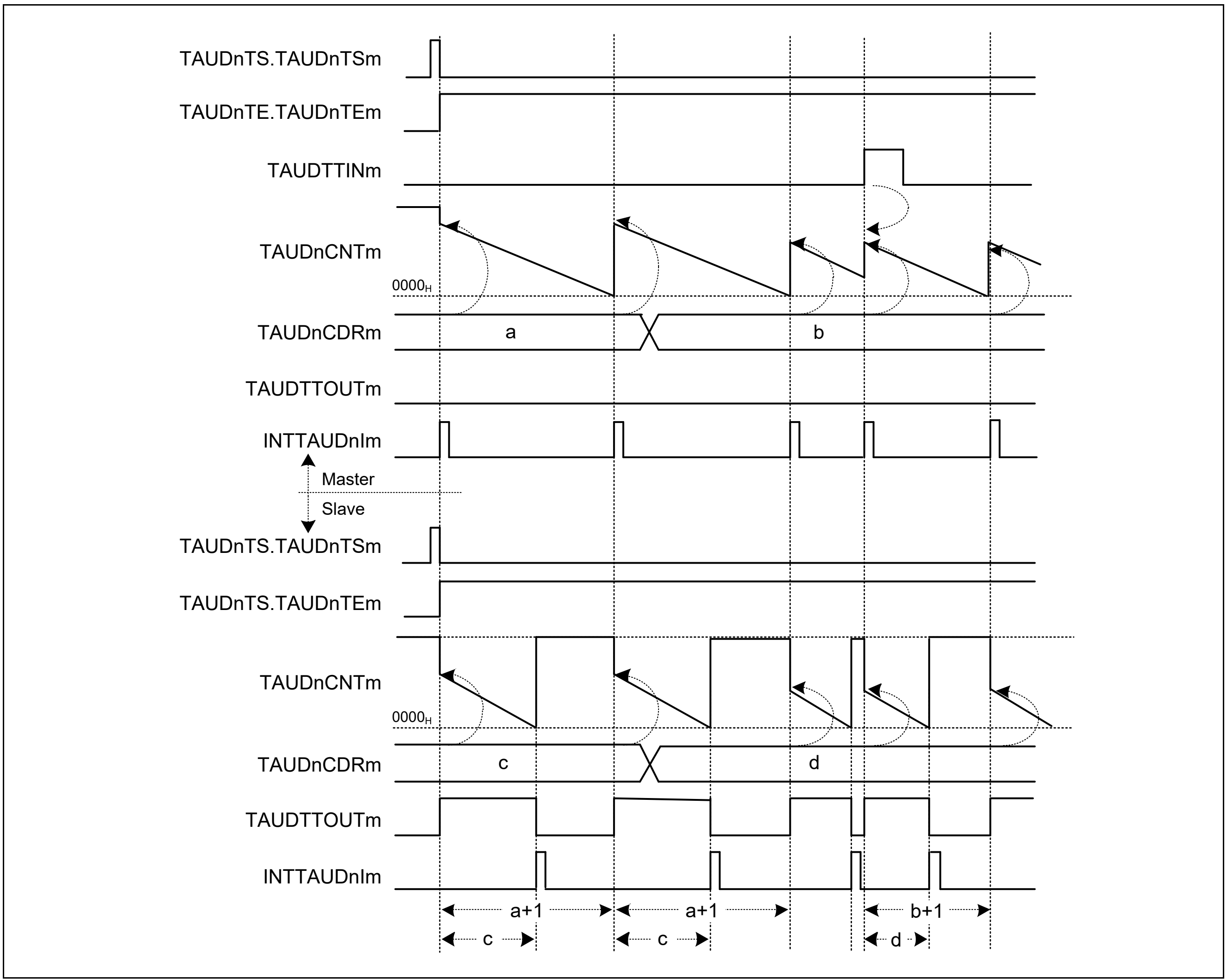


Figure 20.100 General Timing Diagram for Trigger Start PWM Output

Remark: TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDIm of the master channel.

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.117 Contents of the TAUDCMORm Register for the Master Channel of the Trigger Start PWM Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as the start trigger
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.118 Contents of the TAUDCMURm Register for the Master Channel of the Trigger Start PWM Output

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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode

The channel output mode is not used by this function.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.119 Simultaneous Reload Settings for the Master Channel of the Trigger Start PWM Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(5) Register Settings for Slave Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS[1:0]		TAUDCCS[1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS[1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.120 Contents of the TAUDCMORm Register for the Slave Channel of the Trigger Start PWM Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	100: INTTAUDIm of master channel is a start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Start trigger during operation is valid. The value of the TAUDMD[0] bit of the master and slave channel must be identical.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.121 Contents of the TAUDCMURm Register for the Slave Channel of the Trigger Start PWM Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.122 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel operation
TAUDTOC.TAUDTOCm	0: Operating mode 1
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.123 Simultaneous Reload Settings for the Slave Channel of the Trigger Start PWM Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(6) Operating Procedure for Trigger Start PWM Output

Table 20.124 Operating Procedure for Trigger Start PWM Output

		Operation	TAUD Status
	Initial Channel Setting	Master channel: Set TAUDCMORm/ TAUDCMURm register and the channel output mode as described in 20.15.3(4), Register Settings for the Master Channel. Slave channel: Set TAUDCMORm/ TAUDCMURm register and the channel output mode as described in 20.15.3(5), Register Settings for Slave Channels. Set the value of TAUDCDRm register of every channel.	Channel operation is stopped.
Restart	Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDIm is generated on the master channel.
Restart	During Operation	TAUDCDRm can be changed at any time. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	TAUDCNTm of master channel loads TAUDCDRm value and counts down. When the counter reaches 0000H: • INTTAUDIm (master) is generated. • TAUDCDRm value is loaded into TAUDCNTm (master) to continue count operation. • TAUDCNTm (slave) reloads the TAUDCDRm value and starts to count down • TAUDTTOUTm (slave) is set When TAUDCNTm of the slave = 0000H: • INTTAUDIm (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: • TAUDCNTm (master and slave) reloads the TAUDCDRm value and counts down • INTTAUDIm (master) is generated. • TAUDTTOUTm (slave) is set to the active level.
Restart	Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

(7) Specific Timing Diagrams

(a) Duty cycle = 0%

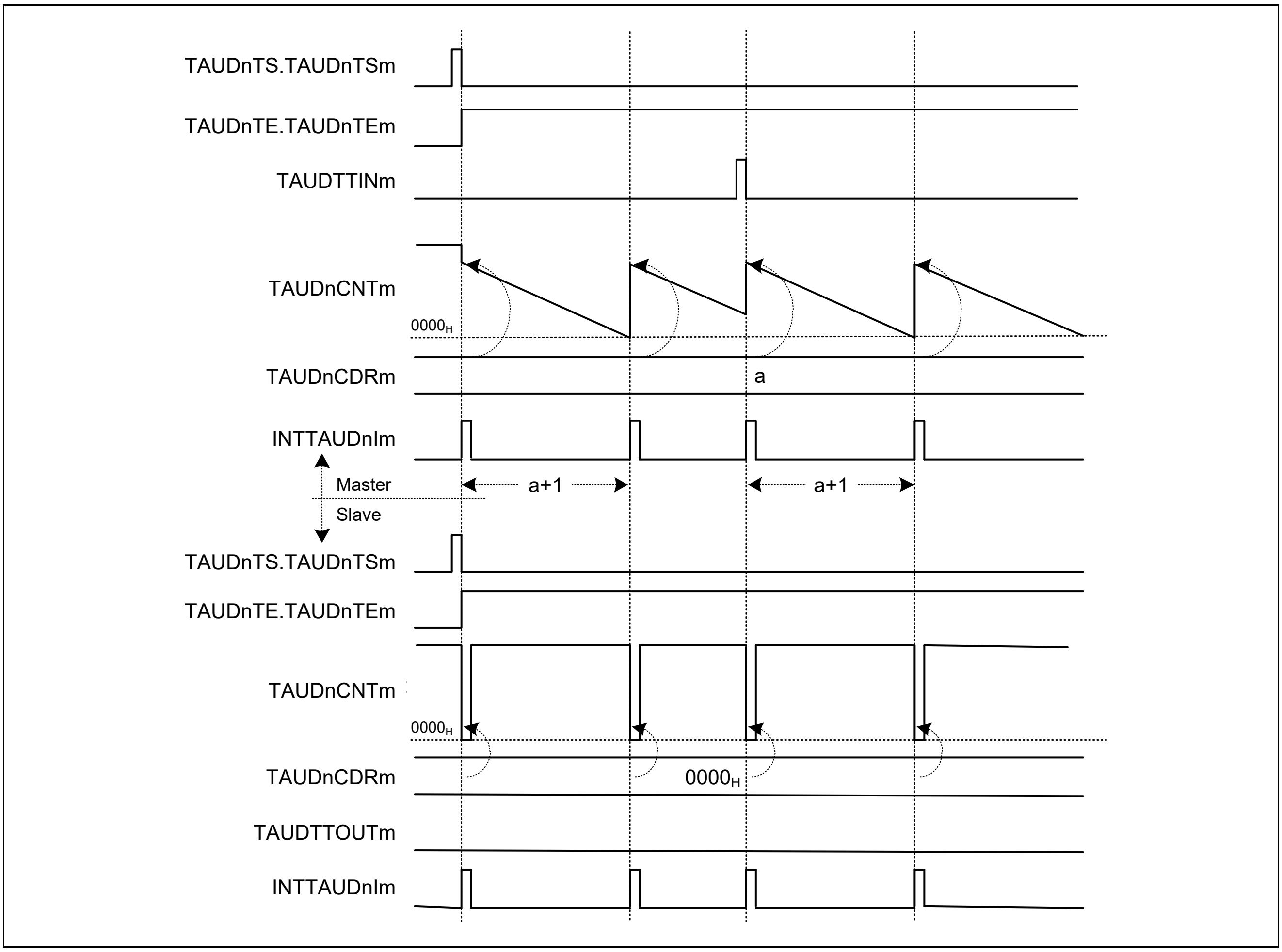


Figure 20.101 TAUDCDRm (Slave) = 0000H, Positive Logic (TAUDTOL.TAUDTOLm (Slave) = 0) Detection of Falling Edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

- Every time the master channel generates an interrupt (INTTAUDIm), 0000H is written to TAUDCNTm (slave). Therefore, TAUDCNTm (slave) cannot start to count and TAUDTTOUTm remains inactive.
- TAUDCNTm (slave) generates an interrupt every time the value of TAUDCDRm is reloaded. The detection of an effective TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

(b) Duty cycle = 100%

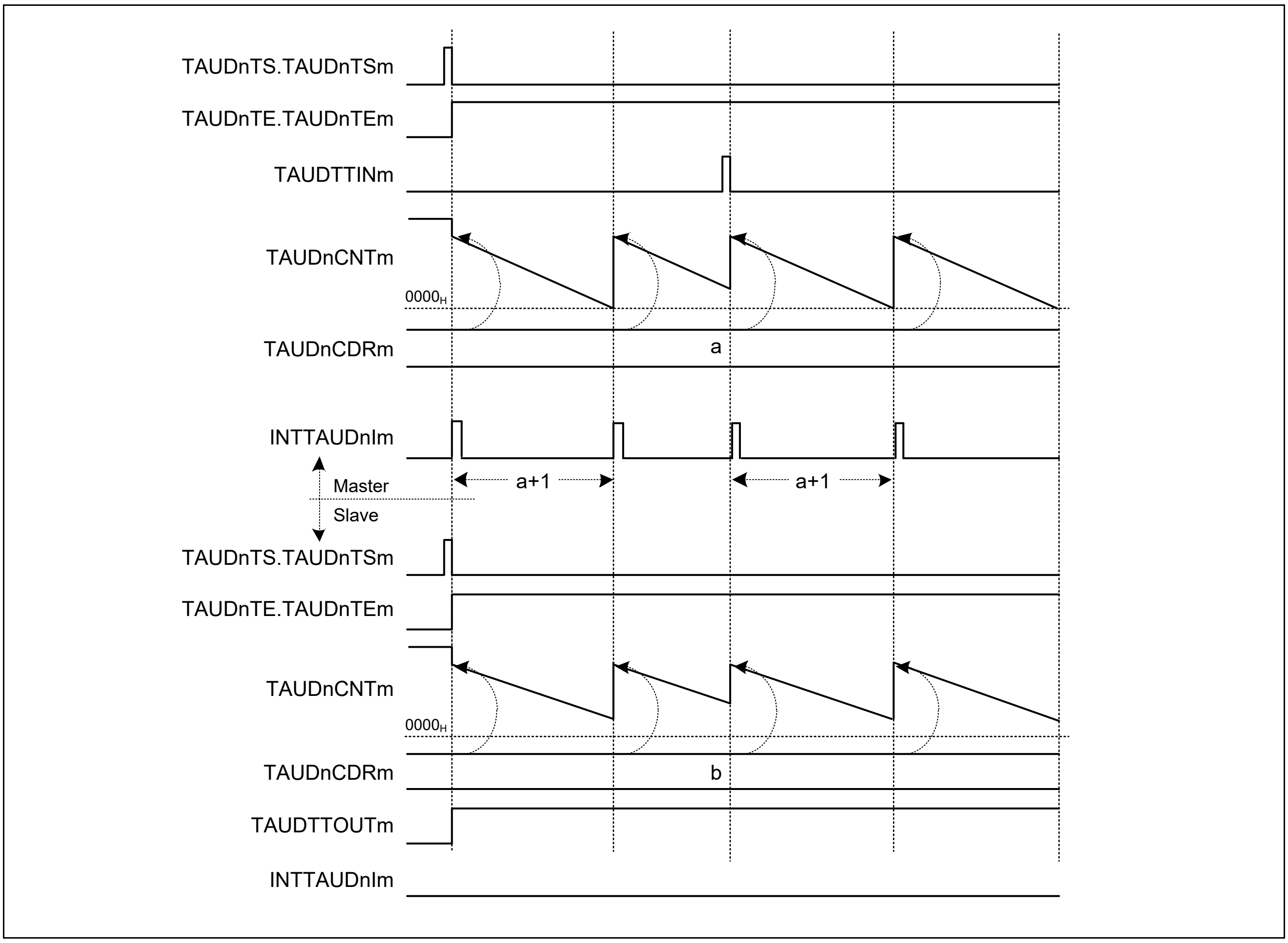


Figure 20.102 TAUDCDRm (Slave) ≥ TAUDCDRm (Master) + 1, Positive Logic (TAUDTOL.TAUDTOLm (Slave) = 0) Falling Edge Detection (TAUDCMURm.TIS[1:0] = 00B)

- If the value TAUDCDRm (slave) is higher than the value TAUDCDRm (master), the counter of the slave channel cannot reach 0000H and cannot generate interrupts.
  The TAUDTTOUTm remains at active state.
  The detection of an effective TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

(c) TAUDTTINm detection and active slave counter

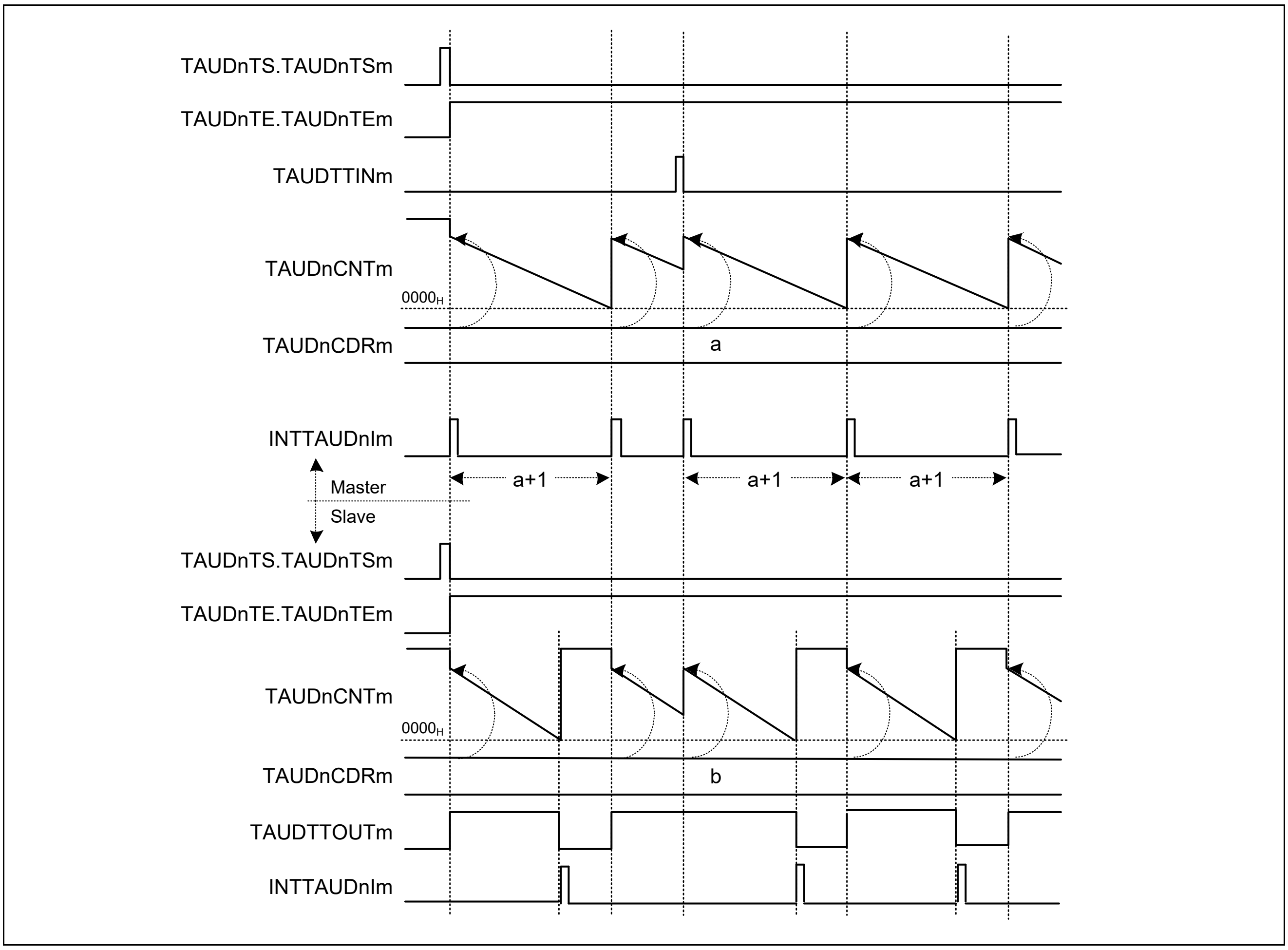


Figure 20.103 Positive Logic (TAUDTOL.TAUDTOLm (Slave) = 0) Detection of Falling Edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

- If TAUDCNTm (slave) reloads the value TAUDCDRm (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.

20.15.4 Delay Pulse Output

(1) Overview

(a) 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.

(b) Prerequisites

- Four channels
- The operating mode for the master channel should be set to interval timer mode. (See Table 20.125, Contents of the TAUDCMORm Register for the Master Channel of the Delay Pulse Output.)
- The operating mode for slave channels 1 and 2 should be set to one-count mode. (See Table 20.128, Contents of the TAUDCMORm Register for Slave Channel 1 of the Delay Pulse Output.)
- The operating mode for slave channel 3 should be set to pulse one-count mode. (See Table 20.132, Contents of the TAUDCMORm Register for Slave Channel 2 of the Delay Pulse Output.)
- 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 20.7, Channel Output Modes.)
- The channel output mode for slave channel 3 should be set to independent channel output mode 2. (See Section 20.7, Channel Output Modes.)

(c) Functional description

The counters of the channel group are enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm to 1, enabling count operation.

- Master channel:

  The current value of TAUDCDRm is loaded into TAUDCNTm and the counter starts to count down from this value. INTTAUDIm is generated on the master channel.

  When the counter value of master channel reaches 0000H and pulse cycle time has elapsed, INTTAUDIm is generated. The TAUDCDRm 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 TAUDCDRm 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 0000H (duty time has elapsed), INTTAUDIm is generated and TAUDTTOUTm signal is reset. The counter is reset to FFFFH and waits for the next INTTAUDIm of master channel.
  - Slave channel 2:

    When the counter of slave channel 2 reaches 0000H and delay time has elapsed, INTTAUDIm is generated. The counter is reset to FFFFH and waits for the next INTTAUDIm of master channel.
    Generating INTTAUDIm (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 TAUDCDRm. INTTAUDIm is generated and the TAUDTTOUTm signal (slave channel 3) is set. When the counter of slave channel 3 reaches 0001H, INTTAUDIm 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 TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDTS.TAUDTSm to 1.

(d) Conditions

Simultaneous reloading can be used with this function. See Section 20.6, Simultaneous Reloading.

(2) Equations

Pulse cycle = (TAUDCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUDCDRm (slave 1)) × count clock cycle

Delay width = (TAUDCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUDCDRm (slave 3)) × count clock cycle

However, the delay width shall be set within the following range:

0000H ≤ TAUDCDRm (slave 2) < TAUDCDRm (master)

Remarks 1. The output waveform of TAUDTTOUTm (slave 3) is delayed by the amount generated by slave 2 from the output waveform of TAUDTTOUTm (slave 1). It cannot be delayed longer than the pulse cycle.

2. If INTTAUD0Im of slave 2 is generated while slave 3 is counting, slave 3 restarts operation. Therefore, the output 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).

(3) Block Diagram and General Timing Diagram

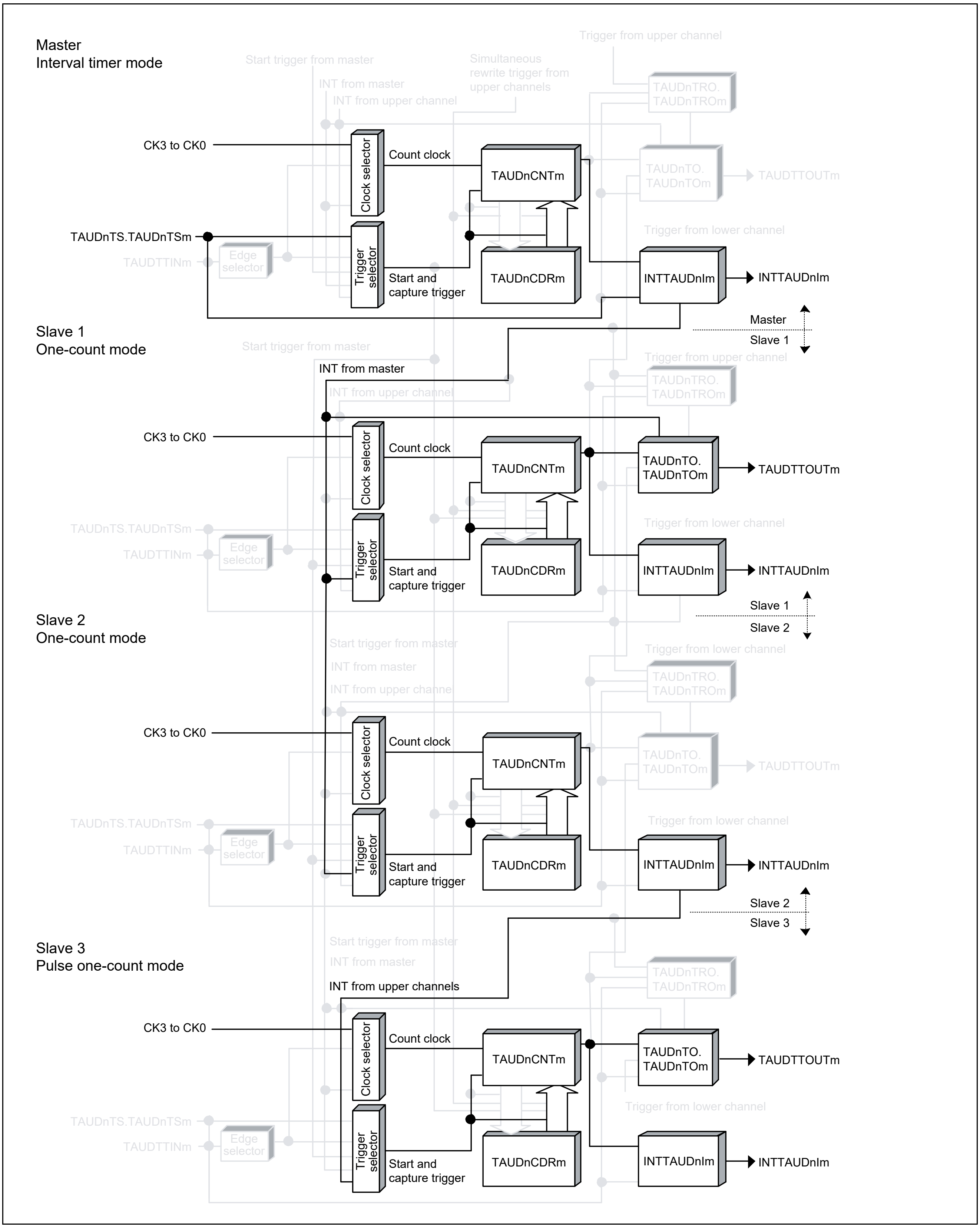


Figure 20.104 Block Diagram of Delay Pulse Output

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUDTOL.TAUDTOLm = 0)
- Slave channel 3: Positive logic (TAUDTOL.TAUDTOLm = 0)

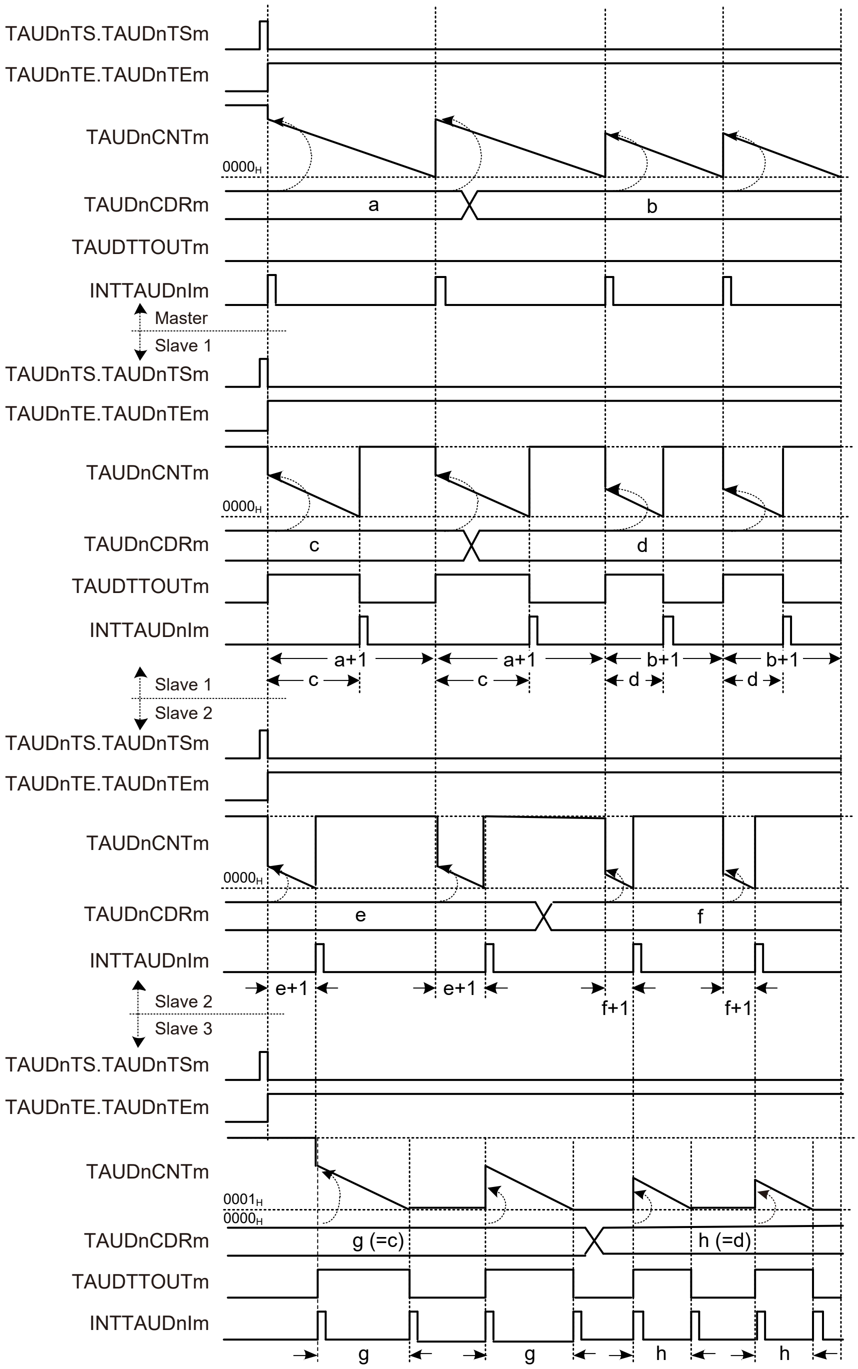


Figure 20.105 General Timing Diagram of Delay Pulse Output

Remark: TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDIm of the master channel.

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.125 Contents of the TAUDCMORm Register for the Master Channel of the Delay Pulse Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.126 Contents of the TAUDCMURm Register for the Master Channel of the Delay Pulse Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because channel output mode is not used for the master channel with this function.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.127 Simultaneous Reload Settings for the Master Channel of Delay Pulse Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Master channel is simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(5) Register Settings for Slave Channel 1

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.128 Contents of the TAUDCMORm Register for Slave Channel 1 of the Delay Pulse Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	100: INTTAUDIm of master channel is a start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Valid start trigger during operation

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.129 Contents of the TAUDCMURm Register for Slave Channel 1 of the Delay Pulse Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.130 Control Bit Settings for Slave Channel 1 in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel operation
TAUDTOC.TAUDTOCm	0: Operating mode 1
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.131 Simultaneous Reload Settings for Slave Channel 1 of Delay Pulse Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Master channel is simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(6) Register Settings for Slave Channel 2

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.132 Contents of the TAUDCMORm Register for Slave Channel 2 of the Delay Pulse Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	100: INTTAUDIm of master channel is a start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Valid start trigger during operation

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.133 Contents of the TAUDCMURm Register for Slave Channel 2 of the Delay Pulse Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because channel output mode is not used with this function.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings

Table 20.134 Simultaneous Reload Settings for Slave Channel 2 of Delay Pulse Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Master channel is simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(7) Register Settings for Slave Channel 3

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.135 Contents of the TAUDCMORm Register for Slave Channel 3 of the Delay Pulse Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	101: INTTAUDIm of upper channel (m - 1) is a start trigger regardless of master setting.
7, 6	TAUDCOS[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	TAUDMD[4:1]	1010: Pulse one-count mode
0	TAUDMD0	1: Valid start trigger during operation

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.136 Contents of the TAUDCMURm Register for Slave Channel 3 of the Delay Pulse Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.137 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.138 Simultaneous Reload Settings for Slave Channel 3 of Delay Pulse Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Master channel is simultaneous reload control channel.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(8) Operating Procedure for Delay Pulse Output

Table 20.139 Operating Procedure for Delay Pulse Output

(1/2)

	Operation	TAUD Status
Initial Channel Setting	Master channel: Set TAUDCMORm/ TAUDCMURm register and the channel output mode as described in 20.15.4(4) Register Settings for the Master Channel. Slave channel 1: Set TAUDCMORm and TAUDCMURm registers and the channel output mode as described in 20.15.4(5) Register Settings for Slave Channel 1. Slave channel 2: Set TAUDCMORm and TAUDCMURm registers and the channel output mode as described in 20.15.4(6) Register Settings for Slave Channel 2. Slave channel 3: Set the TAUDCMORm and TAUDCMURm registers and the channel output mode as described in 20.15.4(7) Register Settings for Slave Channel 3. Set the value of TAUDCDRm register of every channel.	Channel operation is stopped.

(2/2)

	Operation	TAUD Status
Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm (master and slave channels) is set to 1 and the counters of master channel and slave channels 1 and 2 start. INTTAUDIm is generated on the master channel and TAUDTTOUTm (slave channel 1) is set.
During Operation	TAUDCDRm can be changed at any time. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	TAUDCNTm of master channel and slave channels 1 and 2 load TAUDCDRm value and count down. When the counter of master channel reaches 0000H: • INTTAUDIm (master) is generated. • TAUDCDRm value is reloaded into TAUDCNTm (master) to continue count operation. • TAUDCDRm value is reloaded into TAUDCNTm (slave 1/2) to count down. • TAUDTTOUTm (slave 1) is set. When TAUDCNTm (slave 1) reaches 0000H: • INTTAUDIm (slave 1) is generated. • TAUDTTOUTm (slave 1) is reset. When TAUDCNTm (slave 2) reaches 0000H: • INTTAUDIm (slave 2) is generated. • INTTAUDIm (slave 3) is generated. • TAUDTTOUTm (slave 3) is set. • TAUDCDRm value is reloaded into TAUDCNTm (slave 3) to count down operation. When TAUDCNTm (slave 3) reaches 0001H: • INTTAUDIm (slave 3) is generated. • TAUDTTOUTm (slave 3) is reset.
Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

Restart (Stop Operation → Start Operation)

(9) Specific Timing Diagrams

(a) Duty cycle (slave 3) = 100%

The following values apply to 0:

- TAUDCDRm (master) = 000AH
- TAUDCDRm (slave 1) = 000BH
- TAUDCDRm (slave 2) = 0000H
- TAUDCDRm (slave 3) = 000BH

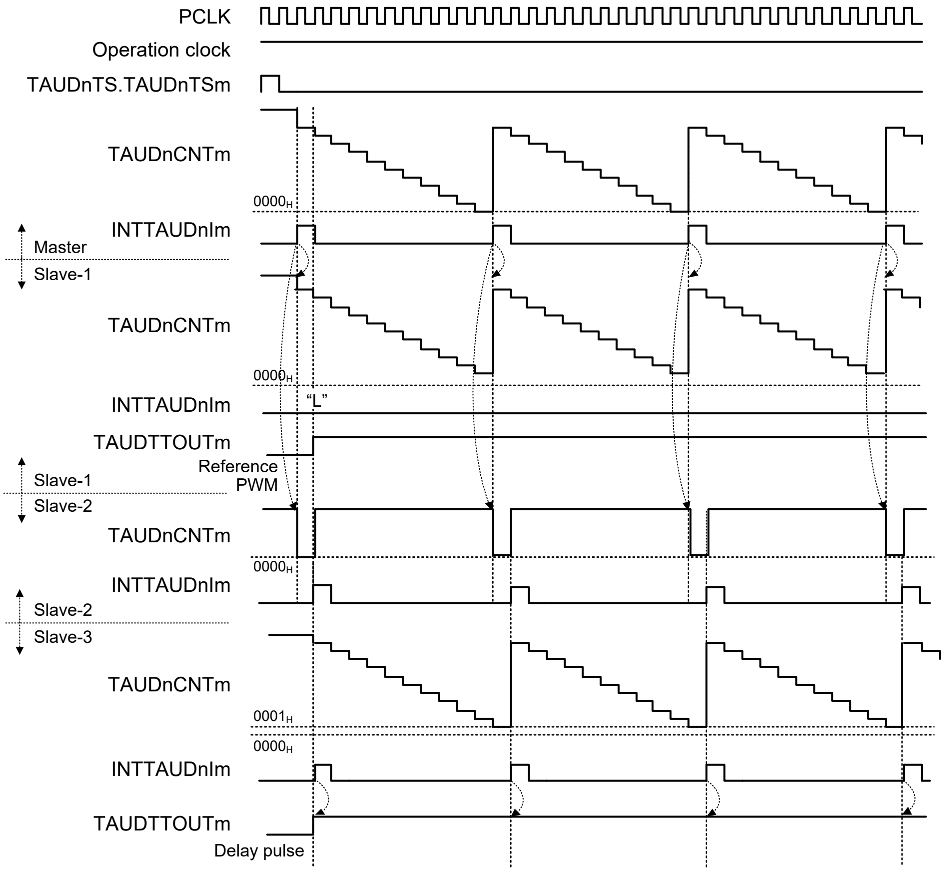


Figure 20.106 Duty Cycle (Slave 3) = 100%

- If the value of TAUDCDRm (slave 1 and 3) is higher than the value of TAUDCDRm (master), the counter of the slave channel 1 cannot reach 0000H and cannot generate interrupts.
  TAUDTTOUTm of channels 1 and 3 remain in the active state.

(b) TAUDTTOUTm (slave 1) = TAUDTTOUTm (slave 3)

The following values apply to Figure 20.107.

- TAUDCDRm (master) = 000AH
- TAUDCDRm (slave 1) = 0005H
- TAUDCDRm (slave 2) = 0000H
- TAUDCDRm (slave 3) = 0005H

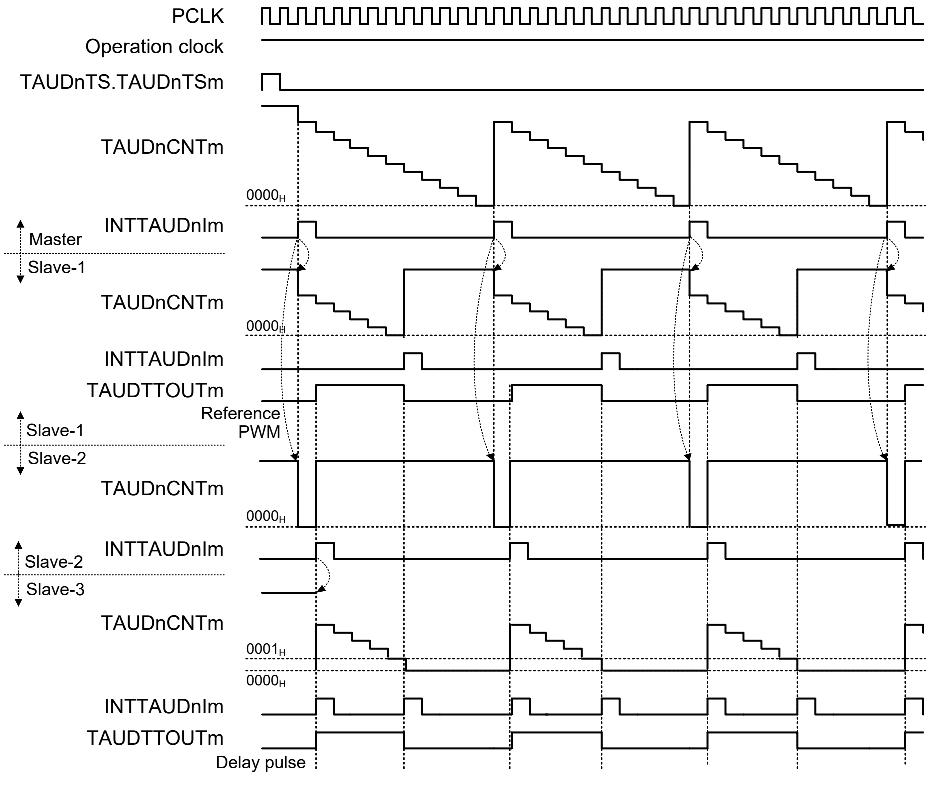


Figure 20.107 TAUDTTOUTm (Slave 1) = TAUDTTOUTm (Slave 3)

- If TAUDCDRm (slave 2) = 0000H, 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.

20.15.5 Offset Trigger Output

(1) Overview

(a) 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 in response to the detection of an effective input edge of master channel. The pulse width is specified on the slave channel.

(b) Prerequisites

- Two channels
- The operating mode for the master channel should be set to capture mode. (See Table 20.140, Contents of the TAUDCMORm Register for the Master Channel of the Offset Trigger Output.
- The operating mode for slave channels should be set to one-count mode. (See Table 20.143, Contents of the TAUDCMORm Register for the Slave Channel of the Offset Trigger Output.
- The output mode for slave channels should be set to synchronous channel output mode 1. (See Section 20.7, Channel Output Modes.
- TAUDTTOUTm is not used with the master channel of this function.

(c) Functional description

The counter can be started by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling the counter to count up. The master channel counter (TAUDCNTm) starts to count up from 0000H.

- Master channel:

  When an effective TAUDTTINm input edge is detected, the current value of the counter (TAUDCNTm) is loaded into the data register of master channel (TAUDCDRm).

  INTTAUDIm is generated and the counter restarts to count up from 0000H.

- Slave channel:

  The INTTAUDIm of master channel sets the TAUDTTOUTm (slave) signal and triggers the counter of the slave channel. The current value of TAUDCDRm (slave) is loaded into TAUDCNTm (slave) and the counter starts to count down from this value.

  When the counter reaches 0000H (duty time has elapsed), INTTAUDIm is generated and TAUDTTOUTm signal is reset. The counter returns to FFFFH and awaits the next INTTAUDIm of the master channel.

The counter can be stopped by setting TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDTS.TAUDTSm to 1.

(2) Equations

Pulse width = (TAUDCDRm (slave) + 1) × count clock cycle

Duty cycle [%] = [TAUDCDRm (slave)/(TAUDCDRm (master) + 1)] × 100

- Duty cycle = 0%

  TAUDCDRm (slave) = 0000H

- Duty cycle = 100%

  TAUDCDRm (slave) ≥ TAUDCDRm (master) + 1

(3) Block Diagram and General Timing Diagram

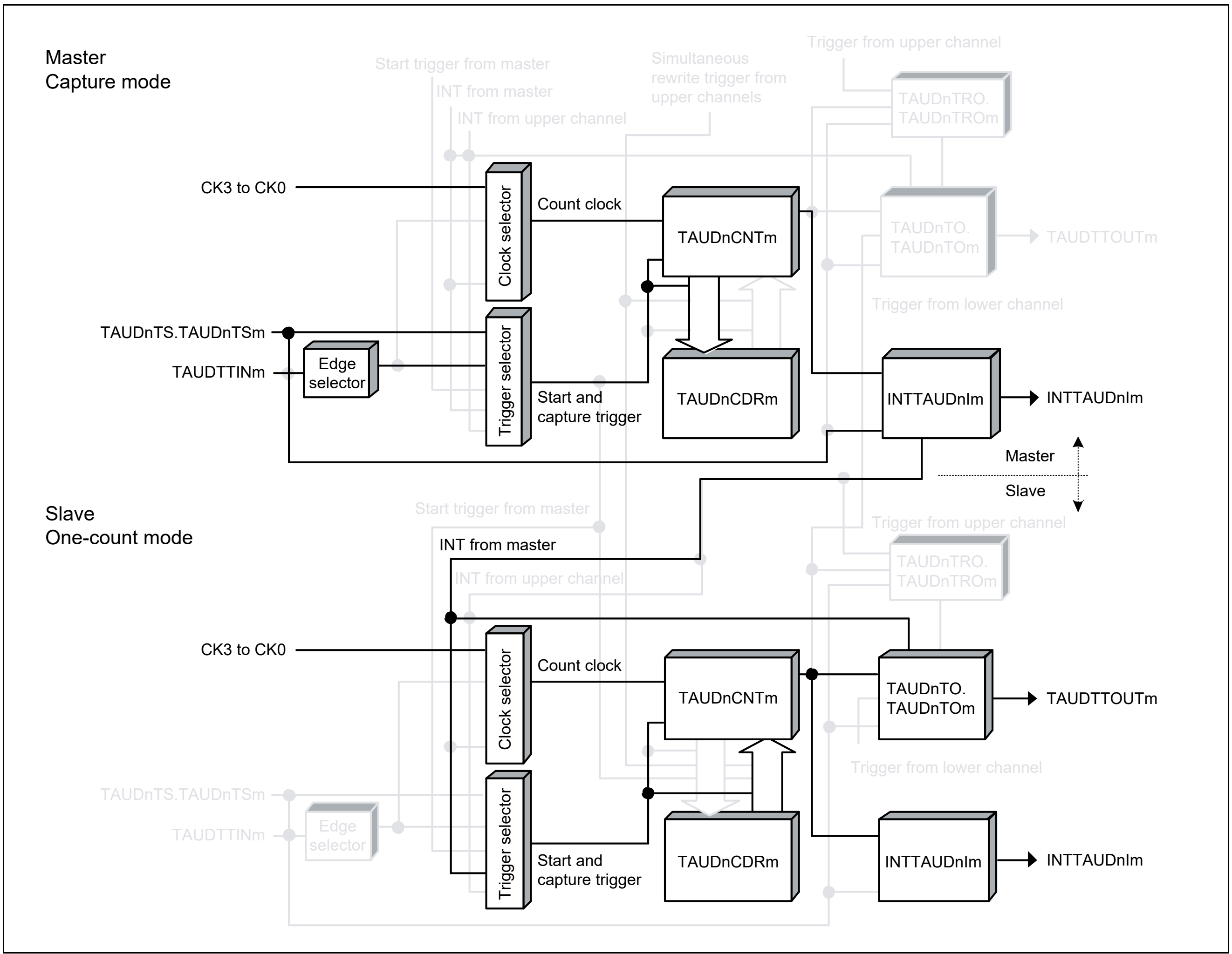


Figure 20.108 Block Diagram of Offset Trigger Output

The following settings apply to the general timing diagram.

- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

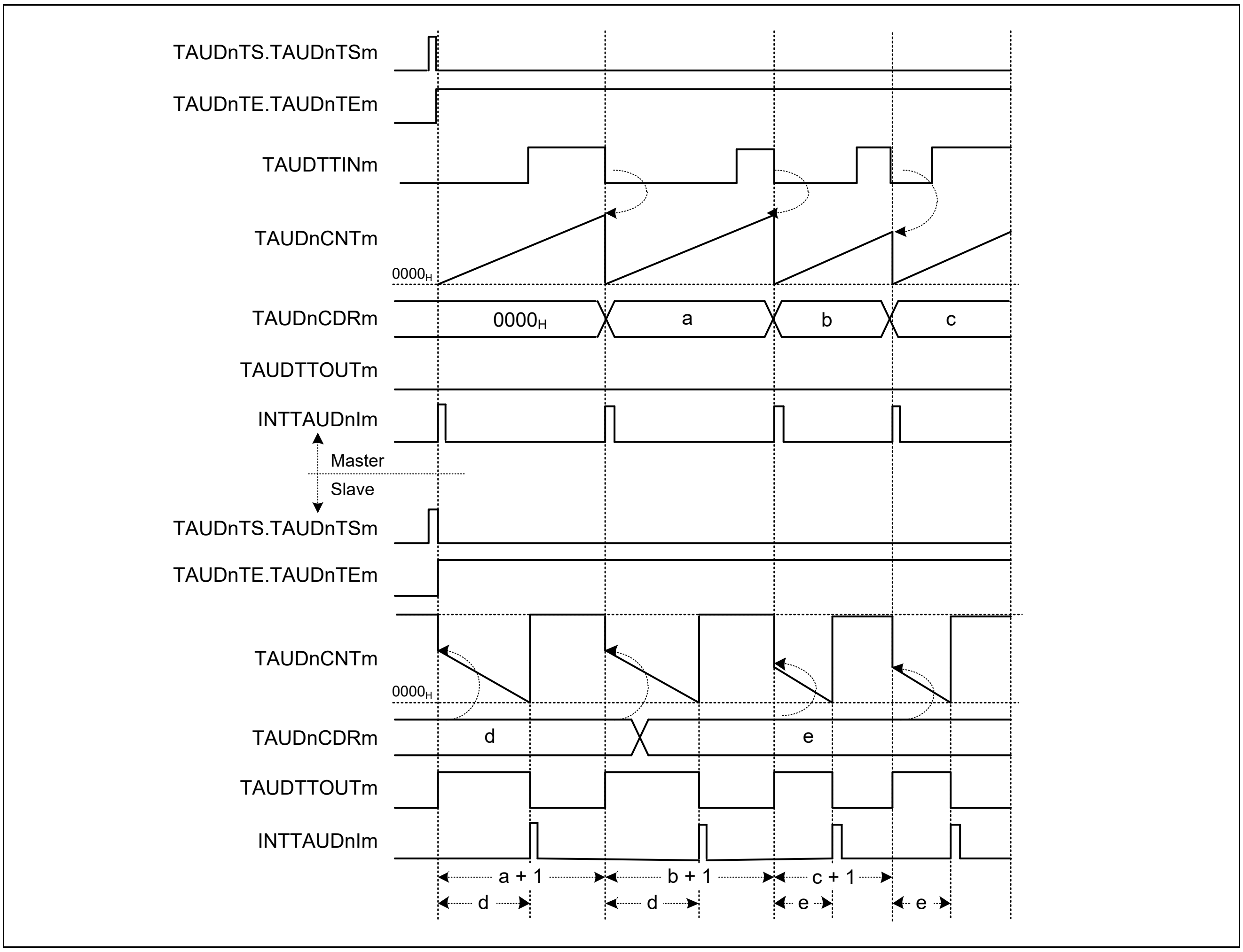


Figure 20.109 General Timing Diagram of Offset Trigger Output

Remark: TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDIm of the master channel.

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.140 Contents of the TAUDCMORm Register for the Master Channel of the Offset Trigger Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	001: Effective edge of the TAUDTTINm input signal is used as the start trigger.
7, 6	TAUDCOS[1:0]	11: Capture register is updated upon detection of an effective TAUDTTINm input edge or when a counter overflow occurs: • Detection of effective TAUDTTINm input edge: The counter value is written into TAUDCDRm. • Occurrence of overflow: FFFFH is written into TAUDCDRm. A effective TAUDTTINm input edge to be detected next will be ignored. TAUDCSRm.TAUDOVF is set when a counter overflow occurs, and cleared by setting TAUDCSCm.TAUDCLOV = 1.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUDMD[4:1]	0010: Capture mode
0	TAUDMD0	1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.141 Contents of the TAUDCMURm Register for the Master Channel of the Offset Trigger Output

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	TAUDTIS[1:0]	00: Detection of falling edges 01: Detection of rising edges 10: Detection of rising and falling edges 11: Setting prohibited

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because channel output mode is not used with this function.

(d) Simultaneous reloading

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 20.142 Simultaneous Reload Settings for the Master Channel of Offset Trigger Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading
TAUDRDS.TAUDRDSm	When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(5) Register Settings for Slave Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.143 Contents of the TAUDCMORm Register for the Slave Channel of the Offset Trigger Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	100: INTTAUDIm of master channel is a start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Enables start trigger detection while counting.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.144 Contents of the TAUDCMURm Register for the Slave Channel of the Offset Trigger Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode for slave channels

Table 20.145 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel operation
TAUDTOC.TAUDTOCm	0: Operating mode 1
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading of slave channels

Simultaneous reload registers (TAUDRDE, TAUDRDS, TAUDRDM, and TAUDRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 20.146 Simultaneous Reload Settings for Slave Channels of Offset Trigger Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading
TAUDRDS.TAUDRDSm	When simultaneous reloading is disabled (TAUDRDE.TAUDRDEm = 0), set these bits to 0
TAUDRDM.TAUDRDMm	
TAUDRDC.TAUDRDCm	

(6) Operating Procedure for Offset Trigger Output

Table 20.147 Operating Procedure for Offset Trigger Output

		Operation	TAUD Status
	Initial Channel Setting	Master channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in 20.15.5(4) Register Settings for the Master Channel. Slave channel: Set TAUDCMORm/TAUDCMURm register and channel output mode as described in 20.15.5(5) Register Settings for Slave Channels . The TAUDCDRm register of master channel functions as a capture register. Set the value of TAUDCDRm register of slave channel.	Channel operation is stopped.
Restart →	Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm (master and slave channels) is set to 1 and the counters of master and slave channels start: • TAUDCNTm (master) counts up. • TAUDCDRm value is loaded into TAUDCNTm (slave) to perform counting down. INTTAUDIm is generated on the master channel and TAUDTTOUTm (slave) is set.
	During Operation	TAUDCDRm can be changed at any time. TAUDCSCm.TAUDCLOV can be set to 1. TAUDCDRm of slave channel can be changed after the generation of INTTAUDIm (master). TAUDCNT.TAUDCNTm and TAUDCSRm can be read at any time.	When TAUDCNTm of the slave = 0000H: • INTTAUDIm (slave) is generated. • TAUDTTOUTm (slave) is reset, and the counter of slave channel stops. When TAUDTTINm input edge is detected on the master channel: • INTTAUDIm (master) is generated. • TAUDCNTm (master) is reset to 0000H and then continues count operation subsequently. • TAUDCDRm value is reloaded into TAUDCNTm (slave) to perform counting down. • TAUDTTOUTm (slave) is set.
	Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

(7) Specific Timing Diagrams

(a) Duty cycle = 0%

The following settings apply to this diagram.

- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

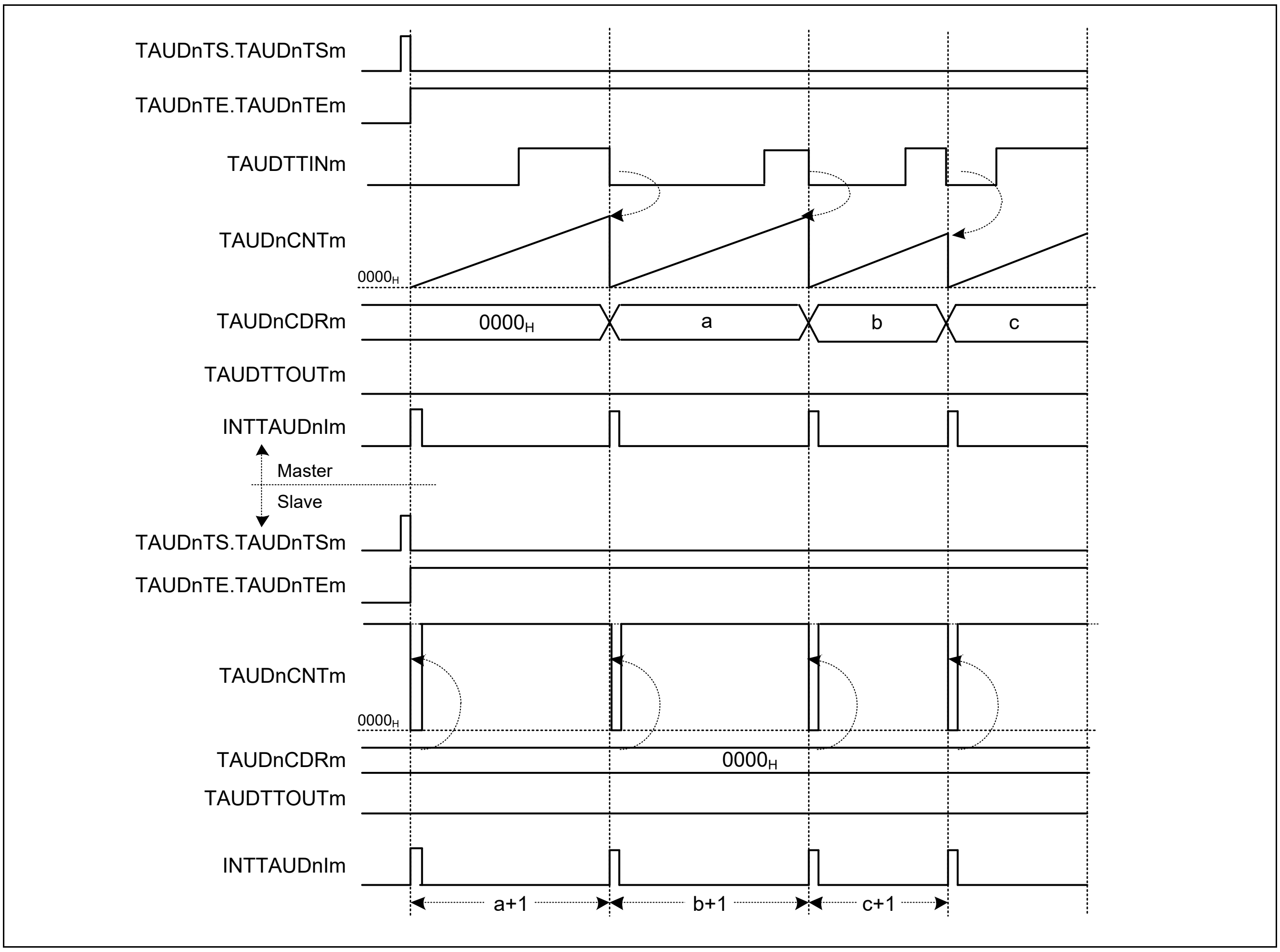


Figure 20.110 TAUDCDRm (Slave) = 0000H

- When TAUDCDRm (slave) = 0000H, 0000H is written to TAUDCNTm every time the master channel generates an interrupt (INTTAUDIm), and TAUDCNTm cannot start to count. The TAUDTTOUTm remains inactive.
- TAUDCNTm (slave) generates an interrupt every time the value of TAUDCDRm is reloaded. The slave and the master channels generate interrupts in the same cycle.

(b) Duty cycle = 100%

The following settings apply to this diagram.

- Detection of falling edges (TAUDCMURm.TAUDTIS[1:0] = 00B)

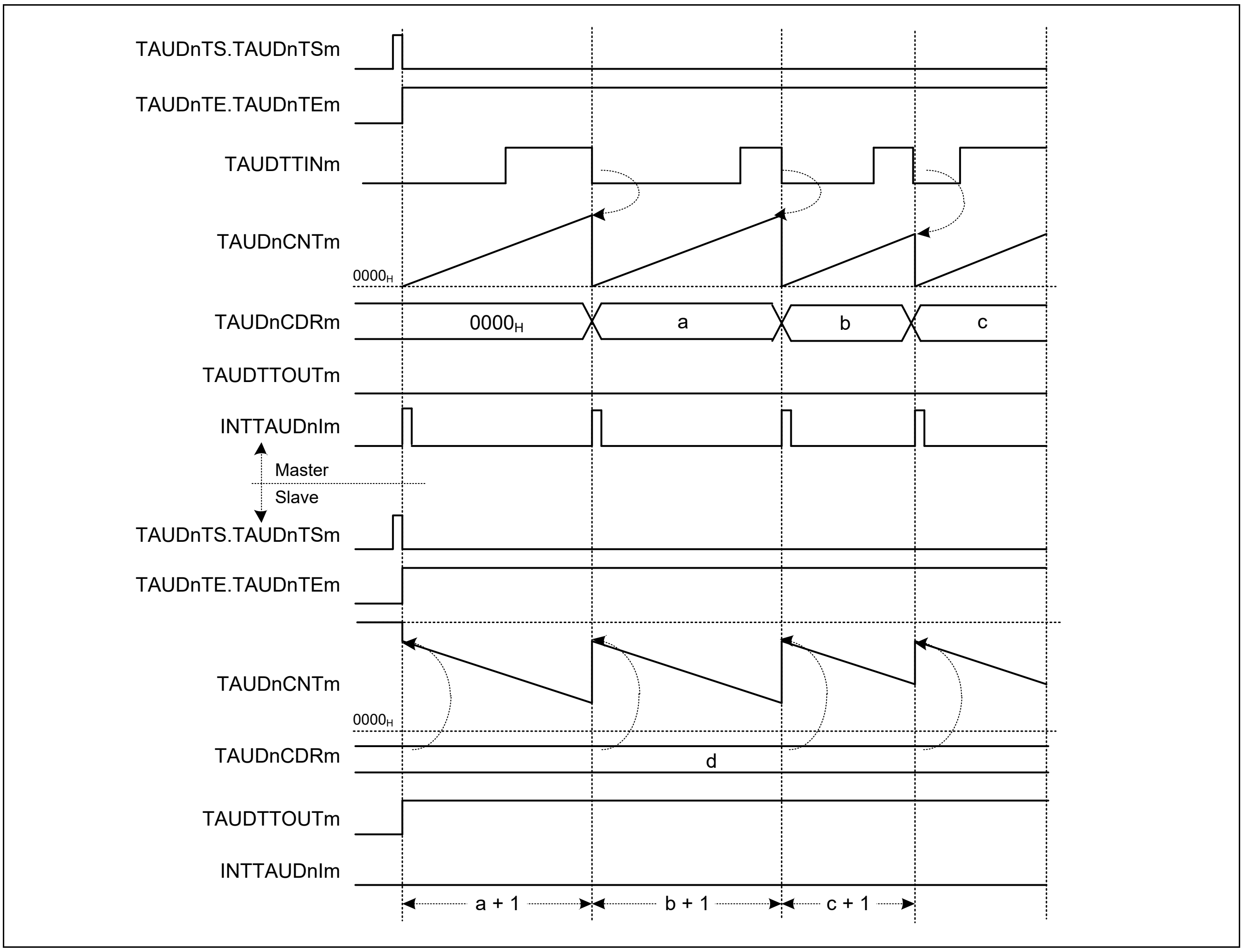


Figure 20.111 TAUDCDRm (Slave) ≥ TAUDCDRm (Master) + 1

- If the value TAUDCDRm (slave) is higher than the interval of effective input edges, the counter of the slave channel cannot reach 0000H and cannot generate interrupts. The TAUDTTOUTm remains at active state.

20.15.6 Triangle PWM Output

(1) Overview

(a) 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.

(b) Prerequisites

- Two channels
- The operating mode for the master channels should be set to interval timer mode. (See Table 20.148, Contents of the TAUDCMORm Register for the Master Channel of the Triangle PWM Output.)
- The operating mode for slave channels should be set to count-up/-down mode. (See Table 20.152, Contents of the TAUDCMORm Register for the Slave Channel of the Triangle PWM Output.)
- The channel output mode for the master channel should be set to independent channel output mode 1. (See Section 20.7, Channel Output Modes.)
- The channel output mode for slave channels should be set to synchronous channel output mode 2. (See Section 20.7, Channel Output Modes.)
- The following settings allows the TAUDTTOUTm signal to be at high level during the down status of a carrier cycle.
  - If TAUDCMORm.TAUDMD0 (master) bit is set to 0, TAUDTO.TAUDTOm should be set to 1 while TAUDTOE.TAUDTOEm is set to 0 (recommended setting).
  - If TAUDCMORm.TAUDMD0 (master) bit is set to 1, TAUDTO.TAUDTOm should be set to 0 while TAUDTOE.TAUDTOEm is set to 0.

(c) Functional description

The counters are enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1 for every channel. This in turn sets TAUDTE.TAUDTEm, enabling count operation. The current values of TAUDCDRm (master and slave) are loaded into TAUDCNTm (master and slave) and the counters start counting down from these values. When the TAUDCMORm.TAUDMD0 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 0000H (pulse cycle time has elapsed), INTTAUDIm is generated and the TAUDTTOUTm signal toggles. TAUDCNTm then reloads the TAUDCDRm value and counts down.
- Slave channel:
  The INTTAUDIm 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 TAUDCDRm value is reloaded and the counter starts to count down.
- When the counter of the slave channel reaches 0001H while counting up or down, INTTAUDIm is generated and the TAUDTTOUTm (slave) signal is set/reset.
  The counter continues count-up/-down and waits for the next INTTAUDIm of the master channel.
  Setting TAUDTOL.TAUDTOLm allows TAUDTTOUTm signal switching between normal phase and reverse phase during operation.

The counter can be stopped by setting TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm = 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but retain their values.

(d) Conditions

This function enables simultaneous reloading. See Section 20.6, Simultaneous Reloading.

(2) Equations

Pulse cycle = (TAUDCDRm (master) + 1) × count clock cycle
  $0000_H \leq$ TAUDCDRm (master) < $FFFF_H$

Carrier cycle (down/up) = (TAUDCDRm (master) + 1) × 2 × count clock cycle
Duty cycle 100 [%] = [(TAUDCDRm (master) + 1 – TAUDCDRm (slave))/(TAUDCDRm (master) + 1)] × 100

- Duty cycle = [%]
  TAUDCDRm (slave) = 0000H
- Duty cycle = 0%
  TAUDCDRm (slave) ≥ TAUDCDRm (master) + 1

(3) Block Diagram and General Timing Diagram

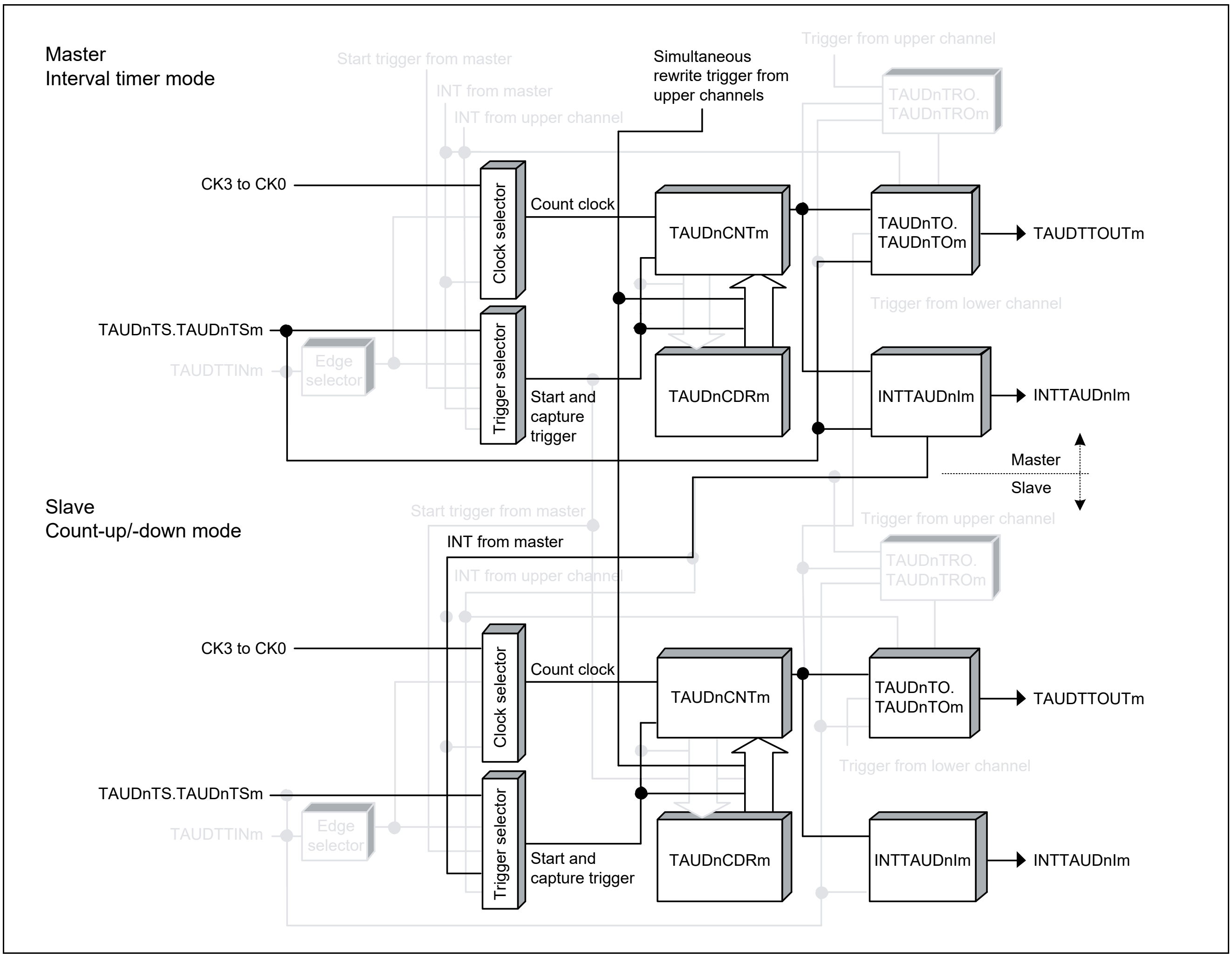


Figure 20.112 Block Diagram of Triangle PWM Output

The following settings apply to the general timing diagram.

- Master channel
  INTTAUDIm is generated at the beginning of operation.

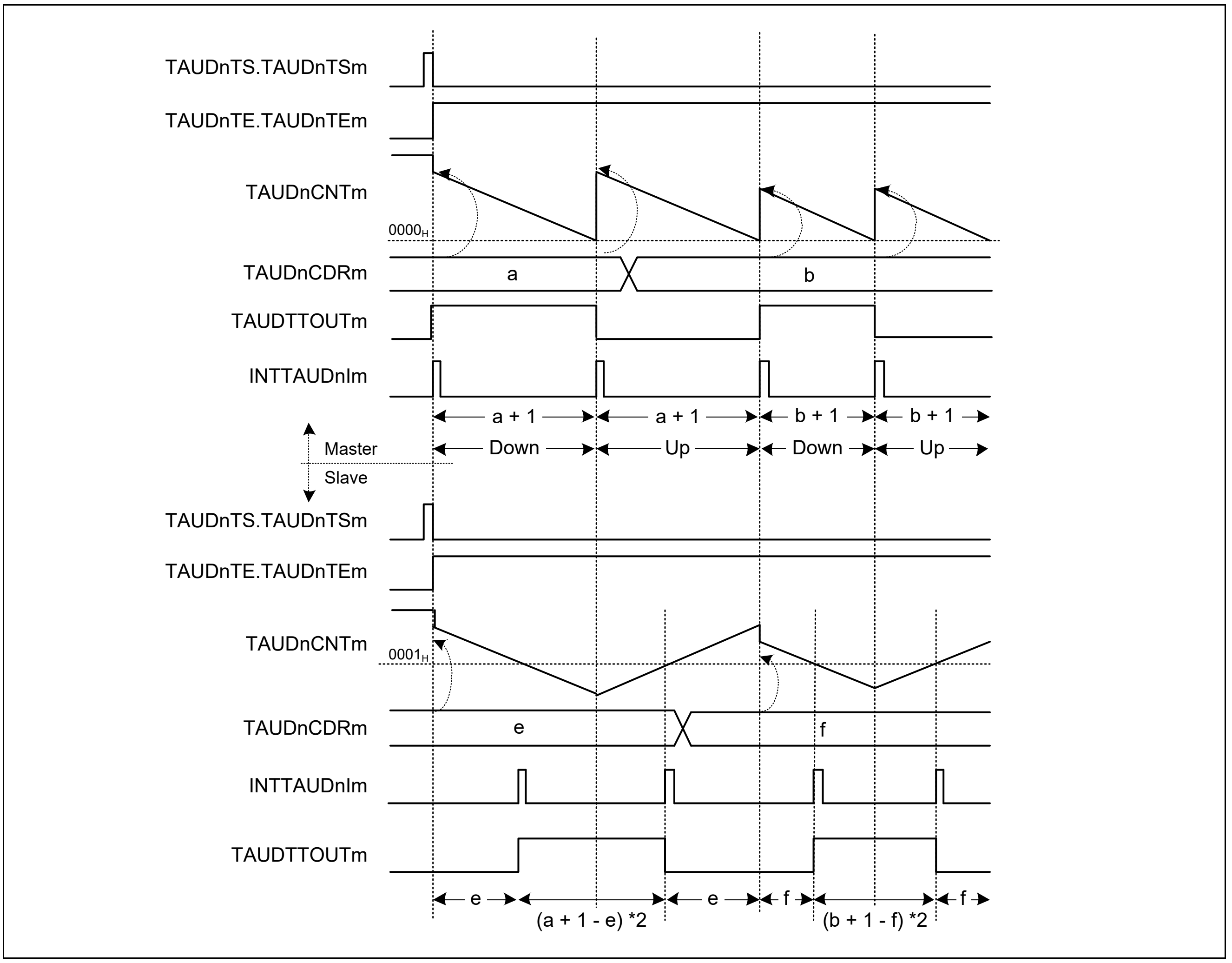


Figure 20.113 General Timing Diagram of Triangle PWM Output

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS[1:0]		TAUDCCS[1:0]		TAUDMAS	TAUDSTS[2:0]			TAUDCOS[1:0]		0	TAUDMD[4:1]				TAUDMD0

Table 20.148 Contents of the TAUDCMORm Register for the Master Channel of the Triangle PWM Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	When read, the value after reset is returned. When writing to this bit, write the value after reset.
0	TAUDMD0	0: INTTAUDIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation. 1: INTTAUDIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.149 Contents of the TAUDCMURm Register for the Master Channel of the Triangle PWM Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.150 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.151 Simultaneous Reload Settings for the Master Channel of Triangle PWM Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: A simultaneous reload trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

Remark: If TAUDRDS.TAUDRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous reload trigger signal.

(5) Register Settings for Slave Channels

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.152 Contents of the TAUDCMORm Register for the Slave Channel of the Triangle PWM Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDCOS[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	TAUDMD[4:1]	1001: Count-up/-down mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.153 Contents of the TAUDCMURm Register for the Slave Channel of the Triangle PWM Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.154 Control Bit Settings in Synchronous Channel Output Mode 2

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel operation
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.155 Simultaneous Reload Settings for Slave Channels of Triangle PWM Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channels. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: A simultaneous reload 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.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(6) Operating Procedure for Triangle PWM Output

Table 20.156 Operating Procedure for Triangle PWM Output

		Operation	TAUD Status
	Initial Channel Setting	Master channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.15.6(4), Register Settings for the Master Channel. Slave channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.15.6(5), Register Settings for Slave Channels. Set the value of TAUDCDRm register of every channel.	Channel operation is stopped.
Restart	Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDIm (master) is generated on the master channel if TAUDCMORm.TAUDMD0 is set to 1.
	During Operation	TAUDCDRm can be changed at any time. TAUDTOL.TAUDTOLm can be changed. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	TAUDCDRm value of master and slave channels is loaded into TAUDCNTm to count down. When the counter of master channel reaches 0000H: • INTTAUDIm (master) is generated. • TAUDTTOUTm (master) is toggled. • TAUDCDRm value is reloaded into TAUDCNTm (master) to continue count operation. • TAUDCDRm value is reloaded into TAUDCNTm (slave) or counting is started in opposite direction. When TAUDCNTm of slave channel reaches 0001H: • INTTAUDIm (slave) is generated. • TAUDTTOUTm (slave) is set in the count down status or reset in count-up status.
	Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

(7) Specific Timing Diagrams

(a) Duty cycle = 0%

The following settings apply to the Figure 20.114.

- Master channel:
  - INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 =1)
  - TAUDCDRm = a = 5H
- Slave channel:
  - TAUDCDRm = 6H

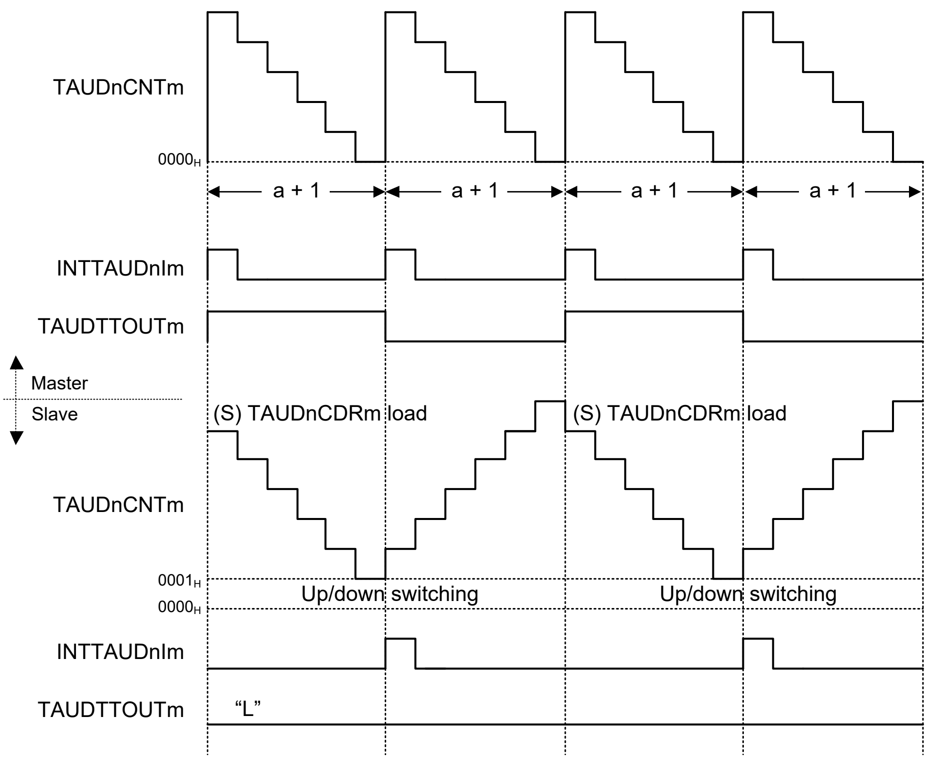


Figure 20.114 TAUDCDRm (Slave) ≥ TAUDCDRm (Master) + 1

- If TAUDCDRm (slave) value ≥ TAUDCDRm (master) value +1, the counter of slave channel does not reach 0001H while counting down. TAUDTTOUTm remains low because there is no set signal to be detected.

(b) Duty cycle = 100%

The following settings apply to the Figure 20.115.

- Master channel:
  - INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 1)
  - TAUDCDRm = a = 5H
- Slave channel:
  - TAUDCDRm = 0H

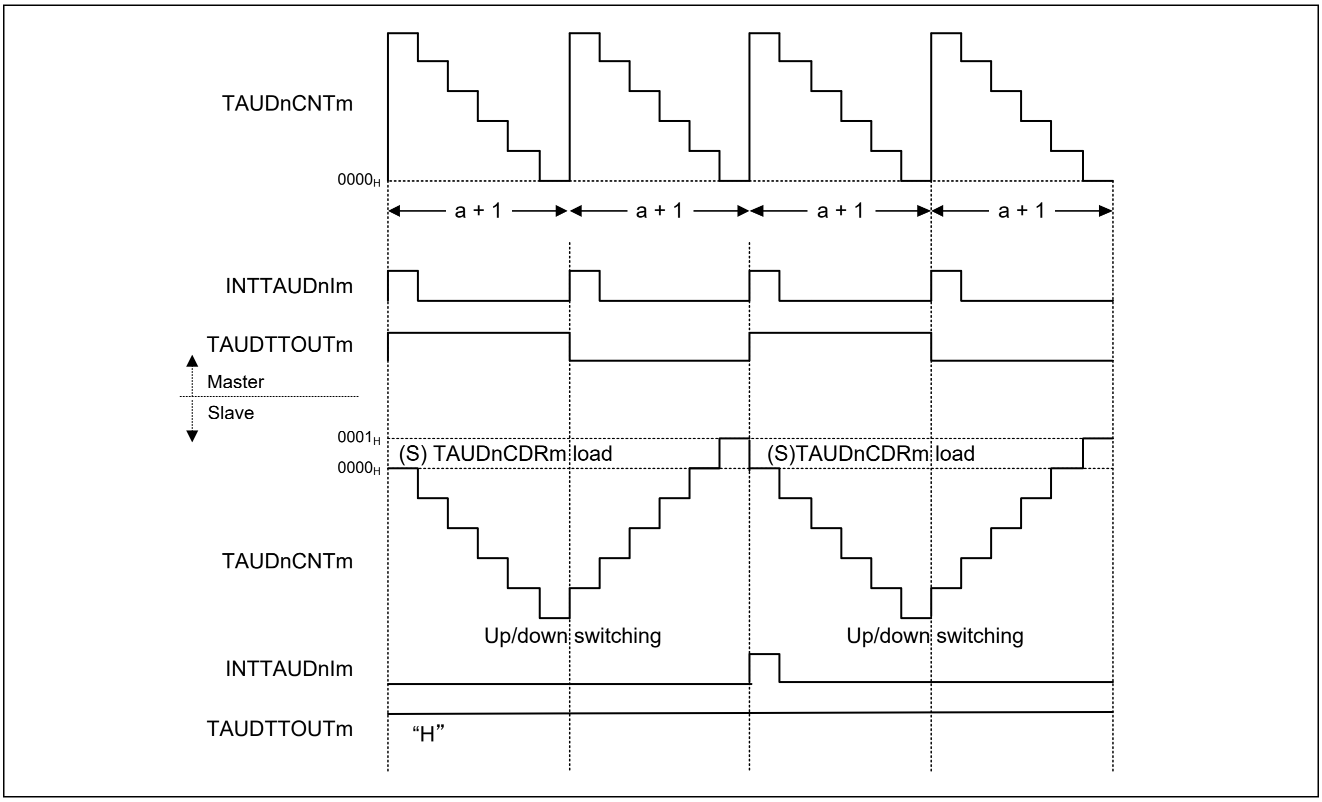


Figure 20.115 TAUDCDRm (Slave) = 0000H

- If TAUDCDRm (slave) = 0000H, the counter of slave channel does not reach 0001H while counting up. TAUDTTOUTm remains high because there is no reset signal to be detected.

20.15.7 Triangle PWM Output with Dead Time

(1) Overview

(a) 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 TAUDTDL.TAUDTDLm, 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.

(b) Prerequisites

- Three channels. For slave channels 2 and 3, select even-numbered channel CH (a) and odd-numbered channel CH (a + 1).
- The operating mode for the master channel should be set to interval timer mode (see Table 20.158, Contents of the TAUDCMORm Register for the Master Channel of the Triangle PWM Output with Dead Time).
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even-numbered channel (a), and slave channel 3 is an odd-numbered channel (a + 1).
- The operating mode for slave channel 2 should be set to count-up/-down mode (see Table 20.162, Contents of the TAUDCMORm Register for Slave Channel 2 of the Triangle PWM Output with Dead Time). Slave channel 2 should be an even-numbered channel.
- The operating mode for slave channel 3 should be set to one-count mode (see Table 20.166, Contents of the TAUDCMORm Register for Slave Channel 3 of the Triangle PWM Output with Dead Time). Slave channel 3 should be an odd-numbered channel.
- The channel output mode for the master channel should be set to independent channel output mode 1 (see Section 20.7, 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 20.7, Channel Output Modes).
- The following settings drive the TAUDTTOUTm signal to the high level during the down status of the carrier cycle:
  - If TAUDCMORm.TAUDMD0 (master) bit is set to 0, TAUDTO.TAUDTOm should be set to 1 while TAUDTOE.TAUDTOEm is set to 0 (recommended setting).
  - If TAUDCMORm.TAUDMD0 (master) bit is set to 1, TAUDTO.TAUDTOm should be set to 0 while TAUDTOE.TAUDTOEm is set to 0.

Remark: The triangle PWM output with dead time does not use slave channel 1.

(c) Functional description

The counter starts by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation. The current value of TAUDCDRm is loaded into TAUDCNTm and the counter starts to count down from the TAUDCDRm value. If TAUDCMORm.TAUDMD0 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 0000H, an INTTAUDIm is generated and the TAUDTTOUTm signal is toggled. The TAUDCDRm value is reloaded to continue counting down.
- Slave channel 2:
  If INTTAUDIm 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 TAUDCDRm value is reloaded and the counter starts counting down.

  The counter continues to count down/up and waits for the next INTTAUDIm of the master channel.
- Slave channel 3:
  If INTTAUDIm is generated on slave channel 2, the counter of slave channel 3 is triggered. The current value of TAUDCDRm (slave 3) is loaded into TAUDCNTm (slave 3) and the counter starts to count down from the TAUDCDRm value.
  When the counter reaches 0000H, INTTAUDIm occurs. The counter returns to FFFFH and waits for the next INTTAUDIm of slave channel 2.

As described in Table 20.157, 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 TAUDTDL.TAUDTDLm setting of the corresponding channel.

The setting of TAUDTOL.TAUDTOLm also determines whether a high level signal (TAUDTOL.TAUDTOLm = 0) or a low level signal (TAUDTOL.TAUDTOLm = 1) is output from the corresponding channel.

The counter can be stopped by setting TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but retain their values.

TAUDTTOUTm can be 100% output by setting the TAUDCDRm value of slave channel 2 to 0000H.

(d) Conditions

This function enables simultaneous reloading. See Section 20.6, Simultaneous Reloading.
TAUDTOL.TAUDTOLm and TAUDTDL.TAUDTDLm should be set before start of count operation. Slave channels 2 and 3 should have the opposite settings of TAUDTDL.TAUDTDLm.

Table 20.157 Operation of TAUDTTOUTm upon Occurrence of an Interrupt on Slave Channel 2

TAUDTDL. TAUDTDLm	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	Reset after elapse of dead time

(e) Equations

Pulse cycle = (TAUDCDRm (master) + 1) × count clock cycle
0000H ≤ TAUDCDRm (master) < FFFFH

Carrier cycle (down/up) = (TAUDCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (normal phase)
= [(TAUDCDRm (master) + 1 – TAUDCDRm (slave 2)) × 2 – (TAUDCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (reverse phase)
= [(TAUDCDRm (master) + 1 – TAUDCDRm (slave 2)) × 2 + (TAUDCDRm (slave 3) + 1)] × count clock cycle

(2) Block Diagram and General Timing Diagram

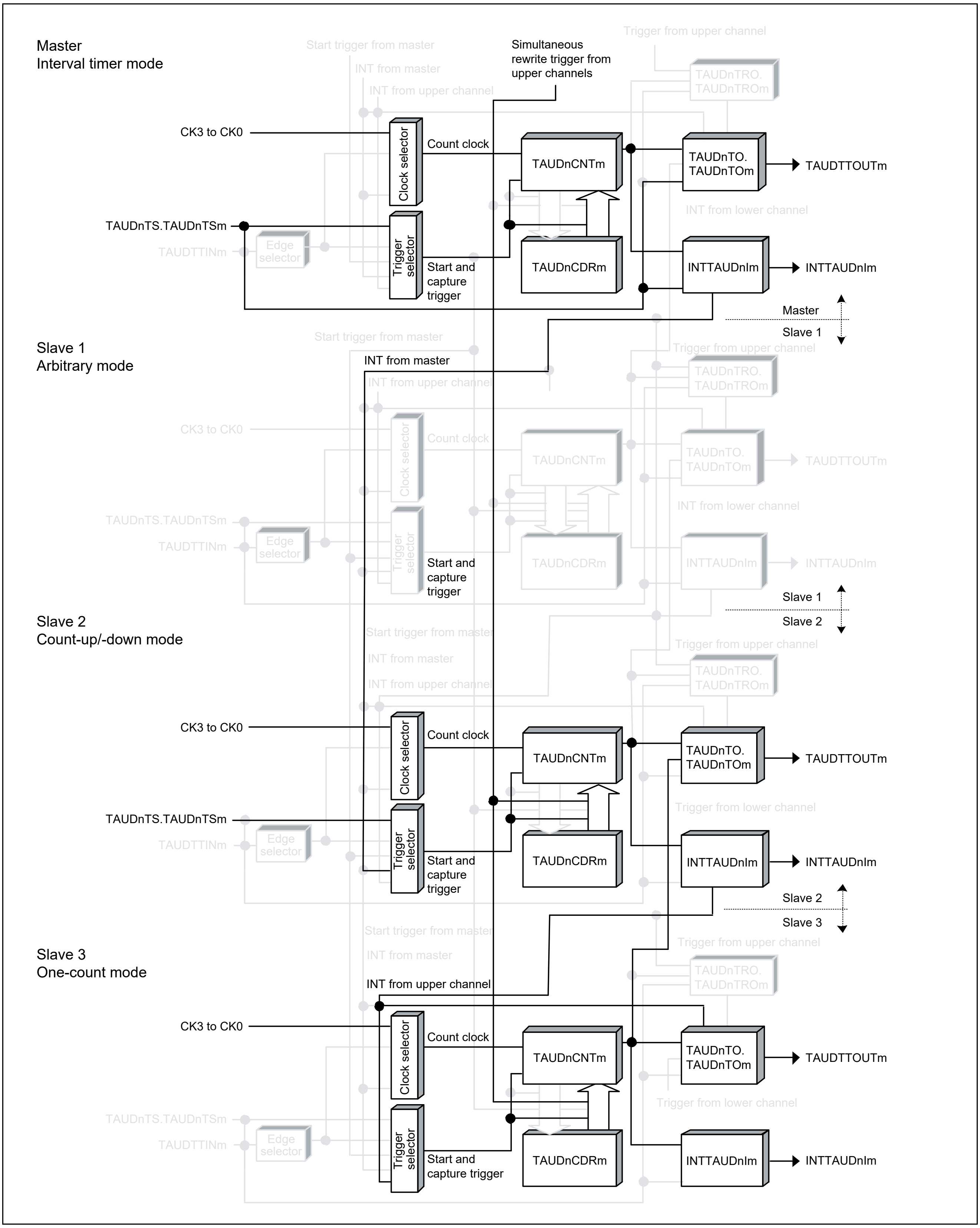


Figure 20.116 Block Diagram of Triangle PWM Output with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  – INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 1)
- Slave channel 2:
  – INTTAUDIm not generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 0)
  – TAUDTDL.TAUDTDLm = 0
  – Positive logic (TAUDTOL.TAUDTOLm = 0)
- Slave channel 3:
  – INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 1)
  – TAUDTDL.TAUDTDLm = 1
  – Positive logic (TAUDTOL.TAUDTOLm = 0)

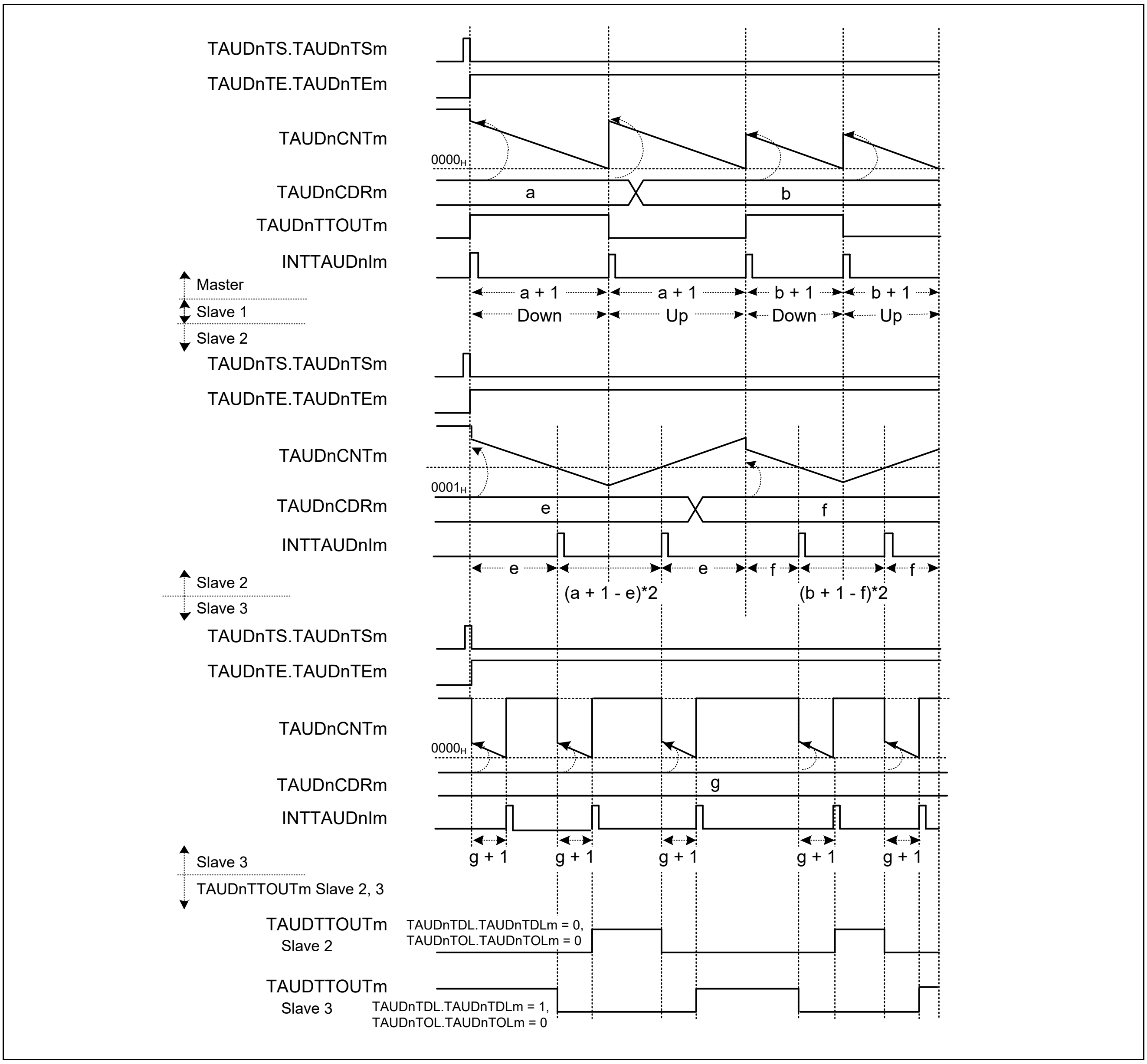


Figure 20.117 General Timing Diagram of Triangle PWM Output with Dead Time

(3) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.158 Contents of the TAUDCMORm Register for the Master Channel of the Triangle PWM Output with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation. 1: INTTAUDIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.159 Contents of the TAUDCMURm Register for the Master Channel of the Triangle PWM Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output the mode

Table 20.160 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.161 Simultaneous Reload Setting for the Master Channel of Triangle PWM Output with Dead Time

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: A simultaneous reload trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

Remark: If TAUDRDS.TAUDRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous reload trigger signal.

(4) Register Settings for Slave Channel 2

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.162 Contents of the TAUDCMORm Register for Slave Channel 2 of the Triangle PWM Output with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDCOS[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	TAUDMD[4:1]	1001: Count-up/-down mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.163 Contents of the TAUDCMURm Register for Slave Channel 2 of the Triangle PWM Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.164 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel operation
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	1: Enables dead time operation.
TAUDTDM.TAUDTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDTDL.TAUDTDLm are satisfied.
TAUDTDL.TAUDTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

Caution: Set TAUDTDLm exclusively from odd-numbered channels.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.165 Simultaneous Reload Settings for Slave Channel 2 of Triangle PWM Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: A simultaneous reload trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(5) Register Settings for Slave Channel 3

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.166 Contents of the TAUDCMORm Register for Slave Channel 3 of the Triangle PWM Output with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	110: Dead time output signal of the TAUDTTOUTm generation unit
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Enables start trigger detection while counting.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.167 Contents of the TAUDCMURm Register for Slave Channel 3 of the Triangle PWM Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.168 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel operation
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	1: Enables dead time operation.
TAUDTDM.TAUDTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDTDL.TAUDTDLm are satisfied.
TAUDTDL.TAUDTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set this bit to 0
TAUDTRC.TAUDTRCm	0: Disables the operation as a real-time output trigger channel
TAUDTME.TAUDTMEm	0: Disables modulation

Caution: Set TAUDTDL.TAUDTDLm exclusively from even-numbered channels.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.169 Simultaneous Reload Settings for Slave Channel 3 of Triangle PWM Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel. 1: Selects a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: A simultaneous reload trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(6) Operating Procedure for Triangle PWM Output with Dead Time

Table 20.170 Operating Procedure for Triangle PWM Output with Dead Time

		Operation	TAUD Status
	Initial Channel Setting	Master channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.15.7(3), Register Settings for the Master Channel. Slave channel 2: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.15.7(4), Register Settings for Slave Channel 2. Slave channel 3: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.15.7(5), Register Settings for Slave Channel 3. Set the value of TAUDCDRm register of every channel.	Channel operation is stopped.
	Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDIm (master) is generated on the master channel if TAUDCMORm.TAUDMD0 is set to 1.
Restart	During Operation	TAUDCDRm can be changed at any time. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	TAUDCDRm value of master channel and slave channel 2 is loaded into TAUDCNTm to perform counting down. When the counter of master channel reaches 0000H: • INTTAUDIm (master) is generated. • TAUDCDRm value is reloaded into TAUDCNTm (master) to continue count operation. • TAUDCDRm value is reloaded into TAUDCNTm (slave 2) or counting is started in opposite direction. When TAUDCNTm of slave channel 2 reaches 0001H: • IINTTAUDIm (slave 2) is generated. • TAUDCDRm value of slave channel 3 is loaded into TAUDCNTm perform counting down. When TAUDCNTm of slave channel 3 reaches 0000H: • INTTAUDIm is generated.
	Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

(7) Specific Timing Diagrams

(a) Duty cycle = 0%

The following settings apply to the 0.

- Slave channel 2:
  – Positive logic (TAUDTDL.TAUDTDLm = 0)
- Slave channel 3:
  – Negative logic (TAUDTDL.TAUDTDLm = 1)

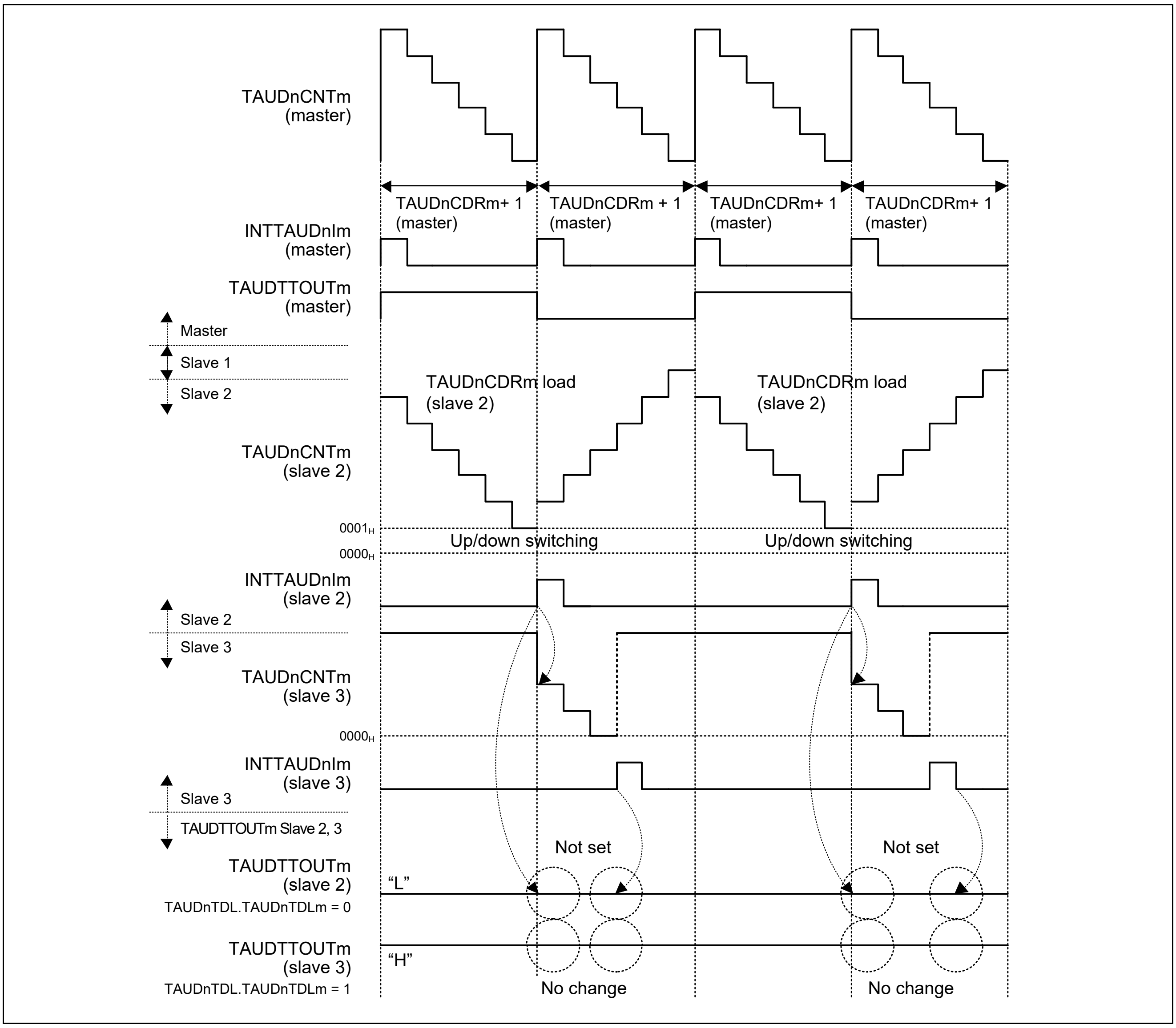


Figure 20.118 TAUDCDRm (Slave 2) ≥ TAUDCDRm (Master) + 1

- If TAUDCDRm (slave 2) is greater than TAUDCDRm (master), the counter of slave channel does not reach 0000H 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.

(b) Duty cycle = 100%

The following settings apply to the Figure 20.119.

- Slave channel 2:
  - Positive logic (TAUDTDL.TAUDTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDTDL.TAUDTDLm = 1)

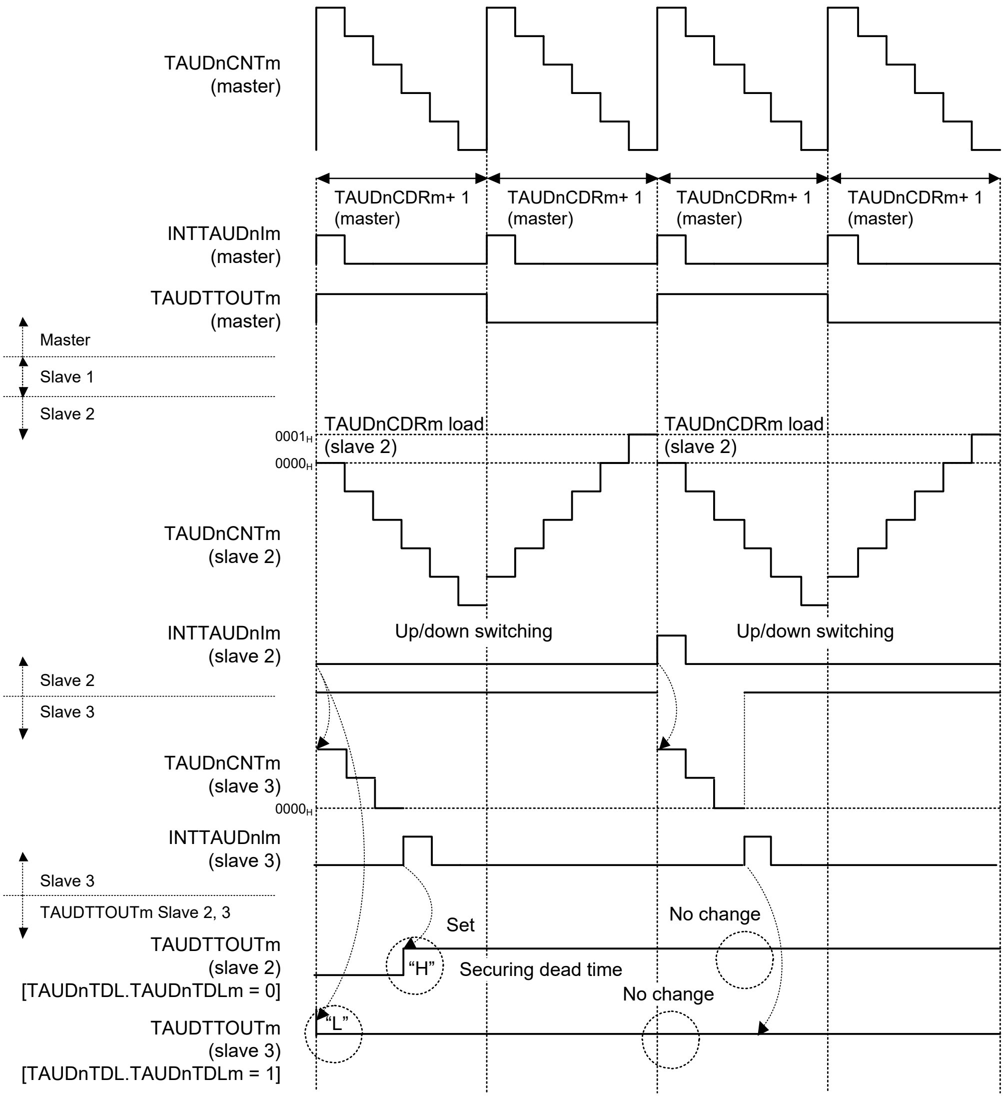


Figure 20.119 TAUDCDRm (Slave) = 0000H

- If TAUDCDRm (slave 2) = 0000H, the slave channel counter does not reach 0001H while counting up. Therefore, no INTTAUDIm is generated during count-up operation.
  - The set conditions for a channel with TAUDTDL.TAUDTDLm = 0 are met after elapse of dead time. Even if TAUDTTOUTm is set or reset, the signal is left in a newly set state 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 TAUDTDL.TDLm = 1 because no reset conditions are satisfied on that channel.

20.15.8 Skipping Interrupt Request Signals

(1) Overview

(a) Summary

This function divides the number of interrupts of the master channel by a specified value using a slave channel. Skipping interrupt request signals is a sub function of the following functions:

- PWM Output (See Section 20.15.1, PWM Output)
- Triangle PWM Output (See Section 20.15.6, Triangle PWM Output)
- Triangle PWM Output with Dead Time (See Section 20.15.7, Triangle PWM Output with Dead Time)

(b) Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode (See Table 20.171, Contents of the TAUDCMORm Register for the Master Channel for Skipping Interrupt Request Signals)
- The operation mode of the slave channel must be set to Event Count Mode. (See Table 20.174, Contents of the TAUDCMORm Register for the Slave Channel for Skipping Interrupt Request Signals)
- This function does not use TAUDTTOUTm.

(c) Functional description

The counters (master and slave) are enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1 for both channels. This in turn sets TAUDTE.TAUDTEm, enabling count operation. The current value of the data register of the master channel and slave channel (TAUDCDRm) are written to the counter (TAUDCNTm).

- Master channel:
  When the counter of the master channel reaches 0000H, INTTAUDIm is generated and TAUDCDRm value is reloaded to TAUDCNTm.
- Slave channel:
  Every time the master channel generates an INTTAUDIm, the counter of the slave channel decrements by one. When the counter reaches 0000H, it awaits the next interrupt from the master channel. This causes TAUDCNTm (slave) to reload the value of TAUDCDRm, and an INTTAUDIm is generated.

Forced restart is not possible for this function. The counter can be stopped by setting TAUDTT.TAUDTTm to 1 for the master and slave channel, which in turn sets TAUDTE.TAUDTEm to 0. TAUDCNTm of master and slave channel stops but retains its value.

(d) Conditions

This function enables simultaneous reloading. See Section 20.6, Simultaneous Reloading.

(2) Equations

Interrupt division operator = TAUDCDRm (slave channel)

- One INTTAUDIm is generated for the INTTAUDIm count of the master channel defined by TAUDCDRm (slave channel) + 1.

(3) Block Diagram and General Timing Diagram

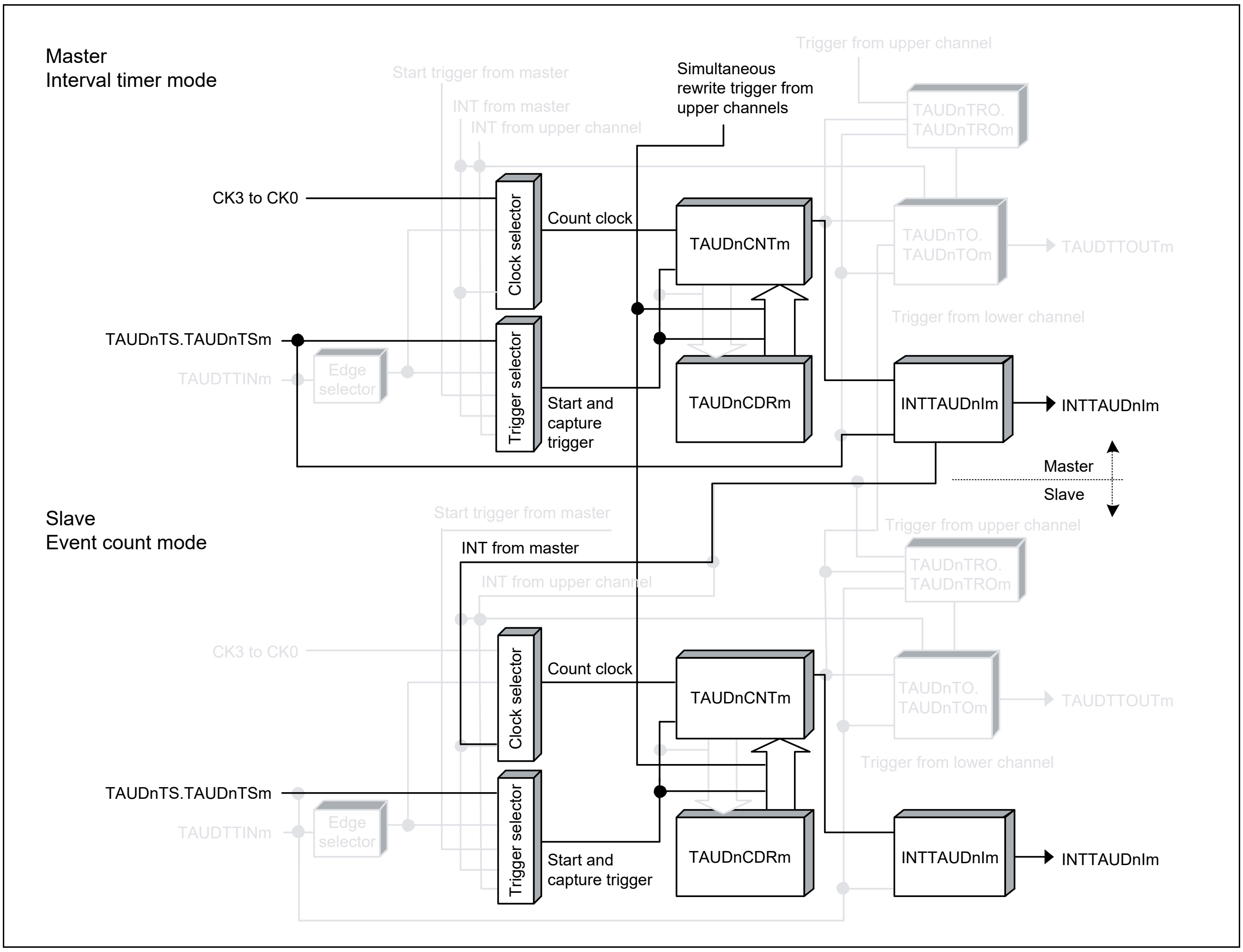


Figure 20.120 Block Diagram of Skipping Interrupt Request Signals

The following settings apply to the general timing diagram.

- Master channel:
  INTTAUDIm is generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 1)

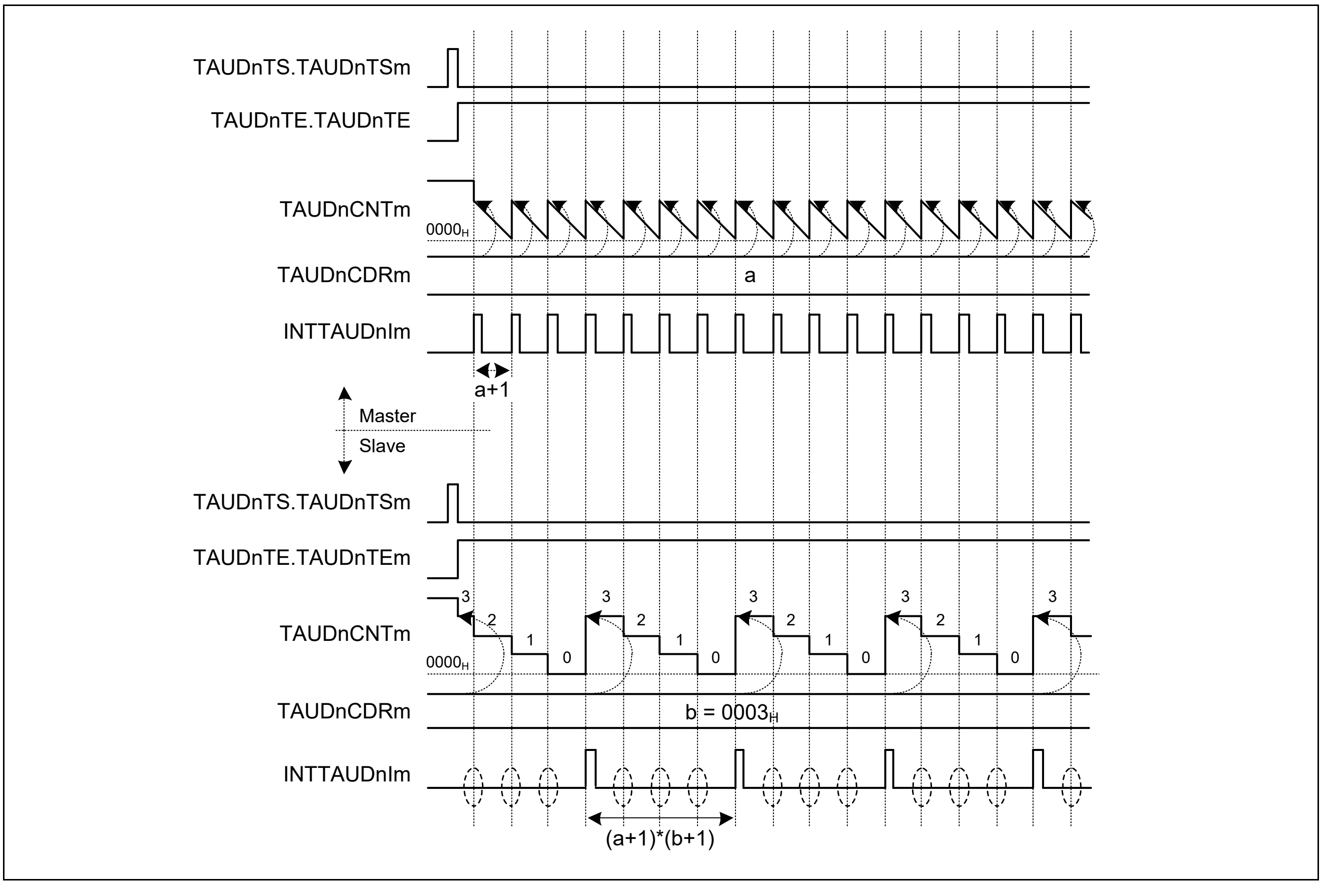


Figure 20.121 General Timing Diagram of Skipping Interrupt Request Signals

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.171 Contents of the TAUDCMORm Register for the Master Channel for Skipping Interrupt Request Signals

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation. 1: INTTAUDIm generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.172 Contents of the TAUDCMURm Register for the Master Channel for Skipping Interrupt Request Signals

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.173 Simultaneous Reload Settings for the Master Channel for Skipping Interrupt Request Signals

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count. 1: Simultaneous reload trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(5) Register Settings for the Slave Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.174 Contents of the TAUDCMORm Register for the Slave Channel for Skipping Interrupt Request Signals

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	11: INTTAUDIm of the master channel is used as the count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0011: Event count mode
0	TAUDMD0	0: INTTAUDIm not generated at the beginning of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.175 Contents of the TAUDCMURm Register for the Slave Channel for Skipping Interrupt Request Signals

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.176 Simultaneous Reload Settings for the Slave Channel for Skipping Interrupt Request Signals

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Selects a trigger for simultaneous reloading of master channel.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count. 1: Simultaneous reload trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

(6) Operating Procedure for Skipping Interrupt Request Signals

Table 20.177 Operating Procedure for Skipping Interrupt Request Signals

	Operation	TAUD Status
Initial Channel Setting	Master channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.15.8(4), Register Settings for the Master Channel. Slave channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.15.8(5), Register Settings for the Slave Channel. Set the value of TAUDCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDIm is generated on the master channel.
During Operation	TAUDCDRm can be changed at any time. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	TAUDCNTm of master channel loads TAUDCDRm value and counts down. When the counter reaches 0000H: • INTTAUDIm (master) is generated. • TAUDCNTm (master) loads TAUDCDRm value and continues count operation. • TAUDCNTm of slave channels counts down each time INTTAUDIm of master channel is detected. When TAUDCNTm of the slave = 0000H: • INTTAUDIm (slave) is generated. • The TAUDCDRm value is loaded in TAUDCNTm (slave) and count operation continues.
Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm stops and retains its current value.

Restart (from Stop Operation back to During Operation)

(7) Specific Timing Diagrams

(a) Interrupt count (master) = interrupt count (slave)

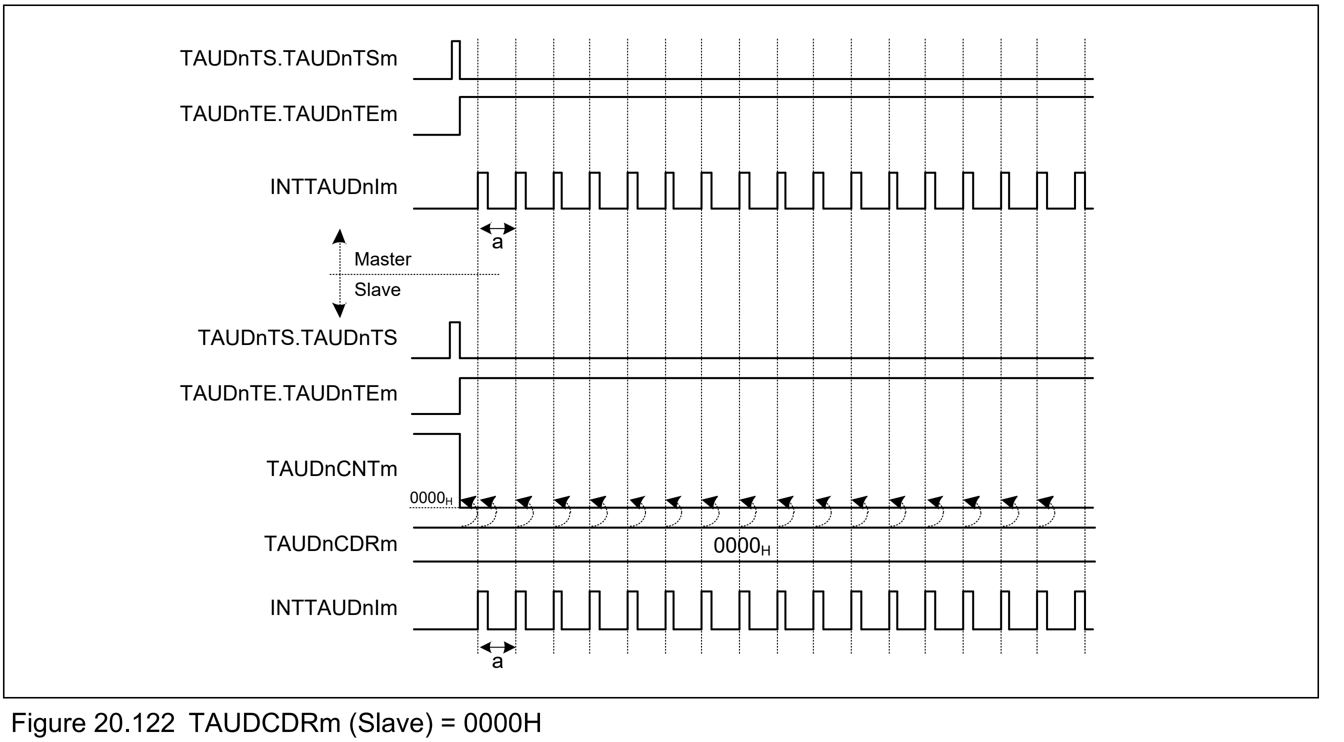


Figure 20.122 TAUDCDRm (Slave) = 0000H

- If TAUDCDRm = 0000H, the TAUDCDRm value of the slave channel is loaded into TAUDCNTm each time INTTAUDIm of master channel is detected. In other words, TAUDCNTm is always 0000H.
- Therefore, an interrupt occurs on the master channel and simultaneously an interrupt occurs on slave channels.

20.16 Synchronous Non-Complementary and Complementary Modulation Output

This section describes functions that generate 6-phase PWM output or triangle PWM output using a master channel and seven slave channels.

20.16.1 Synchronous Non-Complementary Modulation Output Type 1

(1) Overview

(a) 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 (TAUDTRO.TAUDTROm) and the modulation output enable bits (TAUDTME.TAUDTMEm) of a pair of slave channels. Three pairs of channels are typically used.

(b) Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See Table 20.179, Contents of the TAUDCMORm Register for the Master Channel of Non-Complementary Modulation Output Type 1).
- The operating mode for slave channels 1 to 7 should be set to one-count mode (See Table 20.182, Contents of the TAUDCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Type 1, and Table 20.185, Contents of the TAUDCMORm Register for Slave Channel 2 to 7 of Non-Complementary Modulation Output 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 TAUDTRC.TAUDTRCm should be set to 1 (See Section 20.7, 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 20.7, Channel Output Modes).
- TAUDCDRm of slave channel 1 should be set to 0000H.

(c) Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation. The value of data register (TAUDCDRm) is loaded into the counter (TAUDCNTm) and the counter starts to count down. When the counter reaches 0000H, INTTAUDIm is generated.

- Slave channel 1:
  Slave channel 1 is set as a channel that triggers real-time output (TAUDTRC.TAUDTRCm = 1). If an interrupt occurs on slave channel 1 (TAUDCDRm is fixed to 0000H), the value of realtime output bit (TAUDTRO.TAUDTROm) of the channel that monitors the interrupt on slave channel 1 is reflected to the TAUDTTOUTm output. After that, the counter returns to FFFFH 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 FFFFH and awaits the next interrupt from the master channel.

Slave channels 3 to 7 operate like slave channel 2.

As described in Table 20.178, TAUDTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Type 1 (TAUDTOL.TAUDTOLm = 0), a signal output from TAUDTTOUTm depends on the value of the real-time output bit (TAUDTRO.TAUDTROm) and modulation output bit (TAUDTME.TAUDTMEm) of slave channel.

This function cannot use a forced restart. The counter can be stopped by setting TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDTS.TAUDTSm to 1.

(d) Conditions

- If TAUDTME.TAUDTMEm = 0 on slave channels 2 to 7 (TAUDTOL.TAUDTOLm = 0):
  – If the channel's TAUDTRO.TAUDTROm is set to 1, TAUDTTOUTm outputs a high-level signal.
  – If the channel's TAUDTRO.TAUDTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDTME.TAUDTMEm = 1 on slave channels 2 to 7 (TAUDTOL.TAUDTOLm = 0):
  – If the channel's TAUDTRO.TAUDTROm is set to 1, TAUDTTOUTm outputs PWM corresponding to the channel.
  – If the channel's TAUDTRO.TAUDTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDTOL.TAUDTOLm 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 TAUDTOL.TAUDTOLm is permitted (cannot be changed during operation).

Table 20.178 TAUDTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Type 1 (TAUDTOL.TAUDTOLm = 0)

TAUDTME.TAUDTMEm	TAUDTRO.TAUDTROm	TAUDTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous reloading. See Section 20.6, Simultaneous Reloading.
- TAUDCDRm value of slave channel 1 should be set to 0000H so that a real-time output is triggered at the same time with PWM generation on slave channels 2 to 7.
- If TAUDTOL.TAUDTOLm is set to 0 on slave channels 2 to 7, TAUDTO.TAUDTOm is set to 0 (low) before TAUDTE.TAUDTEm is set to 0.
- If TAUDTOL.TAUDTOLm is set to 1 on slave channels 2 to 7, TAUDTO.TAUDTOm is set to 1 (high) before TAUDTE.TAUDTEm is set to 0.

(2) Equations

Slave channels 2 to 7:

Pulse period = [TAUDCDRm (master) + 1] × count clock cycle

Duty time = [TAUDCDRm (slave)] × count clock cycle

(3) Block Diagram and General Timing Diagram

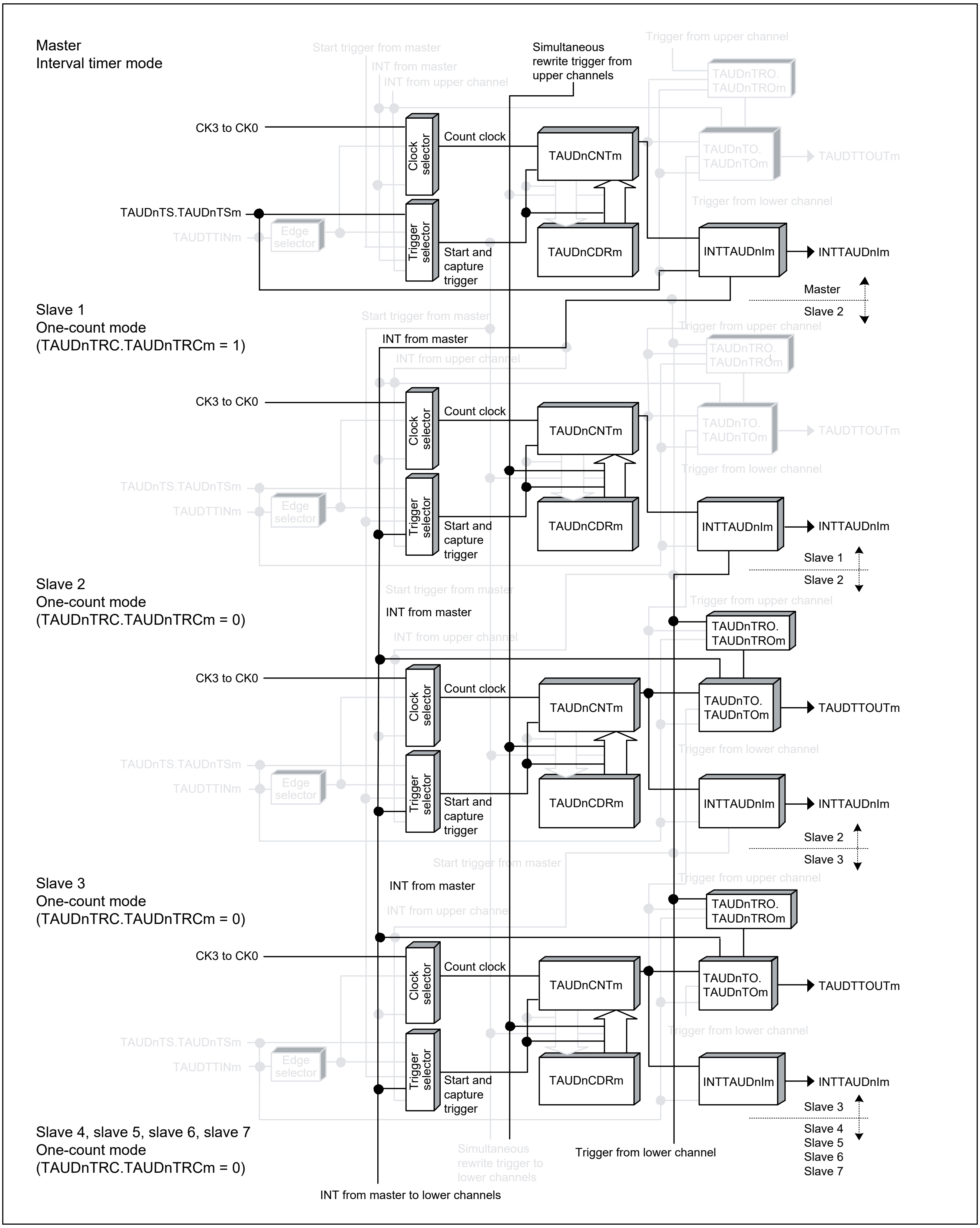


Figure 20.123 Block Diagram of Non-Complementary Modulation Output Type 1

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDTOL.TAUDTOLm = 0)

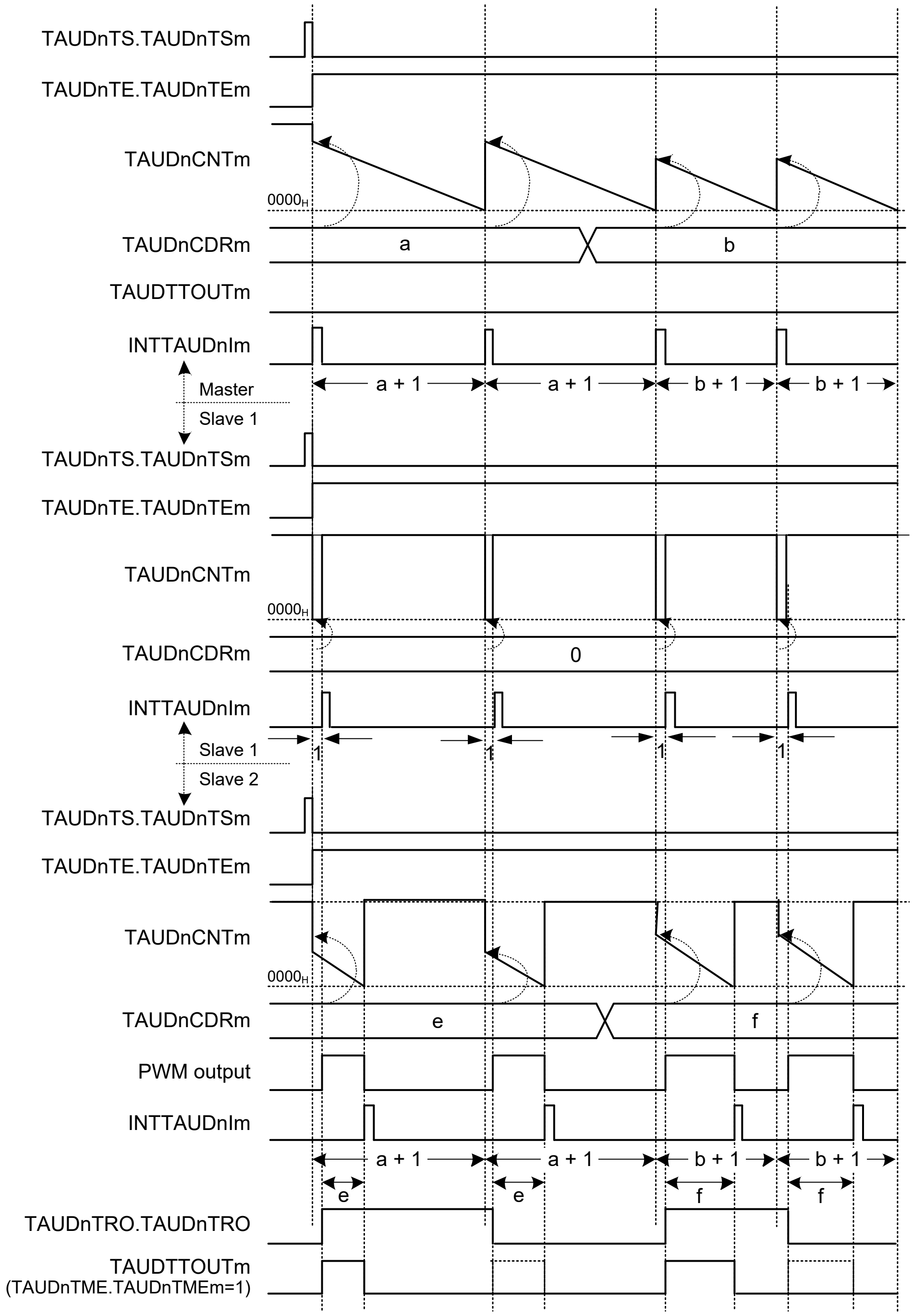


Figure 20.124 General Timing Diagram of Non-Complementary Modulation Output Type 1

Remark: TAUDTTOUTm of slave channel 2 rises with a delay of one clock count after the rise of INTTAUDIm of the master channel.

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.179 Contents of the TAUDCMORm Register for the Master Channel of Non-Complementary Modulation Output Type 1

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10–8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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.
4–1	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	1: INTTAUDIm is generated at the beginning of operation or at a restart time.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.180 Contents of the TAUDCMURm Register for the Master Channel of Non-Complementary Modulation Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm is set to 0 because channel output mode is not used with this function.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.181 Simultaneous Reload Settings for the Master Channel of Non-Complementary Modulation Output Type 1

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

Remark: Use with TAUDRDS.TAUDRDSm bit = 1 requires an upper channel higher than the master channel that operates with Section 20.14.1, Simultaneous Reload Trigger Generation Type 1. Make settings for operation under the following conditions.

- **Simultaneous reload trigger output type 1 setting channel: TAUDRDCm = 1, TAUDRDSm = 1**
  In addition, TAUDCDRm settings for this channel are as follows.
  = ((TAUDCDR setting for the master channel targeted for simultaneous reloading + 1) × Interrupt count) – 1
- **Master channel: TAUDRDCm = 0, TAUDRDSm = 1**
- **Slave channel: TAUDRDCm = 0, TAUDRDSm = 1**

(5) Register Settings for Slave Channel 1

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS[1:0]		TAUDCCS[1:0]		TAUDMAS	TAUDSTS[2:0]			TAUDCOS[1:0]		0	TAUDMD[4:1]				TAUDMD0

Table 20.182 Contents of the TAUDCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Type 1

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	100: INTTAUDIm of master channel is a start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Start trigger during operation is valid.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.183 Contents of the TAUDCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm 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: TAUDTRC.TAUDTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.184 Simultaneous Reload Settings for Slave Channel 1 of Non-Complementary Modulation Output Type 1

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

(6) Register Settings for Slave Channels 2 to 7

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.185 Contents of the TAUDCMORm Register for Slave Channel 2 to 7 of Non-Complementary Modulation Output Type 1

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	100: INTTAUDIm of master channel is a start trigger.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Start trigger during operation is valid.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.186 Contents of the TAUDCMURm Register for Slave Channel 2 to 7 of Non-Complementary Modulation Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.187 Control Bit Settings in Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	1: Enables real-time output.
TAUDTRO.TAUDTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDTRC.TAUDTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDTME.TAUDTMEm	0: Disables modulation 1: Enables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.188 Simultaneous Reload Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Type 1

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	0: Generates a simultaneous reload trigger signal when the master channel starts to count.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

(7) Operating Procedure for Non-Complementary Modulation Output Type 1

Table 20.189 Operating Procedure for Non-Complementary Modulation Output Type 1

(1/2)

	Operation	TAUD Status
Initial Channel Setting	Master channel: Set TAUDCMORm/ TAUDCMURm register and the channel output mode as described in Section 20.16.1(4), Register Settings for the Master Channel. Slave channel 1: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.16.1(5), Register Settings for Slave Channel 1. Slave channels 2 to 7: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.16.1(6), Register Settings for Slave Channels 2 to 7. Set the value of TAUDCDRm register of every channel. Set a pulse cycle with TAUDCDRm of master channel, 0000H in TAUDCDRm of slave channel 1, and duty width with TAUDCDRm of slave channels 2 to 7. Set TAUDTRC.TAUDTRCm to 1 on slave channel 1.	Channel operation is stopped.

(2/2)

	Operation	TAUD Status
Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDCDRm, TAUDTRO.TAUDTROm, and TAUDTMEm.TAUDTMEm can be changed at any time. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	TAUDCDRm value of master channel, slave channel 1 and slave channels 2 to 7 is loaded into TAUDCNTm to perform counting down. When the counter of master channel reaches 0000H: • INTTAUDIm is generated. • TAUDCDRm value is reloaded into TAUDCNTm of master channel to continue counting down. • PWM output signals of slave channels 2 to 7 are set. • TAUDCDRm value of slave channel 1 is reloaded into TAUDCNTm to perform counting down. • TAUDCDRm value of slave channels 2 to 7 is reloaded into TAUDCNTm to perform counting down. • When the counter of slave channel 1 reaches 0000H: - INTTAUDIm is generated. - The TAUDTRO.TAUDTROm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. • When the counter of slave channels 2 to 7 reaches 0000H: - INTTAUDIm 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 (TAUDTRO.TAUDTROm) and modulation output bit (TAUDTME.TAUDTMEm) of a pair of slave channels.
Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

Restart

(8) Specific Timing Diagrams

The following settings apply to the specific timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDTOL.TAUDTOLm = 0)

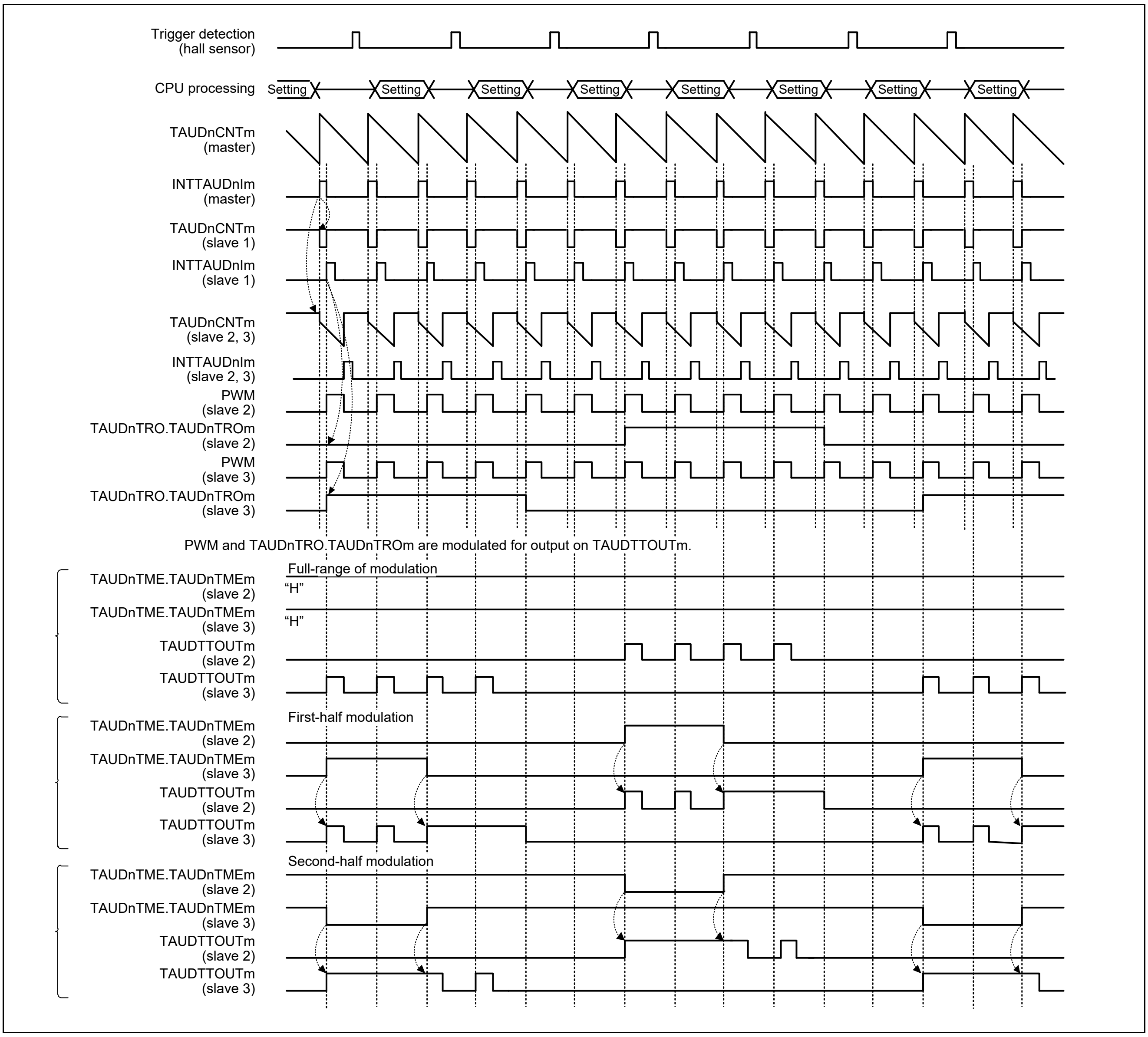


Figure 20.125 Specific Timing Diagram of Non-Complementary Modulation Output Type 1

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDTME.TAUDTMEm bits of lower slave channels during operation.

The "Setting" symbol indicates a time period when the values of TAUDCDRm, TAUDTME.TAUDTMEm, and TAUDTRO.TAUDTROm can be changed.

TAUDTME.TAUDTMEm 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 TAUDTRO.TAUDTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

20.16.2 Synchronous Non-Complementary Modulation Output Type 2

(1) Overview

(a) 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 (TAUDTRO.TAUDTROm) and the modulation output enable bit value (TAUDTME.TAUDTMEm) of a pair of slave channels. Three pairs of channels are typically used.

(b) Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See Table 20.191, Contents of the TAUDCMORm Register for the Master Channel of Non-Complementary Modulation Output Type 2).
- The operating mode for slave channel 1 should be set to event count mode (See Table 20.195, Contents of the TAUDCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Type 2).
- The operating mode for slave channels 2 to 7 should be set to count-up/-down mode (See Table 20.198, Contents of the TAUDCMORm Register for Slave Channel 2 to 7 of Non-Complementary Modulation Output Type 2).
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDTRC.TAUDTRCm should be set to 1 (See Section 20.7, Channel Output Modes.)
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDTRC.TAUDTRCm should be set to 1 (See Section 20.7, 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 20.7, Channel Output Modes.)

(c) Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation. The value of data register (TAUDCDRm) is loaded into the counter (TAUDCNTm).

- Master channel:
  The counter of master channel starts to count down. When the counter reaches 0000H, INTTAUDIm is generated.
- Slave channel 1:
  When slave channel 1 detects an interrupt from the master channel, the TAUDCNTm value is decremented. When an interrupt from the master channel is detected for the (TAUDCDRm + 1) times, INTTAUDIm is generated. Then, the TAUDCDRm value is loaded into TAUDCNTm to continue operation subsequently.
  Since slave channel 1 is set as a real-time output trigger channel (TAUDTRC.TAUDTRCm = 1), the real-time output bit (TAUDTRO.TAUDTROm) 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, TAUDCNTm counts in the reverse direction. When an interrupt is detected during count-up operation, TAUDCDRm value is reloaded and then the counter starts to count down.
  If TAUDCNTm = 0001H, 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 (TAUDTRO.TAUDTROm) and a modulation output bit value (TAUDTME.TAUDTMEm) of the slave channel, as described in Table 20.190, TAUDTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Type 2 (TAUDTOL.TAUDTOLm = 0).

The counter cannot be forcibly restarted with this function. The counter can be stopped by setting TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDTS.TAUDTSm to 1.

(d) Conditions

- If TAUDTME.TAUDTMEm = 0 on slave channels 2 to 7 (TAUDTOL.TAUDTOLm = 0):
  - If the channel's TAUDTRO.TAUDTROm is set to 1, TAUDTTOUTm outputs a high-level signal.
  - If the channel's TAUDTRO.TAUDTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDTME.TAUDTMEm = 1 on slave channels 2 to 7 (TAUDTOL.TAUDTOLm = 0):
  - If the channel's TAUDTRO.TAUDTROm is set to 1, TAUDTTOUTm outputs PWM (positive logic) corresponding to the channel.
  - If the channel's TAUDTRO.TAUDTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDTOL.TAUDTOLm 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 TAUDTOL.TAUDTOLm is permitted (cannot be changed during operation).

Table 20.190 TAUDTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Type 2 (TAUDTOL.TAUDTOLm = 0)

TAUDTME.TAUDTMEm	TAUDTRO.TAUDTROm	TAUDTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous reloading. See Section 20.6, Simultaneous Reloading.
- If TAUDTOL.TAUDTOLm is set to 0 on slave channels 2 to 7, TAUDTO.TAUDTOm is set to 0 (low) before TAUDTE.TAUDTEm is set to 0.
- If TAUDTOL.TAUDTOLm is set to 1 on slave channels 2 to 7, TAUDTO.TAUDTOm is set to 1 (high) before TAUDTE.TAUDTEm is set to 0.

(2) Equations

Slave channels 2 to 7:

Carrier cycle (down/up) = [TAUDCDRm (master) + 1] × 2 × count clock cycle

Duty time = [TAUDCDRm (master) + 1 − TAUDCDRm (slave)] × 2 × count clock cycle

(3) Block Diagram and General Timing Diagram

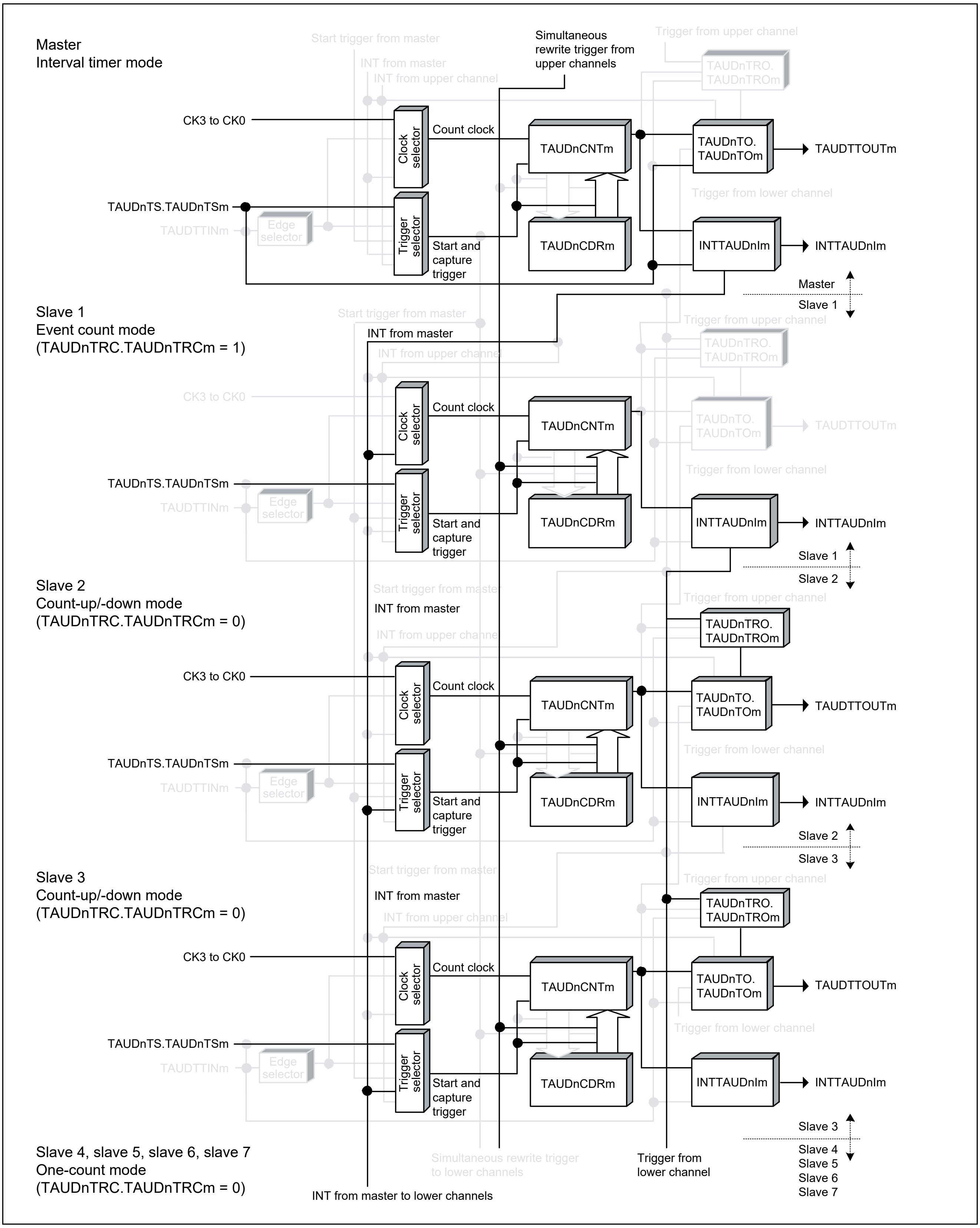


Figure 20.126 Block Diagram and General Timing Diagram

The following settings apply to the general timing diagram.

- Master channel: INTTAUDIm is not generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDTOL.TAUDTOLm = 0)

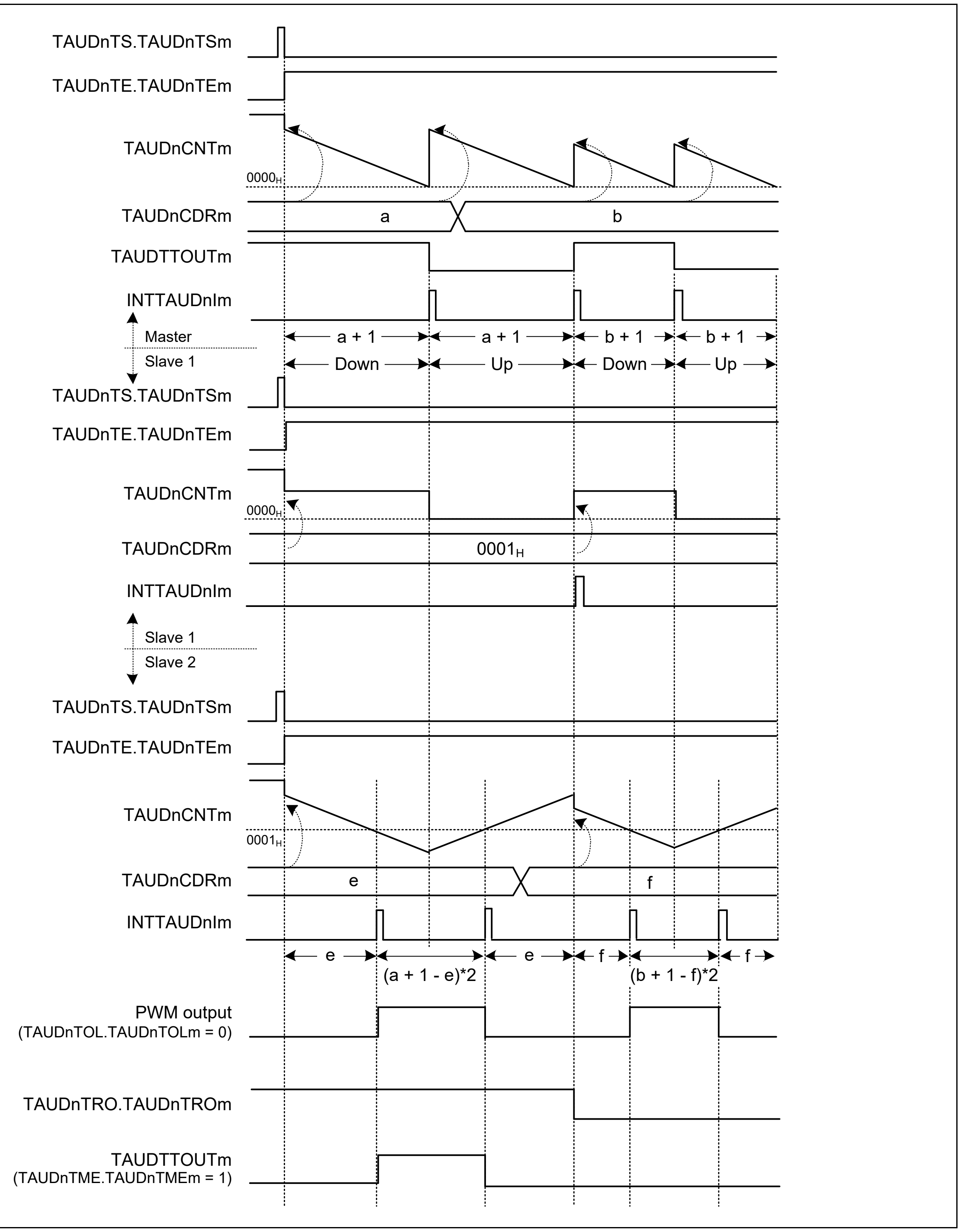


Figure 20.127 General Timing Diagram of Non-Complementary Modulation Output Type 2

(4) Register Settings the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.191 Contents of the TAUDCMORm Register for the Master Channel of Non-Complementary Modulation Output Type 2

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm is not generated at the beginning of operation or at a restart time. 1: INTTAUDIm is generated at the beginning of operation or at a restart time.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.192 Contents of the TAUDCMURm Register for the Master channel of Non-Complementary Modulation Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.193 Control Bit Settings for the Master Channel in Non-Complementary Modulation Output Type 2

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (toggle mode with TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm TAUDTRC.TAUDTRCm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set these bits to 0
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.194 Simultaneous Reload Settings for the Master Channel of Non-Complementary Modulation Output Type 2

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Disables simultaneous reloading
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: A simultaneous reload trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel.

Remark: If TAUDRDS.TAUDRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous reload trigger signal.

(5) Register Settings for Slave Channel 1

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.195 Contents of the TAUDCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Type 2

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	11: INTTAUDIm of the master channel is used as the count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous reloading.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0011: Event count mode
0	TAUDMD0	0: INTTAUDIm is not generated at the beginning of operation or at a restart time.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.196 Contents of the TAUDCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm 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: TAUDTRC.TAUDTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.197 Simultaneous Reload Settings for Slave Channel 1 of Non-Complementary Modulation Output Type 2

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: Simultaneous reload trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

(6) Register Settings for Slave Channels 2 to 7

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS[1:0]		TAUDCCS[1:0]		TAUDMAS	TAUDSTS[2:0]			TAUDCOS[1:0]		0	TAUDMD[4:1]				TAUDMD0

Table 20.198 Contents of the TAUDCMORm Register for Slave Channel 2 to 7 of Non-Complementary Modulation Output Type 2

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	111: The up/down output trigger signal of the master channel
7, 6	TAUDCOS[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	TAUDMD[4:1]	1001: Count-up/-down mode
0	TAUDMD0	0: INTTAUDIm is not generated at the beginning of operation or at a restart time.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.199 Contents of the TAUDCMURm Register for Slave Channel 2 to 7 of Non-Complementary Modulation Output 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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Output mode

Table 20.200 Control Bit Settings in Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel output
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm TAUDTDL.TAUDTDLm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTRE.TAUDTREm	1: Enables real-time output.
TAUDTRO.TAUDTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDTRC.TAUDTRCm	0: The upper channel generates the real-time output trigger for channel m
TAUDTME.TAUDTMEm	0: Disables modulation 1: Enables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.201 Simultaneous Reload Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Type 2

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: Simultaneous reload trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

(7) Operating Procedure for Non-Complementary Modulation Output Type 2

Table 20.202 Operating Procedure for Non-Complementary Modulation Output Type 2

(1/2)

	Operation	TAUD Status
Initial Channel Setting	Master channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.16.2(4), Register Settings the Master Channel. Slave channel 1: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.16.2(5), Register Settings for Slave Channel 1. Slave channels 2 to 7: Set TAUDCMORm/ TAUDCMURm register and the channel output mode as described in Section 20.16.2(6), Register Settings for Slave Channels 2 to 7. Set the value of TAUDCDRm register of every channel. Set pulse cycle in TAUDCDRm of master channel, and in TAUDCDRm 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 TAUDCDRm of slave channels 2 to 7. Set TAUDTRC.TAUDTRCm to 1 on slave channel 1.	Channel operation is stopped.

(2/2)

	Operation	TAUD Status
Start Operation	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDCDRm, TAUDTRO.TAUDTROm, and TAUDTMEm.TAUDTMEm can be changed at any time. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	The TAUDCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDCNTm to perform counting down. The TAUDCDRm 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 0000H: • INTTAUDIm is generated. • TAUDCDRm value is reloaded into TAUDCNTm to continue counting down. • The TAUDCNTm value of slave channel 1 decrements by 1 and the counter waits for a next interrupt from the master channel. • TAUDCNTm of slave channels 2 to 7 reloads the TAUDCDRm value, but performs counting in opposite direction. • At the same timing when the TAUDCDRm value is loaded, the TAUDTME.TAUDTMEm 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 (TAUDCDRm + 1) times: - INTTAUDIm is generated. - The TAUDTRO.TAUDTROm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. • When the counter of slave channels 2 to 7 reaches 0001H: - INTTAUDIm is generated. - PWM output signals of slave channels 2 to 7 are set/reset.
Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

Restart (Stop Operation → Start Operation)

(8) Specific Timing Diagrams

The following settings apply to the general timing diagram.

- Master channel: INTTAUDIm is not generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDTOL.TAUDTOLm = 0)

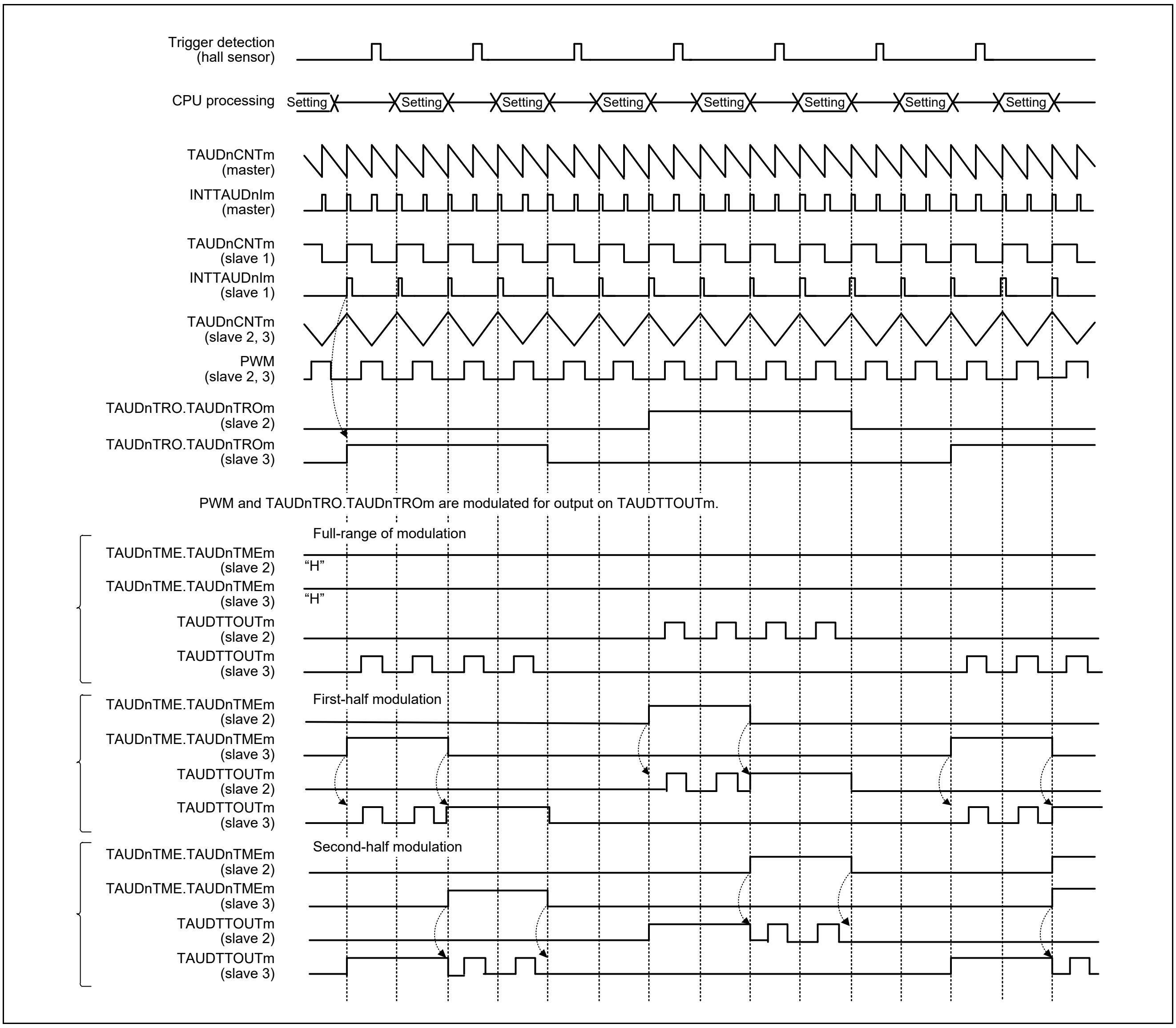


Figure 20.128 Specific Timing Diagram of Non-Complementary Modulation Output Type 2

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDTME.TAUDTMEm bits of lower slave channels during operation.

The "Setting" symbol indicates a time period when the values of TAUDCDRm, TAUDTME.TAUDTMEm, and TAUDTRO.TAUDTROm can be changed.

TAUDTME.TAUDTMEm setting is reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDTRO.TAUDTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

20.16.3 Complementary Modulation Output

(1) Overview

(a) 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 (TAUDTRO.TAUDTROm) and the modulation output bit value (TAUDTME.TAUDTMEm) of a pair of slave channels, and an output level bit value (TAUDTDL.TAUDTDLm). Three pairs of channels are typically used.

(b) Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See Table 20.204, Contents of the TAUDCMORm Register for the Master Channel of the Non-Complementary Modulation Output).
- The operating mode for slave channel 1 should be set to event count mode (See Table 20.208, Contents of the TAUDCMORm Register for Slave Channel 1 of the Non-Complementary Modulation Output).
- The operating mode for slave channels 2, 4 and 6 should be set to count-up/-down mode (See Table 20.211, Contents of the TAUDCMORm Register for Slave Channel 2, 4, and 6 of the Non-Complementary Modulation Output).
- The operating mode for slave channels 3, 5 and 7 should be set to one-count mode (See Table 20.215, Contents of the TAUDCMORm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output). 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 Section 20.7, Channel Output Modes).
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDTRC.TAUDTRCm should be set to 1 (See Section 20.7, 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 20.7, Channel Output Modes).

(c) Functional description

- Master channel:
  The counter of the master channel is enabled by setting the channel trigger bit (TAUDTS.TAUDTSm) to 1. This sets TAUDTE.TAUDTEm = 1, enabling count operation.
  The value of data register (TAUDCDRm) of the master channel is loaded into the counter (TAUDCNTm) and the counter starts to count down from this value.
  When the counter of master channel reaches 0000H, INTTAUDIm 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 0000H, slave channel 1 waits for the next interrupt from the master channel. Then the TAUDCDRm value is reloaded into TAUDCNTm (slave 1) and INTTAUDIm is generated.
  Slave channel 1 is set as a real-time output trigger channel (TAUDTRC.TAUDTRCm = 1). The value of real-time output bit (TAUDTRO.TAUDTROm) 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 0001H, the slave channel 3 counter starts counting down. When the slave channel 3 counter reaches 0000H, 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

(TAUDTRO.TAUDTROm), a modulation output bit value (TAUDTME.TAUDTMEm), and an

output level bit value (TAUDTDL.TAUDTDLm) of the slave channel, as described in Table 20.203, TAUDTTOUTm Output of Slave Channel 1 with Complementary Modulation Output (TAUDTOL.TAUDTOLm = 0).

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 TAUDTT.TAUDTTm of master and slave channels to 1. This sets TAUDTE.TAUDTEm to 0. TAUDCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDTS.TAUDTSm to 1.

(d) Conditions

- If TAUDTME.TAUDTMEm of a pair of channels is set to 1 (TAUDTOL.TAUDTOLm = 0):
  - If TAUDTRO.TAUDTROm of one channel is set to 1, TAUDTTOUTm outputs the corresponding PWM of the channel.
  - If TAUDTRO.TAUDTROm of both channels is set to 0, TAUDTTOUTm of a pair outputs a low-level signal.
- If TAUDTME.TAUDTMEm of a pair of channels is set to 0 (TAUDTOL.TAUDTOLm = 0):
  - If TAUDTRO.TAUDTROm is set to 1, TAUDTTOUTm of the channel outputs a high-level signal.
  - If TAUDTRO.TAUDTROm is set to 0, TAUDTTOUTm of the channel outputs a low-level signal.
- If TAUDTOL.TAUDTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic.

Table 20.203 TAUDTTOUTm Output of Slave Channel 1 with Complementary Modulation Output (TAUDTOL.TAUDTOLm = 0)

TAUDTME. TAUDTME2	TAUDTME. TAUDTME3	TAUDTRO. TAUDTRO2	TAUDTRO. TAUDTRO3	TAUDTDL. TAUDTDL2	TAUDTDL. TAUDTDL3	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

Remarks 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 TAUDTDL.TAUDTDLm.

2. Any settings not listed above are prohibited.

- If TAUDTME.TAUDTMEm is continuously set to 1 while TAUDTRO.TAUDTROm of one of paired channels is set to 1, full modulation is applied.
- If TAUDTME.TAUDTMEm is set to 1 at the first half of the period while TAUDTRO.TAUDTROm of one of paired channels is set to 1, first-half modulation is applied.
- If TAUDTME.TAUDTMEm is set to 1 at the second half of the period while TAUDTRO.TAUDTROm 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 TAUDTDL.TAUDTDLm bit value.
  – If TAUDTDL.TAUDTDLm = 0, dead time is added to a normal phase PWM signal.
  – If TAUDTDL.TAUDTDLm = 1, dead time is added to a reverse phase PWM signal.
  – The operation defined by a TAUDTDL.TAUDTDLm bit value should be conducted by application software during operation. To modify TAUDTDL.TAUDTDLm, rewrite it during the period when TAUDTRO.TAUDTROm is 00B.
- The TAUDCDRm value of slave channel 1 should be set to the value to generate INTTAUDIm of slave channel 1 at a carrier cycle (peak interrupt timing).
- If TAUDTOL.TAUDTOLm is set to 0 on slave channels 2 to 7:
  – If TAUDTDL.TAUDTDLm is set to 0, TAUDTO.TAUDTOm is set to 0 (low) before TAUDTE.TAUDTEm is set to 0.
  – If TAUDTDL.TAUDTDLm is set to 1, TAUDTO.TAUDTOm is set to 1 (high) before TAUDTE.TAUDTEm is set to 0.
- If TAUDTOL.TAUDTOLm is set to 1 on slave channels 2 to 7:
  – If TAUDTDL.TAUDTDLm is set to 0, TAUDTO.TAUDTOm is set to 1 (high) before TAUDTE.TAUDTEm is set to 0.
  – If TAUDTDL.TAUDTDLm is set to 1, TAUDTO.TAUDTOm is set to 0 (low) before TAUDTE.TAUDTEm is set to 0.
- This function enables simultaneous reloading. See Section 20.6, Simultaneous Reloading.

(2) Equations

Pulse period = (TAUDCDRm (master) + 1) × count clock cycle
$0000_H \leq$ TAUDCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDCDRm (master) + 1) × 2 × count clock cycle

For slave channels 2 and 3:

PWM signal width (positive phase)
= [(TAUDCDRm (master) + 1 – TAUDCDRm (slave 2) × 2) – (TAUDCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (negative phase)
= [(TAUDCDRm (master) + 1 – TAUDCDRm (slave 2) × 2) + (TAUDCDRm (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.

(3) Block Diagram and General Timing Diagram

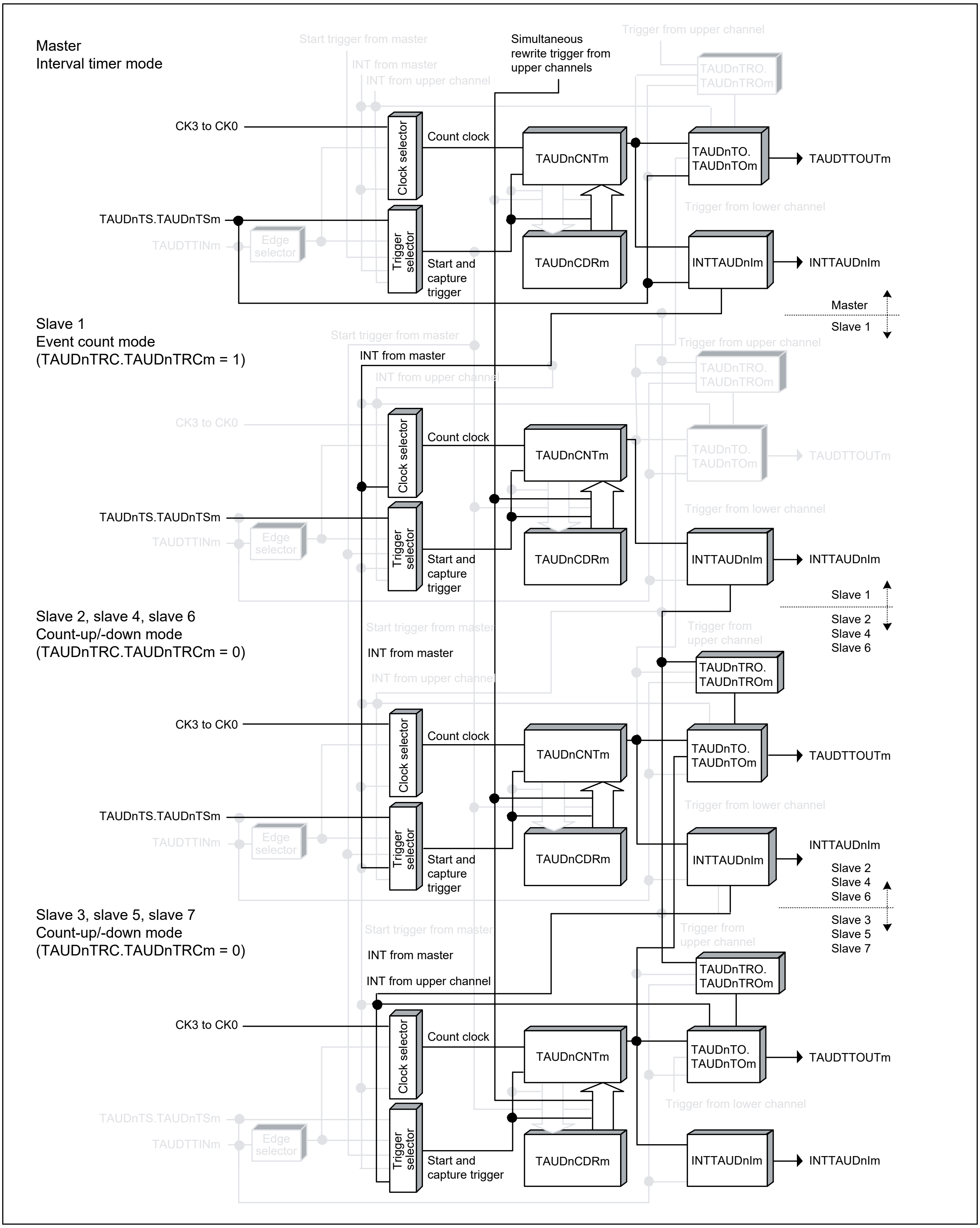


Figure 20.129 Block Diagram of Complementary Modulation Output

The following settings apply to the general timing diagram.

- Master channel: INTTAUDIm is not generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 0)
- Slave channel 1: TAUDCDRm = 0001H
- Slave channels 2 to 7: Positive logic (TAUDTOL.TAUDTOLm = 0)

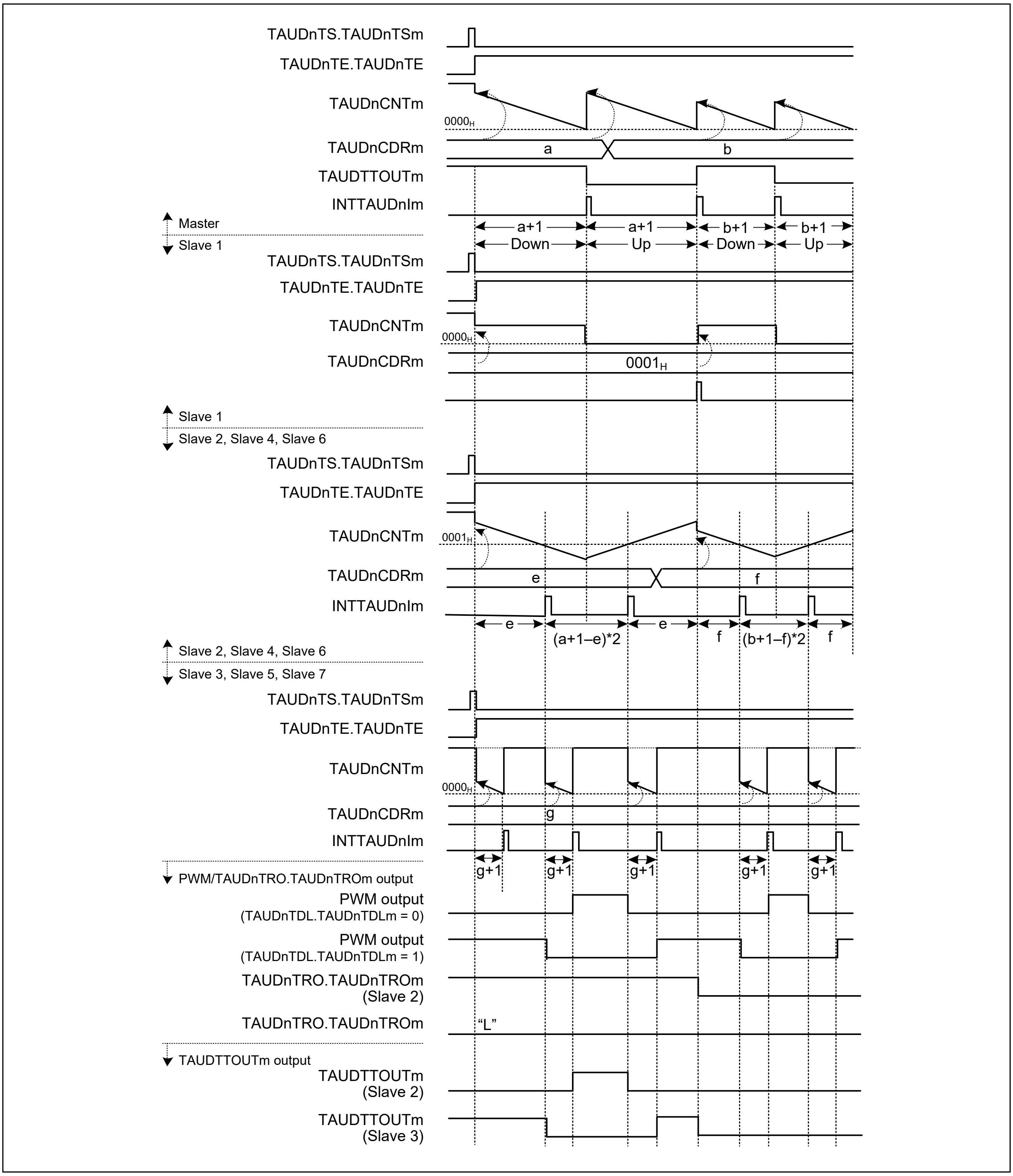


Figure 20.130 General Timing Diagram of Complementary Modulation Output

(4) Register Settings for the Master Channel

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.204 Contents of the TAUDCMORm Register for the Master Channel of the Non-Complementary Modulation Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDMAS	1: Master channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDCOS[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.
4–1	TAUDMD[4:1]	0000: Interval timer mode
0	TAUDMD0	0: INTTAUDIm is not generated and TAUDTTOUTm is not toggled at the beginning and restarting of operation. 1: INTTAUDIm is generated and TAUDTTOUTm is toggled at the beginning and restarting of operation.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.205 Contents of the TAUDCMURm Register for the Master Channel of the Non-Complementary Modulation Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

Table 20.206 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	0: Independent channel output
TAUDTOC.TAUDTOCm	0: Operating mode 1 (Toggle mode if TAUDTOM.TAUDTOMm = 0)
TAUDTOL.TAUDTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDTDE.TAUDTDEm	0: Disables dead time operation
TAUDTDM.TAUDTDMm	0: When dead time operation is disabled (TAUDTDE.TAUDTDEm = 0), set these bits to 0
TAUDTDL.TAUDTDLm	
TAUDTRE.TAUDTREm	0: Disables real-time output
TAUDTRO.TAUDTROm	0: When real-time output is disabled (TAUDTRE.TAUDTREm = 0), set these bits to 0
TAUDTRC.TAUDTRCm	
TAUDTME.TAUDTMEm	0: Disables modulation

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.207 Simultaneous Reload Settings for the Master Channel of Complementary Modulation Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: Simultaneous reload trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

Remark: If TAUDRDS.TAUDRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous reload trigger signal.

(5) Register Settings for Slave Channel 1

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.208 Contents of the TAUDCMORm Register for Slave Channel 1 of the Non-Complementary Modulation Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	11: INTTAUDIm of the master channel is used as the count clock
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous reloading.
7, 6	TAUDCOS[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	TAUDMD[4:1]	0011: Event count mode
0	TAUDMD0	0: INTTAUDIm is not generated at the beginning of operation or at a restart time.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.209 Contents of the TAUDCMURm Register for Slave Channel 1 of the Non-Complementary Modulation Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Channel output mode

TAUDTOE.TAUDTOEm 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: TAUDTRC.TAUDTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(d) Simultaneous reloading of slave channel 1

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.210 Simultaneous Reload Settings for Slave Channel 1 of Complementary Modulation Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: Simultaneous reload trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

(6) Register Settings for Slave Channels 2, 4, and 6

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.211 Contents of the TAUDCMORm Register for Slave Channel 2, 4, and 6 of the Non-Complementary Modulation Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDCCS[1:0]	11: INTTAUDIm of the master channel is used as the count clock.
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDCOS[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	TAUDMD[4:1]	1001: Count-up/-down mode
0	TAUDMD0	0: INTTAUDIm is not generated at the beginning of operation or at a restart time.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.212 Contents of the TAUDCMURm Register for Slave Channel 2, 4, and 6 of the Non-Complementary Modulation Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Output mode

Table 20.213 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel output
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	1: Enables dead time operation.
TAUDTDM.TAUDTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDTDL.TAUDTDLm are satisfied.
TAUDTDL.TAUDTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDTRE.TAUDTREm	1: Enables real-time output.
TAUDTRO.TAUDTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDTRC.TAUDTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDTME.TAUDTMEm	0: Disables modulation 1: Enables modulation

Caution: At the PWM output, set TAUDTDL.TAUDTDLm exclusively from odd-numbered channels.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.214 Simultaneous Reload Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: A simultaneous reload trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

(7) Register Settings for Slave Channels 3, 5, and 7

(a) TAUDCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDCKS [1:0]		TAUDCCS [1:0]		TAUD MAS	TAUDSTS[2:0]			TAUDCOS [1:0]		0	TAUDMD[4:1]				TAUD MD0

Table 20.215 Contents of the TAUDCMORm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output

Bit Position	Bit Name	Function
15, 14	TAUDCKS[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 TAUDCKS[1:0] bits of the master and slave channels must be identical
13, 12	TAUDCCS[1:0]	11: INTTAUDIm of the master channel is used as the count clock.
11	TAUDMAS	0: Slave channel
10 to 8	TAUDSTS[2:0]	110: Dead time trigger
7, 6	TAUDCOS[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	TAUDMD[4:1]	0100: One-count mode
0	TAUDMD0	1: Enables start trigger detection while counting.

(b) TAUDCMURm

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TAUDTIS[1:0]	

Table 20.216 Contents of the TAUDCMURm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output

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	TAUDTIS[1:0]	00: Unused. Set to 00.

(c) Output mode

Table 20.217 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUDTOE.TAUDTOEm	1: Enables independent channel output mode
TAUDTOM.TAUDTOMm	1: Synchronous channel output
TAUDTOC.TAUDTOCm	1: Operating mode 2
TAUDTOL.TAUDTOLm	0: Positive logic 1: Negative logic
TAUDTDE.TAUDTDEm	1: Enables dead time operation.
TAUDTDM.TAUDTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDTDL.TAUDTDLm are satisfied.
TAUDTDL.TAUDTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDTRE.TAUDTREm	1: Enables real-time output.
TAUDTRO.TAUDTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDTRC.TAUDTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUDTME.TAUDTMEm	0: Disables modulation 1: Enables modulation

Caution: At the PWM output, set TAUDTDL.TAUDTDLm exclusively from even-numbered channels.

(d) Simultaneous reloading

Both the master and slave channels should have the same simultaneous reload settings.

Table 20.218 Simultaneous Reload Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output

Bit Name	Setting
TAUDRDE.TAUDRDEm	1: Enables simultaneous reloading.
TAUDRDS.TAUDRDSm	0: Monitors a trigger for simultaneous reloading of master channel. 1: Monitors a trigger for simultaneous reloading of upper channel other than the channel group.
TAUDRDM.TAUDRDMm	1: Simultaneous reload trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDRDC.TAUDRDCm	0: Does not operate as a simultaneous reload trigger generation channel. Monitors a trigger for simultaneous reloading of master channel when TAUDRDS.TAUDRDSm = 0, even if this bit is set to 0.

(8) Operating Procedure for Complementary Modulation Output Type 1

Table 20.219 Operating Procedure for Complementary Modulation Output

(1/2)

	Operation	TAUD Status
Initial Channel Setting	Master channel: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.16.3(4), Register Settings for the Master Channel. Slave channel 1: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.16.3(5), Register Settings for Slave Channel 1. Slave channels 2, 4, and 6: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.16.3(6), Register Settings for Slave Channels 2, 4, and 6. Slave channels 3, 5, and 7: Set TAUDCMORm/TAUDCMURm register and the channel output mode as described in Section 20.16.3(7), Register Settings for Slave Channels 3, 5, and 7. Set the value of TAUDCDRm register of every channel. Set a pulse cycle using TAUDCDRm of master channel, and an interrupt count of master channel ignored using TAUDCDRm of slave channel 1. Also set a duty width in TAUDCDRm of slave channels 2, 4, and 6, and a dead time delay on slave channels 3, 5, and 7. Set TAUDTRC.TAUDTRCm to 1 on slave channel 1.	Channel operation is stopped.
Start Operation (Restart)	Set TAUDTS.TAUDTSm of master and slave channels to 1 simultaneously. TAUDTS.TAUDTSm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm of master and slave channels is set to 1 and the counter starts counting down.

(2/2)

	Operation	TAUD Status
During Operation	TAUDCDRm, TAUDTRO.TAUDTROm, TAUDTME.TAUDTMEm, and TAUDTDL.TAUDTDLm can be changed at any time. TAUDCNTm and TAUDRSF.TAUDRSFm can be read at any time. TAUDRDT.TAUDRDTm can be changed during operation.	TAUDCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDCNTm to perform counting down. TAUDCDRm value of slave channel 1 is loaded and the counter waits for a master channel interrupt. When the counter of master channel reaches 0000H: • INTTAUDIm is generated. • TAUDCDRm value is reloaded into TAUDCNTm to continue counting down. • TAUDCNTm value of slave channel 1 decrements by 1 and the counter waits for the next master channel interrupt. • TAUDCNTm of slave channels 2, 4, and 6 reloads the TAUDCDRm value, but performs counting in opposite direction. • At the same timing when the TAUDCDRm value of slave channels 2, 4, and 6 is loaded, the TAUDTME.TAUDTMEm 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 0000H. When the interrupt is detected: – TAUDCDRm value is reloaded into TAUDCNTm and the counter waits for the next master channel interrupt. – INTTAUDIm is generated. – TAUDTRO.TAUDTROm is changeable. • When the counter of slave channels 2, 4, and 6 reaches 0001H: – INTTAUDIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is satisfied). – TAUDCDRm value of slave channels 3, 5, and 7 is loaded into TAUDCNTm to perform counting down. • When the counter of slave channels 3, 5, and 7 reaches 0000H: – INTTAUDIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is satisfied).
Stop Operation	Set TAUDTT.TAUDTTm of master and slave channels to 1 simultaneously. TAUDTT.TAUDTTm is a trigger bit, which is automatically cleared to 0.	TAUDTE.TAUDTEm is cleared to 0 and the counter stops. TAUDCNTm and TAUDTTOUTm stop and retain their current values.

(9) Specific Timing Diagrams

The following settings apply to the timing diagram.

- Master channel: INTTAUDIm is not generated at the beginning of operation. (TAUDCMORm.TAUDMD0 = 0)
- Slave channel 1: TAUDCDRm = 0001H
- Slave channels 2 to 7: Positive logic (TAUDTOL.TAUDTOLm = 0)

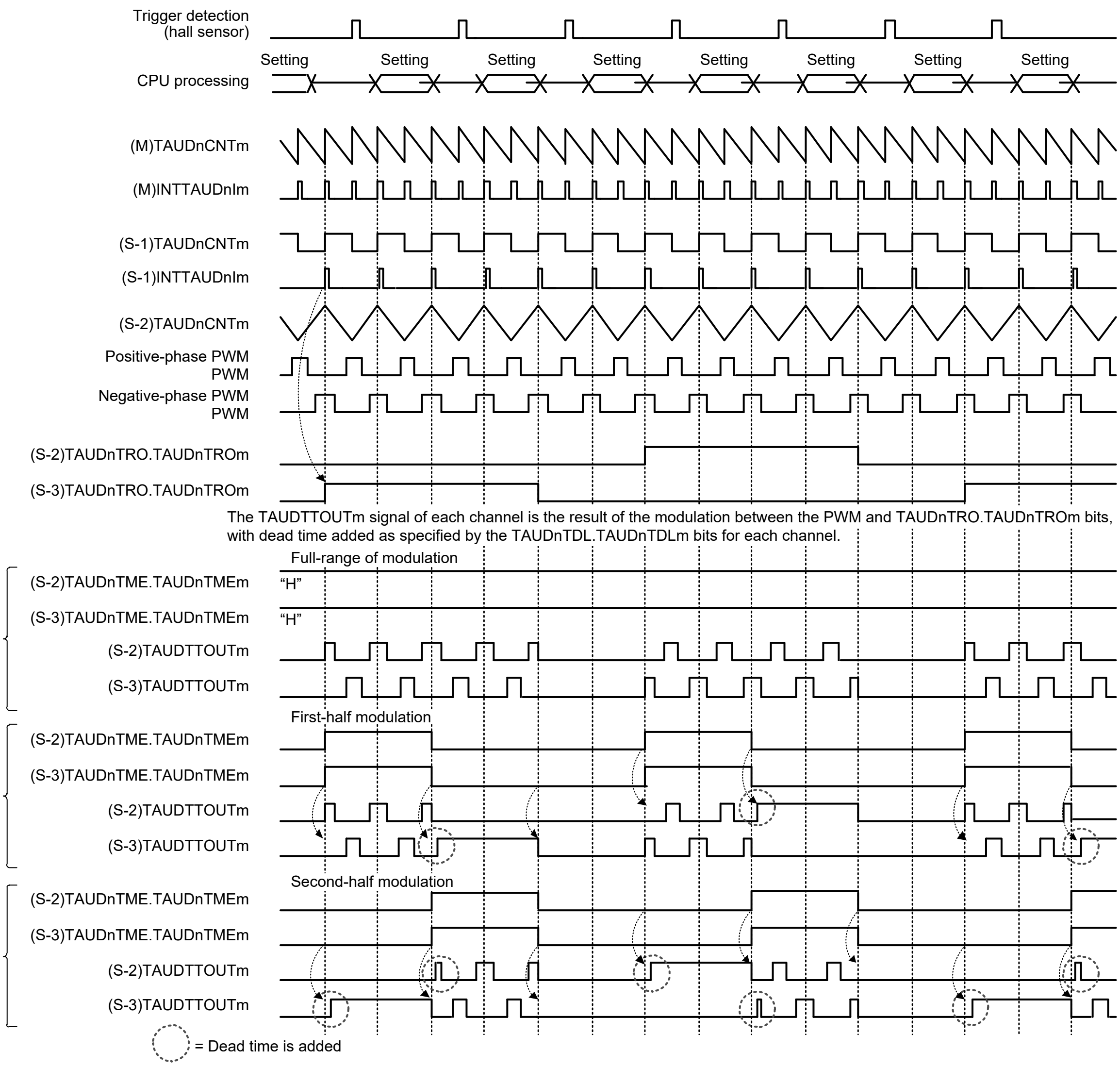


Figure 20.131 Specific Timing Diagram of Complementary Modulation Output

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDTME.TAUDTMEm bits of lower slave channels during operation.

A modulated PWM output signal and TAUDTRO.TAUDTROm bit value are output from slave channels 2 and 3.

TAUDTME.TAUDTMEm and TAUDTDL.TAUDTDLm settings are reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDTRO.TAUDTROm bit value is specified by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

Remark: 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 TAUDCDRm, TAUDTME.TAUDTMEm, TAUDTRO.TAUDTROm, and TAUDTDL.TAUDTDLm can be changed.

21. Window Watchdog Timer A (WDTA)

This section explains window watchdog timer A (WDTA).

21.1 WDTA Features

This microcontroller has the following number of channels of window watchdog timer A.

Table 21.1 Channels of WDTA

Window Watchdog Timer A	
Number of channels	1
Name	WDTA0

- Interrupts and reset outputs
  The interrupts and reset outputs of WDTA0 are listed in the table below.

Table 21.2 WDTA Interrupts and Reset Outputs

WDTA Signals	Function	Connected to
WDTA0		
WDTA0TRES	WDTA0 error reset	Reset Controller WDTA0RES Interrupt controller INTWDTA
WDTA0TNMI	WDTA0 error NMI	Cortex-M4 NMI Input Output as WDTOUTZ through the port (P25) WDTIL input of CC-Link IE Field Network Wdl_n input of CC-Link (remote device station)
INTWDT0	WDTA0 75% interrupt	Interrupt controller INTWDTA

Remark: The WDTA0 error NMI also includes the WDTA0 75% interrupt.

21.2 Functional Overview

The WDTA has the following functions:

- Operation mode after reset selectable by using start-up option
- Fixed software trigger start mode
- Error mode:
  - Generation of NMI request WDTA0TNMI on error detection
  - Generation of reset WDTA0TRES on error detection
- An interrupt request is generated when the counter reaches 75% of its overflow value
- Window function
- Overflow Time
  - 163 µs to 5.36 s (interrupt when the counter reaches 75% of its overflow value)

The following figure shows the main components of WDTA:

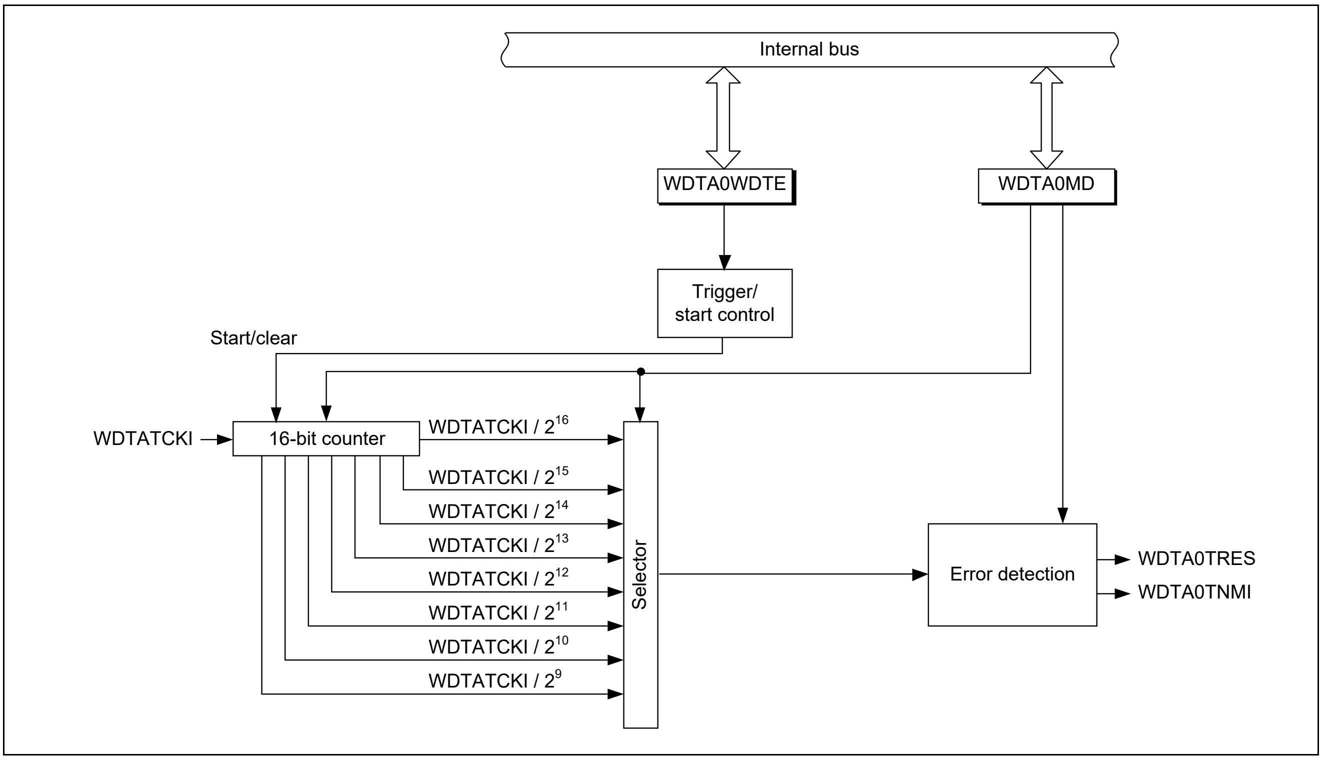


Figure 21.1 Block Diagram of WDTA

21.3 Registers

This section contains a description of all registers of WDTA.

21.3.1 Overview of WDTA Registers

WDTA is controlled and operated by the following registers:

Table 21.3 Overview of WDTA Registers

Register Name	Symbol	Address
WDTA enable register	WDTA0WDTE	4000 0700H
WDTA mode register	WDTA0MD	4000 070CH

21.3.2 Details of WDTA Registers

(1) WDTA enable register (WDTA0WDTE)

This register is the WDTA start control and trigger register.

- WDTA trigger — Writing ACH to this register restarts the counter. Refer to section 21.4.2, WDTA Trigger, for details.
- Access — This register can be read/written in 8-bit units.
- Initial value — WDTA0 is started while it is disabled.

	7	6	5	4	3	2	1	0	Address	Initial value
WDTA0 WDTE	WDTA0 RUN	0	1	0	1	1	0	0	4000 0700H	2CH
R/W	R/W	0	1	0	1	1	0	0		

Bit Position	Bit Name	Function
7	WDTA0RUN	Enables or disables WDTA: 0: WDTA disabled 1: WDTA enabled Since WDTA cannot be stopped once it is started, this bit can only be cleared by a reset.

(2) WDTA mode register (WDTA0MD)

This register specifies the overflow interval time, error mode, and open window size.

It can be updated only once after release from the reset state and before the first trigger. The updated value is effective from the next WDTA trigger.

Changing the value of this register after WDTA has been started leads to an error, but writing the same value to it does not generate an error.

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

	7	6	5	4	3	2	1	0	Address	Initial value
WDTA0MD	0	WDTA0OVF[2:0]			WDTA0 WIE	WDTA0 ERM	WDTA0WS[1:0]		4000 070CH	0FH
R/W	0	R/W			R/W	R/W	R/W			

Bit Position	Bit Name	Description
7	—	Reserved. Writing 0 has no effect. When read, 0 is returned.
6 to 4	WDTA0OVF[2:0]	Selects the overflow interval time: (see table below)
3	WDTA0WIE	Enables or disables the 75% interrupt request. 0: The 75% interrupt request is disabled. 1: The 75% interrupt request is enabled.
2	WDTA0ERM	Specifies the error mode: 0: NMI request mode 1: Reset mode
1, 0	WDTA0WS[1:0]	Select the open window size: (see table below)

WDTA0OVF2	WDTA0OVF1	WDTA0OVF0	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

WDTA0WS1	WDTA0WS0	Open window size
0	0	25%
0	1	50%
1	0	75%
1	1	100%

21.4 Functional Description

WDTA generates a reset or a non-maskable interrupt if the 16-bit counter overflows or if any other error condition is fulfilled. For a description of all error conditions, refer to section 21.4.3, Error Detection.

The counter is cleared and restarted every time a WDTA trigger occurs in the open window period. Refer to section 21.4.2, WDTA Trigger and section 21.4.5, Window Function, for details.

The WDTA can generate an interrupt request (INTWDT0) in response to the value counted reaching 75% of the maximum value. For details, see section 21.4.4, Output of 75% Interrupt.

The start-up option specifies the start mode and WDTA setting after release from the reset state. The setting can be changed by writing to the watchdog timer mode register WDTA0MD. For details, see section 21.4.1, WDTA after Release from the Reset State.

21.4.1 WDTA after Release from the Reset State

(1) Software trigger start mode

The counter value remains 0000H after release from the reset state.
The counter is started with the first WDTA trigger. The first trigger can be generated at any time after release from the reset state.

(2) WDTA settings after release from the reset state

The WDTA settings are between release from the reset state and the first trigger as follows:

Function	Setting	Remark
Counter clock	2^9 / WDTATCKI	For the description of the start modes, refer to section 21.4.1, WDTA after Release from the Reset State.
Error mode	Reset mode	Any error condition before the first trigger generates a reset.
Open window size	100%	If automatic start mode is specified, the first trigger is always valid until the counter overflows.

(a) Changing WDTA settings

After the first trigger, WDTA continues according to the settings of the watchdog timer mode register WDTA0MD.

To change the WDTA settings, data must be written to WDTA0MD before the first trigger. Changing the value of WDTA0MD after the first trigger leads to an error.

If WDTA0MD is not changed before the first trigger, the WDTA mode is specified by the initial value of WDTA0MD.

The new or initial value of WDTA0MD applies after the first trigger.

(b) Software trigger start mode timing

The software trigger start mode timing and changes to the WDTA settings are illustrated in the following figure.

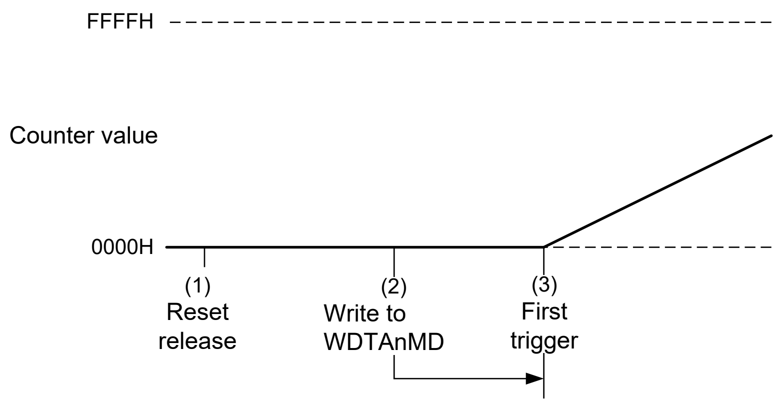


Figure 21.2 Timing Diagram of WDTA Start in Software Trigger Start Mode

The timing diagram above shows the following:

1. After release form the reset state, the counter remains 0000H until the first trigger.
   The counter clock is specified by the start-up options, but it does not have any effect since counting does not proceed.
2. WDTA0MD is written before the first trigger. However, the settings are not applied immediately.
3. The counter starts at the first trigger.
   The counter clock and other settings specified in WDTA0MD are applied.

21.4.2 WDTA Trigger

The following two triggers are selectable as a WDTA trigger:

- Counter start trigger in software trigger start mode
- Counter restart trigger to avoid counter overflow

Writing an activation code to the trigger register leads to generation of a WDTA trigger.

Table 21.4 Trigger Register and Activation Code

Trigger register	Activation code
WDTA0WDTE	ACH

21.4.3 Error Detection

The conditions for error detection are:

- The overflow interval time being exceeded (counter overflow)
- A wrong activation code being written to the trigger register
- Writing to the trigger register in the closed window.
- Illegal update of the watchdog timer mode register WDTA0MD:
  - Writing a new value to WDTA0MD after the first trigger leads to error detection.
  - Writing the same value to WDTA0MD after the first trigger does not lead to error detection.

(1) Error mode

When an error is detected, an NMI request (WDTA0TNMI) or a reset (WDTA0TRES) is generated.

WDTA0MD.WDTA0ERM is used to select the error mode:

- WDTA0MD.WDTA0ERM = 0: NMI mode
- WDTA0MD.WDTA0ERM = 1: reset mode

The following figure shows the reset or NMI request generation when the counter overflows and automatic start mode is selected.

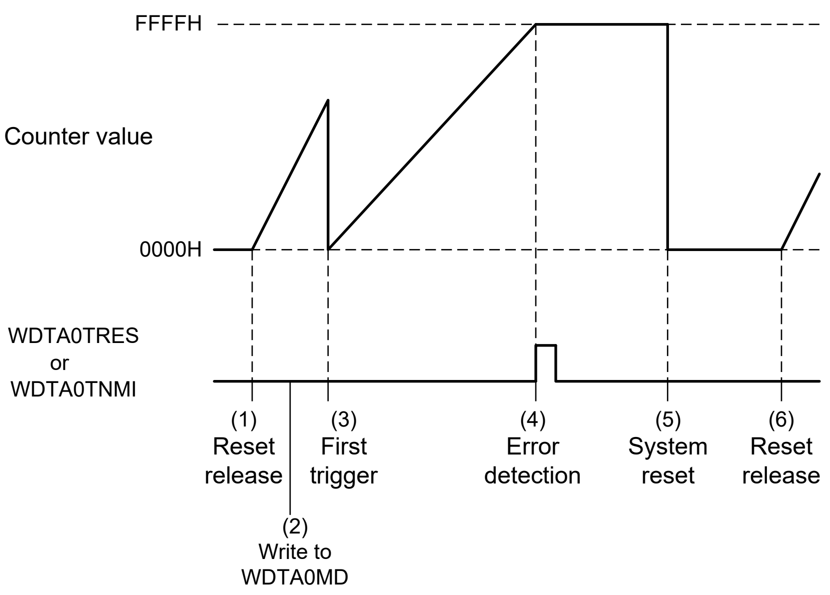


Figure 21.3 Timing Diagram of WDTA NMI Request or Reset Generation

The timing diagram above shows the following:

1. After release from the reset state, the counter starts (if automatic start mode is selected).
2. WDTA0MD is written before the first trigger. However, the settings are not applied immediately.
3. The counter is cleared at the first trigger and the new WDTA settings are applied.
4. When the counter overflows, an error is detected. Depending on the error mode, interrupt request WDTA0TNMI or reset WDTA0TRES is generated.
   The counter value remains until the system is reset.
5. When the system is reset, the counter is cleared and stopped until release from the reset state.

21.4.4 Output of 75% Interrupt

An INTWDT0 interrupt request can be generated when the value counted reaches 75% of the maximum value.

This function is enabled or disabled by the setting of the WDTA0MD.WDTA0WIE bit.

The figure below shows the generation of 75% interrupt requests under the following conditions.

- Default start mode being selected
- The counter clock being changed after the first trigger

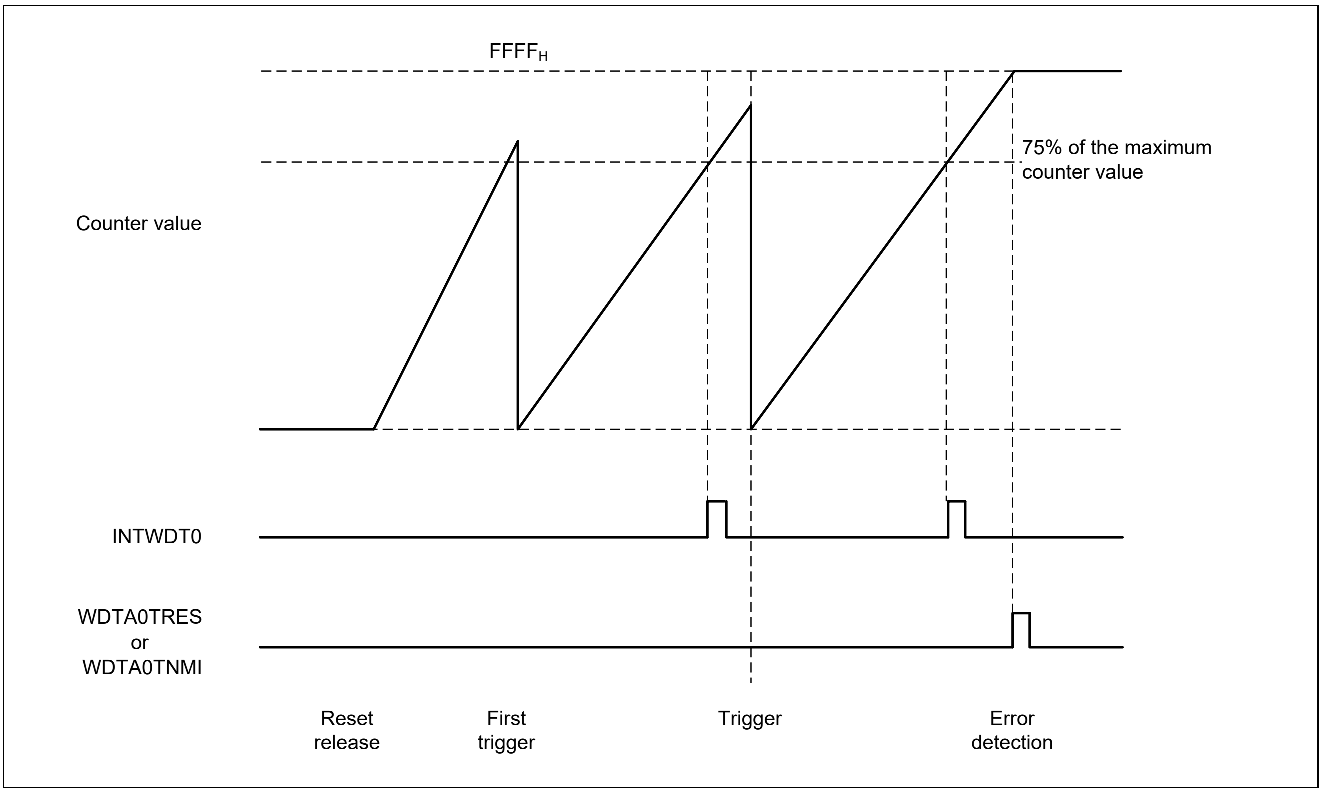


Figure 21.4 Timing Diagram of Output of WDTA 75% Interrupt

21.4.5 Window Function

When the open window size is set to less than 100%, an error is detected if the trigger occurs in the closed window.

The setting of the open window size differs before and after the first trigger:

- After release from the reset state, the open window size is 100%.
  The settings of the OPWDWS[1:0] and WDTA0MD.WDTA0WS[1:0] bits are ineffective.
- After the first trigger, the open window size is the value specified by the WDTA0MD.WDTA0WS[1:0] bits.

The following figure shows WDTA operation with an open window size of 25% and with automatic start mode selected.

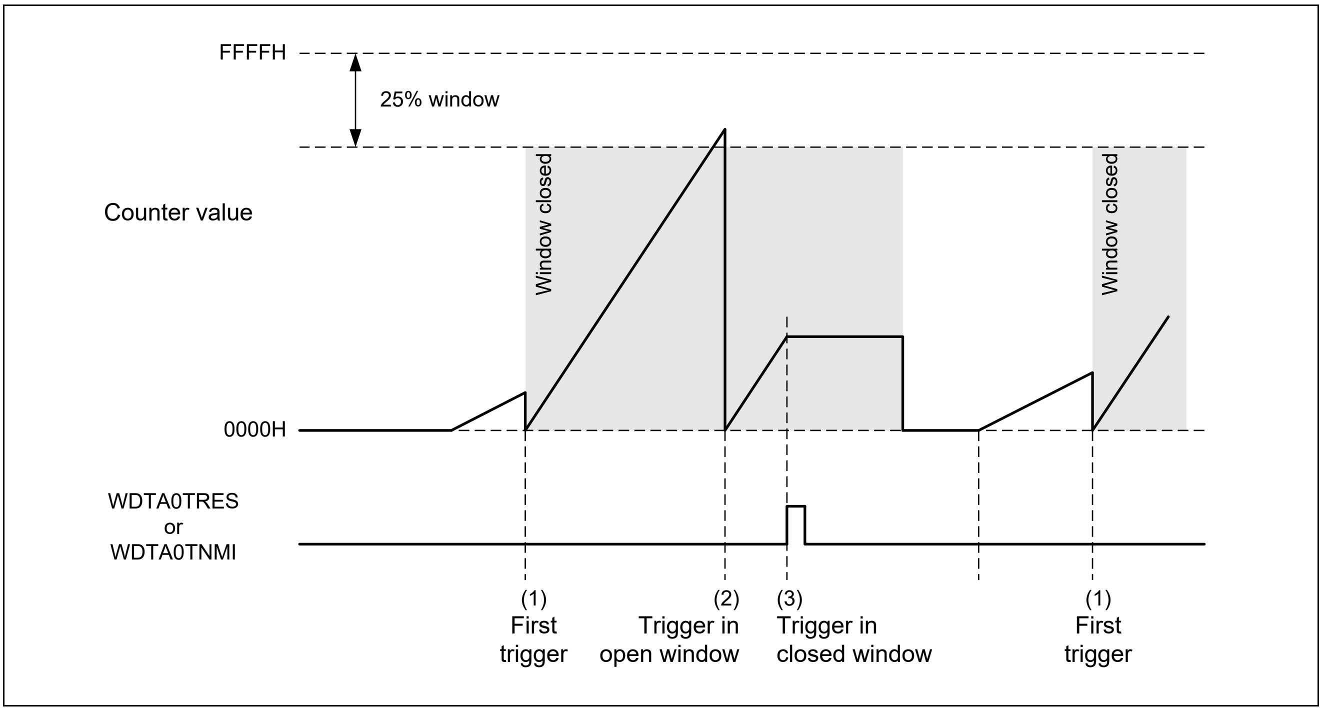


Figure 21.5 Timing Diagram of WDTA Window Function

The timing diagram above shows the following:

1. The open window size is fixed to 100% for the first trigger.
2. A trigger that occurs in the open window does not generate an error.
3. A trigger that occurs in the closed window generates a WDTA0TNMI request or a WDTA0TRES reset, depending on the selected operating mode.

21.5 WDTOUTZ Output

If a WDTA0TNMI interrupt occurs and port pin P25 is set for multiplexed function 1, the low level will be output on this pin. Once the low level is output on WDTOUTZ, the level on the pin will not change from the low level until the input of the reset signal on the RESETZ pin or from the SYSRESET register.

21.6 Notes

(1) An NMI interrupt on a timeout of WDTA may occur successively.
The period during which an NMI interrupt occurs successively depends on the value of the watchdog timer input select register (WDTCLKCFG).

22. Asynchronous Serial Interface J (UARTJ)

This section explains asynchronous serial interface J (UARTJ).

22.1 UARTJ Features

- Number of channels: This microcontroller has two channels of asynchronous serial interface J (UARTJn).

Table 22.1 Channels of UARTJn

Asynchronous Serial Interface J	
Number of channels	2
Name	UARTJ0, UARTJ1

- Bit rate: By setting a prescaler and a bit rate generator, a bit rate is selectable in the range from 300 bps to 12500000 bps.
- Index n: Throughout this section, the individual channels of UARTJ are identified by the index "n" (n = 0 to 3); for example, UARTJnCTL0 for UARTJn control register 0.
- I/O signals: The I/O signals of the asynchronous serial interface J are listed in the table below.

Table 22.2 UARTJn I/O Signals

UARTJn Signals	Function	Connected to
URTJ0TTXD	Transmit data output	Port21 TXD0
URTJ0TRXD	Receive data input	Port20 RXD0
URTJ1TTXD	Transmit data output	Port31 TXD1
URTJ1TRXD	Receive data input	Port30 RXD1

- Interrupts and peripheral modules:

  The following interrupt requests from UARTJ can be used as triggers for interrupt service routines or hardware ISRs (where listed as such), for DMA transfer (by the general-purpose DMAC or real-time port DMAC), for capture by a timer (TAUJ2 or TAUD), and for updating the real-time port pins (RP00–RP37).

Table 22.3 UARTJn Interrupts

UARTJn Signals	Function	Connected to
UARTJ0		
INTUAJ0TIT	Transmission interrupt	• Interrupt controller INTUAJ0TIT • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDFTR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTUAJ0TIR	Reception interrupt	• Interrupt controller INTUAJ0TIR • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDFTR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTUAJ0TIS	Status interrupt	• Interrupt controller INTUAJ0TIS • HW-RTOS (Hardware ISR)
UARTJ1		
INTUAJ1TIT	Transmission interrupt	• Interrupt controller INTUAJ1TIT • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDFTR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTUAJ1TIR	Reception interrupt	• Interrupt controller INTUAJ1TIR • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDFTR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTUAJ1TIS	Status interrupt	• Interrupt controller INTUAJ1TIS

22.2 Functional Overview

- Full-duplex communications via built-in receive and transmit FIFOs:
  - Internal UARTJn 10 bit × 16 receive data FIFO (URTJnFRX)
  - Internal UARTJn 8 bit × 16 transmit data FIFO (URTJnFTX)
- 2-pin configuration:
  - URTJnTTXD: Transmit data output pin
  - URTJnTRXD: Receive data input pin
- Various error detection functions
  - Rx parity error
  - Rx framing error
  - Tx data consistency error
- Tx FIFO overflow error
  - Rx FIFO overrun error
  - Rx timeout error
  - Rx BF receive error
- Various FIFO status information
  - Rx FIFO empty/empty status
  - Tx FIFO empty/empty status
  - Rx FIFO fill level
  - Tx FIFO fill level
- Interrupt requests: 3
  - Transmission interrupt INTUAJnTIT
  - Reception interrupt INTUAJnTIR
  - Status interrupt INTUAJnTIS
- Character length: 7, 8 bits
- Parity function: odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- MSB-/LSB-first transfer selectable
- Transmit/receive data inverted input/output possible
- 13 to 20 bits selectable for the BF (Break Field) in the LIN (Local Interconnect Network) communication format
  - Recognition of 11 bits or more possible for BF reception in LIN communication format
  - BF reception flag provided
- BF reception can be detected during data communications
- Bus monitor function to keep data consistency of the transmit data

22.3 Configuration

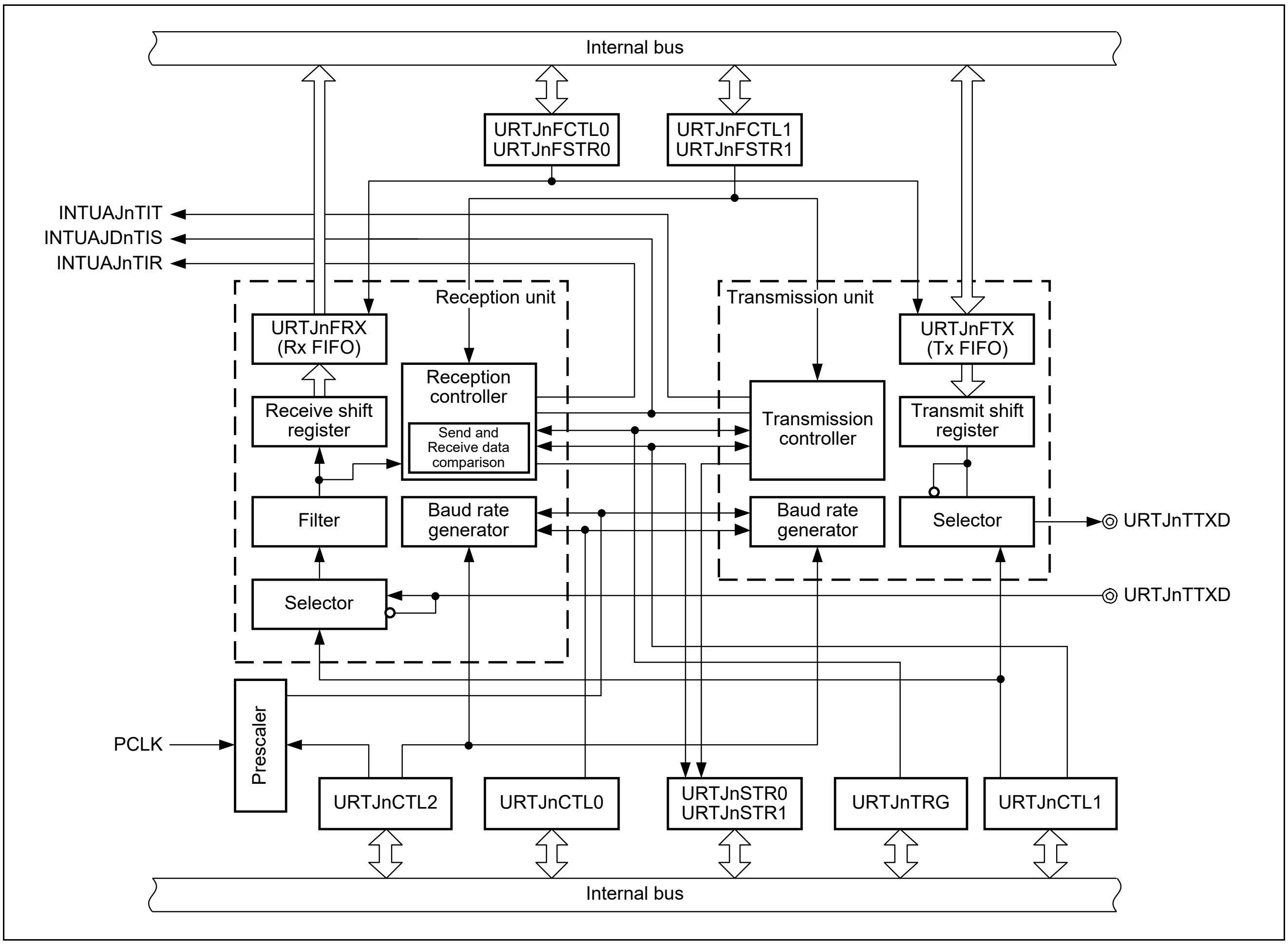


Figure 22.1 Block Diagram of Asynchronous Serial Interface UARTJn

22.4 UARTJn Registers

UARTJn is controlled and operated by using the following registers:

Table 22.4 UARTJn Registers

Register name	Symbol	Address
UARTJ0 control register 0	URTJ0CTL0	4000 0300H
UARTJ0 control register 1	URTJ0CTL1	4000 0320H
UARTJ0 control register 2	URTJ0CTL2	4000 0324H
UARTJ0 trigger register	URTJ0TRG	4000 0304H
UARTJ0 status register 0	URTJ0STR0	4000 0308H
UARTJ0 status register 1	URTJ0STR1	4000 030CH
UARTJ0 Status clear register	URTJ0STC	4000 0310H
UARTJ0 FIFO control register 0	URTJ0FCTL0	4000 0380H
UARTJ0 FIFO control register 1	URTJ0FCTL1	4000 03A0H
UARTJ0 FIFO status register 0	URTJ0FSTR0	4000 0384H
UARTJ0 FIFO status register 1	URTJ0FSTR1	4000 0388H
UARTJ0 FIFO status clear register	URTJ0FSTC	4000 038CH
UARTJ0 FIFO receive data register	URTJ0FRX	4000 0390H
UARTJ0 FIFO transmit data register	URTJ0FTX	4000 0394H
UARTJ1 control register 0	URTJ1CTL0	4000 0400H
UARTJ1 control register 1	URTJ1CTL1	4000 0420H
UARTJ1 control register 2	URTJ1CTL2	4000 0424H
UARTJ1 trigger register	URTJ1TRG	4000 0404H
UARTJ1 status register 0	URTJ1STR0	4000 0408H
UARTJ1 status register 1	URTJ1STR1	4000 040CH
UARTJ1 status clear register	URTJ1STC	4000 0410H
UARTJ1 FIFO control register 0	URTJ1FCTL0	4000 0480H
UARTJ1 FIFO control register 1	URTJ1FCTL1	4000 04A0H
UARTJ1 FIFO status register 0	URTJ1FSTR0	4000 0484H
UARTJ1 FIFO status register 1	URTJ1FSTR1	4000 0488H
UARTJ1 FIFO status clear register	URTJ1FSTC	4000 048CH
UARTJ1 FIFO receive data register	URTJ1FRX	4000 0490H
UARTJ1 FIFO transmit data register	URTJ1FTX	4000 0494H

(1) UARTJn control register 0 (URTJnCTL0)

This register controls UARTJn the basic serial transfer operation.

- Access This register can be read/written in 32-bit and 1-bit units.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJnCTL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJnPW	URTJnTXE	URTJnRXE	0	0	0	0	URTJnSLDC
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	0	0	0	0	R/W

Address
4000 0300H
+100H × n
Initial Value
0000 0000H

Bit Position	Bit Name	Function
31 to 8	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
7	URTJnPW	UARTJn enable 0: Stop UARTJn operation 1: Enable UARTJn operation Changing the value of this bit initializes all transmission and reception units.
6	URTJnTXE	Transmission operation enable 0: Disable transmission operation 1: Enable transmission operation • To start transmission, set URTJnPW and then set URTJnTXE. To stop transmission, clear URTJnTXE, and then clear URTJnPW (they can be cleared at the same time). • To initialize the transmission unit, clear URTJnTXE, wait for two prescaler clock cycles, and then set URTJnTXE again. For details about the prescaler clock, see (3), UARTJn control register 2 (URTJnCTL2).
5	URTJnRXE	Reception enable 0: Disable reception operation 1: Enable reception operation • To enable reception, set URTJnPW, and then set URTJnRXE. To stop reception, clear URTJnRXE, and then clear URTJnPW (they can be cleared at the same time). • To initialize the reception unit, clear URTJnRXE, wait for two prescaler clock cycles, and then set URTJnRXE again. Reception is enabled when the time of two prescaler clock cycles has elapsed since URTJnRXE is set. The rising edge detection of the URTJnTRXD signal is enabled when four prescaler clock cycles has elapsed after URTJnRXE is set. For details about the prescaler clock, see (3), UARTJn control register 2 (URTJnCTL2).
4 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
0	URTJnSLDC	Data consistency check enable 0: Disable consistency check 1: Enable consistency check This bit selects the handling of data consistency error checks when transmitting data. When this bit is set to 1, the transmit data and receive data are compared, and if a mismatch is detected, URTJnSTR1.URTJnDCE is set to 1 and a status interrupt request INTUAJnTIS is issued. This bit is referenced only when starting transmission. Consequently, if this bit value is changed later on during transmission processing, the subsequent transmission processing proceeds in accord with the setting at the start of transmission.

Remark: n = 0, 1

Cautions 1. Disable transmission if UARTJn meets all the conditions below:

- **Transmission and reception are enabled (URTJnCTL0.URTJnPW = URTJnRXE = URTJnTXE = 1).**
- **Data consistency check is enabled (URTJnCTL0.URTJnSLDC = 1).**
- **Data is being transmitted or has been transmitted.**

Use the following procedure to keep reception enabled:

- **Check that no data is pending for transmission (URTJnSTR0.URTJnSSBT = URTJnSST = 0).**
- **Check that no data is pending for reception (URTJnSTR0.URTJnSSBR = URTJnSSR = 0).**
- **Disable transmission by clearing URTJnCTL0.URTJnTXE.**

The reason why this procedure is required is that the data consistency error flag URTJnSTR1.URTJnDCE is cleared if URTJnCTL0.URTJnTXE is cleared.

Thus a potential data consistency error would not occur if transmission is disabled during a data transfer or after its completion.

2. Disable reception if UARTJn meets all the conditions below:

- **Transmission and reception are enabled (URTJnCTL0.URTJnPW = URTJnRXE = URTJnTXE = 1).**
- **Data consistency check is enabled (URTJnCTL0.URTJnSLDC = 1).**
- **Data is being transmitted or has been transmitted.**

Use the following procedure to keep transmission enabled:

- **Check that no data is pending for transmission (URTJnSTR0.URTJnSSBT = URTJnSST = 0).**
- **Check that no data is pending for reception (URTJnSTR0.URTJnSSBR = URTJnSSR = 0).**
- **Disable reception by clearing URTJnCTL0.URTJnRXE.**

The reason why this procedure is required is that the data consistency error flag URTJnSTR1.URTJnDCE is cleared and invalid if URTJnCTL0.URTJnTXE is cleared. Thus a potential data consistency error of already transmitted data would not occur.

3. Do not start data transmission if all the conditions below are met:

- **Data consistency check is enabled (URTJnCTL0.URTJnSLDC = 1).**
- **BF reception is enabled (URTJnSTR0.URTJnSSBR = 1).**
- **BF detection during reception is disabled (URTJnCTL1.URTJnSLBM = 0).**

A data consistency error will occur under above conditions when BF reception is completed. The status interrupt INTUAJnTIS will be generated and completion of BF reception will not be reported (URTJnSTR1.URTJnBSF remains 0). Consequently BF reception completion will not be recognized.

(2) UARTJn control register 1 (URTJnCTL1)

This register defines the data frame properties of UARTJn serial data transfers.

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

(1/3)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJnCTL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJnSLBM	URTJnBLG[2:0]			0	0	0	URTJnCLG	URTJnSLP[1:0]		URTJnTDL	URTJnRDL	0	URTJnSLG	URTJnSLD	URTJnSLIT
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W			0	0	0	R/W	R/W		R/W	R/W	0	R/W	R/W	R/W

Address: 4000 0320H +100H × n

Initial Value: 0000 5002H

Bit Position	Bit Name	Function
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15	URTJnSLBM	BF receive mode selection 0: BF reception during data reception disabled. 1: BF reception during data reception enabled. • Changing this bit is only allowed if reception is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = 0).
14 to 12	URTJnBLG[2:0]	BF bit length during transmission (see table below) Changing these bits is only allowed if transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnTXE = 0).
11 to 9	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
8	URTJnCLG	Receive/transmit data bit length 0: 7 bits 1: 8 bits • When the transmission/reception is performed in the LIN format, set URTJnCLG to 1. • Changing this bit is only allowed if reception and transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = URTJnCTL0.URTJnTXE = 0).

URTJnBLG2	URTJnBLG1	URTJnBLG0	BF length
1	0	1	13 bits
1	1	0	14 bits
1	1	1	15 bits
0	0	0	16 bits
0	0	1	17 bits
0	1	0	18 bits
0	1	1	19 bits
1	0	0	20 bits

Remark: n = 0, 1

(2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>URTJnSLP[1:0]</td><td>Parity bit selection
<table><tr><th rowspan="2">URTJnSLP1</th><th rowspan="2">URTJnSLP0</th><th colspan="2">Operation</th></tr><tr><th>Transmission</th><th>Reception</th></tr><tr><td>0</td><td>0</td><td>Output without parity bit</td><td>Received with no parity</td></tr><tr><td>0</td><td>1</td><td>Output 0 parity (0-fixed)</td><td>No parity judgment</td></tr><tr><td>1</td><td>0</td><td>Output odd parity</td><td>Judged as odd parity</td></tr><tr><td>1</td><td>1</td><td>Output even parity</td><td>Judged as even parity</td></tr></table>• If "Reception with no parity judgment" is selected during reception, a parity check is not performed. Therefore, since the URTJnSTR1.URTJnPE bit is not set, no error interrupt is output.
• When transmission/reception is performed in the LIN format, set URTJnSLP[1:0] to 00B.
• Changing these bits is only allowed if reception and transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = URTJnCTL0.URTJnTXE = 0).</td></tr>
<tr><td>5</td><td>URTJnTDL</td><td>Transmission data level control
0: No inverted output of transmit data
1: Inverted output of transmit data
• The output level of the URTJnTTXD pin can be inverted using this bit. It inverts the URTJnTTXD output level immediately, regardless of the values of URTJnCTL0.URTJnPW and URTJnCTL0.URTJnTXE. Therefore, if URTJnTDL is set to 1 while the operation is disabled, the URTJnTTXD outputs low level.
• Changing this bit is only allowed if transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnTXE = 0).</td></tr>
<tr><td>4</td><td>URTJnRDL</td><td>Reception data level control
0: No inverted output of receive data
1: Inverted output of receive data
• The output level of the URTJnTRXD pin can be inverted using this bit. It inverts the URTJnTRXD input level immediately, regardless of the values of URTJnCTL0.URTJnPW and URTJnCTL0.URTJnRXE. Therefore, if URTJnRDL is set to 1 while the operation is disabled, the URTJnTRXD inputs low level.
• Changing this bit is only allowed if reception is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = 0).</td></tr>
<tr><td>3</td><td>—</td><td>Reserved. When writing to this bit, write 0. When read, 0 is returned.</td></tr>
<tr><td>2</td><td>URTJnSLG</td><td>Stop bit number selection for transmission data
0: 1 bit
1: 2 bits
• The stop bit length during data or BF reception is always handled as "1".
• Changing this bit is only allowed if transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnTXE = 0).</td></tr>
<tr><td>1</td><td>URTJnSLD</td><td>Transfer direction selection
0: MSB-first transfer
1: LSB-first transfer
• When the transmission/reception is performed in the LIN format, set URTJnSLD to 1.
• Changing this bit is only allowed if reception and transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = URTJnCTL0.URTJnTXE = 0).</td></tr>
</table>

Remark: n = 0, 1

(3/3)

Bit Position	Bit Name	Function
0	URTJnSLIT	Transmission interrupt request (INTUAJnTIT) timing selection 0: INTUAJnTIT generated at the start of transmission, i.e. when the transmit data is stored in the transmission shift register 1: INTUAJnTIT generated at transmission completion • Changing this bit is only allowed if transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnTXE = 0).

Remark: n = 0, 1

(3) UARTJn control register 2 (URTJnCTL2)

This register specifies the bit rate for UARTJn serial data transfer.

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

Caution: Writing to this register is only allowed if the UARTJn operation is disabled (URTJnCTL0.URTJnPW = 0).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–13	12	11–0
URTJnCTL2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJnPRS[2:0]	0	URTJnBRS[11:0]
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	0	R/W

Address 4000 0324H +100H × n

Initial Value 0000 EFFFH

Bit Position	Bit Name	Function
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 13	URTJnPRS[2:0]	Prescaler clock (PRSCLK) division value 0: PRSCLK = PCLK / 2^0 1: PRSCLK = PCLK / 2^1 2: PRSCLK = PCLK / 2^2 3: PRSCLK = PCLK / 2^3 4: PRSCLK = PCLK / 2^4 5: PRSCLK = PCLK / 2^5 6: PRSCLK = PCLK / 2^6 7: PRSCLK = PCLK / 2^7
12	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.
11 to 0	URTJnBRS[11:0]	Bit rate clock (BRCLK) division value

URTJnBRS[11:0]	Transmit/receive BRCLK	BF receive clock
000H	PRSCLK/(2 × 4)	PRSCLK/4
001H		
002H		
003H		
004H		
005H	PRSCLK/(2 × 5)	PRSCLK/5
...	PRSCLK/(2 × URTJnBRS[11:0])	PRSCLK/URTJnBRS[11:0]
FFEH	PRSCLK/(2 × 4094)	PRSCLK/4094
FFFH	PRSCLK/(2 × 4095)	PRSCLK/4095

Remark: n = 0, 1

(4) UARTJn trigger register (URTJnTRG)

This register controls the UARTJn transmission/reception trigger of BF.

- Access This register can be read/written in 32-bit and 1-bit units.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJnTRG	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	URTJnBRT	URTJnBTT	0	0	0	0	0
R/W	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	R/W	R/W	0	0	0	0	0

Address
4000 0304H
+100H × n
Initial Value
0000 0000H

Bit Position	Bit Name	Function
31 to 7	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
6	URTJnBRT	BF reception trigger 0: Read value is always 0, writing 0 is ignored 1: Enable BF reception • When reception is enabled, writing 1 to this bit enables BF reception (URTJnSTR0.URTJnSSBR = 1) and BF reception processing begins when the falling edge of the receive serial signal is detected. • If 1 is written to this bit during reception processing, the current reception processing is terminated. Consequently, the received data is not stored, the framing, parity and overflow error bits are not updated based on the data that was being received and no interrupts are generated. Meanwhile, the BF counter value is continuously being used. • After BF reception, the reception status is set according to the URTJnCTL1.URTJnSLBM setting. • Setting this bit to 1 is only allowed if reception is enabled (URTJnCTL0.URTJnPW = URTJnCTL0.URTJnRXE = 1). After URTJnBRT is set to 1, completion of BF reception is reported by either of the following two methods, based on the URTJnCTL1.URTJnSLBM setting: • if URTJnCTL1.URTJnSLBM = 0 When BF reception is complete, the reception interrupt request INTUAJnTIR is output. • if URTJnCTL1.URTJnSLBM = 1 When BF reception is complete, URTJnSTR1.URTJnBSF is set to 1 and a status interrupt request INTUAJnTIS is output.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
5	URTJnBTT	BF transmission trigger 0: Read value is always 0, writing 0 is ignored 1: Enable BF transmission • When this bit is set while URTJnSTR0.URTJnSSBT = 0 and transmission is enabled (URTJnDCE = 0), a BF transmit request is set, and URTJnSSBT is set to 1. • When this bit is set during data transmission, a BF is transmitted after the current transmission processing is completed. Even if this bit is set before the BF transmission is completed, a BF is transmitted only once. • When transmission is enabled (URTJnPW = URTJnTXE = 1), setting this bit clears all previously set data transmit requests (which have not been transmitted), leaving only BF transmit requests. If the URTJnTX7 to URTJnTX0 bits are written after writing 1 to this bit, data is transmitted after the BF is transmitted. • If both a BF transmit request and a data transmit request have been set when transmission starts, the BF transmission takes priority. • When URTJnDCE = 1, writing 1 to this bit is ignored. • Setting this bit to 1 is only allowed if transmission is enabled (URTJnCTL0.URTJnPW = URTJnCTL0.URTJnTXE = 1).
4 to 0	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.

Remark: n = 0, 1

(5) UARTJn status register 0 (URTJnSTR0)

This register indicates the current status of serial data transmissions.

- Access This register can be read in 32-bit and 1-bit units.

Writing to this register is only allowed if UARTJn operation is disabled (URTJnCTL0.URTJnPW = 0). If UARTJn operation is enabled (URTJnCTL0.URTJnPW = 1), any written values are ignored and the initial values are restored.

This register is initialized by any reset and when URTJnCTL0.URTJnPW is set or cleared.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJnSTR0	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	URTJnSSBR	URTJnSSBT	0	0	0	URTJnSSR	URTJnSST
R/W	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	R	R	0	0	0	R	R

Address
4000 0308H
+100H × n
Initial Value
0000 0000H

Bit Position	Bit Name	Function
31 to 7	—	Reserved. These bits are read as 0.
6	URTJnSSBR Note1	BF reception enable status 0: BF reception is disabled 1: BF reception is enabled by setting URTJnTRG.URTJnBRT to 1 (BF reception standby mode or BF reception busy).
5	URTJnSSBT Note2	BF transmission enable status 0: BF transmission is disabled 1: BF transmission is enabled by setting URTJnTRG.URTJnBTT to 1 (BF transmission standby mode or BF transmission busy).
4 to 2	—	Reserved. These bits are read as 0.
1	URTJnSSR Note2	Data reception status 0: No data reception ongoing 1: Data reception ongoing (data reception busy)
0	URTJnSST Note2	Data transmission status 0: No transmission pending or ongoing 1: Data in URTJnTX[7:0] pending to be transmitted or transmission ongoing

Notes 1. This bit is also initialized when reception is disabled by setting URTJnCTL0.URTJnRXE = 0.

2. This bit is also initialized when transmission is disabled by setting URTJnCTL0.URTJnTXE = 0.

Remark: n = 0, 1

(6) UARTJn status register 1 (URTJnSTR1)

This register indicates results of serial data transmission.

- Access This register can be read in 32-bit and 1-bit units.

Writing to this register is only allowed if UARTJn operation is disabled (URTJnCTL0.URTJnPW = 0). If UARTJn operation is enabled (URTJnCTL0.URTJnPW = 1), any written values are ignored and the initial values are restored.

This register is initialized by any reset and when URTJnCTL0.URTJnPW is set or cleared.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJnSTR1	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	URTJnBSF	URTJnDCE	URTJnPE	URTJnFE	0
R/W	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	R	R	R	R	0

Address: 4000 030CH +100H × n

Initial Value: 0000 0000H

Bit Position	Bit Name	Function
31 to 5	—	Reserved. These bits are read as 0.
4	URTJnBSF Note1	BF reception successful flag 0: BF transmission is disabled by clearing URTJnTRG.URTJnBTT. 1: BF transmission is enabled by setting URTJnTRG.URTJnBTT (BF transmission standby mode or BF transmission busy). The URTJnBSF bit is cleared by the following: • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnRXE = 0 • URTJnSTC.URTJnCLBS = 1
3	URTJnDCE Note2	Data consistency error flag 0: Transmit/receive data (transmit/receive BF) mismatch was not detected. 1: Transmit/receive data (transmit/receive BF) mismatch was detected. When the BF receive mode selection bit is set during LIN communication, it is necessary to read this bit by using status interrupt processing and to confirm the beginning of a new frame slot. The URTJnDCE bit is cleared by the following: • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnTXE = 0 • URTJnSTC.URTJnCLDC = 1

Notes 1. These bits are also initialized when reception is disabled by setting URTJnCTL0.URTJnRXE = 0.

2. This bit is also initialized when transmission is disabled by setting URTJnCTL0.URTJnTXE = 0.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
2	URTJnPE Note1	Parity error flag 0: No parity error was detected in the received data. 1: A parity error was detected in the received data. The operation of URTJnPE is controlled by the settings of URTJn.URTJnSLP[1:0]. The URTJnPE bit is cleared by the following: • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnRXE = 0 • URTJnSTC.URTJnCLP = 1
1	URTJnFE Note1	Framing error flag 0: No framing error was detected in the received data. 1: A framing error was detected in the received data. The URTJnFE bit is cleared by the following: • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnRXE = 0 • URTJnSTC.URTJnCLF = 1
0	—	Reserved. This bit is read as 0.

Notes 1. These bits are also initialized when reception is disabled by setting URTJnCTL0.URTJnRXE = 0.

Remark: n = 0, 1

Remark: If the bits of these registers are set (1) and cleared (0) at the same time, setting takes priority over clearing.
For further information concerning error detections, refer to section 22.6.5, UARTJn Transmission and section 22.6.7, Reception Errors.

Caution: In case reception and transmission is enabled and a consistency check error occurs (URTJnSTR1.URTJnDCE = 1), follow the procedure below prior next data transmission:

- **disable transmission by URTJnCTL0.URTJnTXE = 0**
- **initiate transmission by URTJnTRG.URTJnBTT = 1 (BT transmission trigger) or writing any data to URTJnFTX**
- **enable transmission by URTJnCTL0.URTJnTXE = 1 Afterwards new transmissions can be started.**

(7) UARTJn status clear register (URTJnSTC)

This register is used to clear the status bits of the status register 1 (URTJnSTR1).

- Access This register can be read/written in 32-bit and 1-bit units.

Reading this register always returns 00H.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJnSTC	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	URTJnCLBS	URTJnCLDC	URTJnCLP	URTJnCLF	0
R/W	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	R/W	R/W	R/W	R/W	0

Address: 4000 0310H +100H × n

Initial Value: 0000 0000H

Bit Position	Bit Name	Function
31 to 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4	URTJnCLBS	Clear BF reception successful flag 0: writing 0 is ignored 1: writing 1 clears URTJnSTR1.URTJnBSF
3	URTJnCLDC	Clear data consistency error flag 0: writing 0 is ignored 1: writing 1 clears URTJnSTR1.URTJnDCE If URTJnDCE is cleared by setting this bit, any pending data or BF transmit requests will be ignored.
2	URTJnCLP	Clear parity error flag 0: writing 0 is ignored 1: writing 1 clears URTJnSTR1.URTJnPE
1	URTJnCLF	Clear framing error flag 0: writing 0 is ignored 1: writing 1 clears URTJnSTR1.URTJnFE
0	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.

Remark: n = 0, 1

(8) FIFO control register 0 (URTJnFCTL0)

This register defines the fill stage of the Rx and Tx FIFOs, at which the reception (INTUAJnTIR) and transmission (INTUAJnTIT) interrupt requests are generated.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJn FCTL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJnSLRP[3:0]				0	0	0	0	URTJnSLTP[3:0]			
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W				0	0	0	0	R/W			

Address: 4000 0380H +100H × n

Initial Value: 0000 0F00H

Bit Position	Bit Name	Function
31 to 12	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
11 to 8	URTJnSLRP[3:0]	Rx FIFO level interrupt setting URTJnSLRP[3:0] defines the Rx FIFO pointer status, at which the reception interrupt request INTUAJnTIR is generated. INTUAJnTIR is generated if URTJnFSTR0.URTJnSSRW[4:0] = (10H - URTJnSLRP[3:0]), in other words, readable data of (10H - URTJnSLRP[3:0]) words still remains in the Rx FIFO.
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	URTJnSLTP[3:0]	Tx FIFO level interrupt setting URTJnSLTP[3:0] defines the Tx FIFO pointer status, at which the transmission interrupt request INTUAJnTIT is generated. INTUAJnTIT is generated if URTJnFSTR0.URTJnSSTW[4:0] = (10H - URTJnSLTP[3:0]), in other words, writable space of (10H - URTJnSLTP[3:0]) words still remains in the Tx FIFO.

Remark: n = 0, 1

(9) FIFO control register 1 (URTJnFCTL1)

This register controls the Rx time-out detection.

- Access This register can be read/written in 32-bit and 1-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
URTJn FCTL1	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	URTJnSLRT[5:0]						4000 03A0H +100H × n Initial Value 0000 003FH
R/W	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	R/W						

Bit Position	Bit Name	Function
31 to 6	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
5 to 0	URTJnSLRT[5:0]	Rx time-out control: 00H: Time-out detection disabled 01H to 3FH: Time-out time = (URTJnSLRT[5:0] × 8) cycles of bit-rate clock BRCLK

Remark: n = 0, 1

(10) FIFO status register 0 (URTJnFSTR0)

This register informs about the fill status of the Rx and Tx FIFOs.

- Access This register can be read in 16-bits units, and can only be written if URTJnPW = 0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJn FSTR0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJn SSRW[4:0]					0	0	0	URTJn SSTW[4:0]				
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R					0	0	0	R				

Address: 4000 0384H +100H × n

Initial Value: 0000 0010H

Bit Position	Bit Name	Function
31 to 13	—	Reserved. These bits are read as 0.
12 to 8	URTJnSSRW[4:0]	Rx FIFO status URTJnSSRW[4:0] indicates the number of unread received data words in the Rx FIFO. These bits can be written only if URTJnPW = 0, in a range from 00H to 10H. If these bits are read, a value of (10H - value written to URTJnSSRW[4:0]) is read. If 01H is written, for example, the value read will be 0FH.
7 to 5	—	Reserved. These bits are read as 0.
4 to 0	URTJnSSTW[4:0]	Tx FIFO status URTJnSSTW[4:0] indicates the amount of writable space (in words) in the Tx FIFO. These bits can be written only if URTJnPW = 0, in a range from 00H to 10H. If these bits are read, a value of (10H - value written to URTJnSSTW[4:0]) is read. If 01H is written, for example, the value read will be 0FH.

Remark: n = 0, 1

(11) FIFO status register 1 (URTJnFSTR1)

This register controls the Rx time-out detection.

- Access This register can be read in 32-bits units, and can be only written if URTJnPW = 0

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJn FSTR1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJnTMOE	URTJnTOFE	URTJnROVE	0	URTJnSSTF	URTJnSSTE	URTJnSSRF	URTJnSSRE
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	R	R	0	R	R	R	R

Address
4000 0388H
+100H × n

Initial Value
0000 0005H

Bit Position	Bit Name	Function
31 to 8	—	Reserved. These bits are read as 0.
7	URTJnTMOE	Time-out error detection 0: No time-out error was detected 1: A time-out error was detected To clear this bit after a time-out error has been detected, set URTJnFSTC.URTJnCLTM to 1. If a timeout error is detected at the same time as this bit is cleared, this bit remains set. URTJnTMOE can be written only if URTJnPW = 0. If 1 is written, the value read will be 1.
6	URTJnTOFE	Tx FIFO overflow error detection 0: No Tx FIFO overflow error was detected 1: A Tx FIFO overflow error was detected To clear this bit after a Tx FIFO overflow error has been detected, set URTJnFSTC.URTJnCLTO to 1. If a timeout error is detected at the same time as this bit is cleared, this bit remains set. URTJnTOFE can be written only if URTJnPW = 0. If 1 is written, the value read will be 1.
5	URTJnROVE	Rx FIFO overrun error detection 0: No Rx FIFO overrun error was detected 1: A Rx FIFO overrun error was detected To clear this bit after an Rx FIFO overrun error has been detected, set URTJnFSTC.URTJnCLOV to 1. If a timeout error is detected at the same time as this bit is cleared, this bit remains set. URTJnROVE can be written only if URTJnPW = 0. If 1 is written, the value read will be 1.
4	—	Reserved. This bit is read as 0.
3	URTJnSSTF	Tx FIFO full status 0: Tx FIFO is not full 1: Tx FIFO is full
2	URTJnSSTE	Tx FIFO empty status 0: Tx FIFO is not empty 1: Tx FIFO is empty

Note: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
1	URTJnSSRF	Rx FIFO full status 0: Rx FIFO is not full 1: Rx FIFO is full
0	URTJnSSRE	Rx FIFO empty status 0: Rx FIFO is not empty 1: Rx FIFO is empty

Remark: n = 0, 1

(12) FIFO status clear register (URTJnFSTC)

By using this register the error flags of URTJnFSTR1 can be cleared. Furthermore, the pointers of the Rx and Tx FIFOs can be cleared, thus indicating empty status for both FIFOs.

- Access This register can be read/written in 32-bit and 1-bit units.

Reading this register always returns 00H.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJnFSTC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJnCLTM	URTJnCLTO	URTJnCLRO	0	0	0	URTJnCLTP	URTJnCLRP
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	0	0	0	R/W	R/W

Address 4000 038CH +100H × n

Initial Value 0000 0000H

Bit Position	Bit Name	Function
31 to 8	—	When writing to these bits, write 0. When read, 0 is returned.
7	URTJnCLTM	Time-out error flag clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets URTJnFSTR1.URTJnTMOE = 0
6	URTJnCLTO	Tx FIFO overflow error flag clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets URTJnFSTR1.URTJnTOFE = 0
5	URTJnCLRO	Rx FIFO overrun error flag clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets URTJnFSTR1.URTJnROVE = 0
4 to 2	—	When writing to these bits, write 0. When read, 0 is returned.
1	URTJnCLTP	Tx FIFO pointer clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets the Tx FIFO pointer to 00H, thus - URTJnFSTR0.URTJnSSTW[4:0] = 00H (Tx FIFO pointer) - URTJnFSTR1.URTJnTOFE = 0 (no Tx FIFO overflow error status) - URTJnFSTR1.URTJnSSTF = 0 (Tx FIFO not full status) - URTJnFSTR1.URTJnSSTE = 1 (Tx FIFO empty status)
0	URTJnCLRP	Rx FIFO pointer clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets the Rx FIFO pointer to 00H, thus - URTJnFSTR0.URTJnSSRW[4:0] = 00H (Rx FIFO pointer) - URTJnFSTR1.URTJnROVE = 0 (no Rx FIFO overrun error status) - URTJnFSTR1.URTJnSSRF = 0 (Rx FIFO not full status) - URTJnFSTR1.URTJnSSRE = 1 (Rx FIFO empty status)

Remark: n = 0, 1

(13) FIFO receive data register (URTJnFRX)

By using this register the reception data is read from the Rx FIFO.

Each reception data is accompanied by flags, which indicate parity and framing error during reception.

- 7-bit transfer — If the data length has been specified as 7 bits (URTJnCTL1.URTJnCLG = 0) and
  - reception is LSB-first (URTJnCTL1.URTJnSLD = 1):
    The receive data is transferred to the Rx FIFO URTJnFRX.URTJnRX[6:0] and the data MSB URTJnFRX.URTJnRX[7] always becomes 0.
  - reception is MSB-first (URTJnCTL1.URTJnSLD = 0):
    The receive data is transferred to the Rx FIFO URTJnFRX.URTJnRX[7:1] and the data LSB URTJnFRX.URTJnRX[0] always becomes 0.
    For further information on data formats, refer to section 22.6.1, Data Formats.
- Access — This register can be read in 32-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URTJnFRX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJnPE	URTJnFE	0	0	0	0	0	0	URTJnRX[7:0]							
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	R	0	0	0	0	0	0	R							

Address
4000 0390H
+100H × n
Initial Value
0000 00FFH

Bit Position	Bit Name	Function
31 to 16	—	These bits are read as 0.
15	URTJnPE	Parity error flag 0: No parity error was detected during URTJnRX[7:0] reception 1: A parity error was detected during URTJnRX[7:0] reception
14	URTJnFE	Framing error flag 0: No framing error was detected during URTJFnRX[7:0] reception 1: A framing error was detected during URTJnRX[7:0] reception
13 to 8	—	These bits are read as 0.
7 to 0	URTJnRX[7:0]	Reception data

Remark: n = 0, 1

- Change in the pointer — Each read from URTJnFRX decreases the amount of unread data in the Rx FIFO and thus decreases URTJnFSTR0.URTJnSSRW[4:0].

(14) FIFO transmit data register (URTJnFTX)

By this register the transmission data is written to the Tx FIFO.

- 7-bit transfer — If the data length has been specified as 7 bits (URTJnCTL1.URTJnCLG = 0) and
  - transmission is LSB-first (URTJnCTL1.URTJnSLD = 1):
    The URTJnFTX.URTJnTX[6:0] value is transferred to the shift register as the Tx FIFO data.
  - transmission is MSB-first (URTJnCTL1.URTJnSLD = 0):
    The URTJnFTX.URTJnTX[7:1] value is transferred to the shift register as the Tx FIFO data.
    For further information on data formats, refer to section 22.6.1, Data Formats.
- Access — This register can be read/written in 8-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7 to 0
URTJnFTX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	URTJnTX[7:0]
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W

Address: 4000 0394H +100H × n

Initial Value: 0000 00FFH

Bit Position	Bit Name	Function
31 to 8	—	When writing to these bits, write 0. When read, 0 is returned.
7 to 0	URTJnTX[7:0]	Transmission data

- Read access — Reading URTJnFTX returns the most recent data that was written to the Tx FIFO.
- Change in the pointer — Each write to URTJnFTX decreases the amount of writable space (words) in the Tx FIFO and thus decreases URTJnFSTR0.URTJnSSTW[4:0].
- Overflow error — If URTJnFTX is written while the Tx FIFO is full (URTJnFSTR1.URTJnSSTF = 1, the written data is discarded, an overflow error is detected (URTJnFSTR1.URTJnROVE = 1) and the status interrupt INTUAJnTIS is generated.

22.5 Interrupt Request Signals

The following three interrupt request signals are generated by UARTJn.

- Transmission interrupt request INTUAJnTIT
- Reception interrupt request INTUAJnTIR
- Status interrupt request INTUAJnTIS

22.5.1 Transmission Interrupt Request INTUAJnTIT

A transmit interrupt request INTUAJnTIT can be configured to be generated upon a certain fill level of the Tx FIFO.

The Tx FIFO fill level for the transmit interrupt request can be set by URTJnFCTL0.URTJnSLTP[3:0], whereas the interrupt is generated if
URTJnFSTR0.URTJnSSTW[4:0] = 10H - URTJnSLTP[3:0]

The amount of writable space in the Tx FIFO, at which INTUAJnTIT is generated, depends on the selected timing of the transmission interrupt request:

- if URTJnCTL1.URTJnSLIT = 0: 10H - URTJnSLTP[3:0]
- if URTJnCTL1.URTJnSLIT = 1: 0FH - URTJnSLTP[3:0]

Writable space of the number of words shown above remained in the Tx FIFO when the interrupt was generated.

(1) INTUAJnTIT timing

The time to generate the transmission interrupt INTUAJnTIT, and thus indicating the specified amount of writable space in the Tx FIFO, depends on the setting of the URTJnCTL1.URTJnSLIT bit:

- URTJnCTL1.URTJnSLIT = 0: at start of transmission
  The transmission interrupt request INTUAJnTIT is issued when transmission of the first bit is starting. In case of data transmission, this indicates the transmission start of the FIFO data of fill level URTJnFCTL0.URTJnSLTP[3:0]. In case of BF transmission every transmission start of a BF generates INTUAJnTIT.
- URTJnCTL1.URTJnSLIT = 1: at end of transmission
  INTUAJnTIT is generated when the entire data transmission process is completed, i.e. when the last bit of the transmit data has been transmitted.
  The transmission interrupt request INTUAJnTIT is issued when transmission of the last bit is completed.
  In case of data transmission, this indicates the transmission end of the FIFO data of fill level URTJnFCTL0.URTJnSLTP[3:0].
  In case of BF transmission every transmission completion of a BF generates INTUAJnTIT.

The following diagram show the timing of the transmission interrupt request INTUAJnTIT during data transmission for both cases.

(2) INTUAJnTIT at transmission errors

If an error is detected during data consistency checking, the interrupt INTUAJnTIT is not generated.

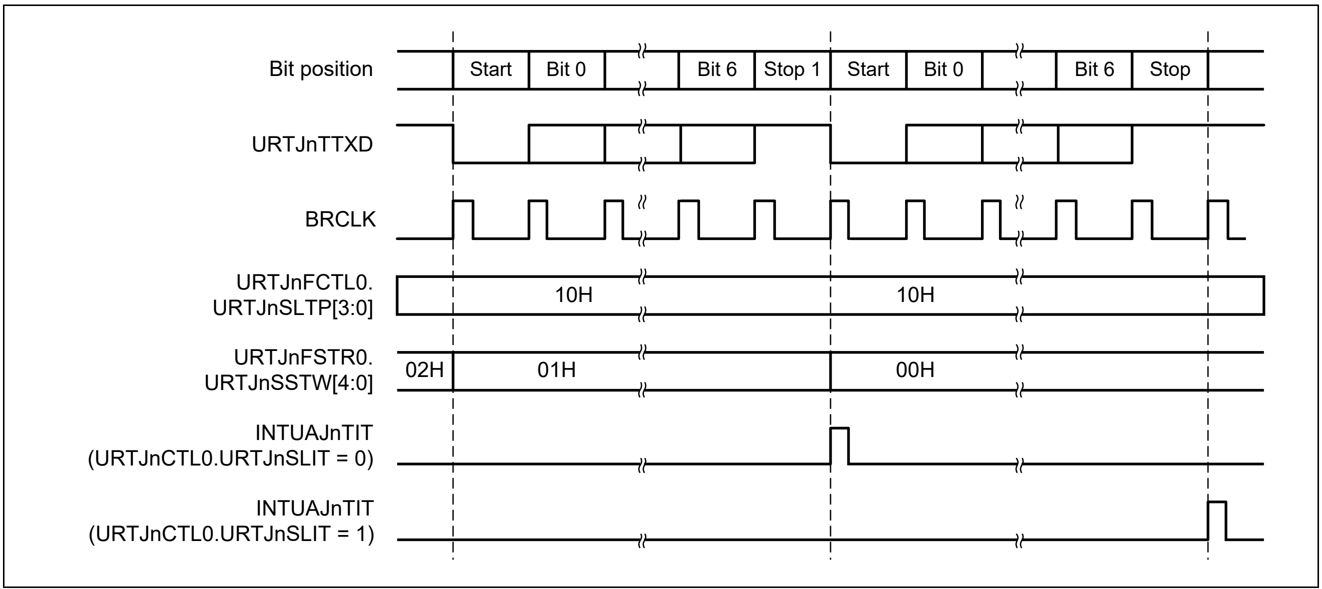


Figure 22.2 Transmission Interrupt Request Timing

Caution: After a transmission interrupt is generated, an additional transmission interrupt might be generated in a system that keeps transmission waiting for at least one frame when the FIFO is empty. Therefore, you must clear the interrupt request flag (EICn.EIRFn) in the interrupt routine to 0.

22.5.2 Reception Interrupt Request INTUAJnTIR

A reception interrupt request INTUAJnTIR can be configured to be generated upon a certain fill level of the Rx FIFO.

The Rx FIFO fill level for the reception interrupt request can be set by URTJnFCTL0.URTJnSLRP[3:0], whereas the interrupt is generated if URTJnFSTR0.URTJnSSRW[4:0] = 10H - URTJnSLRP[3:0]

URTJnFSTR0.URTJnSSRW[4:0] = (10H - URTJnSLRP[3:0])

(1) INTUAJnTIR at reception errors

INTUAJnTIR is also generated if parity or framing error was detected, provided the above Rx FIFO fill condition is met.

In case of an Rx FIFO overrun error, no data is stored in the Rx FIFO, thus INTUAJnTIR is not generated.

The figure below shows the timing of the reception interrupt request during data reception.

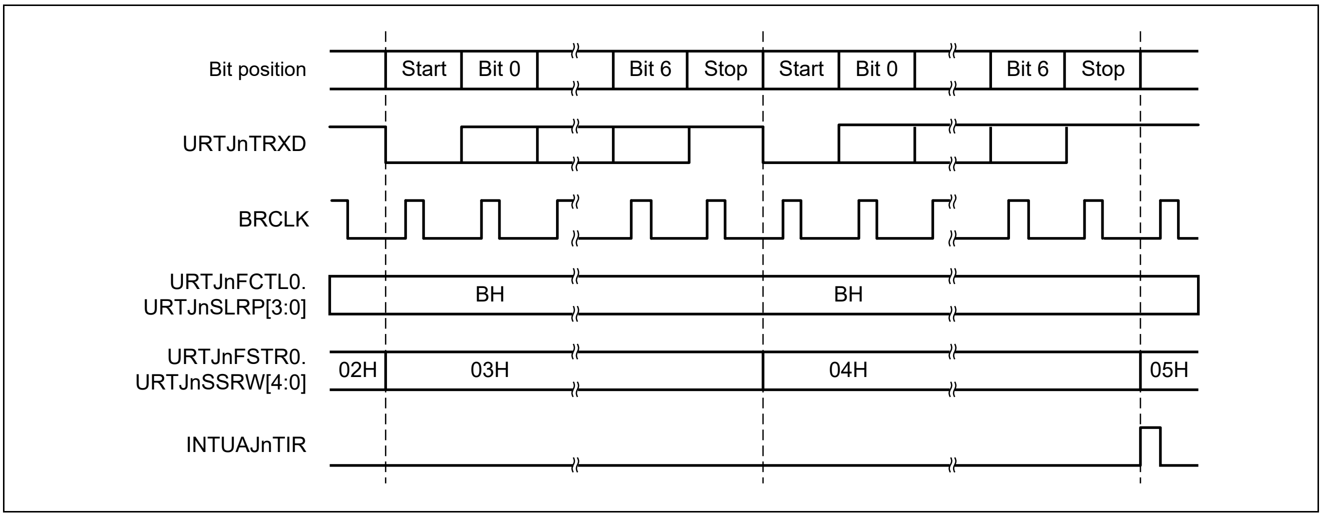


Figure 22.3 Reception Interrupt Request Timing

Caution: After a reception interrupt is generated, an additional reception interrupt might be generated in a system that keeps reception waiting for at least one frame when the FIFO is full. Therefore, you must clear the interrupt request flag (EICn.EIRFn) in the interrupt routine to 0.

(2) BF reception

In case of BF reception, INTUAJnTIR is always generated upon completion of the BF reception.

22.5.3 Status Interrupt Request INTUAJnTIS

A status interrupt request is generated if an error condition occurred during reception or transmission:

- transmission data consistency check error (URTJnSTR1.URTJnDCE = 1)
- reception data parity error (URTJnSTR1.URTJnPE = 1)
- reception data framing error (URTJnSTR1.URTJnFE = 1)
- time-out error (URTJnFSTR1.URTJnTMOE = 1)
- time-out error (URTJnFSTR1.URTJnTMOE = 1)
- Rx FIFO overrun error (URTJnFSTR1.URTJnROVE = 1)

if BF length of more than 10 bits is detected while BF reception is enabled during data reception (URTJnCTL1.URTJnSLBM = 1)

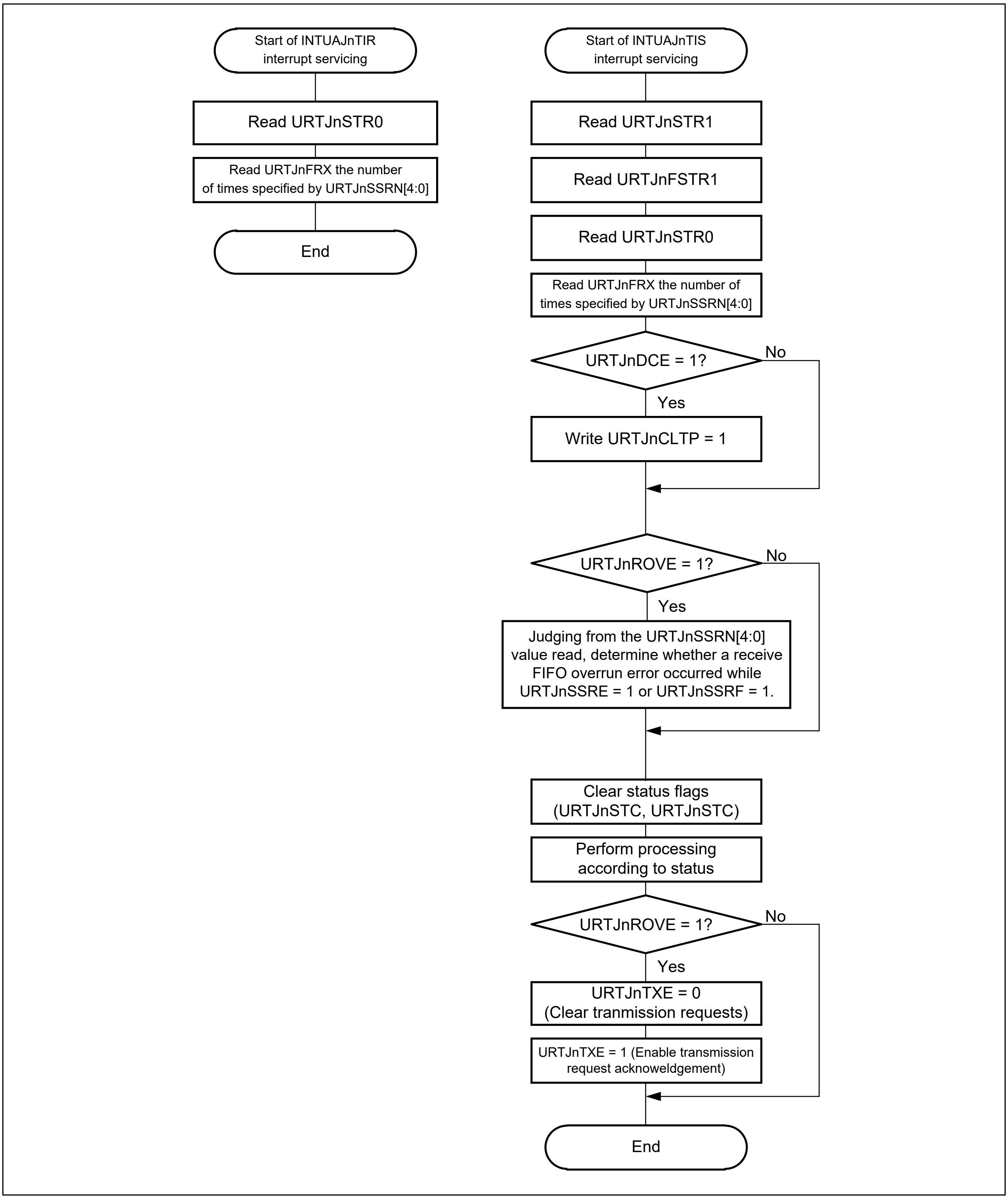


Figure 22.4 Processing Flow after Interrupt Generation

22.6 Operation

22.6.1 Data Formats

Full-duplex serial data reception and transmission is performed.

As shown in the figures below, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

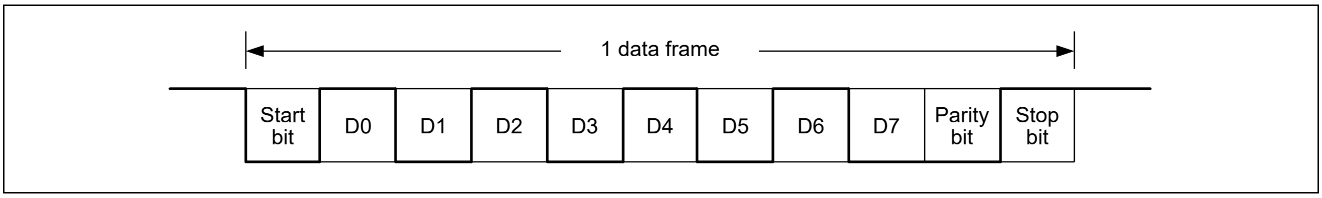
Several properties of a transmit/receive data frame can be specified by control bits of the URTJnCTL1 register:

Table 22.5 Data Format Specification

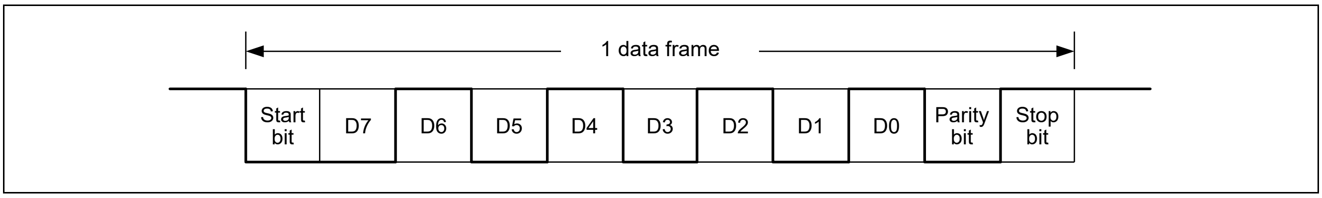
Item	Options	Control Bits
Start bit	1 bit	Fixed
Character bits	7 bits / 8 bits	URTJnCTL1.URTJnCLG
Parity	Even parity / odd parity / 0 parity / no parity	URTJnCTL1.URTJnSLP[1:0]
Stop bit	1 bit / 2 bits	URTJnCTL1.URTJnSLG
Data order	MSB first / LSB first	URTJnCTL1.URTJnSLD
Tx data level	inverted / not inverted	URTJnCTL1.URTJnTDL
Tx data level	inverted / not inverted	URTJnCTL1.URTJnRDL

(1) UARTJn transmit/receive data format

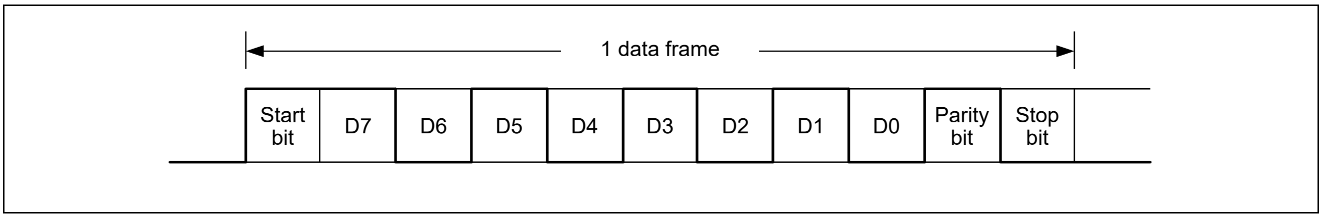
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



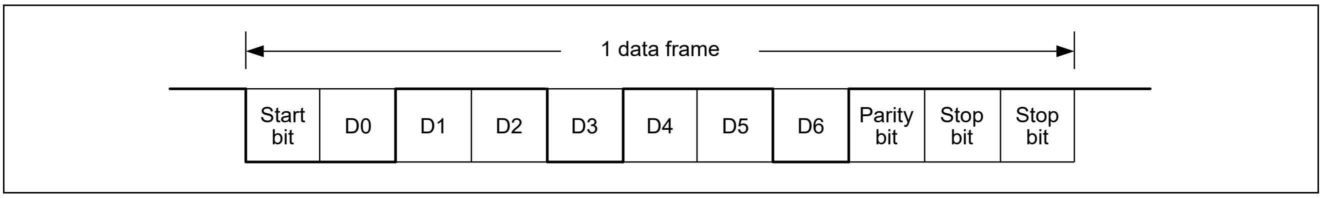
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, URTJnTTXD inversion



(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H



(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H

22.6.2 BF Transmission/Reception Format

The UARTJn has a BF (Break Field) transmission/reception control function to enable use of the LIN functions.

(1) About LIN

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with bit rate information and the slave receives it and corrects the error in the bit rate. Therefore, communication is possible when the error in the bit rate of the slave is ±14% or less.

Figure 22.5, LIN Transmission Outline and Figure 22.6, LIN Reception Outline outline the transmission and reception operations of LIN.

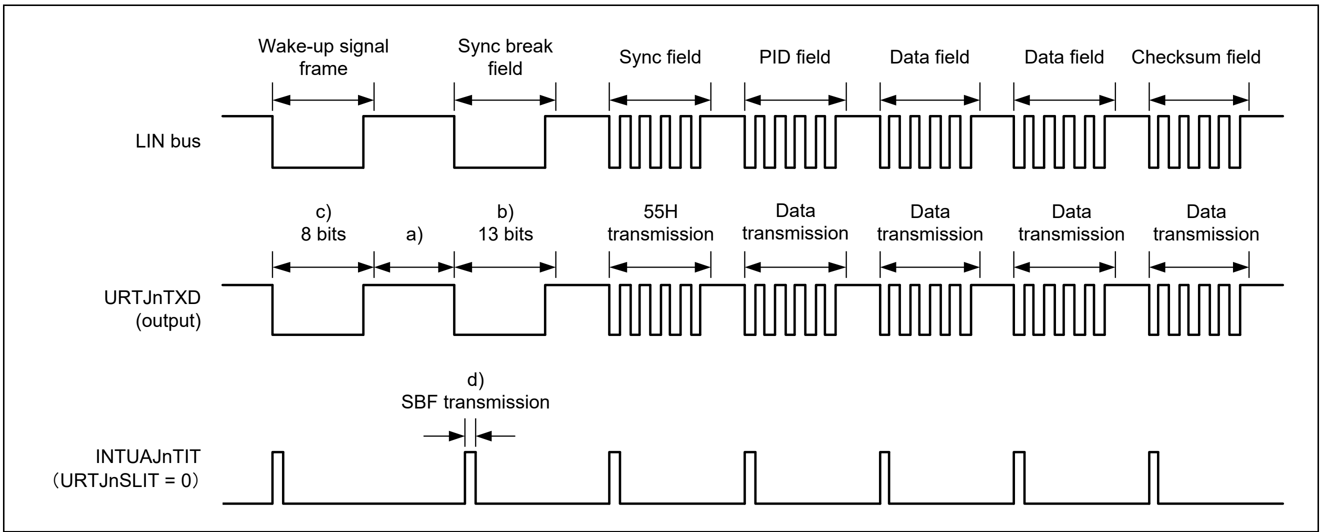


Figure 22.5 LIN Transmission Outline

a) The interval between fields is controlled by software.

b) BF output is performed by hardware. The output width is the bit length set by URTJnCTL1.URTJnBLG[2:0]. If even finer output width adjustments are required, such adjustments can be performed using URTJnCTLn.URTJnBRS[11:0].

c) 80H transfer in the 8-bit mode is substituted for the wakeup signal frame.

d) A transmission enable interrupt INTUAJnTIT is generated at the start of each transmission. INTUAJnTIT is also generated at the start of each BF transmission.

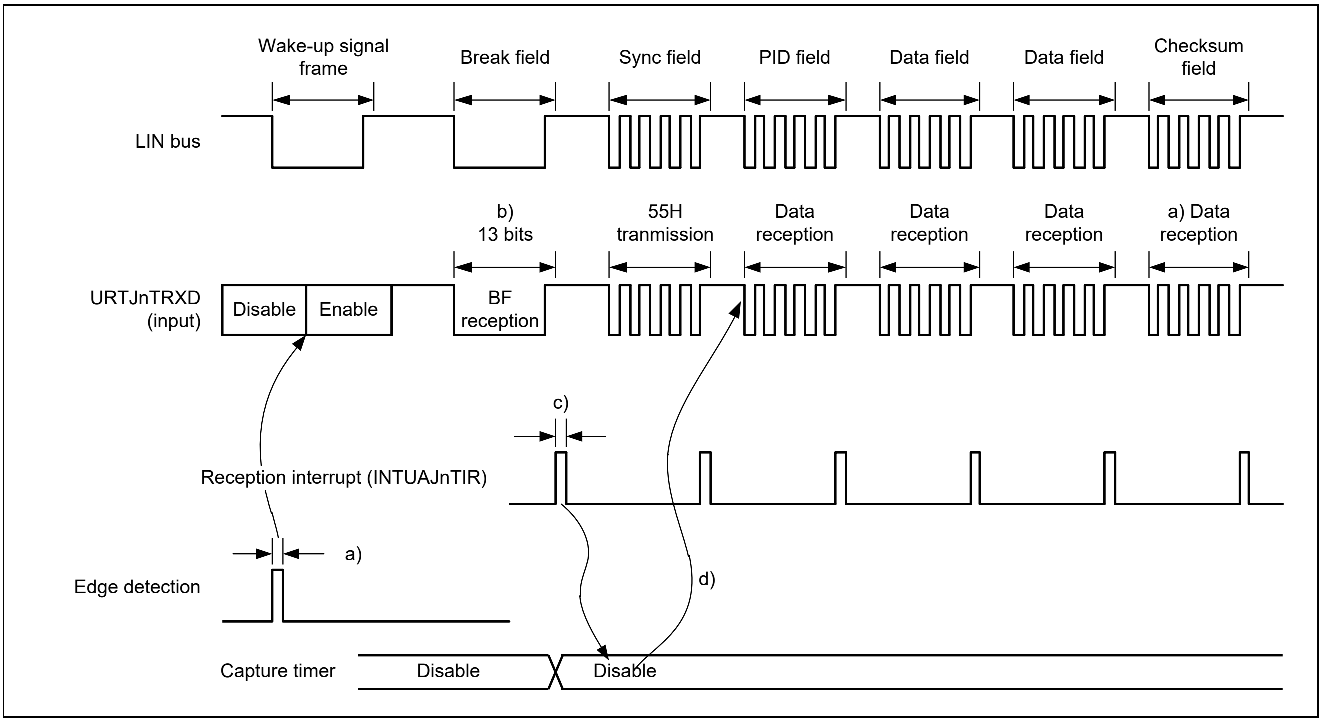


Figure 22.6 LIN Reception Outline

a) The wakeup signal sent by the pin edge detector enables UARTJn, and sets BF reception mode.

b) BF reception is judged to end normally when a BF of 11 or more bits is received.
An interrupt is generated as shown in the table below, according to the setting of the BF reception mode selection bit URTJnCTL1.URTJnSLBM and the value of the URTJnSTR0.URTJnSSBR bit.

URTJnSLBM	URTJnSSBR	Interrupts
1	×	INTUAJnTIS
0	1	INTUAJnTIR
0	0	A framing error has occurred, so INTUAJnTIS is generated.

c) When BF reception ends normally, an interrupt is generated as follows according to the setting of the BF reception mode selection bit URTJnCTL1.URTJnSLBM:
- When URTJnDTL1.URTJnSLBM is 0, the reception interrupt INTUAJnTIR is generated.
- When URTJnDTL1.URTJnSLBM is 1, the status interrupt INTUAJnTIS is generated and the BF reception success flag URTJnSTR1.URTJnBSF is set.

If the BF reception trigger bit URTJnTRG.URTJnBRT is 1, error detection for the overrun, parity, and framing errors is not performed during BF reception. Also, data transfer from the receive shift register to the receive data register URTJnRX is not performed. URTJnRX holds the previous value at this time.

d) In order to adjust the bit-rate clock properly, the URTJnTRXD signal must be connected to the timer capture input. The transfer rate and bit rate error can be calculated by measuring the time between URTJnTRXD edges, and the bit rate can be adjusted by specifying a value for the bit rate setting bits URTJnCTL2.URTJnBRS[11:0].

e) A checksum field is identified by software. When a checksum field is received, UARTJn is initialized and set to BF reception mode again by software. If URTJnCTL1.URTJnSLBM is 1 at this time, UARTJn automatically starts BF reception without being set to BF reception mode again.

22.6.3 BF Transmission

When the URTJnCTL0 bits URTJnPW = URTJnTXE = 1, the transmission enabled status is entered, and BF transmission is started by setting the BF transmission trigger URTJnTRG.URTJnBTT = 1.

Thereafter, URTJnSTR0.URTJnSSBT is set to "1" and a low level width of 13 to 20 bits, as specified by URTJnCTL1.URTJnBLG[2:0], is output.

A transmission interrupt INTUAJnTIT) is generated upon BF

- transmission start, if URTJnCTL1.URTJnSLIT = 0
- transmission end, if URTJnCTL1.URTJnSLIT = 1

Following the end of BF transmission, URTJnSTR0.URTJnSSBT is automatically cleared. Thereafter, the UARTJn transmission mode is restored.

Transmission is suspended until the data to transmit next is written to the URTJnTX register and URTJnSTR0.URTJnSST is set, or until the BF transmission trigger URTJnTRG.URTJnBTT is set and URTJnSTR0.URTJnSSBT changes to 1.

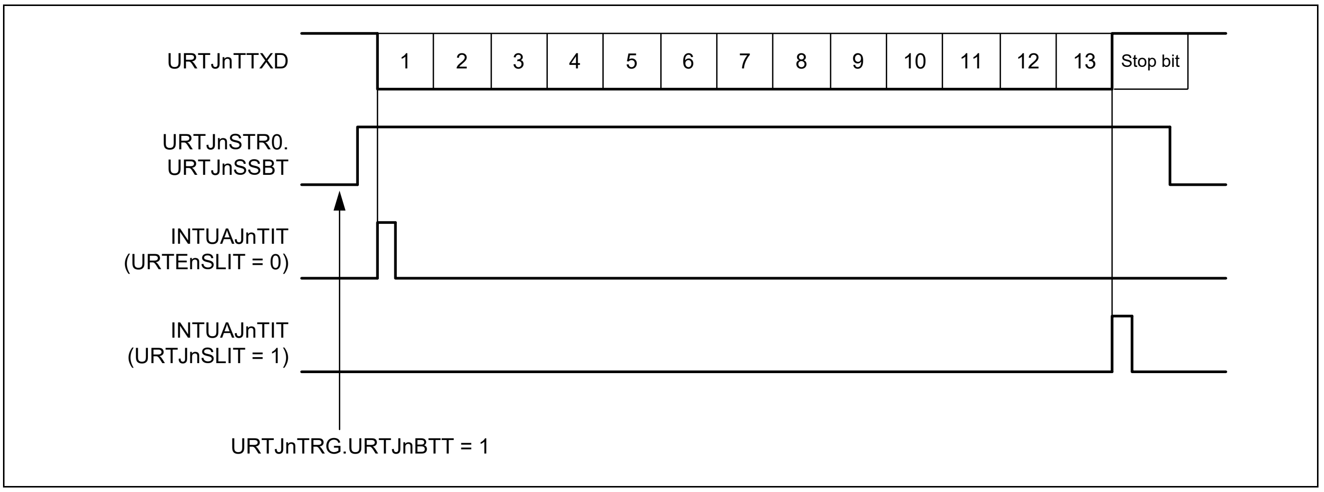
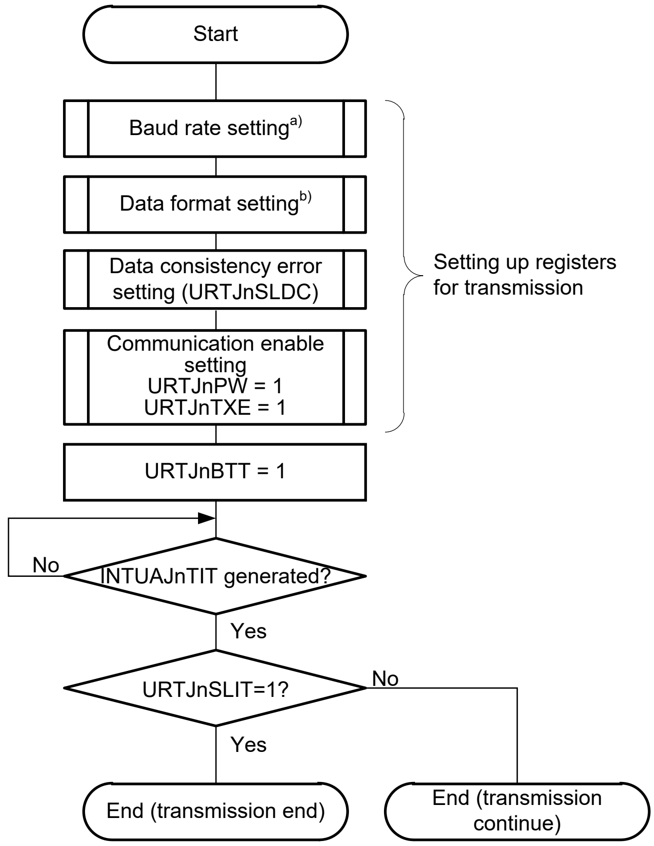


Figure 22.7 BF Transmission



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnBLG[2:0], URTJnCLG, URTJnSLP[1,0], URTJnTDL, URTJnSLG, URTJnSLD, and URTJnSLIT bits.

Figure 22.8 Flowchart of BF Transmission

22.6.4 BF Reception

Reception is enabled by setting the URTJnCTL0.URTJnPW bit to 1 and then setting the URTJnCTL0.URTJnRXE bit to 1.

The BF reception wait state is entered by setting the BF reception trigger URTJnTRG.URTJnBRT = 1.

In the BF reception wait state, the URTJnTRXD pin is monitored and start bit detection is performed.

Following detection of the low level, reception is started and the internal counter counts up according to the bit rate setting.

When a high level is received and if the BF width is 11 or more bits, while the BF receiving mode selection bit

- URTJnCTL1.URTJnSLBM = 0, the reception interrupt INTUAJnTIR is generated.
- URTJnCTL1.URTJnSLBM = 1, the status interrupt INTUAJnTIS is generated and BF reception success flag URTJnSTR1.URTJnBSF is set at the same time. The URTJnSTR0.URTJnSSBR bit is automatically cleared and BF reception ends.

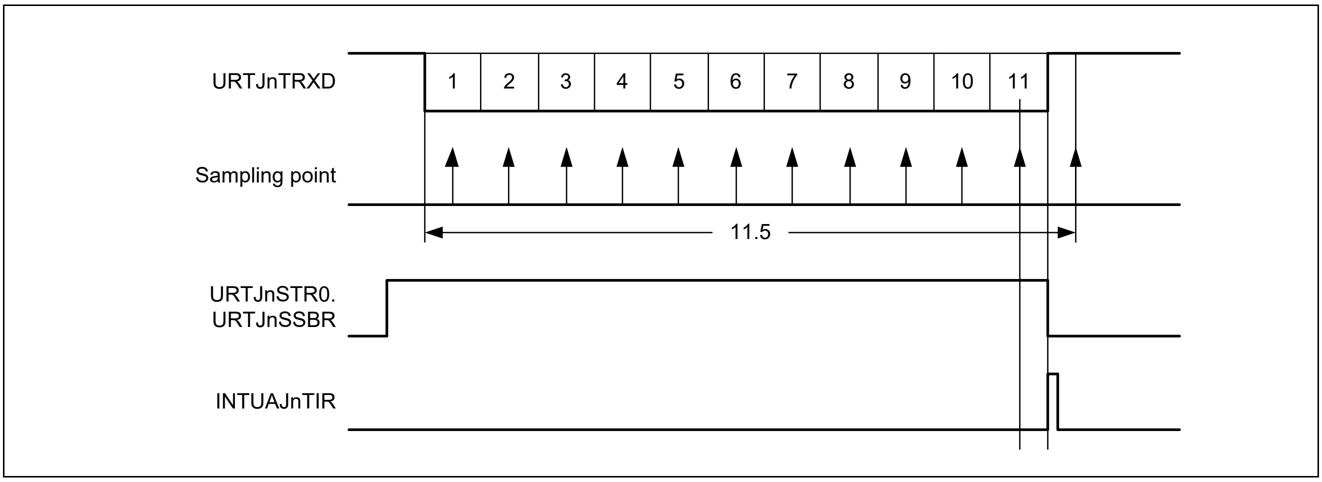


Figure 22.9 Normal BF Reception (Stop Bit after More Than 10.5 "L" Bits)

Error detection for the URTJnSTR1 error flags URTJnOVE, URTJnPE, and URTJnFE is suppressed and UARTJn communication error detection processing is not performed.

Moreover, the erroneous data is not stored in URTJnRX, but the initial value FFH is held.

If the BF width is 10 or fewer bits, reception is terminated as error processing without generating an interrupt, and BF reception mode is restored. URTJnSTR0.URTJnSSBR is not cleared at this time.

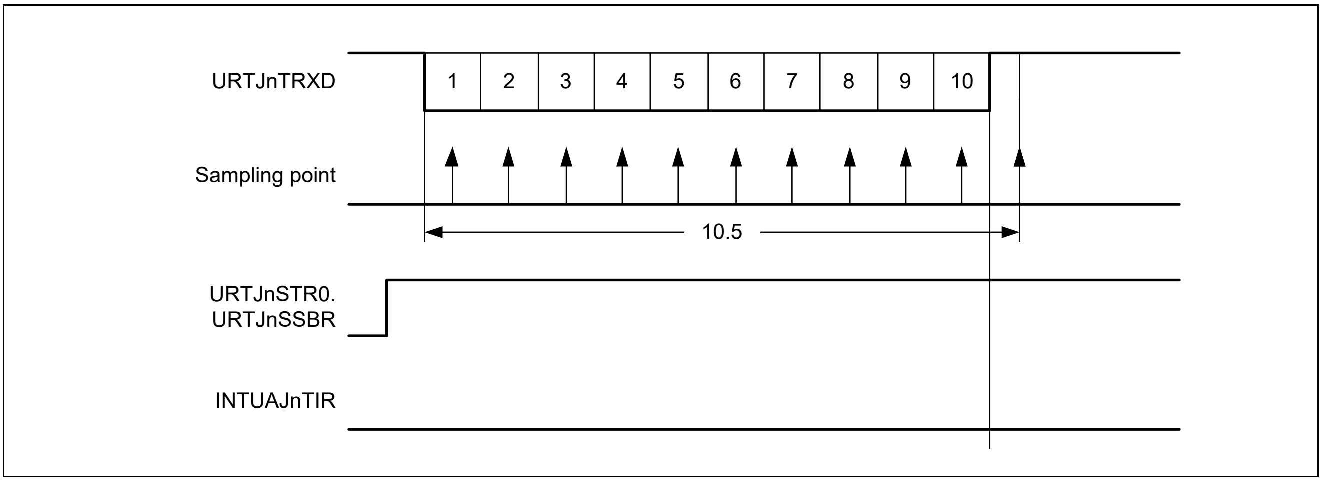


Figure 22.10 BF Reception Error (Stop Bit within 10.5 "L" Bits)

The BF mode can be selected between a single BF receive mode and any time BF receive mode in by URTJnCTL1.URTJnSLBM. The status of a successful reception of the BF is indicated by URTJnSTR1.URTJnBSF.

Remark: URTJnSTR0.URTJnSSBR is set to "1" when

- **URTJnTRG.URTJnBRT is set to "1", or**
- **the error is cleared by normal BF reception.**

22.6.5 UARTJn Transmission

(1) Transmission FIFO

The Tx FIFO comprises 8 bit × 16 levels to hold the 8-bit data to be transmitted consecutively.

The Tx FIFO is filled by writing to the URTJnFTX register.

- Various status information are provided to check the fill level of the Tx FIFO:
  - The amount of writable space in the Tx FIFO can be checked by reading the Tx FIFO pointer URTJnFSTR0.URTJnSSTW[4:0]
  - FIFO full/not full status is indicated by URTJnSTR1.URTJnSSTF (= 1: full)
  - FIFO empty/not empty status is indicated by URTJnSTR1.URTJnSSTE (= 1: empty)
- Change in the pointer
  Each write to URTJnFTX decreases the amount of writable space in the Tx FIFO and thus decreases URTJnFSTR0.URTJnSSTW[4:0].
- Overflow error
  If URTJnFTX is written while the Tx FIFO is full (URTJnFSTR1.URTJnSSTF = 1), the written data is discarded, an overflow error is detected (URTJnFSTR1.URTJnTOFE = 1) and the status interrupt INTUAJnTIS is generated.
- URTJnFTX read
  Reading URTJnFTX returns the most recent data that was written to the Tx FIFO.

(2) Transmission start and stop

- Transmission start
  Set the transmission enabled status by performing the following procedures.
  - Specify the bit rate by URTJnCTL2.
  - Specify the transmit parity, data character length, stop bit length, transmit data order, transmission interrupt request timing and output logic level by URTJnCTL1.
  - Enable UARTJn operation and transmission by URTJnCTL0.URTJnPW = URTJnCTL0.URTJnTXE = 1)

  Writing the transmit data to the Tx FIFO via URTJnFTX starts transmission. The data which is saved in the Tx FIFO is transferred to the transmit shift register. Then, the start, parity and stop bits are added and the data frame is output serially via URTJnTTXD.
- Transmission stop
  When URTJnCTL0.URTJnPW or URTJnCTL0.URTJnTXE is set to 0, transmission operations are stopped immediately, even during transmission processing.
- Concurrent transmission of BF and data
  When a BF transmit request and a data transmit request have both been set, BF transmission is given priority.

(3) Transmission data consistency checking

The UARTJn handles data consistency checking to detect a mismatch between the transmit data output via the signal URTJnTTXD and the data received from the URTJnTRXD signal, when UARTJn operates in transmission mode.

Remark: To perform data consistency checking, the URTJnTTXD signal must be fed back to URTJnTRXD externally.

Data consistency checking is enabled by URTJnCTL0.URTJnSLDC = 1.

In case of a mismatch between the URTJnTTXD and URTJnTRXD signals the data consistency error flag URTJnSTR1.URTJnDCE is set and a status interrupt request INTUAJnTIS occurs.

Data consistency checking can be performed with reception enabled or disabled.

If reception is disabled (URTJnCTL0.URTJnRXE = 0) the reception completion interrupt request INTUAJnTIR, the URTJnSTR1 status bits URTJnBSF, URTJnFE, URTJnPE and the status interrupt request signal INTUAJnTIS will not be generated. Receive data is not stored in the Rx FIFO.

If reception is enabled (URTJnCTL0.URTJnRXE = 1) the receive data is treated as in normal reception mode, i.e. all status bits and interrupts are handled, and the data is stored in the Rx FIFO.

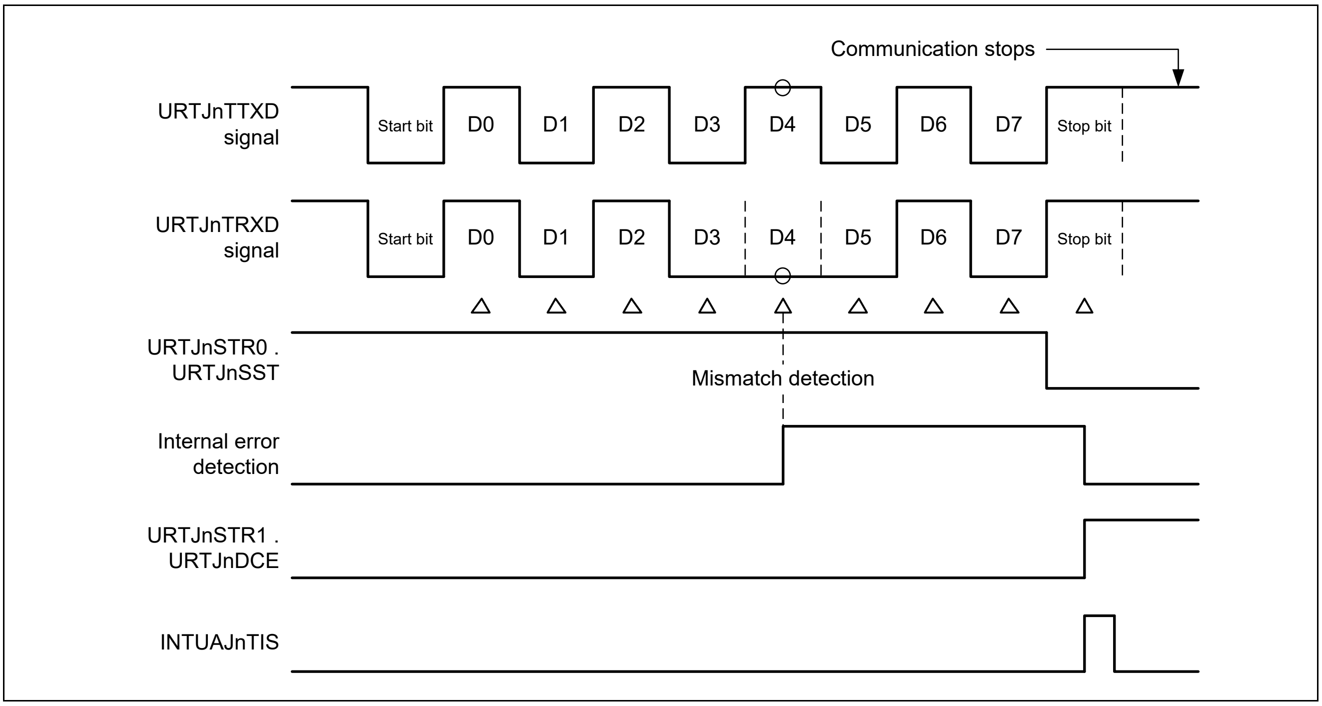


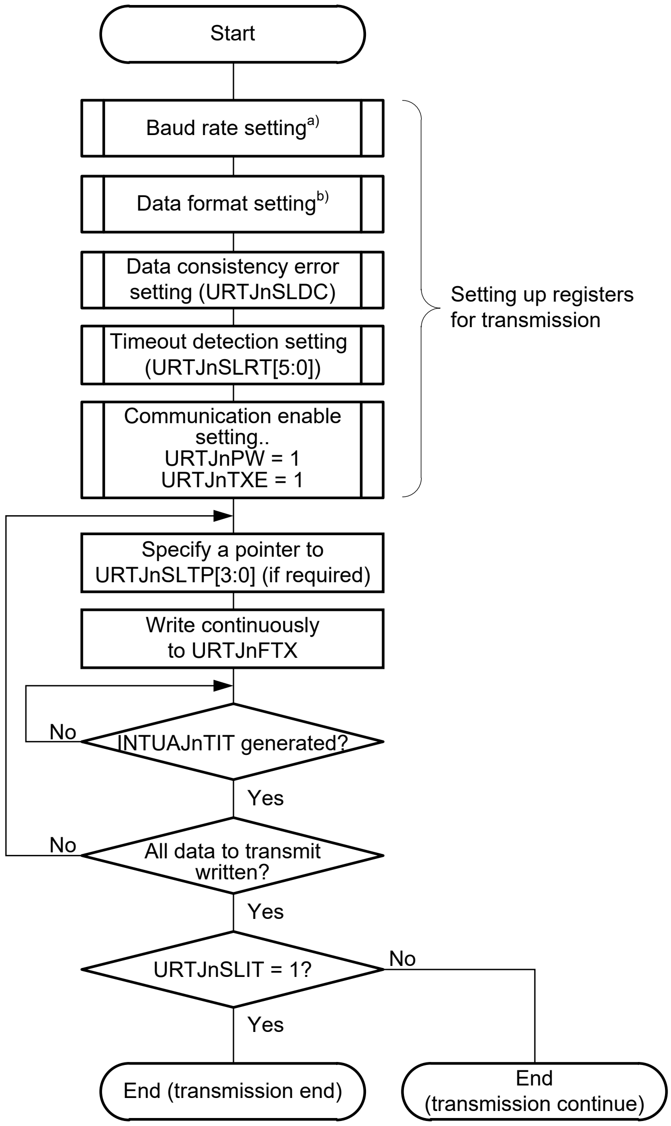
Figure 22.11 Timing Example of Data Consistency Error (No BF Reception Active, i.e. URTJnSTR0.URTJnSSBR = 0)

If a data consistency error was detected (URTJnSTR1.URTJnDCE = 1), the subsequent data is not transmitted until the data consistency error flag is cleared (URTJnSTC.URTJnCLDC = 1) or transmission is disabled (URTJnCTL0.URTJnPW = 0, or URTJnCTL0.URTJnTXE = 0).

(4) Continuous transmission procedure

Continuous transmission is achieved by maintaining a certain fill level of the Tx FIFO.

This means in particular to set the generation of the transmission interrupt INTUAJnTIT that indicates the Tx FIFO fill level, appropriately via the Tx FIFO level interrupt setting URTJnSLTP [3:0].



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnCLG, URTJnSLP[1,0], URTJnTDL, URTJnSLG, URTJnSLD, and URTJnSLIT bits.

Figure 22.12 Flowchart of Data Transmission

22.6.6 UARTJn Reception

(1) Reception FIFO

The Rx FIFO comprises 10 bit × 16 levels to store the 8-bit data that has been received and additionally two error flags, indicating parity and framing errors.

The Rx FIFO is emptied by reading from the URTJnFRX register.

- Rx FIFO status
  Various status information are provided to check the fill level of the Rx FIFO:
  - The number of received words in the Rx FIFO can be checked by reading the Rx FIFO pointer URTJnFSTR0.URTJnSSRW[4:0]
  - FIFO full/not full status is indicated by URTJnFSTR1.URTJnSSRF (= 1: full)
  - FIFO empty/not empty status is indicated by URTJnFSTR1.URTJnSSRE (= 1: empty)
- Change in the pointer
  Each reception increases the number of data words in reception FIFO and thus increases URTJnSTR0.URTJnSSRW[4:0].
- Overrun error
  If new data is received while the Rx FIFO is full (URTJnFSTR1.URTJnSSRF = 1), the received data is discarded, an overrun error is detected (URTJnFSTR1.URTJnROVE = 1) and the status interrupt INTUAJnTIS is generated.

(2) Reception start and stop

- Reception start
  Set the reception enabled status by the following procedure:
  - Specify the bit rate by URTJnCTL2.
  - Specify the receive parity, data character length, stop bit length, receive data order and output logic level by URTJnCTL1.
  - Enable UARTJn operation and reception by URTJnCTL0.URTJnPW =URTJnCTL0.URTJnRXE = 1).

  When the sampling of the input level of the URTJnTRXD pin is performed and the falling edge is detected, the data sampling of the URTJnTRXD input is started. The start bit is recognized if the URTJnTRXD pin is low level after the time of a half bit is passed after the detection of the falling edge (shown in the figure below). After a start bit has been recognized, the receive operation starts, and serial data is stored in the receive shift register according to the bit rate setting. When the reception interrupt INTUAJnTIR is generated upon reception of the stop bit, the data stored in the receive shift register is written to the Rx FIFO.
- Reception stop
  When URTJnCTL0.URTJnPW or URTJnCTL0.URTJnRXE is set to 0, reception operations are stopped immediately, even during reception processing.
- Rx format change
  When the receive data order, parity, data character length, and the stop bit length are changed, clear the power bit (URTJnCTL0.URTJnPW = 0) or clear both the transmission enabled bit and the reception enabled bit (URTJnTXE = 0, URTJnRXE = 0), and then change the setting.

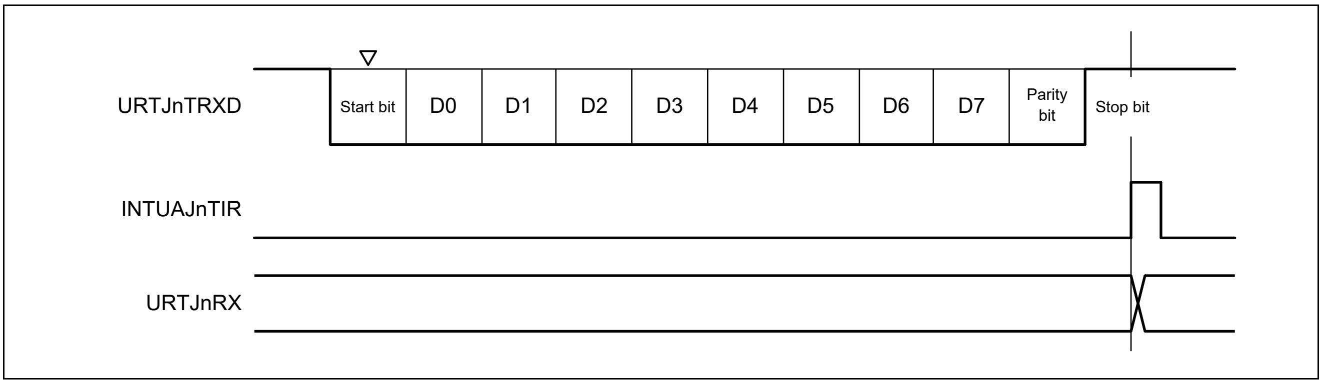
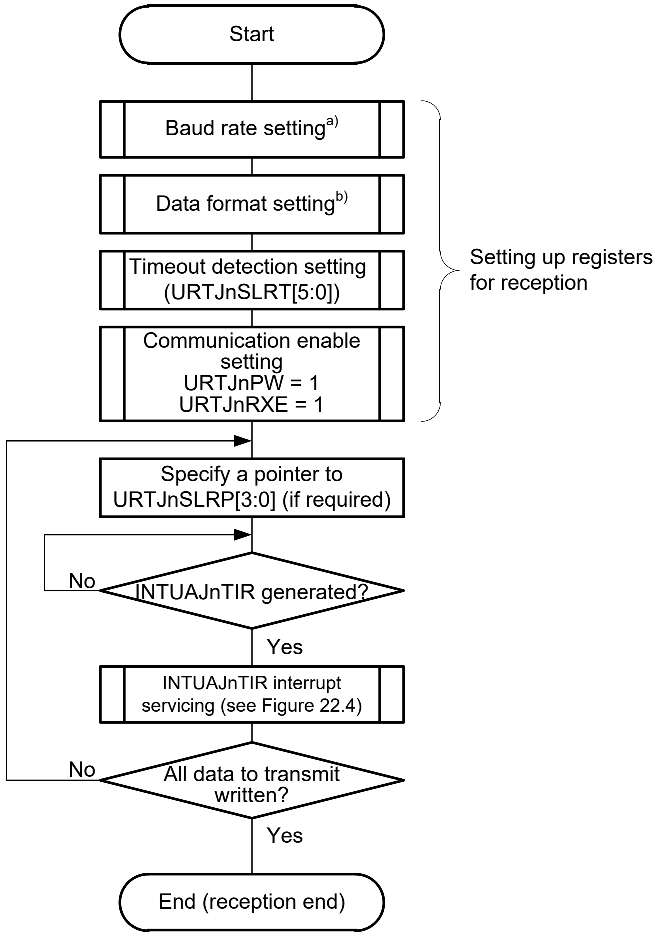


Figure 22.13 UARTJn Reception

Caution: During reception, operation is performed based on the assumption that there is only one stop bit. Accordingly, the second stop bit is ignored.

Remarks 1. If the low level is always input to the URTJnTRXD pin, the input is not judged as the start bit.

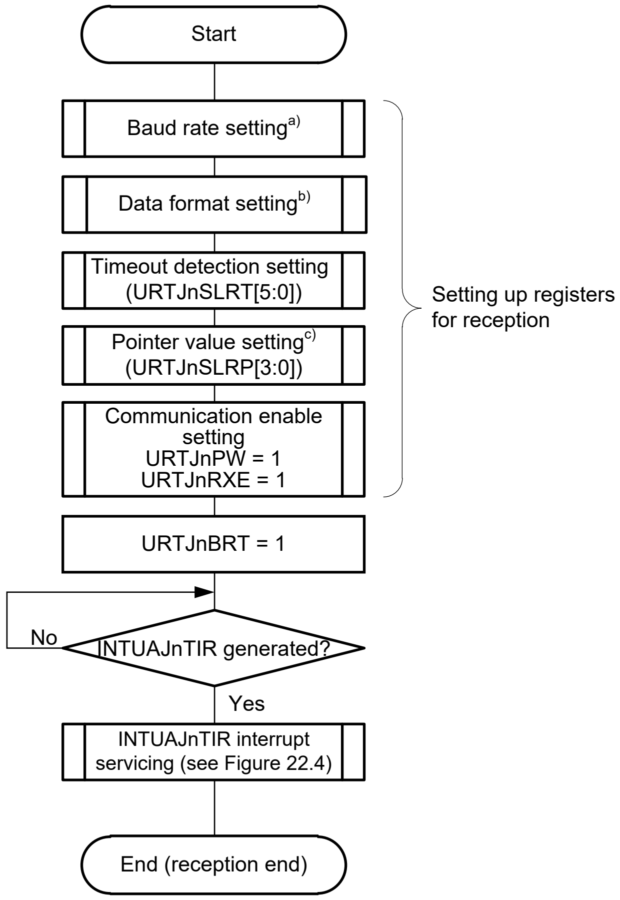
2. In continuous reception, immediately after the stop bit is detected at the first received bit (when the reception interrupt is generated), the next start bit may be detected.



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnSLBM, URTJnCLG, URTJnSLP[1:0], URTJnRDL, and URTJnSLD bits.

Figure 22.14 Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 0

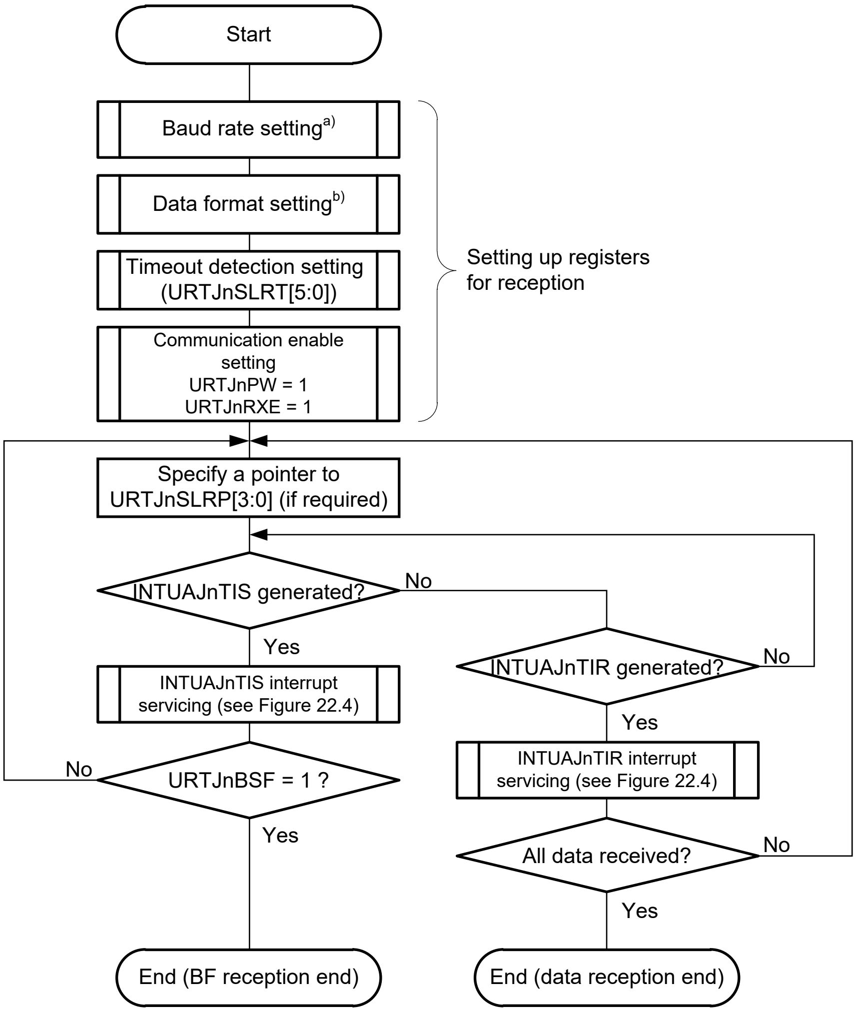


a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnSLBM, URTJnCLG, URTJnSLP[1:0], URTJnRDL, and URTJnSLD bits.

c) When receiving data after BF reception, pointer value setting.

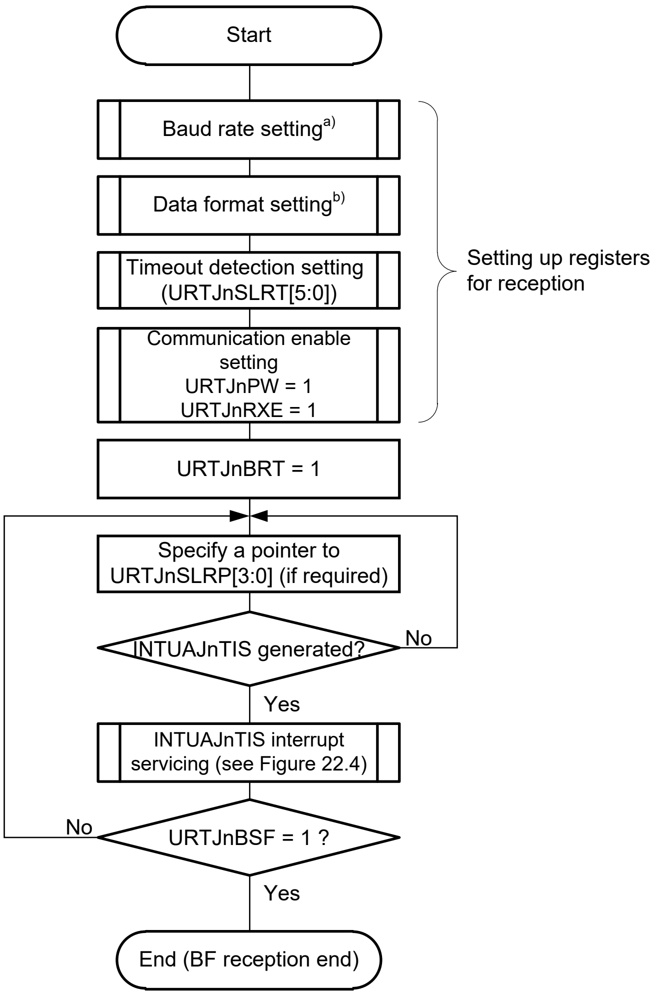
Figure 22.15 Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 1



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnSLBM, URTJnCLG, URTJnSLP[1:0], URTJnRDL, and URTJnSLD bits.

Figure 22.16 Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 0



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnSLBM, URTJnCLG, URTJnSLP[1:0], URTJnRDL, and URTJnSLD bits.

Figure 22.17 Flowchart of Data Reception when URTJnSLBM = 1, URTJnSSBR = 1

22.6.7 Reception Errors

Errors during a receive operation are of four types:

- parity errors
- framing errors
- overrun errors
- timeout errors

Various data reception result error flags are provided to identify the error cause and the status interrupt request signal INTUAJnTIS is generated when an error occurs.

Table 22.6 Reception Error Causes and Indicators

Reception Error	Error Flags	Cause
Parity error	URTJnSTR1.URTJnPE = 1 upon first parity error URTJnFRX.URTJnPE = 1 for each data in Rx FIFO	Received parity bit does not match the setting
Framing error	URTJnSTR1.URTJnFE = 1 upon first framing error URTJnFRX.URTJnFE = 1 for each data in Rx FIFO	Stop bit not detected
Overrun error	URTJnFSTR1.URTJnROVE = 1	Reception of next data completed while Rx FIFO is full
Timeout error	URTJnFSTR1.URTJnTMOE = 1	No Rx FIFO access within a certain time period

(1) Overrun error

An overrun error occurs (URTJnFSTR1.URTJnROVE = 1), when data has been received while the Rx FIFO is full. The received data is not transferred to the Rx FIFO, but is discarded.

(2) Parity and framing error

If a parity error or a framing error occurs during reception

- the associated error bit is set:
  - for a parity error: URTJnSTR1.URTJnPE = 1
  - for a framing error: URTJnSTR1.URTJnFE = 1
- reception continues until the reception position of the first stop bit,
- the reception data and the error flag URTJnFRX.URTJnPE respectively URTJnFRX.URTJnFE is transferred to the Rx FIFO,
- the status interrupt INTUAJnTIS is generated,
- in case the Rx FIFO fill level reaches the predefined level URTJnFSTR0.URTJnSSRW[4:0], the reception interrupt INTUAJnTIR is generated.

Remark: The error flags URTJnFRX.URTJnPE and URTJnFRX.URTJnFE are set upon detection of the first parity respectively framing error and stay at 1 until they are cleared by URTJnSTC.URTJnCLP = 1 respectively URTJnSTC.URTJnCLF = 1.

(3) Time-out error

A time-out error occurs under following conditions:

- The Rx FIFO is not empty.
- Neither received data has been stored in nor data has been read from the Rx FIFO within a certain time period.

The time period is programmable by setting URTJnFCTL1.URTJnSLRT[5:0]. This value specifies the time period that is a multiple of the bit-rate clock BRCLK period.

If a time-out error occurs, the flag URTJnFSTR1.URTJnTMOE is set to 1 and the status interrupt request INTUAJnTIS is generated.

22.6.8 Parity Types and Operations

Caution: When using the LIN function, fix the URTJnCTL1.URTJnSLP[1:0] to 00B.

A parity bit is used to detect bit errors in transferred data. Usually, the same type of parity will be used in transmission and reception.

Even parity and odd parity can be used to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(1) Even parity

- During transmission:
  The parity bit is controlled so that the number of 1-valued bits in the data for transmission, including the parity bit itself, is even. The value of the parity bit is as follows:
  - Odd number of 1-valued bits in the data for transmission: 1
  - Even number of 1-valued bits in the data for transmission: 0
- During reception:
  The number of 1-valued bits in the received data, including the parity bit itself, is counted. If the result is an odd number, a parity error has occurred.

(2) Odd parity

- During transmission:
  Opposite to even parity, the parity bit is controlled so that the number of 1-valued bits in the data for transmission, including the parity bit itself, is odd. The value of the parity bit is as follows:
  - Odd number of 1-valued bits in the data for transmission: 0
  - Even number of 1-valued bits in the data for transmission: 1
- During reception:

  The number of 1-valued bits in the received data, including the parity bit itself, is counted. If the result is an even number, a parity error has occurred.

(3) 0 parity

During transmission, the parity bit is always set to 0, regardless of the data for transmission.

During reception, the parity bit is not checked. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(4) No parity

A parity bit is not appended to data for transmission.

Reception proceeds with no parity bit. Since received data do not include parity bits, parity errors will not occur.

22.6.9 Digital Receive Data Noise Filter

The received-data signal input URTJnTRXD has the digital noise filter from which a noise and a glitch are removed. This filter samples an URTJnTRXD signal using PCLK (HCLK).

For details, please refer to Section 28.11, Noise Elimination Circuit.

22.7 Bit-Rate Generator

The transmission and reception bit-rate clock BRCLK are derived from the APB bus clock PCLK by use of a prescaler and a bit-rate generator, as shown in the figure below.

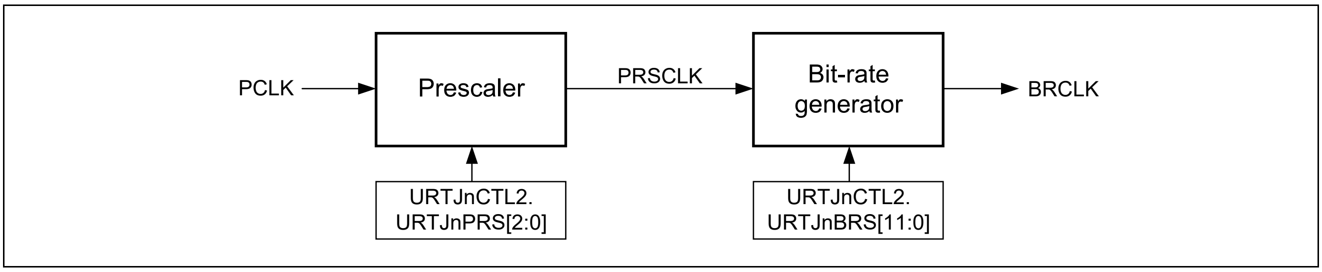


Figure 22.18 Configuration of Bit-Rate Generator

The prescaler output clock PRSCLK is a fraction of PCLK, the divisor is set up the value URTEnCTL2.URTJnPRS[2:0]:

$$PRSCLK = PCLK / 2^{URTJnPRS[2:0]}$$

PRSCLK is further divided by the bit-rate generator by a value, determined by URTJnCTL2.URTJnBRS[11:0].

The bit-rate generator distinguishes between the bit rate for data frames and BF receptions, as listed in the table below. The BF reception clock is double the bit-rate clock BRCLK.

Table 22.7 Bit-Rate Generator Clocks Output

URTJnCTL2.URTJnBRS[11:0]	Transmit/Receive BRCLK	BF Receive Clock
000H	PRSCLK/(2 × 4)	PRSCLK/4
001H		
002H		
003H		
004H		
005H	PRSCLK/(2 × 5)	PRSCLK/5
...	PRSCLK/(2 × URTJnBRS[11:0])	PRSCLK/URTJnBRS[11:0]
FFEH	PRSCLK/(2 × 4094)	PRSCLK/4094
FFFH	PRSCLK/(2 × 4095)	PRSCLK/4095

The clock setting for the bit rate is calculated from the following formula. For details of the register, see section 22.4(3), UARTJn control register 2 (URTJnCTL2).

$$\text{Bit rate} = \frac{\text{PCLK frequency}}{2 \times (\text{URTJnCTL2.URTJnBRS11}-0) \times 2^{\text{URTJnCTL2.URTJnPRS2}-0}} = \text{BRT[bps]}$$

$$\text{Error in the bit rate} = \left\{ \frac{\text{Bit rate (BRT)}}{\text{Target bit rate (target BRT)}} - 1 \right\} \times 100 = \text{ERR[\%]}$$

Remark: The settings of the register when the bit rate is set to 2.94 Mbps are as follows (decimal notation).
URTJnCTL2.URTJnBRS = 2125
URTJnCTL2.URTJnRRS = 3

The following table lists the relationships between the transmission side of an R-IN32M4-CL3 and the allowable scope of error in the bit rate.

Table 22.8 Allowable Scope of Error in Bit Rate

URTJnCTL2.URTJnBRS[11:0] (decimal notation) Note	Maximum Bit Rate	Minimum Bit Rate
4	+ 2.32%	- 2.43%
8	+ 3.52%	- 3.61%
16	+ 4.14%	- 4.19%
32	+ 4.45%	- 4.47%
64	+ 4.60%	- 4.62%
128	+ 4.68%	- 4.69%
256	+ 4.72%	- 4.72%
512	+ 4.74%	- 4.74%
1024	+ 4.75%	- 4.75%
2048	+ 4.75%	- 4.75%
4095	+ 4.75%	- 4.75%

Note: When the setting of URTJnCTL2.URTJnBRS[11:0] is 0 to 3, refer to the entry for 4.

Remark: When URTJnCTL2.URTJnBRS[11:0] = "2125", the allowable scope of error in the bit rate error is ±4.75%.

Table 22.9 Example of Bit Rate Generator Settings (PCLK = 100 MHz)

Bit Rate (bps)	URTJnPRS	URTJnBRS	ERR (%)
300	6	2604	0.01
600	5	2604	0.01
1200	4	2604	0.01
2400	3	2604	0.01
4800	2	2604	0.01
9600	1	2604	0.01
19200	0	2604	0.01
31250	0	1600	0.01
38400	0	1302	0.01
76800	0	651	0.01
115200	0	434	0.01
153600	0	326	- 0.15
312500	0	160	0.00
1000000	0	50	0.00
2000000	0	25	0.00
2500000	0	20	0.00
3125000	0	16	0.00
5000000	0	10	0.00
6250000	0	8	0.00
10000000	0	5	0.00
12500000	0	4	0.00

23. Clocked Serial Interface H (CSIH)

This section contains a generic description of clocked serial interface (CSIH).

23.1 Features of CSIH

R-IN32M4-CL3 products incorporate two channels of clocked serial interface H (CSIHn) listed below.

Table 23.1 Channels of CSIH

Clocked Serial Interface H	
Number of channels	2
Name	CSIH0, CSIH1

- Index n: Throughout this section, the individual channels of CSIH is identified by the index "n" (n = 0, 1); for example, CSIHnCTL0 for CSIHn control register 0.
- Index x: CSIH has two chip select signals. Throughout this section, the individual chip select signals are identified by the index "x" (x = 0, 1), thus a certain chip select signal is denoted as CSx. The number of chip select signals for each channel of CSIH is given in the following table:

Table 23.2 Number of Chip Select Signals of CSIH

CSIHn Channel	Number of Chip Select Signals
CSIH0	CS0, CS1
CSIH1	CS0, CS1

- Maximum transfer speed (baud rate):
  Clocked serial interface H (CSIH) can communicate at the following maximum transfer rates.

Table 23.3 Maximum Transfer Speed (Baud Rate) of CSIH

Mode	Maximum Transfer Speed (Baud Rate)
Master mode	25.0 Mbps (Max.)
Slave mode	16.6 Mbps (Max.)

- Interrupts and peripheral modules:

The following interrupt requests from CSIH can be used as triggers for interrupt service routines or hardware ISRs (where listed as such), for DMA transfer (by the general-purpose DMAC or real-time port DMAC), for capture by a timer (TAUJ2 or TAUD), and for updating the real-time port pins (RP00–RP37).

Table 23.4 CSIHn Interrupts and Requests to Peripheral Modules

CSIHn Signals	Function	Connected to
CSIH0		
CSIHTIC	Communication status interrupt	• Interrupt controller (INTCSIH0IC) • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
CSIHTIR	Reception status interrupt	• Interrupt controller (INTCSIH0IR) • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
CSIHTIRE	Reception error interrupt	• Interrupt controller (INTCSIH0IRE)
CSIHTIJC	Job completion interrupt	• Interrupt controller (INTCSIH0IJC) • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
CSIH1		
CSIHTIC	Communication status interrupt	• Interrupt controller (INTCSIH1IC) • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
CSIHTIR	Reception status interrupt	• Interrupt controller (INTCSIH1IR) • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
CSIHTIRE	Reception error interrupt	• Interrupt controller (INTCSIH1IRE)
CSIHTIJC	Job completion interrupt	• Interrupt controller (INTCSIH1IJC) • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)

- I/O signals The I/O signals of CSIH are listed in the following table.

Table 23.5 CSIHn I/O Signals

CSIHn Signal	Function	Connected to
CSIH0		
CSIHTSCK	Serial clock signal	Port 45 CSISCK0
CSIHTSI	Serial data input signal	Port 46 CSISI0
CSIHTSO	Serial data output signal	Port 47 CSISO0
CSIHTCSS1	Chip select signal 1	Port 42 CSICS00
CSIHTCSS0	Chip select signal 0	Port 43 CSICS01
CSIH1		
CSIHTSCK	Serial clock signal	Port 35 CSISCK1
CSIHTSI	Serial data input signal	Port 36 CSISI1
CSIHTSO	Serial data output signal	Port 37 CSISO1
CSIHTCSS1	Chip select signal 1	Port 70 CSICS10
CSIHTCSS0	Chip select signal 1	Port 71 CSICS11

23.2 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 two configurable chip select output signals
- Built-in baud rate generator
- Baud rate adjustable; in slave mode determined by input clock
- Maximum transfer speed:
  - in master mode: PCLK/4
  - in slave mode: PCLK/6

Caution: There might be restrictions on the maximum baud rate that can actually be used depending on the product. Specify the baud rate so as not to exceed the maximum rate of the product you are using.

- Phase of clock and data selectable
- Data transfer with MSB or LSB first selectable
- Transfer data length selectable from 7 to 16 bits in 1-bit units
- Extended data length (EDL) function for transferring data of more than 16 bits
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Error detection (data consistency check, parity, timeout, overflow, overrun)
- Full support of job concept
- 128-word I/O buffer memory
- Memory mode selectable (FIFO, dual buffer, Tx-only buffer, direct access)
- Four interrupt request signals (CSIHnTIC, CSIHnTIR, CSIHnTIRE, CSIHnTIJC)
- Loop back mode (LBM) function for self-test

The block diagram shows the main components of the CSIH.

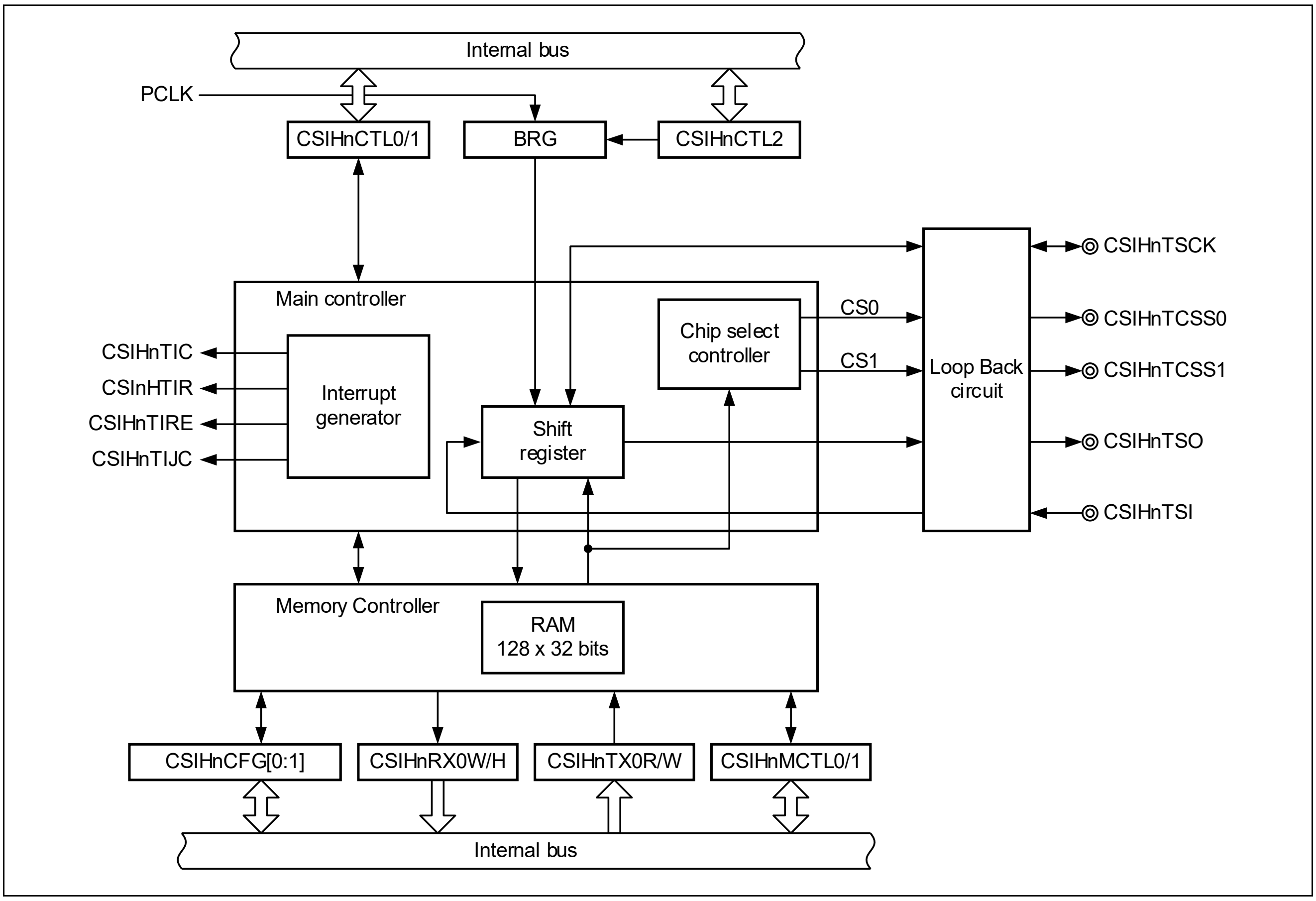


Figure 23.1 CSIH Block Diagram

In master mode, the serial clock CSIHnTSCK is generated by the internal baud rate generator (BRG). In slave mode, the serial clock is supplied 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.

Remark: This section describes the following modes:

- **The "operating mode" is divided into master and slave mode. Only a master can control and communicate with several slaves (for details, see section 23.4.1, Operating Modes (Master/Slave)).**
- **"Job mode" is related to the Autosar job concept (for details, see section 23.4.5, Job Concept).**
- **In the "memory mode", various settings for the associated buffer memory are available (for details, see section 23.4.7, CSIH Buffer Memory).**
- **"Data transfer mode" specifies the mode of communications – transmit only, receive only, or both (for details, see section 23.4.8, Data Transfer Modes).**

23.3 CSIH Control Registers

CSIHn is controlled and operated by means of the following registers:

Table 23.6 CSIH0 Register Overview

Register Name	Shortcut	Address
Control register 0	CSIH0CTL0	4000 0100H
Control register 1	CSIH0CTL1	4000 0110H
Control register 2	CSIH0CTL2	4000 0114H
Status register 0	CSIH0STR0	4000 0104H
Status clear register 0	CSIH0STCR0	4000 0108H
Memory control register 0	CSIH0MCTL0	4000 01C0H
Memory control register 1	CSIH0MCTL1	4000 0180H
Memory control register 2	CSIH0MCTL2	4000 0184H
Configuration register 0	CSIH0CFG0	4000 01C4H
Configuration register 1	CSIH0CFG1	4000 01C8H
Transmit data register 0 for word access	CSIH0TX0W	4000 0188H
Transmit data register 0 for half word access	CSIH0TX0H	4000 018CH
Receive data register 0 for word access	CSIH0RX0W	4000 0190H
Receive data register 0 for half word access	CSIH0RX0H	4000 0194H
Memory read/write pointer register 0	CSIH0MRWP0	4000 0198H

Table 23.7 CSIH1 Register Overview

Register Name	Shortcut	Address
Control register 0	CSIH1CTL0	4000 0200H
Control register 1	CSIH1CTL1	4000 0210H
Control register 2	CSIH1CTL2	4000 0214H
Status register 0	CSIH1STR0	4000 0204H
Status clear register 0	CSIH1STCR0	4000 0208H
Memory control register 0	CSIH1MCTL0	4000 02C0H
Memory control register 1	CSIH1MCTL1	4000 0280H
Memory control register 2	CSIH1MCTL2	4000 0284H
Configuration register 0	CSIH1CFG0	4000 02C4H
Configuration register 1	CSIH1CFG1	4000 02C8H
Transmit data register 0 for word access	CSIH1TX0W	4000 0288H
Transmit data register 0 for half word access	CSIH1TX0H	4000 028CH
Receive data register 0 for word access	CSIH1RX0W	4000 0290H
Receive data register 0 for half word access	CSIH1RX0H	4000 0294H
Memory read/write pointer register 0	CSIH1MRWP0	4000 0298H

23.3.1 CSIH Register Details

(1) CSIH control register 0 (CSIHnCTL0)

This register controls CSIHn. It mainly enables or disables the operation clock, transmission/reception, and the memory assigned to transmission/reception. It forcibly stops communications at the end of the current job.

- Access This register can be read/written in 32-bit or 1-bit units.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
CSIHnCTL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHnPWR	CSIHnTXE	CSIHnRXE	0	0	0	CSIHnJOBE	CSIHnMBS	4000 0100H +100H × n Initial Value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	0	0	0	R/W	R/W	

Bit Position	Bit Name	Function
31 to 8	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
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 the CSIH to standby state. Clock supply to internal circuits is stopped. If CSIHnPWR is cleared during communication, ongoing communication is immediately aborted. In this case, it is necessary to restart communication from the beginning.
6	CSIHnTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIHnRXE	Enables/disables reception. 0: Receive disabled 1: Receive enabled
4 to 2	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
1	CSIHnJOBE	Stops the communication at the end of the current job (Communication ends when data is written to the transmission buffer while CSIHnTX0W.CSIHnEOJ = 1 (indicating that the job has ended).). 0: Communication stop is not required 1: Communication stop This bit can be used to abort an ongoing job. It is automatically cleared. Even if this bit is set, 0 is always returned when it is read. In FIFO mode, the next communication should then be started after clearing the pointers by setting CSIHnSTCR0.CSIHnPCT = 1. Caution: CSIHnJOBE is only valid when CSIHnCTL0.CSIHnPWR = 1 and CSIHnCTL1.CSIHnJE = 1, and while data transfer is in progress. This bit is automatically cleared to 0 at the end of transfer. Setting this bit while in slave mode is prohibited. This bit is always read as 0.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
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: In slave mode, perform rewriting at the same time that CSIHnCTL0.CSIHnPWR changes from 0 to 1.

Remark: n = 0, 1

Cautions 1. When CSIHnPWR = 0, do not change the CSIHnTXE, CSIHnRXE, CSIHnJOBE, or CSIHnMBS bit. However, the CSIHnTXE, CSIHnRXE, or CSIHnMBS bit can be changed at the same time as the CSIHnPWR bit changes from 0 to 1.

2. Do not modify CSIHnTXE or CSIHnRXE or CSIHnMBS while a data transmission is pending or in progress, i.e. if CSIHnSTR0.CSIHnTSF = 1.

(2) CSIH control register 1 (CSIHnCTL1)

This register controls CSIHn. It mainly specifies the clock phase, interrupt timing, and interrupt delay mode, controls the extended data length, and enables or disables the data consistency check, loopback mode, and job mode. This register also selects the active output level of each chip select signal and the chip select signal operation to perform after the last data is transferred.

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

Caution: Changing the contents of this register is only permitted when CSIHnCTL0.CSIHnPWR = 0.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
CSIHnCTL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHnCKR	CSIHnSLIT	0	0	0	0	0	0	CSIHnCSL[1:0]		CSIHnEDLE	CSIHnJE	CSIHnDCS	CSIHnCSRI	CSIHnLBM	CSIHnSIT	0	0	4000 0110H +100H × n Initial Value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	0	0	0	0	0	0	R/W		R/W	R/W	R/W	R/W	R/W	R/W	0	0	

Bit Position	Bit Name	Function
31 to 18	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
17	CSIHnCKR	Selects the CSIHnTSCK clock phase. 0: The default CSIHnTSCK level is the high level. 1: The default CSIHnTSCK level is the low level. Caution: When using this bit without using the chip select function, clear CSIHnCFGx.CSIHnCKPx to 0.
16	CSIHnSLIT	Selects the timing of interrupt CSIHnTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: When the contents of the CSIHnTX0W or CSIHnTX0H register are transferred to the shift register, an interrupt is immediately generated (This only functions in direct access mode). For details, refer to 23.4.12(1), CSIHnTIC (communication interrupt).
15 to 10	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
9, 8	CSIHnCSL[1:0]	Selects the active output level of chip select signal x (CSIHnTCSSx; x = 0, 1)). 0: Chip select is active low 1: Chip select is active high For details, refer to section 23.4.3, Chip Selection (CS) Features.
7	CSIHnEDLE	Enables/disables extended data length (EDL) mode. 0: Extended data length mode disabled 1: Extended data length mode enabled For details, refer to 23.4.9(2), Data length greater than 16 bits.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
6	CSIHnJE	Enables/disables job mode. 0: Job mode disabled 1: Job mode enabled For details, refer to section 23.4.5, Job Concept. The CSIHnCTL0.CSIHnJOBE, CSIHnTX0W.CSIHnEOJ, and CSIHnTX0W.CSIHnCIRE bits are only valid when this bit is 1. Setting this bit is prohibited in slave mode.
5	CSIHnDCS	Enables/disables data consistency check. 0: Data consistency check disabled 1: Data consistency check enabled For details, refer to 23.4.13(1), Data consistency checking.
4	CSIHnCSRI	Defines chip select behavior after last data transfer. 0: Chip select holds active level 1: Chip select returns to inactive level The last data is identified at the interrupt timing while in direct access mode or FIFO mode. Direct access mode is used while CSIHnCTL1.CSIHnSLIT = 1.
3	CSIHnLBM	Controls loop-back mode (LBM). 0: Loop-back mode deactivated 1: Loop-back mode activated For details, refer to section 23.4.14, Loop-Back Mode. Setting this bit is prohibited in slave mode.
2	CSIHnSIT	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, refer to 23.4.12(5), Delay for all interrupts.
1, 0	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.

Remark: n = 0, 1

(3) CSIH control register 2 (CSIHnCTL2)

This register controls CSIHn. It selects the operating mode, prescaler, and baud rate.

For details, refer to section 23.4.6, Serial Clock Selection.

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

Caution: Changing the contents of this register is only permitted when CSIHnCTL0.CSIHnPWR = 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnCTL2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHnPRS[2:0]			0	CSIHnBRS[11:0]											
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W			0	R/W											

Address 4000 0114H +100H × n

Initial Value 0000 E000H

Bit Position	Bit Name	Function
31 to 16, 12	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 13	CSIHnPRS [2:0]	Selects the operating mode and the value of the base clock.
11 to 0	CSIHnBRS [11:0]	Selects the baud rate.

CSIHnPRS[2:0]:

CSIHn PRS2	CSIHn PRS1	CSIHn PRS0	Base clock (PRSOUT) selection
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 by way of CSIHnTSCK (in) (slave mode)

CSIHnBRS[11:0]:

CSIHnBRS[11:0]	Baud rate at CSIHnTBLK
0	BRG is stopped
1	PCLK/ (2^m × 1 × 2)
2	PCLK/ (2^m × 2 × 2)
3	PCLK/ (2^m × 3 × 2)
...	...
4095	PCLK/ (2^m × 4095 × 2)

Remark: m = 0–6: Value set in CSIHnPRS[2:0]

Remark: n = 0, 1

(4) CSIH status register 0 (CSIHnSTR0)

This register indicates the status of the CSIH.

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

(1/4)

	31 to 24	23 to 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnSTR0	CSIHnSRP[7:0]	CSIHnSPF[7:0]	CSIHnTMOE	CSIHnOFE	0	0	0	0	0	0	CSIHnTSF	0	CSIHnFLF	CSIHnEMF	CSIHnDCE	0	CSIHnPE	CSIHnOVE
R/W	R	R	R	R	0	0	0	0	0	0	R	0	R	R	R	0	R	R

Address: 4000 0104H +100H × n

Initial Value: 0000 0010H

Bit Position	Bit Name	Function
31 to 24	CSIHnSRP [7:0]	Indicates the number of received words in FIFO mode. CSIHnSRP[7:0] | Description 00H : 80H | Number of received words (0 to 128) Other than the above | Setting prohibited These bits are cleared by CSIHnSTCR0.CSIHnPCT. In dual buffer mode or transmit-only buffer mode, because the number of data items is managed according to CSIHnMCTL2.CSIHnND[7:0], these bits are fixed to 00H. They are also fixed to 00H in direct access mode because there is no pointer.
23 to 16	CSIHnSPF [7:0]	Indicates the number of unsent data in FIFO mode. CSIHnSPF[7:0] | Description 00H : 80H | Number of unsent data (0–128) Other than the above | Setting prohibited These bits are cleared by CSIHnSTCR0.CSIHnPCT. In dual buffer mode or transmit-only buffer mode, because the number of data items is managed according to CSIHnMCTL2.CSIHnND[7:0], these bits are fixed to 00H. They are also fixed to 00H in direct access mode because there is no pointer.

Remark: n = 0, 1

(2/4)

Bit Position	Bit Name	Function
15	CSIHnTMOE	Timeout error flag in FIFO mode Indicates whether a timeout error was detected in FIFO mode. 0: No timeout error was detected in FIFO mode. 1: A timeout error was detected in FIFO mode. For details, see 23.4.13(3), Timeout error. This bit is cleared by CSIHnSTCR0.CSIHnTMOEC. This bit can be written to when CSIHnSTCR0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. If this bit is set due to a timeout error being detected and cleared by CSIHnSTCR0.CSIHnTMOEC at the same time, setting the bit is prioritized.
14	CSIHnOFE	Overflow error flag in FIFO mode Indicates whether an overflow error was detected in FIFO mode. 0: No overflow error was detected in FIFO mode. 1: An overflow error was detected in FIFO mode. For details, see 23.4.13(4), Overflow error. This bit is cleared by CSIHnSTCR0.CSIHnOFEC. This bit can be written to when CSIHnSTCR0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. If 129 transmission data items are written to the CSIHnTX0W or CSIHnTX0H register when CSIHnCTL0.CSIHnPWR = 0, an overflow error occurs. If this bit is set due to an overflow error being detected and cleared by CSIHnSTCR0.CSIHnOFEC at the same time, setting the bit is prioritized.

Remark: n = 0, 1

(3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>CSIHnTSF</td><td>Transfer status flag
0: Idle state
1: Transmission is in progress or being prepared
Setting and clearing conditions of this bit are as follows:
<table>
<tr><th rowspan="2">Master Mode</th><th colspan="2">Setting Condition</th><th rowspan="2">Clearing Condition</th></tr>
<tr><th>Direct Access Mode, FIFO Mode</th><th>Dual Buffer Mode, Transmit Only Buffer Mode</th></tr>
<tr><td>Transmission mode</td><td rowspan="3">Writing to transmit data register</td><td rowspan="3">Setting CSIHnMCTL2. CSIHnBTST</td><td rowspan="3">Within 0.5 clock cycles from the last CSIHnTSCK edge</td></tr>
<tr><td>Transmission / reception mode</td></tr>
<tr><td>Reception mode</td></tr>
</table>
<table>
<tr><th rowspan="2">Slave Mode</th><th colspan="2">Setting Condition</th><th rowspan="2">Clearing Condition</th></tr>
<tr><th>Direct Access Mode, FIFO Mode</th><th>Dual Buffer Mode, Transmit Only Buffer Mode</th></tr>
<tr><td>Transmission mode</td><td rowspan="2">Writing to transmit data register</td><td rowspan="3">Setting CSIHnMCTL2. CSIHnBTST</td><td rowspan="3">Within 0.5 clock cycles from the last CSIHnTSCK edge</td></tr>
<tr><td>Transmission / reception mode</td></tr>
<tr><td>Reception mode</td><td>CSIHnTSCK input timing</td></tr>
</table>
</td></tr>
<tr><td>5</td><td>CSIHnFLF</td><td>Indicates whether the buffer is full while in FIFO mode.
0: FIFO buffer is not full
1: FIFO buffer is full
This bit is set when the sum of the CSIHnSTR0.CSIHnSRP[7:0] bit value and CSIHnSTR0.CSIHnSPF[7:0] bit value matches 80H, and is cleared when this sum does not match 80H.
This bit is cleared by CSIHnSTCR0.CSIHnPCT.
The FIFO buffer may be filled up with data that has not been transmitted and reception data.</td></tr>
</table>

Remark: n = 0, 1

(4/4)

Bit Position	Bit Name	Function
4	CSIHnEMF	Indicates whether the buffer is empty while in FIFO mode. 0: FIFO buffer is not empty 1: FIFO buffer is empty This bit is set by SCIHnSTCR0.CSIFnPCT. This bit is set when the sum of the CSIHnSTR0.CSIHnSRP[7:0] bit value and CSIHnSTR0.CSIHnSPF[7:0] bit value matches 00H, and is cleared when this sum does not match 00H. The FIFO buffer may not contain any data that has not been transmitted or reception data.
3	CSIHnDCE	Data consistency error flag 0: No data consistency error detected 1: Data consistency error detected This bit is cleared by setting CSIHnSTCR0.CSIHnDCEC. This bit can be written to when CSIHnCTL0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. If this bit is set due to a data consistency error being detected and cleared by CSIHnSTCR0.CSIHnDCEC at the same time, setting the bit is prioritized.
1	CSIHnPE	Parity error flag 0: No parity error detected 1: Parity error detected This bit is cleared by setting CSIHnSTCR0.CSIHnPEC. This bit can be written to when CSIHnCTL0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. If this bit is set due to a parity error being detected and cleared by CSIHnSTCR0.CSIHnPEC at the same time, setting the bit is prioritized.
0	CSIHnOVE	Overrun error flag (fixed to 0 in dual buffer mode) 0: No overrun error detected 1: Overrun error detected This bit is cleared by setting CSIHnSTCR0.CSIHnOVEC. This bit can be written to when CSIHnCTL0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. This bit is fixed to 0 in dual buffer mode. If this bit is set due to an overrun error being detected and cleared by CSIHnSTCR0.CSIHnOVEC at the same time, setting the bit is prioritized.
13 to 8, 6, 2	—	Reserved. These bits are read as 0.

Remark: n = 0, 1

Table 23.8 Operation in Memory Mode

Bit Name	Bit Position	Direct Access Mode	FIFO Mode	Transmit-Only Buffer Mode	Dual Buffer Mode
CSIHnSRP [7:0]	31 to 24	Fixed to 0	Number of received data items	Fixed to 0	Fixed to 0
CSIHnSPF [7:0]	23 to 16	Fixed to 0	Number of data items that have not been transmitted	Fixed to 0	Fixed to 0
CSIHnTMOE	15	Fixed to 0	0: No error has been detected. 1: An error has been detected.	Fixed to 0	Fixed to 0
CSIHnOFE	14	Fixed to 0	0: No error has been detected. 1: An error has been 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: Not full 1: Full	Fixed to 0	Fixed to 0
CSIHnEMF	4	Fixed to 1	0: Not empty 1: Empty	Fixed to 1	Fixed to 1
CSIHnDCE	3	0: No error has been detected. 1: An error has been detected.			
CSIHnPE	1	0: No error has been detected. 1: An error has been detected.			
CSIHnOVE	0	0: No error has been detected. 1: An error has been detected.			Fixed to 0

Remark: n = 0, 1

(5) CSIH status clear register 0 (CSIHnSTCR0)

This register clears the status flags of the CSIHnSTR0 status register.

- Access This register can be written in 32-bit units.
  When read, the value 0000 0000H is always returned.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHn STCR0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHnTMOEC	CSIHnOFEC	0	0	0	0	0	CSIHnPCT	0	0	0	0	CSIHnDCEC	0	CSIHnPEC	CSIHnOVEC
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W	W	0	0	0	0	0	W	0	0	0	0	W	0	W	W

Address 4000 0108H +100H × n
Initial Value 0000 0000H

Bit Position	Bit Name	Function
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15	CSIHnTMOEC	Controls the timeout error flag clear command. 0: No operation. Read value is always 0. 1: Clear time out error flag (CSIHnSTR0.CSIHnTMOE)
14	CSIHnOFEC	Controls the overflow error flag clear command 0: No operation. Read value is always 0. 1: Clear overflow error flag (CSIHnSTR0.CSIHnOFE)
13 to 9	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
8	CSIHnPCT	Controls the FIFO buffer pointers. 0: No operation. Read value is always 0. 1: In dual buffer mode, transmit-only buffer mode, or FIFO mode, clear all the following FIFO buffer pointers: - CSIHnMRWP0.CSIHnTRWA[6:0] - CSIHnMRWP0.CSIHnRRA[6:0] - CSIHnMCTL2.CSIHnSOP[6:0] Only in FIFO mode, also clear all the following status bits: - CSIHnSTR0.CSIHnSPF[7:0] - CSIHnSTR0.CSIHnSRP[7:0] - CSIHnSTR0.CSIHnFLF - CSIHnSTR0.CSIHnTSF Also, CSIHnSTR0.CSIHnEMF is set (indicating an empty FIFO buffer). Caution: When this bit is set during communication, the communication stops.
7 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3	CSIHnDCEC	Controls the data consistency error flag clear command. 0: No operation. Read value is always 0. 1: Clear data consistency error flag (CSIHnSTR0.CSIHnDCE)
2	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
1	CSIHnPEC	Controls the parity error flag clear command. 0: No operation. Read value is always 0. 1: Clear parity error flag (CSIHnSTR0.CSIHnPE)
0	CSIHnOVEC	Controls the overrun error flag clear command. 0: No operation. Read value is always 0. 1: Clear overrun error flag (CSIHnSTR0.CSIHnOVE)

Remark: n = 0, 1

(6) CSIH memory control register 0 (CSIHnMCTL0)

This register selects the memory mode and the timeout setting.

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

CSIHn MCTL0 — Address: 4000 01C0H +100H × n; Initial Value: 0000 001FH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHn MCTL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHnMMS[1:0]		0	0	0	CSIHnTO[4:0]				
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W		0	0	0	R/W				

Bit Position	Bit Name	Function
31 to 10	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
9, 8	CSIHnMMS [1:0]	Selects the memory mode. CSIHnMMS1 / CSIHnMMS0 / Description: 0 / 0 / FIFO mode 0 / 1 / Dual buffer mode 1 / 0 / Transmit-only buffer mode 1 / 1 / Prohibited After changing the memory mode, set the CSIHnSTCR0.CSIHnPCT bit and clear the individual buffer pointers and other data. Caution: The memory mode can only be changed when CSIHnCTL0.CSIHnPWR and CSIHnCTL0.CSIHnMBS = 0.
7 to 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4 to 0	CSIHnTO [4:0]	Select the number of clock cycles until the timeout is reached. CSIHnTO[4:0] / Description: 00000B / No timeout detection 00001B / Timeout is (1 × 8 × BRG output clock cycles) 00010B / Timeout is (2 × 8 × BRG output clock cycles) ... / ... 11111B / Timeout is (31 × 8 × BRG output clock cycles) Caution: The timeout setting can only be changed when CSIHnCTL0.CSIHnPWR = 0. Clear these bits to 00000B when in master mode or a memory mode other than FIFO mode (direct access mode, dual buffer mode, or transmission mode). For details about timeout detection, see also 23.4.13(3), Timeout error.

Remark: n = 0, 1

(7) CSIH Memory control register 1 (CSIHnMCTL1)

This register selects the conditions to generate the interrupt requests CSIHnTIC and CSIHnTIR in FIFO mode.

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

	31	30	29	28	27	26	25	24	23	22 to 16	15	14	13	12	11	10	9	8	7	6 to 0
CSIHn MCTL1	0	0	0	0	0	0	0	0	0	CSIHnFES[6:0]	0	0	0	0	0	0	0	0	0	CSIHnFFS[6:0]
R/W	0	0	0	0	0	0	0	0	0	R/W	0	0	0	0	0	0	0	0	0	R/W

Address: 4000 0180H +100H × n

Initial Value: 0000 0000H

Bit Position	Bit Name	Function
31 to 23	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
22 to 16	CSIHnFES [6:0]	Select the condition for generating the CSIHnTIC interrupt (no transmission data). When the number of transmission data items in the FIFO buffer that have not been transmitted (checked by using the CSIHnSTR0.CSIHnSPF[7:0] bits) matches CSIHnMCTL1.CSIHnFES[6:0], a CSIHnTIC interrupt request is generated.
15 to 7	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
6 to 0	CSIHnFFS [6:0]	Select the condition for generating the CSIHnTIR interrupt (a full reception buffer). When the number of received data items in the FIFO buffer (checked by using the CSIHnSTR0.CSIHnSRP[7:0] bits) matches (128 - CSIHnMCTL1.CSIHnFFS[6:0]), a CSIHnTIR interrupt request is generated.

Remark: n = 0, 1

(8) CSIH Memory control register 2 (CSIHnMCTL2)

This register controls memory operations while in dual buffer mode or transmit-only buffer mode and generates triggers to start communication.

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

Cautions 1. Writing to this register is prohibited when CSIHnSTR0.CSIHnTSF = 1 (during a transfer).

2. Writing to the CSIHnMCTL2 register is prohibited in the following cases:

• When CSIHnCTL0.CSIHnPWR = 0

• When CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0

• When in direct access mode or FIFO mode

(1/2)

	31	30	29	28	27	26	25	24	23 to 16	15	14	13	12	11	10	9	8	7	6 to 0
CSIHn MCTL2	CSIHnBTST	0	0	0	0	0	0	0	CSIHnND[7:0]	0	0	0	0	0	0	0	0	0	CSIHnSOP[6:0]
R/W	R/W	0	0	0	0	0	0	0	R/W	0	0	0	0	0	0	0	0	0	R/W

Address: 4000 0184H +100H × n

Initial Value: 0000 0000H

Bit Position	Bit Name	Function
31	CSIHnBTST	Provides a start trigger for buffer transfer. 0: No operation 1: 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. In direct access mode and FIFO mode, this bit is disabled.
30 to 24	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
23 to 16	CSIHnND [7:0]	Specify the number of data items. When read, these bits indicate the number of remaining communication data items. CSIHnND [7:0] / Dual Buffer Mode / Transmit-Only Buffer Mode / FIFO Mode / Direct Access Mode: 00H / Transmit 0 data items. / Transmit 0 data items. / No effect / No effect 01H / Transmit 1 data items. / Transmit 1 data items. / No effect / No effect ... / ... / ... / No effect / No effect 3FH / Transmit 63 data items. / Transmit 63 data items. / No effect / No effect 40H / Transmit 64 data items. / Transmit 64 data items. / No effect / No effect ... / Prohibited / ... / No effect / No effect 7FH / Prohibited / Transmit 127 data items. / No effect / No effect 80H / Prohibited / Transmit 128 data items. / No effect / No effect other / Setting prohibited The value of these bits is automatically decremented after transferring the data. During a transfer, the number of remaining data items can be read from these bits. The value of these bits is not decremented while in direct access mode.
15 to 7	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
6 to 0	CSIHnSOP [6:0]	Select the transmission data pointer. CSIHnSOP [6:0] / Dual Buffer Mode / Transmit-Only Buffer Mode / FIFO Mode / Direct Access Mode: 00H / 0000H / 0000H / 0000H / No effect 01H / 0004H / 0004H / 0004H / No effect ... / ... / ... / ... / ... 3FH / 00FCH / 00FCH / 00FCH / No effect 40H / Prohibited / 0100H / 0100H / No effect ... / Prohibited / ... / ... / No effect 7FH / Prohibited / 01FCH / 01FCH / No effect When CSIHnCTL0.CSIHnPWR = 0 or CSIHnSTR0.CSIHnPCT is set to forcibly stop communication, these bits are cleared by the hardware. Note: In FIFO mode, these bits indicate the transmission address. The value of these bits is not decremented while in direct access mode.

CSIHnND [7:0] values:

CSIHnND [7:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00H	Transmit 0 data items.	Transmit 0 data items.	No effect	No effect
01H	Transmit 1 data items.	Transmit 1 data items.	No effect	No effect
...	...	...	No effect	No effect
3FH	Transmit 63 data items.	Transmit 63 data items.	No effect	No effect
40H	Transmit 64 data items.	Transmit 64 data items.	No effect	No effect
...	Prohibited	...	No effect	No effect
7FH	Prohibited	Transmit 127 data items.	No effect	No effect
80H	Prohibited	Transmit 128 data items.	No effect	No effect
other	Setting prohibited			

CSIHnSOP [6:0] values:

CSIHnSOP [6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00H	0000H	0000H	0000H	No effect
01H	0004H	0004H	0004H	No effect
...	...	...	...	...
3FH	00FCH	00FCH	00FCH	No effect
40H	Prohibited	0100H	0100H	No effect
...	Prohibited	...	...	No effect
7FH	Prohibited	01FCH	01FCH	No effect

Remark: n = 0, 1

(9) CSIH memory read/write/pointer register 0 (CSIHnMRWP0)

This register sets the pointers for reading from and writing to the dual or transmit-only buffer.

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

Caution: This register can be written to during communication.
Writing to this register in direct access mode or FIFO mode is prohibited.

(1/2)

	31	30	29	28	27	26	25	24	23	22 to 16	15	14	13	12	11	10	9	8	7	6 to 0
CSIHn MRWP0	0	0	0	0	0	0	0	0	0	CSIHnRRA[6:0]	0	0	0	0	0	0	0	0	0	CSIHnTRWA[6:0]
R/W	0	0	0	0	0	0	0	0	0	R/W	0	0	0	0	0	0	0	0	0	R/W

Address: 4000 0198H +100H × n
Initial Value: 0000 0000H

Bit Position	Bit Name	Function
31 to 23	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
22 to 16	CSIHnRRA [6:0]	Selects the read pointer of the Rx buffer.

CSIHnRRA [6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00H	0000H	No effect	0000H	No effect
01H	0004H	No effect	0004H	No effect
...	...	No effect	...	No effect
3FH	00FCH	No effect	00FCH	No effect
40H	Prohibited	No effect	0100H	No effect
...	Prohibited	No effect	...	No effect
7FH	Prohibited	No effect	01FCH	No effect

These bits are automatically incremented when reception data is read.

If an overrun error occurs while reading the CSIHnRX0W or CSIHnRX0H register (when the CPU reads the CSIHnRX0W or CSIHnRX0H register while there is no data), the read pointer is not incremented.

These bits are cleared when CSIHnSTCR0.CSIHnPCT is set.

These bits are not incremented in direct access mode or transmit-only buffer mode.

When writing in transmit-only buffer mode, clear these bits to 0000H.

In FIFO mode, these bits indicate the read address of the reception data.

Remark: n = 0, 1

(2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 7</td><td>—</td><td>Reserved. When writing to these bits, write 0. When read, 0 is returned.</td></tr>
<tr><td>6 to 0</td><td>CSIHnTRWA [6:0]</td><td>Selects the read/write pointer of the Tx buffer.
<table>
<tr><th>CSIHnTRWA [6:0]</th><th>Dual Buffer Mode</th><th>Transmit-Only Buffer Mode</th><th>FIFO Mode</th><th>Direct Access Mode</th></tr>
<tr><td>00H</td><td>0000H</td><td>0000H</td><td>0000H</td><td>No effect</td></tr>
<tr><td>01H</td><td>0004H</td><td>0004H</td><td>0004H</td><td>No effect</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>No effect</td></tr>
<tr><td>3FH</td><td>00FCH</td><td>00FCH</td><td>00FCH</td><td>No effect</td></tr>
<tr><td>40H</td><td>Prohibited</td><td>0100H</td><td>0100H</td><td>No effect</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>...</td><td>No effect</td></tr>
<tr><td>7FH</td><td>Prohibited</td><td>01FCH</td><td>01FCH</td><td>No effect</td></tr>
</table>
When transmission data is read or written from the CPU, these bits are automatically incremented.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set.
In direct access mode, these bits are not incremented.
In FIFO mode, these bits indicate the read/write address of the transmission data.</td></tr>
</table>

Remark: n = 0, 1

(10) CSIH configuration register x (CSIHnCFGx)

These two registers specify for each chip select signal CSIHnTCSSx prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock phase and data phase, setting for the forced idle state, idle timing, hold timing, inter-data time, and setup timing.

- Slave mode: In slave mode, the transmission protocol settings of the CSIHnCFG0 register are valid:
  - CSIHnPS0: Parity usage
  - CSIHnDLS0: Data length selection
  - CSIHnDIR0: Data direction
  - CSIHnCKP0, CSIHnDAP0: Clock phase and data phase

  In slave mode, clear the CSIHnCFG0 register bits other than the above and the CSIHnCFG1 register to 0.
- Access: This register can be read/written in 32-bit units.
- Address: CSIH0CFG0: 4000 01C4H
CSIH0CFG1: 4000 01C8H
CSIH1CFG0: 4000 02C4H
CSIH1CFG1: 4000 02C8H

Caution: Writing is only possible while CSIHnCTL0.CSIHnPWR = 0 (writing is possible while CSIHnCTL0.CSIHnPWR = 1 if the same value is written).

(1/5)

Address: refer to above
Initial Value: 0000 0000H

Bit	31, 30	29, 28	27 to 24	23	22	21	20	19	18	17	16	15	14 to 12	11 to 8	7 to 4	3 to 0
CSIHnCFGx	CSIHnPSCLx[1:0]	CSIHnPSx[1:0]	CSIHnDLSx[3:0]	0	0	0	0	CSIHnRCBx	CSIHnDIRx	CSIHnCKPx	CSIHnDAPx	CSIHnIDLx	CSIHnIDx[2:0]	CSIHnHDx[3:0]	CSIHnINx[3:0]	CSIHnSPx[3:0]
R/W	R/W	R/W	R/W	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
31, 30	CSIHnPSCLx [1:0]	Selects the prescaler for chip select x. CSIHnPSCLx1 / CSIHnPSCLx0 / Prescaler Output: 0 / 0 / CSIHnBCLK 0 / 1 / CSIHnBCLK / 2 1 / 0 / CSIHnBCLK / 4 1 / 1 / CSIHnBCLK / 8 These bits are only available in master mode. For details about CSIHnBPCLK, see section 23.4.6, Serial Clock Selection.

Prescaler selection table for CSIHnPSCLx[1:0] (nested in the Function cell above):

CSIHnPSCLx1	CSIHnPSCLx0	Prescaler Output
0	0	CSIHnBCLK
0	1	CSIHnBCLK / 2
1	0	CSIHnBCLK / 4
1	1	CSIHnBCLK / 8

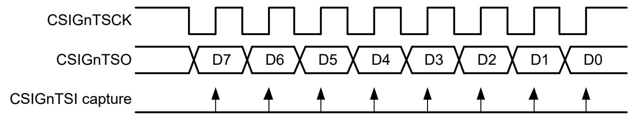
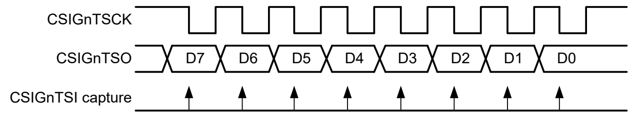
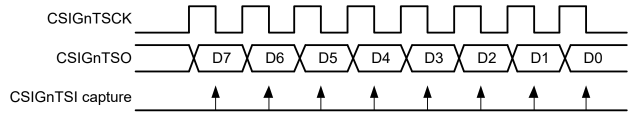
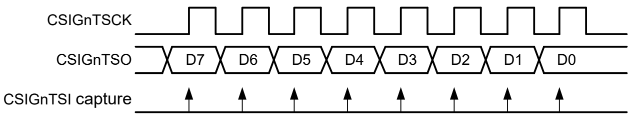
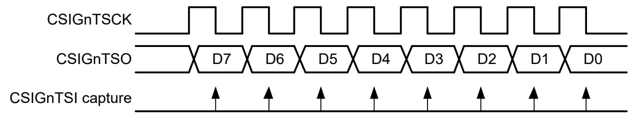
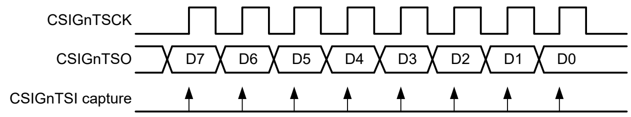
Remark: n = 0, 1; x = 0, 1

(2/5)

Bit Position	Bit Name	Function
29, 28	CSIHnPSx [1:0]	Selects the parity for chip select x for transmission and reception. <table><tr><th>CSIHnPSx1</th><th>CSIHnPSx0</th><th>Transmission</th><th>Reception</th></tr><tr><td>0</td><td>0</td><td>No parity transmitted</td><td>Parity reception is not expected.</td></tr><tr><td>0</td><td>1</td><td>Add parity bit fixed at 0</td><td>Parity bit reception is expected, but parity judgment is not performed.</td></tr><tr><td>1</td><td>0</td><td>Add odd parity</td><td>Odd parity bit reception is expected.</td></tr><tr><td>1</td><td>1</td><td>Add odd parity</td><td>Even parity bit reception is expected.</td></tr></table>
27 to 24	CSIHnDLSx [3:0]	Selects the data length for chip select x. <table><tr><th>CSIHnDLSx[3:0]</th><th>Data Length</th></tr><tr><td>0000B</td><td>16 bits</td></tr><tr><td>0001B</td><td>1 bit</td></tr><tr><td>0010B</td><td>2 bits</td></tr><tr><td>...</td><td>...</td></tr><tr><td>1111B</td><td>15 bits</td></tr></table>Note: For details about the CSIHnDLSx[3:0] bit setting, see section 23.4.9, Data Length Selection. For the CSIHnDLSx[3:0] bits, 0001B (1 bit) to 0110B (6 bits) can be specified only when the data length is 16 bits or more.
23 to 20	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
19	CSIHnRCBx	Selects the recessive configuration for broadcasting for chip select x. 0: Dominant (higher priority) 1: Recessive (lower priority) For details, see 23.4.3(1), Configuration registers.
18	CSIHnDIRx	Selects the serial data direction for chip select x. 0: Data is sent/received with MSB first 1: Data is sent/received with LSB first For details, see section 23.4.10, Serial Data Direction Selection.

Note: n = 0, 1; x = 0, 1

(3/5)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>17, 16</td><td>CSIHnCKPx,
CSIHnDAPx</td><td>
CSIHnCKPx: Clock phase select bit
CSIHnDAPx: Data phase select bit
CSIHnCTL1.CSIHnCKR = 0
<table>
<tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Specifying the timing of transmission or reception for CSIHnTSCK</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>Specifying the timing of transmission or reception for CSIHnTSCK</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>X</td><td>Setting prohibited</td></tr>
</table>
Caution: When not using the chip select function, fix the CSIHnCKPx bit to 0, and use the CSIHnCTL1.CSIHnCKR bit to specify the clock phase.
</td></tr>
</table>

Remark: n = 0, 1; x = 0, 1

(4/5)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15</td><td>CSIHnIDLx</td><td>Selects the setting of the forced idle state for chip select x.
0: If the chip select value did not change, the chip select signal stays active. If a different chip select value is defined, chip select signal x becomes idle.
1: An idle state is inserted after every transfer to chip select x.

This bit is only available in master mode.
If CSIHnCTL1.CSIHnJE = 1 and CSIHnTX0W.CSIHnEOJ = 1, chip select signal x definitely becomes idle even if CSIHnCFG0–1.CSIHnIDLn is cleared to 0.
For details about the idle state, see section 23.4.3, Chip Selection (CS) Features.</td></tr>
<tr><td>14 to 12</td><td>CSIHnIDx
[2:0]</td><td>Selects the idle time for chip select x.
<table>
<tr><th>CSIHnIDx[2:0]</th><th>Idle Timing</th></tr>
<tr><td>000B</td><td>0.5 serial clock cycles</td></tr>
<tr><td>001B</td><td>1.0 serial clock cycles</td></tr>
<tr><td>010B</td><td>1.5 serial clock cycles</td></tr>
<tr><td>011B</td><td>2.5 serial clock cycles</td></tr>
<tr><td>100B</td><td>3.5 serial clock cycles</td></tr>
<tr><td>101B</td><td>4.5 serial clock cycles</td></tr>
<tr><td>110B</td><td>6.5 serial clock cycles</td></tr>
<tr><td>111B</td><td>8.5 serial clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
<tr><td>11 to 8</td><td>CSIHnHDx
[3:0]</td><td>Selects the hold time for chip select x in transmission clock cycles.
<table>
<tr><th>CSIHnHDx
[3:0]</th><th>Hold Timing with
CSIHnCTL1.CSIHnSIT = 0</th><th>Hold Timing with
CSIHnCTL1.CSIHnSIT = 1</th></tr>
<tr><td>0000B</td><td>0.5 serial clock cycles</td><td>1.0 serial clock cycles</td></tr>
<tr><td>0001B</td><td>1 serial clock cycles</td><td>1.5 serial clock cycles</td></tr>
<tr><td>0010B</td><td>1.5 serial clock cycles</td><td>2.0 serial clock cycles</td></tr>
<tr><td>0011B</td><td>2.5 serial clock cycles</td><td>3.0 serial clock cycles</td></tr>
<tr><td>0100B</td><td>3.5 serial clock cycles</td><td>4.0 serial clock cycles</td></tr>
<tr><td>0101B</td><td>4.5 serial clock cycles</td><td>5.0 serial clock cycles</td></tr>
<tr><td>0110B</td><td>6.5 serial clock cycles</td><td>7.0 serial clock cycles</td></tr>
<tr><td>0111B</td><td>8.5 serial clock cycles</td><td>9.0 serial clock cycles</td></tr>
<tr><td>1000B</td><td>9.5 serial clock cycles</td><td>10.0 serial clock cycles</td></tr>
<tr><td>1001B</td><td>10.5 serial clock cycles</td><td>11.0 serial clock cycles</td></tr>
<tr><td>1010B</td><td>11.5 serial clock cycles</td><td>12.0 serial clock cycles</td></tr>
<tr><td>1011B</td><td>12.5 serial clock cycles</td><td>13.0 serial clock cycles</td></tr>
<tr><td>1100B</td><td>14.5 serial clock cycles</td><td>15.0 serial clock cycles</td></tr>
<tr><td>1101B</td><td>16.5 serial clock cycles</td><td>17.0 serial clock cycles</td></tr>
<tr><td>1110B</td><td>18.5 serial clock cycles</td><td>19.0 serial clock cycles</td></tr>
<tr><td>1111B</td><td>20.5 serial clock cycles</td><td>21.0 serial clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
</table>

Remark: n = 0, 1; x = 0, 1

(5/5)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7 to 4</td><td>CSIHnINx[3:0]</td><td>Selects the inter-data time for chip select x in transmission clock cycles.
<table>
<tr><th>CSIHnINx [3:0]</th><th>Inter-Data Time when CSIHnCTL1.CSIHnSIT = 0</th><th>Inter-Data Time when CSIHnCTL1.CSIHnSIT = 1</th></tr>
<tr><td>0000B</td><td>0.0 serial clock cycles</td><td>0.5 serial clock cycles</td></tr>
<tr><td>0001B</td><td>0.5 serial clock cycles</td><td>1.0 serial clock cycles</td></tr>
<tr><td>0010B</td><td>1.0 serial clock cycles</td><td>1.5 serial clock cycles</td></tr>
<tr><td>0011B</td><td>2.0 serial clock cycles</td><td>2.5 serial clock cycles</td></tr>
<tr><td>0100B</td><td>3.0 serial clock cycles</td><td>3.5 serial clock cycles</td></tr>
<tr><td>0101B</td><td>4.0 serial clock cycles</td><td>4.5 serial clock cycles</td></tr>
<tr><td>0110B</td><td>6.0 serial clock cycles</td><td>6.5 serial clock cycles</td></tr>
<tr><td>0111B</td><td>8.0 serial clock cycles</td><td>8.5 serial clock cycles</td></tr>
<tr><td>1000B</td><td>9.0 serial clock cycles</td><td>9.5 serial clock cycles</td></tr>
<tr><td>1001B</td><td>10.0 serial clock cycles</td><td>10.5 serial clock cycles</td></tr>
<tr><td>1010B</td><td>11.0 serial clock cycles</td><td>11.5 serial clock cycles</td></tr>
<tr><td>1011B</td><td>12.0 serial clock cycles</td><td>12.5 serial clock cycles</td></tr>
<tr><td>1100B</td><td>14.0 serial clock cycles</td><td>14.5 serial clock cycles</td></tr>
<tr><td>1101B</td><td>16.0 serial clock cycles</td><td>16.5 serial clock cycles</td></tr>
<tr><td>1110B</td><td>18.0 serial clock cycles</td><td>18.5 serial clock cycles</td></tr>
<tr><td>1111B</td><td>20.0 serial clock cycles</td><td>20.5 serial clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
<tr><td>3 to 0</td><td>CSIHnSPx [3:0]</td><td>Selects the setup time for chip select x in transmission clock cycles.
<table>
<tr><th>CSIHnSPx[3:0]</th><th>Setup Delay</th></tr>
<tr><td>0000B</td><td>0.5 serial clock cycles</td></tr>
<tr><td>0001B</td><td>1.0 serial clock cycles</td></tr>
<tr><td>0010B</td><td>1.5 serial clock cycles</td></tr>
<tr><td>0011B</td><td>2.5 serial clock cycles</td></tr>
<tr><td>0100B</td><td>3.5 serial clock cycles</td></tr>
<tr><td>0101B</td><td>4.5 serial clock cycles</td></tr>
<tr><td>0110B</td><td>6.5 serial clock cycles</td></tr>
<tr><td>0111B</td><td>8.5 serial clock cycles</td></tr>
<tr><td>1000B</td><td>9.5 serial clock cycles</td></tr>
<tr><td>1001B</td><td>10.5 serial clock cycles</td></tr>
<tr><td>1010B</td><td>11.5 serial clock cycles</td></tr>
<tr><td>1011B</td><td>12.5 serial clock cycles</td></tr>
<tr><td>1100B</td><td>14.5 serial clock cycles</td></tr>
<tr><td>1101B</td><td>16.5 serial clock cycles</td></tr>
<tr><td>1110B</td><td>18.5 serial clock cycles</td></tr>
<tr><td>1111B</td><td>20.5 serial clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
</table>

Remark: n = 0, 1; x = 0, 1

(11) CSIH transmit data register 0 for word access (CSIHnTX0W)

This register stores the 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.

Cautions 1. Reading this register is prohibited during communication in FIFO mode.

2. Reading from and writing to this register are prohibited in FIFO mode when CSIHnCTL0.CSIHnPWR = 0.

3. Writing to this register is prohibited in direct access mode when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0.

(1/2)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17–16	15–0
CSIHnTX0W	CSIHnCIRE	CSIHnEOJ	CSIHnEDL	0	0	0	0	0	1	1	1	1	1	1	CSIHnCS[1:0]	CSIHnTX[15:0]
R/W	R/W	R/W	R/W	0	0	0	0	0	1	1	1	1	1	1	R/W	R/W

Address: 4000 0188H +100H × n

Initial Value: Undefined

Bit Position	Bit Name	Function
31	CSIHnCIRE	Enables the communication interrupt request CSIHnTIC in dual or transmit-only buffer mode or the job completion interrupt CSIHnTJC request in FIFO mode. 0: No interrupt requested 1: Interrupt requested. Generates interrupt CSIHnTIC or CSIHnTJC after transmission. For details, see 23.4.12(1), CSIHnTIC (communication interrupt) and 23.4.12(4), CSIHnTIJC 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: No end-of-job data 1: End-of-job data Caution: This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). For use in slave mode, clear this bit.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
29	CSIHnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Extended data length activated The associated data is transmitted as of 16 bits. The inter-data delay time and idle time are not inserted after data transmission. When CSIHnCTL1.CSIHnEDLE = 1 and CSIHnTX0W.CSIHnEDL = 1, the same CS must also be selected for the second data. If the CS for the second data is changed, the correct operation is not guaranteed. This bit can only be used when CSIHnCTL1.CSIHnEDLE = 1.
28 to 24	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
23 to 18	—	Reserved. When writing to these bits, write 1. The value read is undefined.
17,16	CSIHnCS[1:0]	Activates one or several chip select signals. 0: Chip select x is activated for the associated transmission 1: Chip select x is deactivated for the associated transmission Setting CSIHnTX0W.CSIHnCS[1:0] to 3H 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 selects must be set to precisely the same configuration. For use in slave mode, set the CSIHnCS[1:0] bits to 2H.
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

Remark: n = 0, 1; x = 0, 1

(12) CSIH transmit data register 0 for half word access (CSIHnTX0H)

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

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

Cautions 1. Reading this register is prohibited during communication in FIFO mode.

2. Reading from and writing to this register are prohibited in FIFO mode when CSIHnCTL0.CSIHnPWR = 0.

3. Writing to this register is prohibited in direct access mode when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 to 0
CSIHnTX0H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHnTX[15:0]
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W

Address
4000 018CH
+100H × n
Initial Value
Undefined

Bit Position	Bit Name	Function
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

Remark: n = 0, 1

(13) CSIH receive data register 0 for word access (CSIHnRX0W)

This register stores the received data.

- Access This register is read-only, in 32-bit units.

Cautions 1. This register can be read when CSIHnCTL0.CSIHnPWR = 1, and can be written to when CSIHnCTL0.CSIHnPWR = 0.

2. This register is Initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

3. Reading from and writing to this register are prohibited in FIFO mode when CSIHnCTL0.CSIHnPWR = 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 to 0
CSIHnRX0W	0	0	0	0	0	0	CSIHnRPE	CSIHnTDCE	1	1	1	1	1	1	CSIHnCSx		CSIHnRX[15:0]
R/W	0	0	0	0	0	0	R	R	1	1	1	1	1	1	R		R

Address: 4000 0190H +100H × n
Initial Value: Undefined

Bit Position	Bit Name	Function
31 to 26	—	Reserved. These bits are read as 0.
25	CSIHnRPE	Indicates whether a reception data parity error was detected. 0: No parity error has been detected in the received data. 1: A parity error has been detected in the received data.
24	CSIHnTDCE	Indicates whether a transmission data consistency error was detected. A data consistency check is performed on transmission data. The result of the check performed on the data transmitted at the same time as saving received data to CSIHnRX0W.CSIHnRX[15:0] is applied to this bit. 0: No data consistency error has been detected in the transmitted data. 1: A data consistency error has been detected in the transmitted data.
23 to 18	—	Reserved. When read, the value read is undefined.
17,16	CSIHnCSx	Indicate whether the chip select signal is active. When in master mode, the status of the chip select signal upon receiving the data saved to CSIHnRX0W.CSIHnRX[15:0] (that is, which CS to perform communication for) is stored in these bits. 0: Chip select signal x was active upon receiving the data. 1: Chip select signal x was inactive upon receiving the data. When in slave mode, because it is necessary to specify CS0 (CSIHnTX0W.CSIHnCS[1:0] = 02H) as the communication partner when transmission is enabled, 02H is saved in transmission mode or transmission/reception mode. The value is always 00H when in reception mode.
15 to 0	CSIHnRX [15:0]	Store the reception data. Read the value of the CSIHnRX0W or CSIHnRX0H register at least one serial clock cycle before the interrupt is generated.

Remark: n = 0, 1; x = 0, 1

(14) CSIH receive data register 0 for half word access (CSIHnRX0H)

This register stores the reception data. This register is the same as bits 15 to 0 of register CSIHnRX0W.

- Access This register is only readable in 32-bit units.

Cautions 1. This register can be read when CSIHnCTL0.CSIHnPWR = 1, and can be written to when CSIHnCTL0.CSIHnPWR = 0.

2. This register is initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or 1 to 0.

3. Reading from and writing to this register are prohibited in FIFO mode when CSIHnCTL0.CSIHnPWR = 0.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 to 0
CSIHnRX0H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHnRX[15:0]
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R

Address 4000 0194H +100H × n

Initial Value Undefined

Bit Position	Bit Name	Function
31 to 16	—	Reserved. These bits are read as 0.
15 to 0	CSIHnRX[15:0]	Stores the received data.

Remark: n = 0, 1

23.4 Functional Description

The clocked serial interface H uses three signals for communications:

- Serial clock CSIHnTSCK (output in master mode or input in slave mode)
- Data output signal CSIHnTSO
- Data input signal CSIHnTSI

Additional signals are available for external control.

- CSIHnTCSS1, 0: Chip select signals

Data transmission is bit-wise and serial and synchronous to the serial clock.

The most important registers for setting up the CSIH are:

Register	Function
CSIHnCTL0	Enables or disables the operation clock (PCLK) and enables or disables data transmission and 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, etc.
CSIHnCTL2	Selects master/slave mode and – effective in master mode – the baud rate of the Internal baud rate generator (BRG)
CSIHnMCTL0	Selects memory mode and specifies timeout
CSIHnMCTL1	Controls the memory in FIFO mode
CSIHnMCTL2	Controls the memory in dual buffer mode or transmit-only buffer mode
CSIHnCFG0,1	Registers to configure the communication protocol for each chip select signal

23.4.1 Operating Modes (Master/Slave)

Master/slave selection is performed by using the CSIHnCTL2.CSIHnPRS[2:0] bits, and, when the master is selected, the source clock of the transmission clock must also be selected.

(1) Master mode

In master mode, the serial clock is generated by the internal baud rate generator (BRG) and provided to the slave(s) by signal CSIHnTSCK.

Master mode is enabled by setting CSIHnCTL2.CSIHnPRS[2:0] to anything but 111B. In master mode, the BRG frequency can be specified by specifying values for the CSIHnCTL2.CSIHnPRS[2:0] and CSIHnCTL2.CSIHnBRS[11:0] bits in combination.

- 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 23.4.3, Chip Selection (CS) Features.

- Clock defaults

  The default level of CSIHnTSCK depends on the clock phase selection bit: It is high when CSIHnCFGx.CSIHnCKPx = 0, and is low when CSIHnCFGx.CSIHnCKPx = 1.

  The example below shows the communication in master mode for 8 data bits, CSIHnCTL1.CSIHnCKR = 0, CSIHnCFGx.CSIHnDAPx = 0, and MSB first:

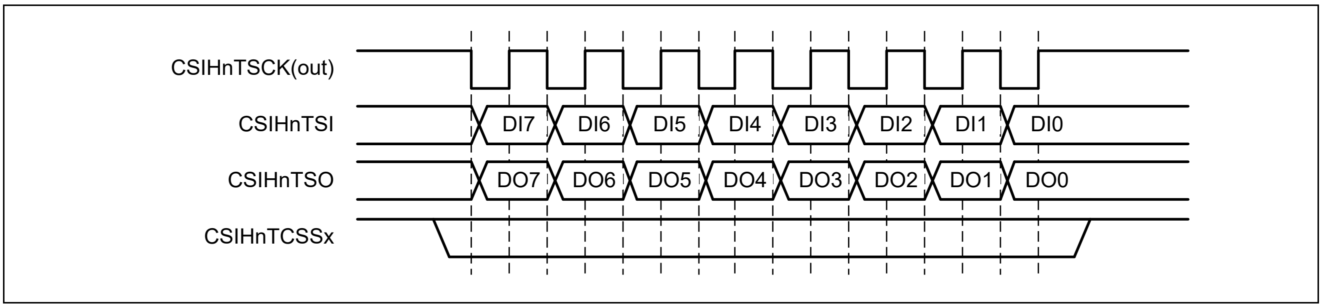


Figure 23.2 Transmission/Reception in Master Mode

(2) Slave mode

In slave mode, another device is the communication master. The serial clock is supplied through the CSIHnTSCK signal. When the serial clock signal is detected, a transmission or reception operation immediately starts.

Slave mode is selected by setting CSIHnCTL2.CSIHnPRS[2:0] to 111B.

In slave mode, the transmission protocol settings of the CSIHnCFG0 register are valid (the settings of the CSIHnCFG1 and CSIHnCFG2 registers are invalid.):

- CSIHnPS0[1:0]: Parity usage
- CSIHnDLS0[3:0]: Data length selection
- CSIHnCFG0.CSIHnDIR0: Data direction
- CSIHnCFG0.CSIHnCKP0, CSIHnCFG0.CSIHnDAP0: Clock phase and data phase

Remark: When using slave mode, the baud rate generator (BRG) can be disabled by clearing the CSIHnCTL2.CSIHnBRS[11:0] bits, reducing power consumption.
However, when using the timeout error function, the BRG must be set to a value other than 0.

The example below shows the communication in the save mode for eight data bits when CSIHnCTL1.CSIHnCKR = 0, CSIHnCFGx.CSIHnDAPx = 0, and the MSB is first.

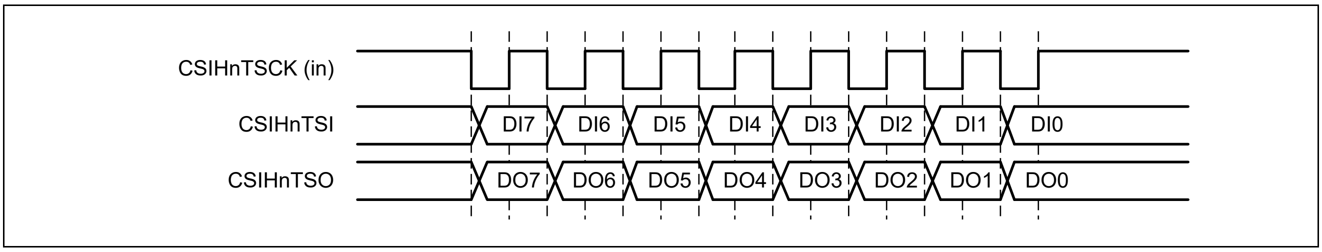


Figure 23.3 Transmission/Reception in Slave Mode

23.4.2 Master/Slave Connections

(1) One master and one slave

The following figure illustrates the connections between one master and one slave.

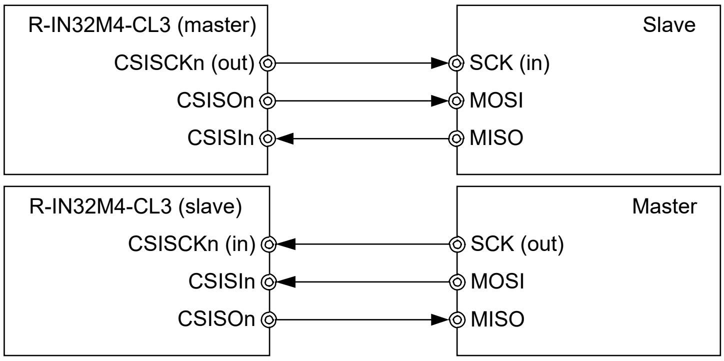


Figure 23.4 Direct Master/Slave Connection

(2) One master and two slaves

The following figure illustrates the connections between an R-IN32M4-CL3 as a master and two slaves. In this example, an R-IN32M4-CL3 can be configured to supply one chip select (CS) signal to each slave. This signal is connected to the slave select input SSI of the slave.

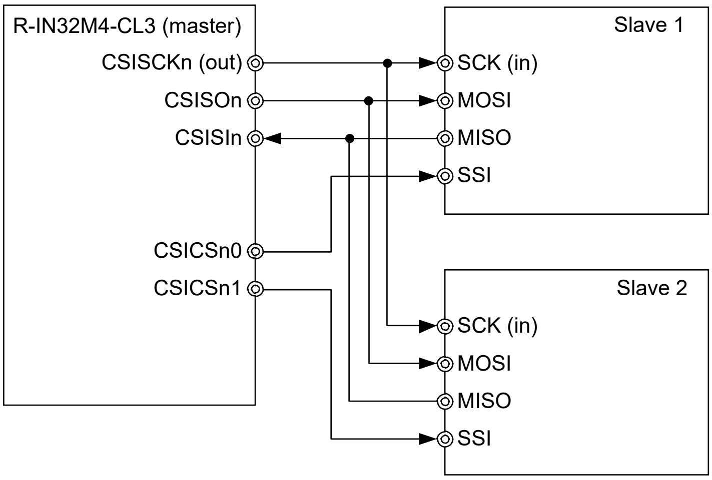


Figure 23.5 Connection between One Master and Two Slaves

The default chip select level is active low. In other words, when the slave select input signal (SSI) of a slave is at the low level, that slave is selected as a CSIH slave (and enabled). However, to use a chip select signal (CS) for another device, programming that sets the chip select signal output level to active high is possible.

If a slave is not selected, it will neither receive nor transmit data. In addition, its output MISO is set to input mode in order to avoid interference with the output of another slave that was selected.

(3) CSISOn output control

The CSIH can output CSISOn when all of the following conditions are satisfied:

- The CSIH is enabled (CSIHnCTL0.CSIHnPWR = 1).
- The CSIH is operated in transmit-only or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1).

By using this function, signal congestions on the external CSISOn signal line can be avoided.

23.4.3 Chip Selection (CS) Features

The chip select signals CSIHnTCSSx can be used by the master to select one or several slaves for communication.

(1) Configuration registers

The parameters for each chip select signal CSIHnTCSSx 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.
- Transfer direction: MSB or LSB first.
- Parity usage: Odd, even, 0 parity, or none.
- Clock phase and data phase.

Additional parameters for each chip select signal only available in master mode are:

- Prescaler selection of the baud rate generator separately for each chip select signal
- Chip select priority: Separates between "dominant" and "recessive" chip select signals. The priority applies if two or more chip selects with different configurations are simultaneously activated for message broadcasting. In this case, the configuration that is set as dominant is used.
  The principle is also called "Recessive Configuration for Broadcasting" (RCB).

Caution: Do not specify several chip select signals as dominant with different configurations unless all dominant chip select signals have the same configuration.

- Chip select timing:
  - Setup time T_{setup}: The time from setting the CS signal active to starting data output.
  - Inter-data time T_{inter}: The time between data while the same CS signal is active.
  - Hold time T_{hold}: Hold time of CS active level before changing the CS.
  - Idle time T_{idle}: Inactive time after terminating a CS signal or after every data transfer to the same CSx.

The figure below shows the timing of the chip select (CSx) signal setup time, inter-data time, hold time, and idle time. No matter which CSIHnCFGx.CSIHnIDLx bit is set (to 1), idle time is added to all CS segments.

Figure 23.6 shows an example in which the default active low setting is specified for the CS0 and CS1 signals (CSIHnCTL1.CSIHnCSL0 = 0, CSIHnCTL1.CSIHnCSL1 = 0). The active level can be separately specified for each CS.

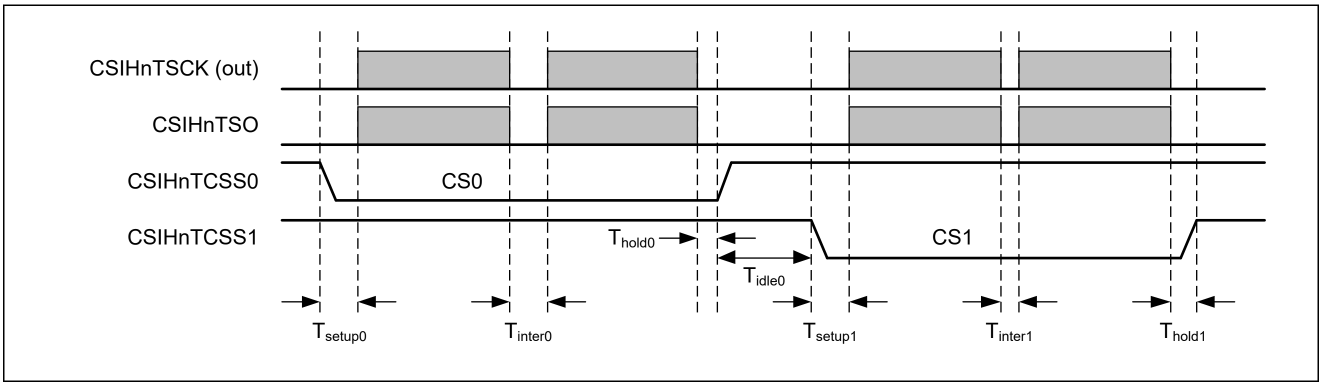


Figure 23.6 Chip Select Timings

Note that each CS can have a different value for setup time, inter-data time, hold time, and idle time.

A particular chip select signal is activated by setting the appropriate bit in the transmission data register CSIHnTX0W.CSIHnCS[1:0].

CSIHnRX0W.CSIHnCS[1:0] of the reception data register indicate the chip select signal associated with the data for transmission.

(2) CS example

The following figure shows an example of two consecutive transmissions.

The first communication uses CS0 to address one single slave. The second (for which communication is performed using the dominant-side communication settings) 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".

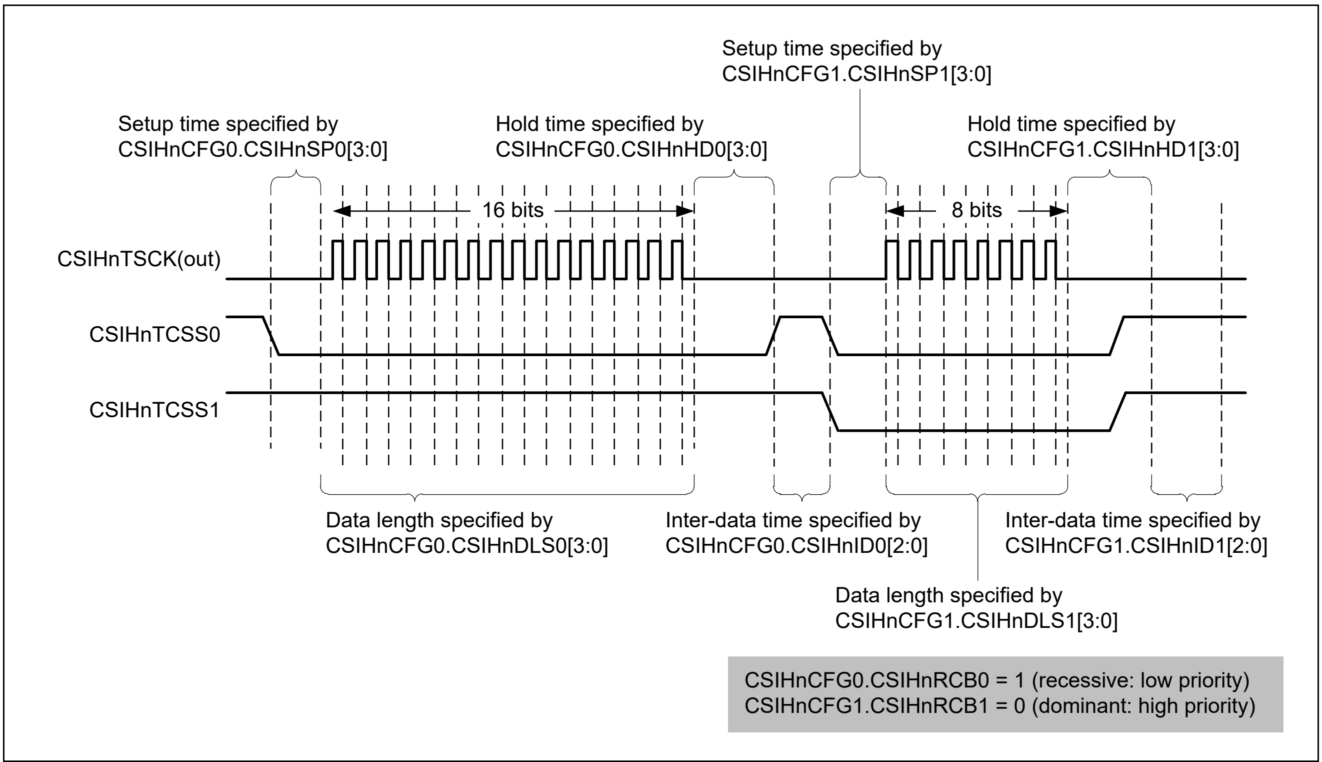


Figure 23.7 Chip Select and RCB Example

23.4.4 Chip Select Timing Details

(1) Changing the clock phase

The serial clock level is specified for each chip select according to CSIHnCFGx.CSIHnCKPx. The chip select or serial clock level is switched during the idle time. The minimum idle time is 1/2 of a serial clock (CSIHnTSCK) cycle (0.5 SCK).

If the idle time is set to 0.5 transmission clock cycles (in CSIHnCFGx.CSIHnIDx[2:0]) and two consecutive data are sent with different CSIHnCFGx.CSIHnCKPx configuration, the idle time is automatically extended to one cycle of CSIHnTSCK.

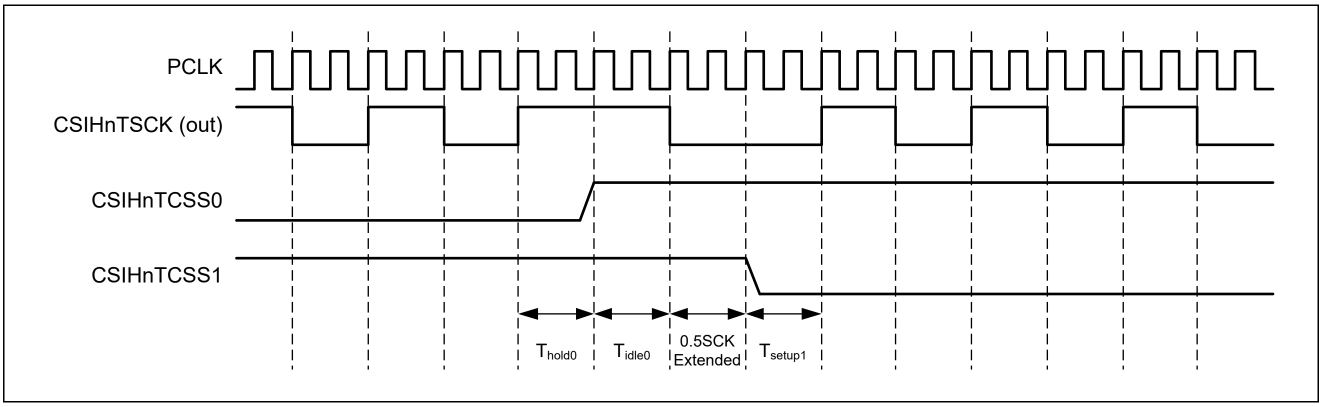


Figure 23.8 Clock Phase Timing (in the case of PCLK/4, T_{hold0} = T_{setup1} = 0.5 SCK, T_{idle0} = 0.5 SCK, CKP0 = 0 (CSIHnTCSS0) → CKP1 = 1 (CSIHnTCSS1))

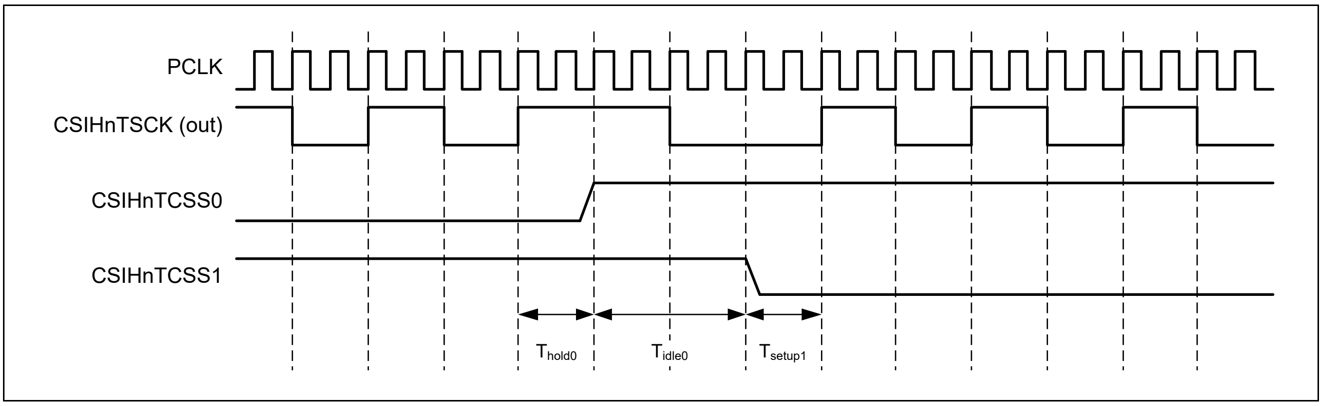


Figure 23.9 Clock Phase Timing (in the case of PCLK/4, T_{hold0} = T_{setup1} = 0.5 SCK, T_{idle0} = 1.0 SCK, CKP0 = 0 (CSIHnTCSS0) → CKP1 = 1 (CSIHnTCSS1))

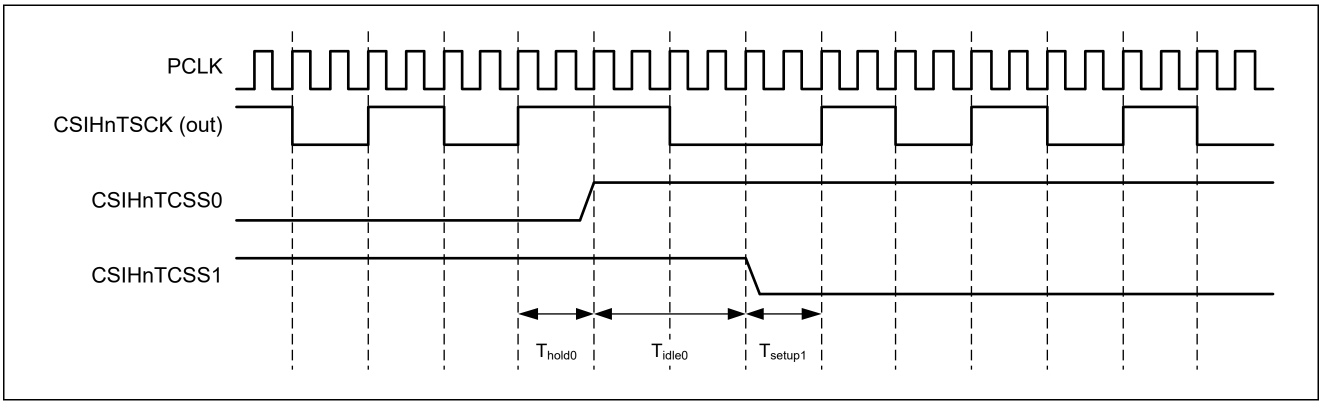


Figure 23.10 Clock Phase Timing (in the case of PCLK/4, T_{hold0} = T_{setup1} = 0.5 SCK, T_{idle0} = 0.5 SCK, CKP0 = 0 (CSIHnTCSS0) → CKP1 = 0 (CSIHnTCSS1))

(2) Changing the data phase

The bit CSIHnCFGx.CSIHnDAPx defines the phase of the data bits relative to the clock.

If CSIHnCFGx.CSIHnDAPx = 0, the transmission clock CSIHnTSCK holds its level after the last bit of a data is transferred.

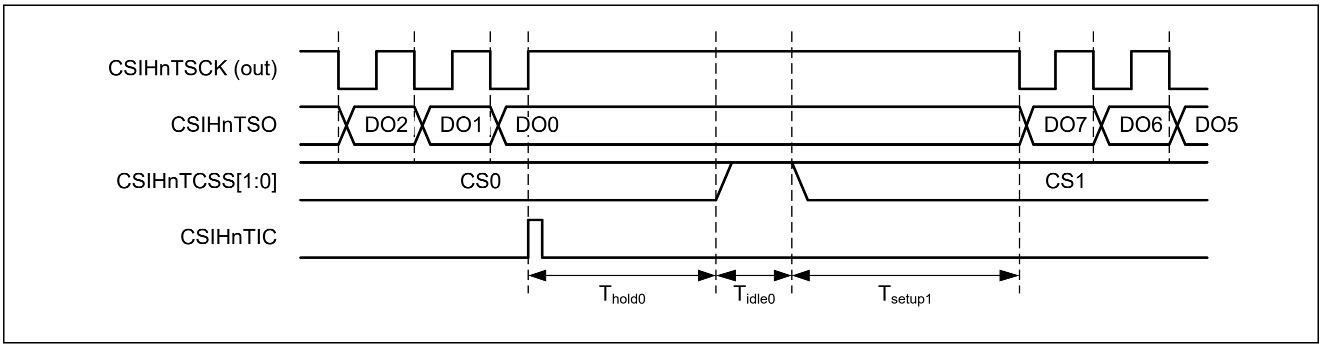


Figure 23.11 Data Phase Timing with CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0 and CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0

If the default clock phase changes between two consecutive chip selects, the transmission clock CSIHnTSCK changes its level after the last bit of the first data is transferred:

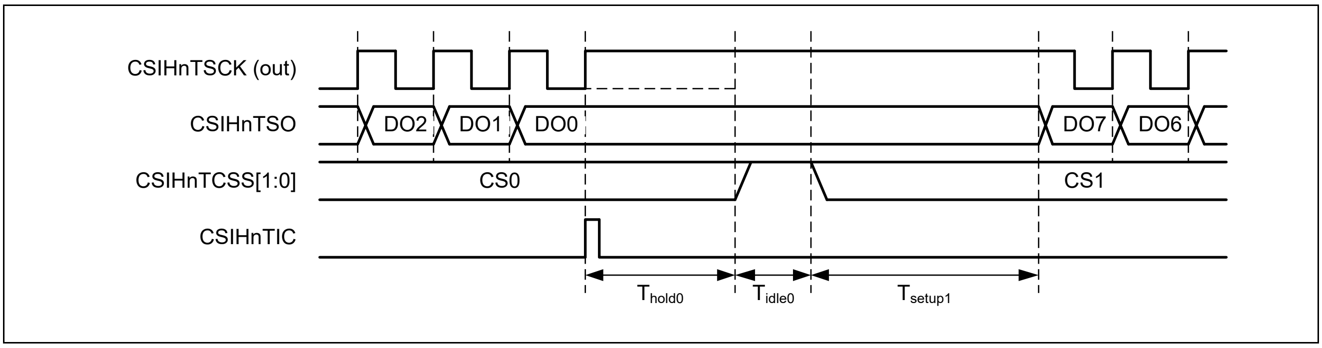


Figure 23.12 Data Phase Timing with CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 1 and CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 1

Note that the minimum idle time of one CSIHnTSCK cycle is automatically inserted, if CSIHnCFGx.CSIHnIDx[2:0] = 0 (Tidle1 = 0.5 CSIHnTSCK cycles).

23.4.5 Job Concept

In terms of CSIH, a job consists of a number of data that are transferred.

- Job mode enabled
  Job mode is enabled and disabled by CSIHnCTL1.CSIHnJE, while the CSIH is disabled by CSIHnCTL0.CSIHnPWR = 0.

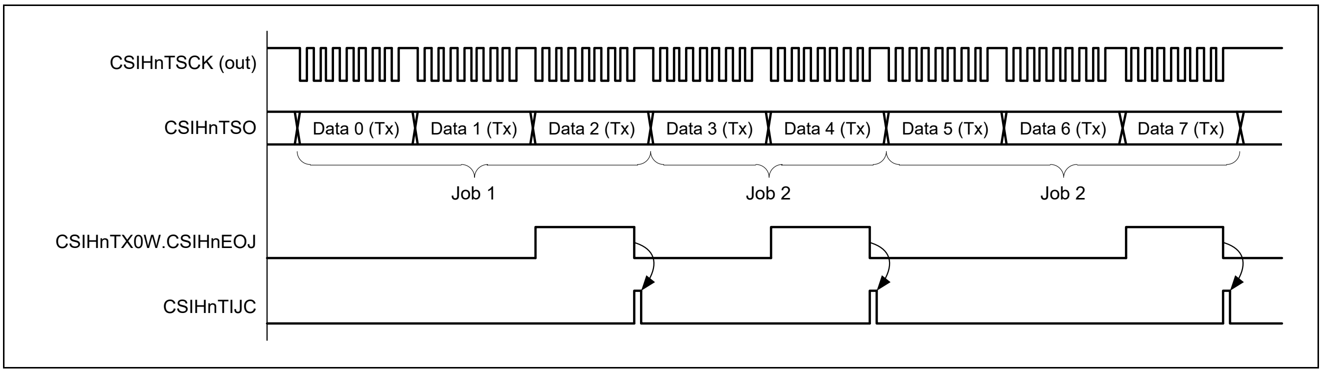


Figure 23.13 Job Examples

A job ends when a data with the end-of-job bit set, i.e. 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 a data is sent, for which the CSIHnEOJ bit was set. After this data is sent, the communication is stopped and the end-of job interrupt CSIHnTIJC is generated.

23.4.6 Serial Clock Selection

In master mode, the transmission baud rate is selectable using

- CSIHnCTL2.CSIHnPRS[2:0]
- CSIHnCTL2.CSIHnBRS[11:0]
- CSIHnCFGx.CSIHnPSCLx[1:0]

While the settings in the CSIHnCTL2 register determine the transmission base clock CSIHnBPCLK, a chip select dedicated prescaler, controlled by CSIHnCFGx.CSIHnPSCLx[1:0], allows generating different baud rates for different chip selects.

The following figure shows a block diagram of the baud rate generator.

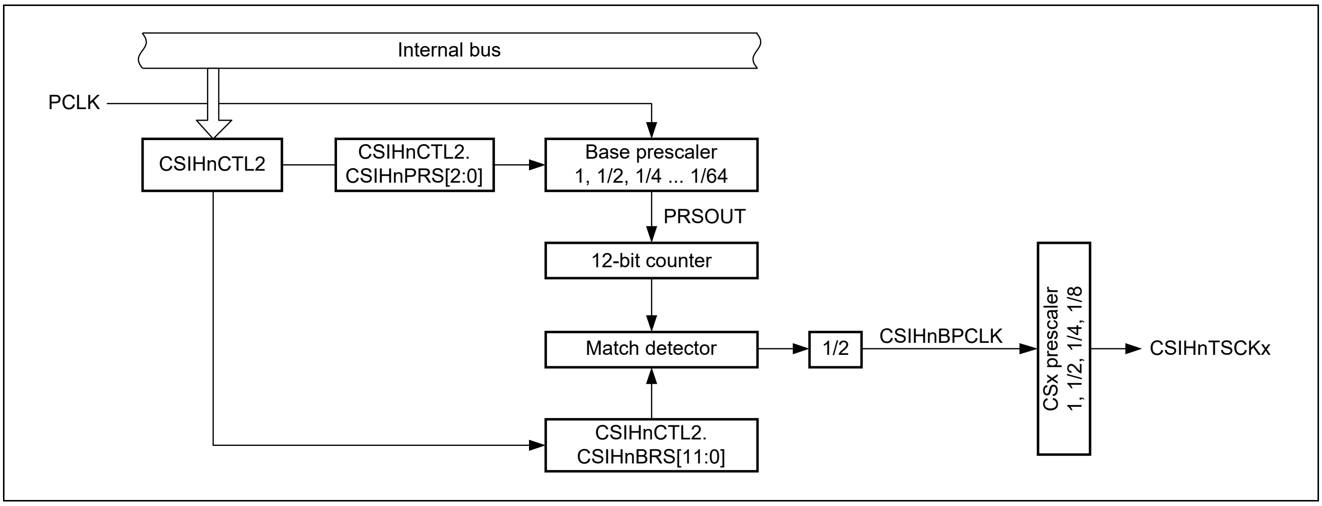


Figure 23.14 Baud Rate Generator Block Diagram

Clearing CSIHnCTL2.CSIHnBRS[11:0] disables the baud rate generator, and thus all CSIHnTSCKx are stopped.

The baud rate is calculated from the following formula:

$$CSIHnTSCKx = PCLK / (2^m \times k \times 2 \times 2^j)$$

where

m = CSIHnCTL2.CSIHnPRS[2:0] = 0–6
k = CSIHnCTL2.CSIHnBRS[11:0] = 1–4095
j = CSIHnCFGx.CSIHnPSCLx[1:0]= 0–3

23.4.6.1 Baud Rate Limits

When setting the baud rate, please note:

- Maximum acceptable baud rate in master mode is PCLK/4.
- Maximum acceptable baud rate in slave mode is PCLK/6 (confirm that the baud rate of the external master is within this range).
- Minimum baud rate in both modes is PCLK/524160.

Caution: There might be restrictions on the maximum baud rate that can actually be used depending on the product. Specify the baud rate so as not to exceed the maximum rate of the product you are using.

[Example]

If PCLK = 100 MHz, the maximum baud rate is

• 25 Mbps (PCLK/4) in master mode

• 16.66 Mbps (PCLK/6) in slave mode

The minimum baud rate is 190.78 bps (PCLK/524160).

23.4.7 CSIH Buffer Memory

The CSIH has a configurable RAM that can be used for buffered I/O. The size is 128 words. One word consists of 32 bits of data.

The following configurations are available:

Mode	CSIHnCTL0.CSIHnMBS	CSIHnMCTL0.CSIHnMMS[1:0]
FIFO mode	0	00B
Dual buffer mode		01B
Transmit-only buffer mode		10B
Direct access mode	1	X

(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, if the FIFO is not full.

Data to be transmitted is stored to the FIFO memory. Transmission and reception occur simultaneously – one bit is sent, one bit is received. That means, received data overwrites the transmitted data in the FIFO.

CSIH automatically updates the respective FIFO memory pointers when a data package is processed, sent or received:

Pointer Description	Control Bits	Range
Number of words that have not been transmitted	CSIHnSTR0.CSIHnSPF[7:0]	0–128
Number of words stored in the reception FIFO buffer	CSIHnSTR0.CSIHnSRP[7:0]	0–128
Address of data to be sent	CSIHnMRWP0.CSIHnTRWA[6:0]	0000H–01FCH
Address of received data	CSIHnMRWP0.CSIHnRRA[6:0]	0000H–01FCH

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. This resets all FIFO pointers and flags.

(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:

Pointer Description	Pointers Note	Range
Destination address for data written to or read from CSIHnTX0W/H	CSIHnMRWP0.CSIHnTRWA[6:0]	0000H–00FCH
Address of data read from CSIHnRX0W/H	CSIHnMRWP0.CSIHnRRA[6:0]	0000H–00FCH
Transmission pointer	CSIHnMCTL2.CSIHnSOP[6:0]	0000H–00FCH

Note: Each pointer is automatically incremented after each read or write.

(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 pointers indicate:

Pointer Description	Pointers Note	Range
Destination address for data written to or read from CSIHnTX0W/H	CSIHnMRWP0.CSIHnTRWA[6:0]	0000H–01FCH
Transmission pointer	CSIHnMCTL2.CSIHnSOP[6:0]	0000H–01FCH

Note: Each pointer is automatically incremented after each read or write.

(4) Direct access mode

In direct access mode, the CSIH memory is completely bypassed:

- Transmission data provided by the CPU to the transmission data register CSIHnTX0W or CSIHnTX0H is directly copied to the shift register.
- Reception data is directly copied from the shift register to the reception data register CSIHnRX0W or CSIHnRX0H.

23.4.8 Data Transfer Modes

(1) Transmit-only mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 0 places 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.

(2) Receive-only mode

Setting CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 1 places the CSIH in receive-only mode.

In master mode, the start of reception depends on the memory mode:

- In case of FIFO, transmit-only buffer or direct access mode, reception starts when dummy data is written in the CSIHnTX0W or CSIHnTX0H register.
- In the dual buffer or transmit-only buffer mode, transmission starts when the CSIHnMCTL2.CSIHnBTST bit is set.

In slave mode, reception starts as soon as the transmission clock CSIHnTSCK from the master is received. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

(3) Transmit & receive mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 1 places the CSIH in transmit/receive mode.

The start of the 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.

(4) Summary

The following table provides a summary. It shows how the data transfer is started in the various memory modes, operating modes, and transfer modes.

Memory Modes	Transfer Modes	Operating Modes	Condition for Starting a Data Transfer
FIFO mode, direct access mode	Transmission mode Transmission/reception mode	Master, Slave	When transmission data is written to the CSIHnTX0W or CSIHnTX0H register
	Reception mode	Master	When dummy data is written to the CSIHnTX0W or CSIHnTX0H register
		Slave	When the serial clock CSIHnTSCK is received from the master
Transmit-only buffer mode, dual buffer mode	Transmission mode Transmission/reception mode Reception mode	Master, Slave	When 1 is written to CSIHnMCTL2.CSIHnBTST

23.4.9 Data Length Selection

(1) Data length between 7 and 16 bits

CSIHnCFGx.CSIHnDLSx[3:0] can be used to select the data packet length for each chip select signal in the range from 7 to 16 bits. The examples below show the communication with MSB first (CSIHnCFGx.CSIHnDIRx = 0).

- Data length = 16 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 0000B):

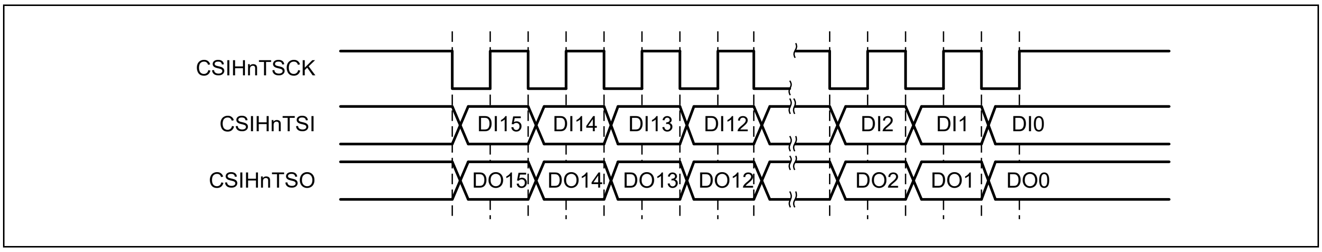


Figure 23.15 16-Bit Data, MSB First

- Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110B):

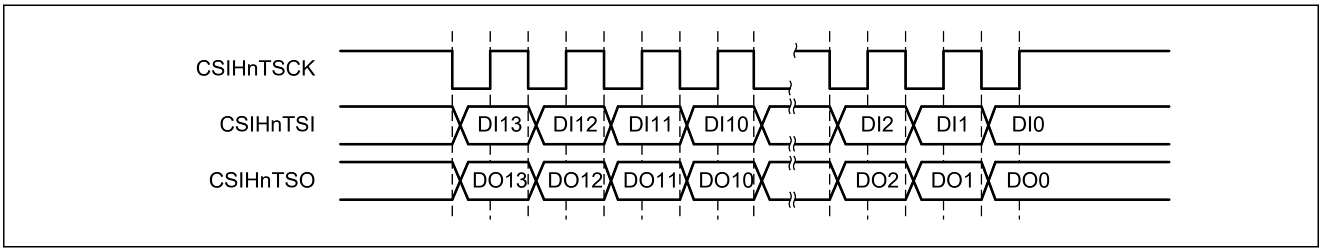


Figure 23.16 14 Bit-Data, MSB First

(2) Data length greater than 16 bits

If the length of 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 CSIHnCTL1.CSIHnEDLE.

The operation and setup procedure of the EDL function are described below:

- Data is divided into 16-bit blocks and a remainder. For example, a 42-bit character string is divided into two 16-bit blocks and a 10-bit remainder.
- For the remainder, the data length is specified by the CSIHnCFGx.CSIHnDLSx[3:0] bits.
- When transmitting 16-bit blocks, set the CSIHnTX0W.CSIHnEDL bit. In this case, the data written to CSIHnTX0W is sent as a 16-bit data length regardless of the CSIHnCFGx.CSIHnDLSx[3:0] bits.
- When the specified length of data (the remainder when CSIHnTX0W.CSIHnEDL = 0) is transmitted, the transfer ends.

[Example] Example of transmitting the 40-bit data 123456789AH to CS0

The 40-bit data is divided into two 16-bit blocks of data and one 8-bit block of data.

- Initialize CSIHnCFGx.CSIHnDLSx[3:0] = 8H.
- To send the string 123456789AH with MSB first, write the following sequence to CSIHnTX0W:
  - 2000 1234H (CSIHnTX0W.CSIHnEDL = 1)
  - 2000 5678H (CSIHnTX0W.CSIHnEDL = 1)
  - 0000 009AH (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

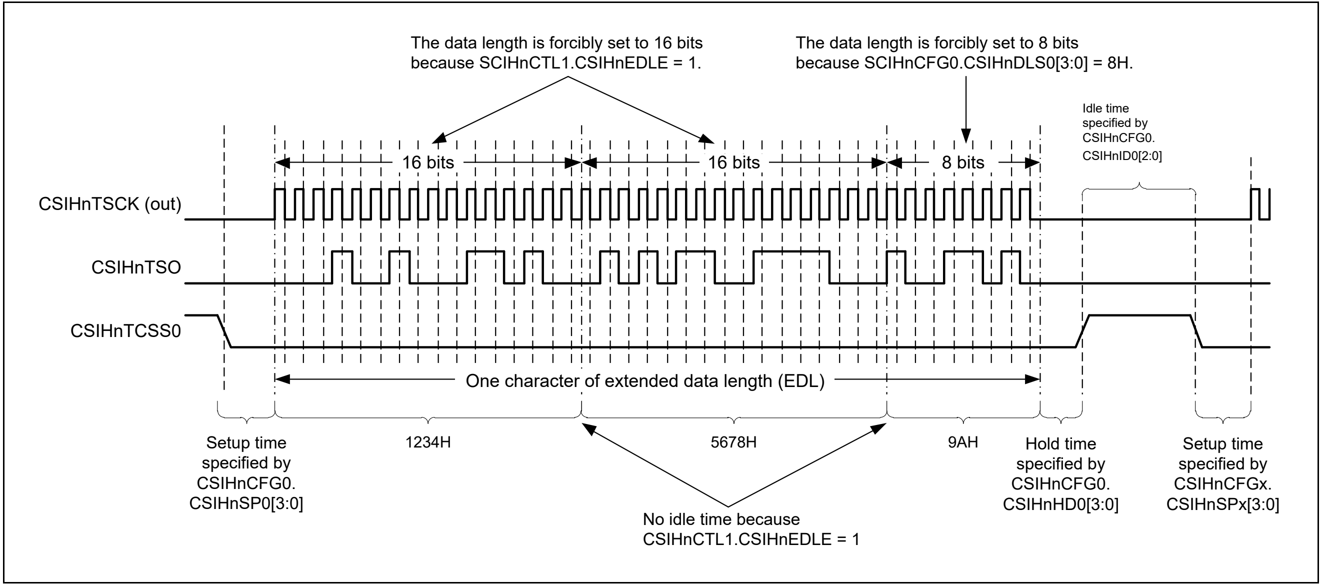


Figure 23.17 EDL Timing Chart

Remarks 1. A data length less than 7 bits can be specified only when the EDL function 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. The following describes an example where the transmitted data is 123456H.

- **CSIHnCFGx.CSIHnDIR is cleared to 0 (MSB first).**
  2000 1234H is written to CSIHnTX0W (CSIHnTX0W.CSIHnEDL = 1).
  0000 0056H is written to CSIHnTX0W (CSIHnTX0W.CSIHnEDL = 0).
- **CSIHnCFGx.CSIHnDIR is set to 1 (LSB first).**
  2000 3456H is written to CSIHnTX0W (CSIHnTX0W.CSIHnEDL = 1).
  0000 0012H is written to CSIHnTX0W (CSIHnTX0W.CSIHnEDL = 0).

5. The EDL function cannot be used in slave mode (CSIHnCTL1.CSIHnPRS[2:0] = 111B) and reception mode (CSIHnCTL0.CSIHnTXE = 0, CSIHnCTL0.CSIHnRXE = 1).

23.4.10 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 the communication for a data length of 8 bit (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B):

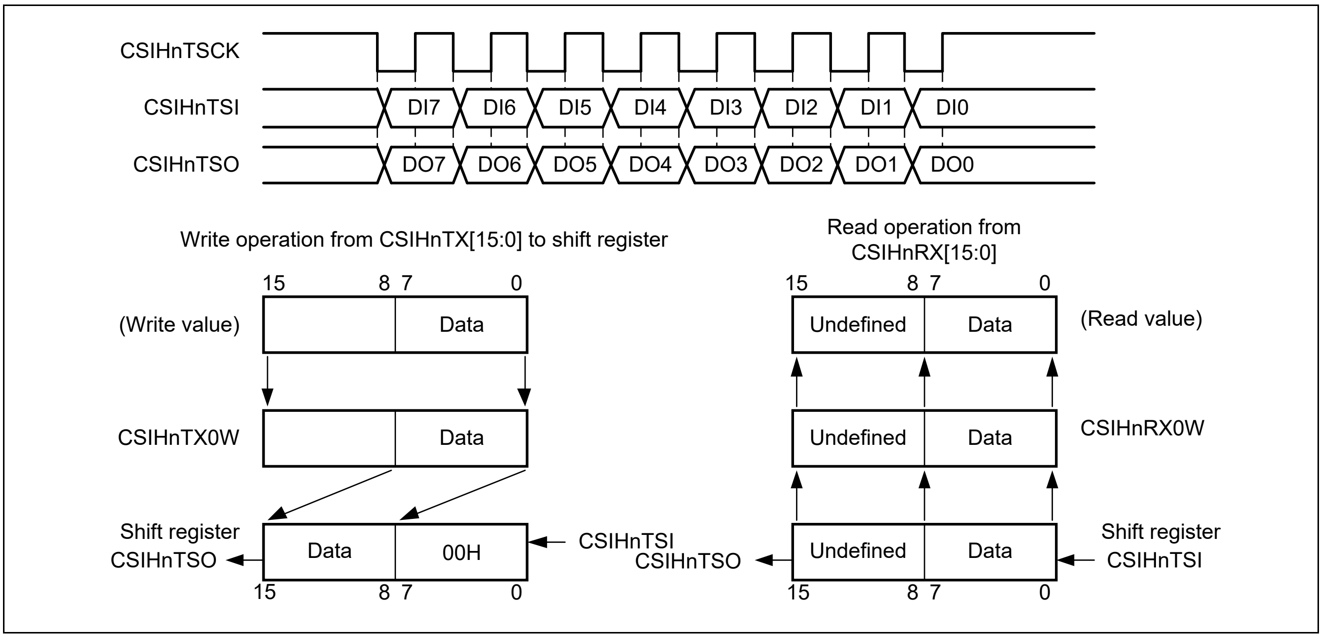


Figure 23.18 Serial Data Direction Select Function ─ MSB First (CSIHnDIR = 0)

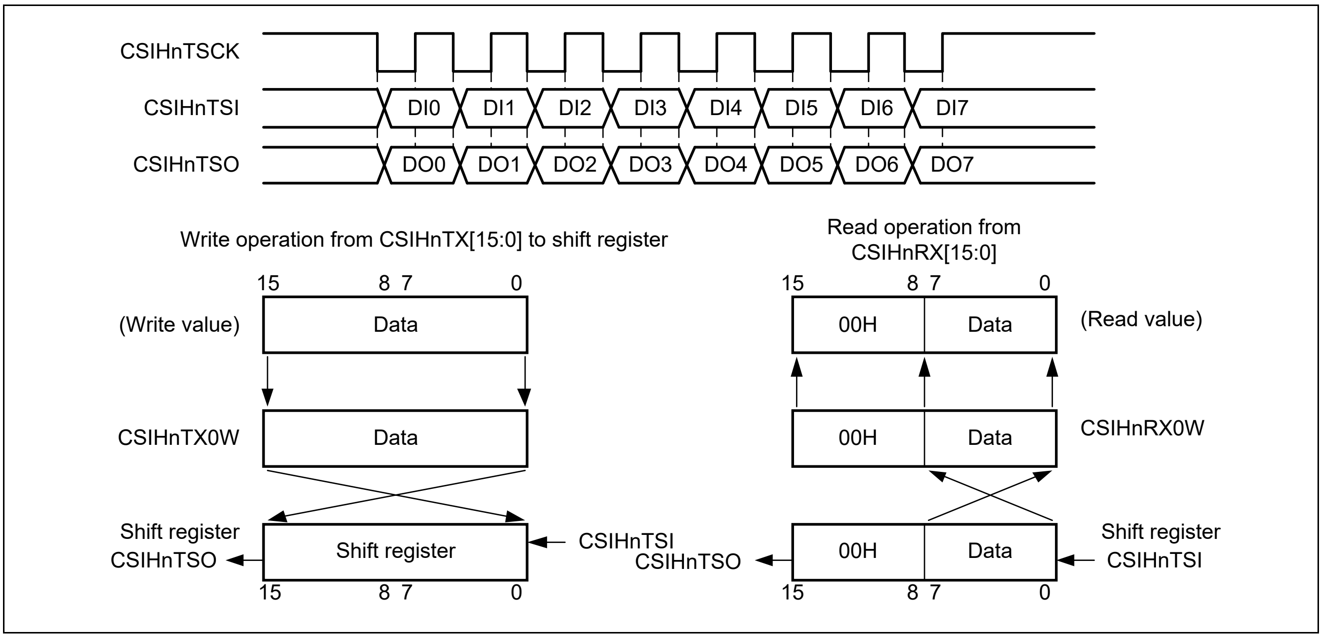


Figure 23.19 Serial Data Direction Select Function ─ LSB First (CSIHnDIR = 1)

23.4.11 Communication in Slave Mode

The following figure illustrates the communication signals and timings in slave mode.

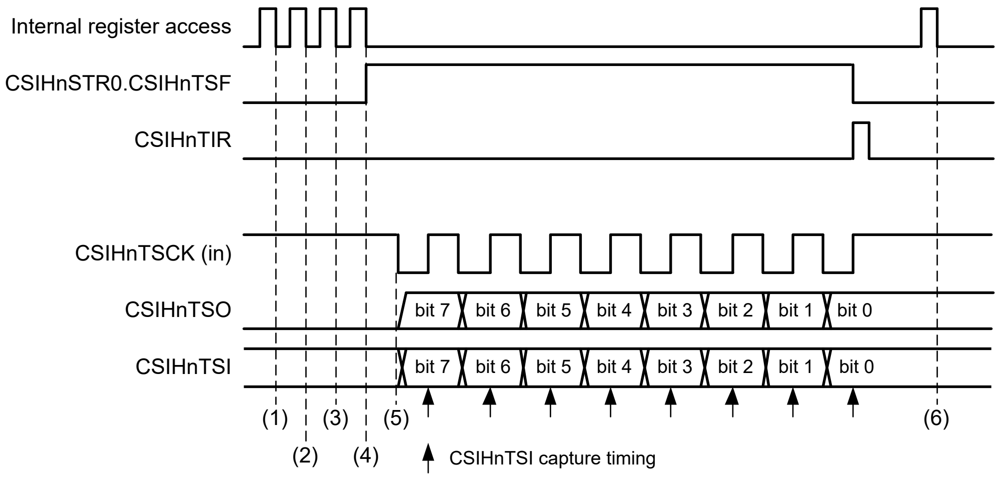


Figure 23.20 Transmission/Reception Timing in Slave Mode

1. CSIH is placed in slave mode by setting CSIHnCTL2.CSIHnPRS[2:0] to 111B.
2. CSIHnCTL1.CSIHnCKR and CSIHnCFG0.CSIHnDAP0 are 0, the data length is 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B), and the data direction is MSB first (CSIHnCFG0.CSIHnDIR0 = 0).
3. CSIH is set to the transmission/reception mode (CSIHnCTL0.CSIHnPWR = 1, CSIHnCTL0.CSIHnTXE = 1, and CSIHnCTL0.CSIHnRXE = 1).
   Start of communication is enabled.
4. If transfer data is written to the transmission data register CSIHnTX0W or CSIHnTX0H, the transfer status flag CSIHnSTR0.CSIHnTSF is automatically set, and the system is ready for reception.
5. If a serial clock is input, transmission data will be transmitted from CSIHnTSO synchronizing with a serial clock, and input to CSIHnTSI is ignored.
6. The CSIHnRX0W or CSIHnRX0H register is read.

Remark: For details about the operating procedure in slave mode for each operation mode, see section 23.5, Operating Procedures.

23.4.12 CSIH Interrupt Requests

CSIH can generate the following interrupt requests:

- CSIHnTIC (communication interrupt)
- CSIHnTIR (reception interrupt)
- CSIHnTIRE (error interrupt)
- CSIHnTIJC (job completion interrupt)

(1) CSIHnTIC (communication interrupt)

The conditions for generating CSIHnTIC differ depending on the memory mode and whether job mode is enabled.

<table>
<tr><th rowspan="2">Memory Modes</th><th colspan="2">Interrupt Source</th></tr>
<tr><th>Job Mode Disabled
CSIHnCTL1.CSIHnJE = 0</th><th>Job Mode Enabled
CSIHnCTL1.CSIHnJE = 1</th></tr>
<tr><td rowspan="2">FIFO mode</td><td colspan="2">CSIHnTIC is generated immediately before the transmission data in the FIFO buffer disappears to inform the application that new data must be added.
CSIHnTIC is generated when the number of transmission data items remaining in the FIFO buffer, CSIHnSTR0.CSIHnSPF[7:0], becomes equal to CSIHnMCTL1.CSIHnFES[6:0].</td></tr>
<tr><td>However, CSIHnTIC is not generated if the job is interrupted [Note].</td><td>—</td></tr>
<tr><td>Transmit-only buffer mode, dual buffer mode</td><td>CSIHnTIC is generated when communication ends (as specified by the CSIHnMCTL2.CSIHnND[7:0] bits).</td><td>CSIHnTIC is generated when data is transmitted while CSIHnTX0W.CSIHnCIRE is "1".
However, when the data and a job interrupt request [Note] are transmitted while CSIHnTX0W.CSIHnCIRE is "1", CSIHnTIJC is generated instead of CSIHnTIC.</td></tr>
<tr><td>Direct access mode</td><td>CSIHnTIC is generated each time a data transfer is performed. However, CSIHnTIC is not generated if the job is interrupted [Note].</td><td>Except when communication is interrupted, CSIHnTIC is generated each time a data transfer is performed.
However, when the data and a job interrupt request are transmitted while CSIHnTX0W.CSIHnCIRE is "1", CSIHnTIJC is generated instead of CSIHnTIC.</td></tr>
</table>

Note: Job abortion condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

(a) CSIHnTIC in direct access mode

The following example shows the CSIHnTIC behavior in direct access mode.

The following example assumes:

- Master mode
- Direct access memory mode
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

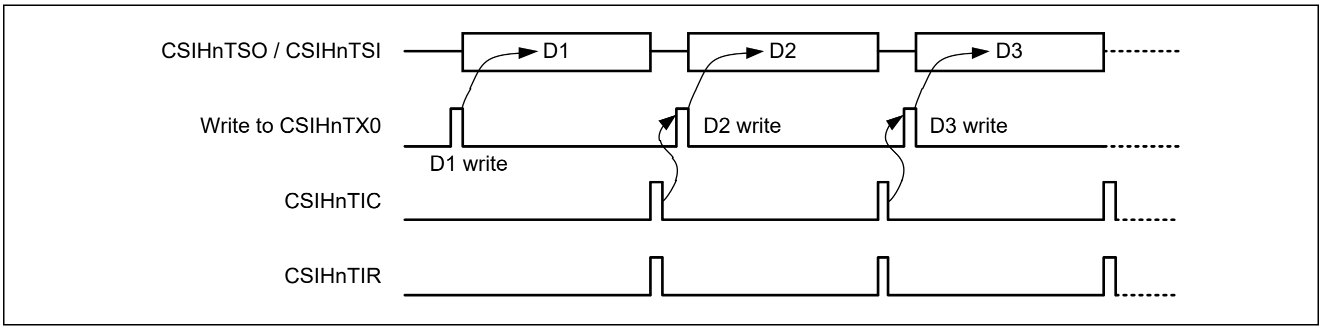


Figure 23.21 Generation of CSIHnTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is sent with CSIHnTX0W.CSIHnEOJ = 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then CSIHnTIC is replaced by the job completion interrupt CSIHnTIJC.

CSIHnTIC can also be set up to occur as soon as the CSIHnTX0 register is free for the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

Remark: This mode allows faster data transfer but is only available in direct access memory mode.

The effect is illustrated in the figure below.

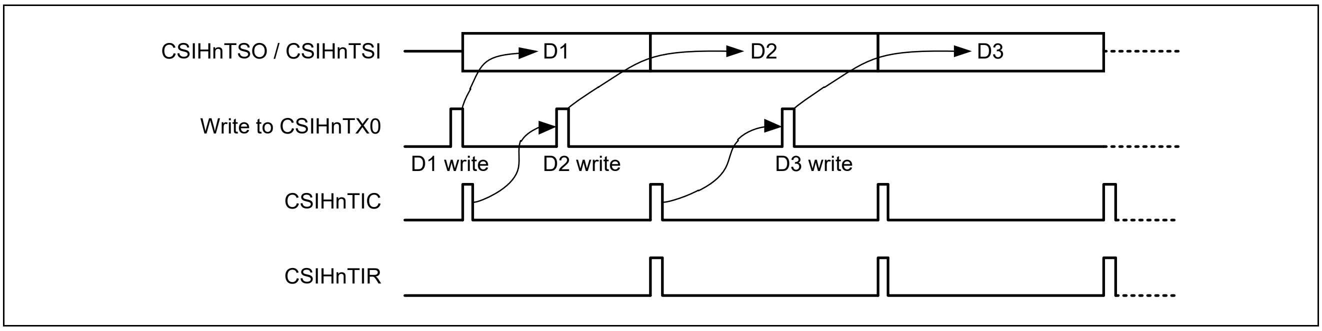


Figure 23.22 Immediate Generation of CSIHnTIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

(b) CSIHnTIC in FIFO mode

The following example shows the CSIHnTIC behavior in FIFO mode. The following example assumes:

- Master mode
- FIFO memory mode
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)

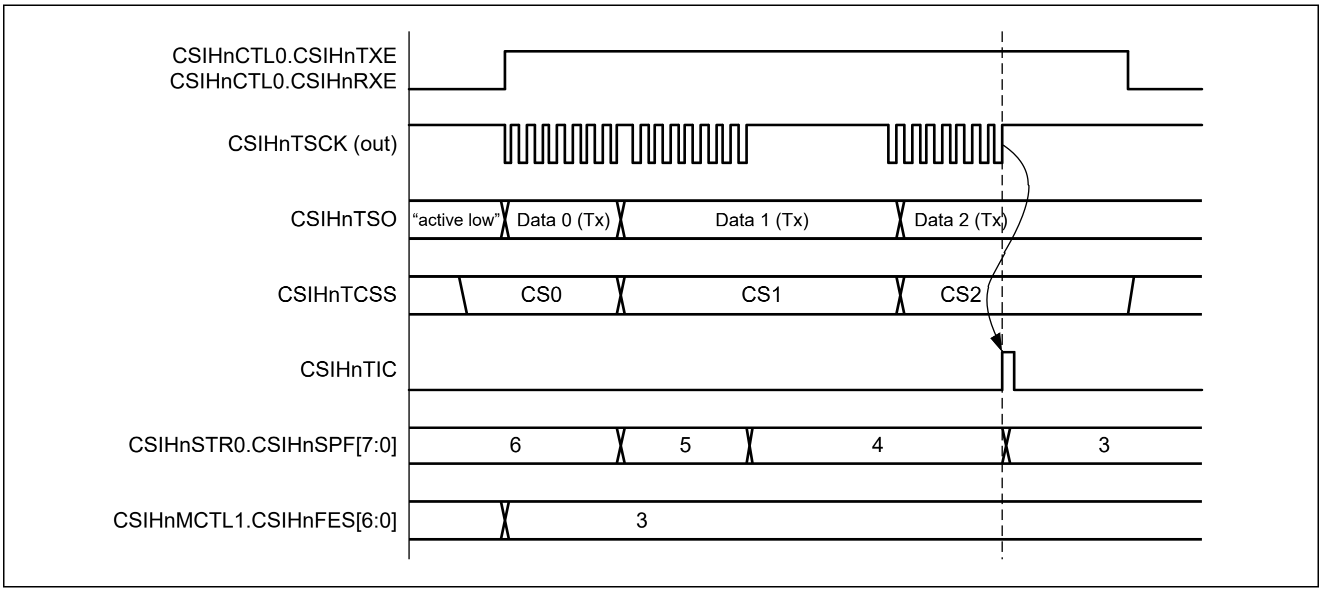


Figure 23.23 Generation of CSIHnTIC in FIFO Memory Mode

The condition for generating CSIHnTIC in FIFO mode (an empty reception buffer) is specified by using CSIHnMCTL1.CSIHnFES[6:0]. For the example in the above figure, three data items are specified as the condition. The CSIHnSTR0.CSIHnSPF[7:0] bits indicate the number of data items that remain in the FIFO buffer and have not been transmitted. When the number of remaining items matches the condition, the interrupt CSIHnTIC is generated.

(c) CSIHnTIC in job mode

The following example shows the CSIHnTIC behavior in job mode.

The following example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01H)

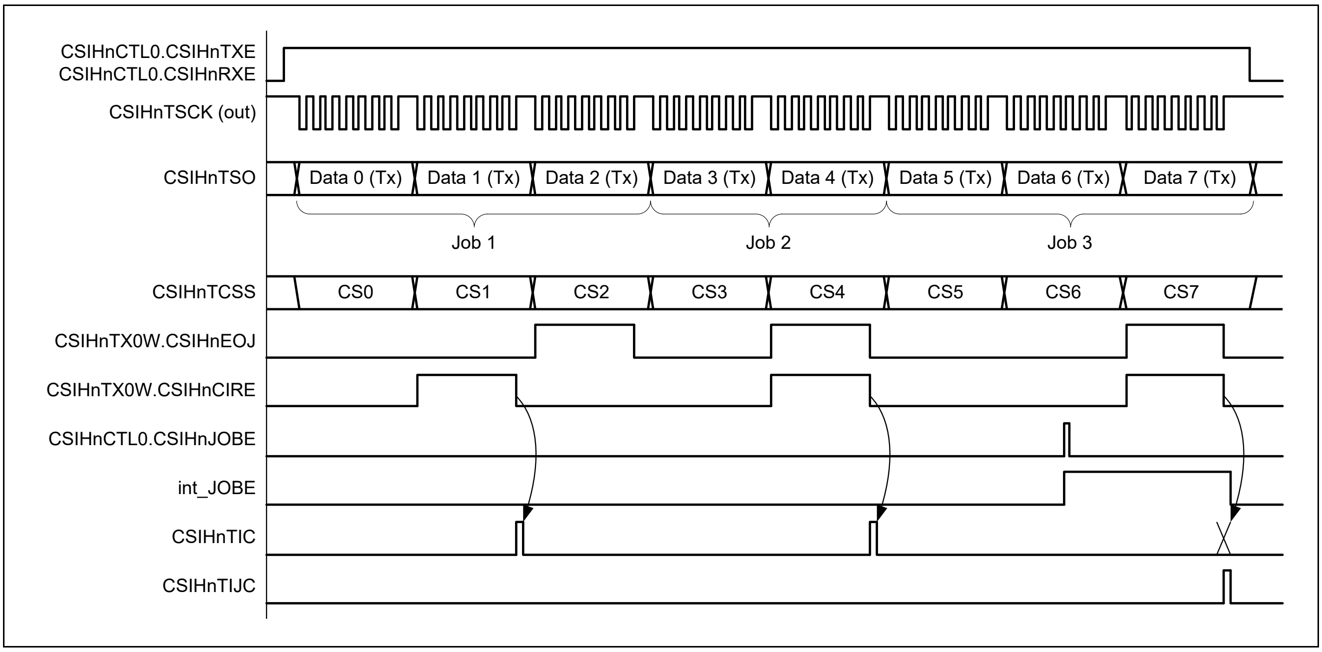


Figure 23.24 Generation of CSIHnTIC in Job Mode

Remark: The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

The rules for generating CSIHnTIC in job mode are:

Table 23.9 Generation of CSIHnTIC in Job Mode

Memory Modes	CSIHnTX0W. CSIHnCIRE	CSIHnTX0W. CSIHnEOJ	CSIHnTIC
FIFO mode (CSIHnCTL1. CSIHnJE=1)	0 (FIFO empty [Note])	0	Generated
		1	CSIHnCTL0.CSIHnJOBE = 0: Generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
	1 (FIFO empty [Note])	0	Generated
		1	CSIHnCTL0.CSIHnJOBE = 0: Generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
	0 (Data in FIFO)	0	Not generated
		1	Not generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
	1 (Data in FIFO)	0	Not generated
		1	CSIHnTIJC is generated instead of CSIHnTIC.
Dual buffer mode, transmit-only buffer mode (CSIHnCTL1. CSIHnJE = 1)	0	0	Not generated
		1	Not generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
	1	0	Generated
		1	CSIHnCTL0.CSIHnJOBE = 0: Generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
Direct access mode (CSIHnCTL1. CSIHnJE = 1)	—	0	Generated
	—	1	CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.

Note: The value of CSIHnSTR0.CSIHnSPF7–0 is the same as that of CSIHnMCTL1.CSIHnFE6–0.

(2) CSIHnTIR reception interrupt

Depending on the memory mode and job mode, this interrupt is generated according to the following conditions:

Table 23.10 CSIHnTIR Interrupt Generation

Memory Mode	Master and Slave	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO mode	This interrupt occurs when the FIFO buffer is almost full with received data, indicating to the application that the FIFO must be emptied. CSIHnTIR is generated, if the number of received data in the FIFO CSIHnSTR0.CSIHnSRP[7:0] equals CSIHnMCTL1.CSIHnFFS[6:0].	
Dual buffer mode	The interrupt is generated when communication ends (as specified by the CSIHnMCTL2.CSIHnND[7:0] bits) and CSIHnCTL0.CSIHnRXE = 1.	The interrupt is generated each time data is received if CSIHnCTL0.CSIHnRXE = 1.
Transmit-only buffer, direct access	The interrupt is generated each time data is received if CSIHnCTL0.CSIHnRXE = 1.	

In transmit-only or dual buffer mode, this interrupt is generated in receive-only and transmit/receive mode after each data has been received.

(a) CSIHnTIR in direct access mode

The following example shows the CSIHnTIR behavior in direct access mode.

The following example assumes:

- Master mode
- Direct access mode
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)

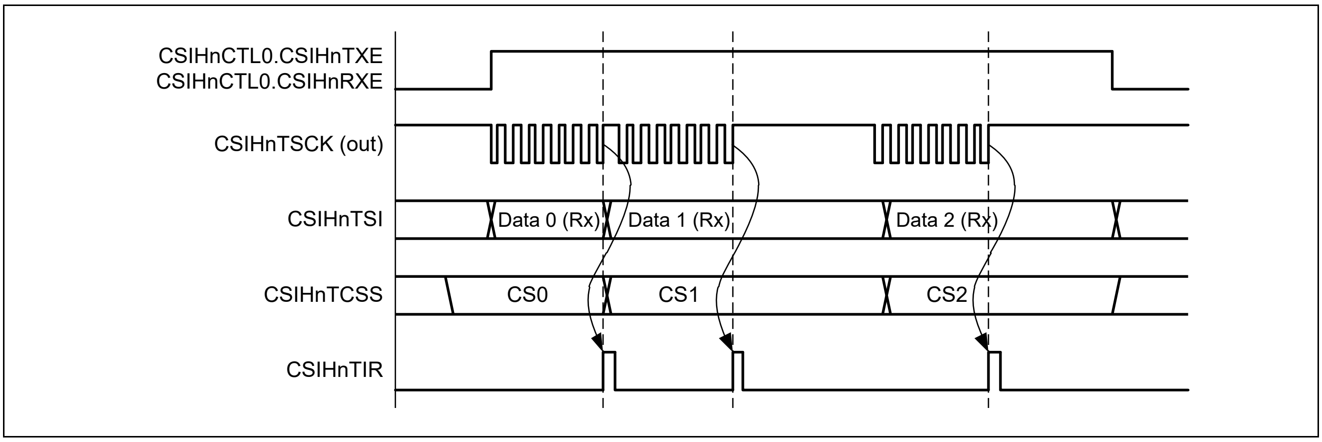


Figure 23.25 Generation of CSIHnTIR in Direct Access Memory Mode

(b) CSIHnTIR in dual buffer mode

The following example shows the CSIHnTIR behavior in buffer mode.

The following example assumes:

- Master mode
- Transmit-only or dual buffer mode
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Default clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0,CSIHnCFGx.CSIHnDAPx = 0)
- 8-bit data length (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Three data items transmitted (CSIHnMCTL2.CSIHnND[7:0] = 03H)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)

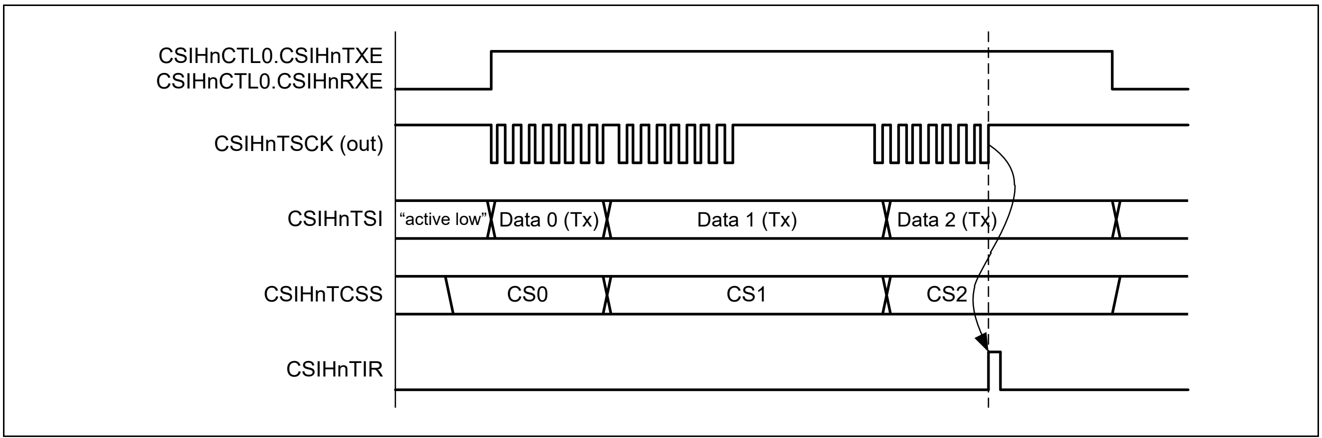


Figure 23.26 CSIHnTIR Generation in Dual Buffer Mode

(3) CSIHnTIRE reception error interrupt

This interrupt is generated whenever an error is detected.

Table 23.11 Data Error Types

Error Type	Communication Status after Error Interrupt	Comment
FIFO overflow error	Interrupt is generated and communication continues	The data written to the FIFO is lost, but previously started communications are continued.
Parity error	Interrupt is generated and communication continues	—
Data consistency error	Interrupt is generated and communication continues	—
Timeout error	Interrupt is generated and communication continues	—
Overrun error	Communication continues after the interrupt is generated. (Communication does not stop.)	This error occurs (but only for FIFO mode) if the CPU reads reception data after the number of reception data items reaches 0.

The type of error that caused the generation of CSIHnTIRE is flagged in register CSIHnSTR0.

Additionally a parity and data consistency error flag is attached to the reception data in CSIHnRX0W.

For details about the various error types, refer to section 23.4.13, Error Detection.

(4) CSIHnTIJC job completion interrupt

This interrupt supports the handling of jobs – refer to section 23.4.5, Job Concept. This interrupt is only available in master mode.

Job mode is enabled by setting CSIHnCTL1.CSIHnJE = 1. When CSIHnCTL1.CSIHnJE = 0, CSIHnTIJC is not generated.

Depending on the memory mode, this interrupt is generated according to the following conditions:

Table 23.12 CSIHnTIJC Interrupt Generation

Memory Mode	Interrupt Source	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO mode	Not applicable	After job abortion Note is triggered, communication stops on job completion.
Transmit-only buffer mode		
Dual buffer mode		
Direct access mode		

Note: Job abortion condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

(5) Delay for all interrupts

In master mode, all interrupts generated by the master can be delayed one half cycle of the serial clock CSIHnTSCK. This function cannot be used in slave mode.

To specify this delay, set the CSIHnCTL1.CSIHnSIT bit to 1.

The figure below shows an example of using the interrupt delay function with the following settings:

CSIHnCTL1.CSIHnSIT = 1 (interrupt delay enabled),
CSIHnCFGx.CSIHnCKPx = 0,
CSIHnCFGx.CSIHnDAPx = 0 (normal clock phase and data phase)
, and CSIHnCFGx.CSIHnDLSx[3:0] = 1000B (8-bit data length).

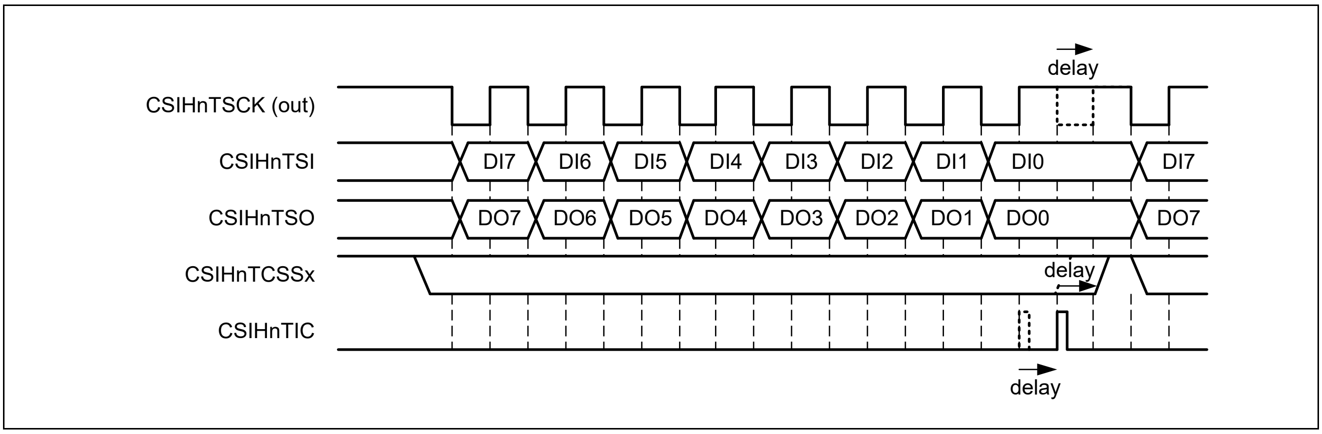


Figure 23.27 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

When CSIHnCTL1.CSIHnSIT is set to 1, a delay of half a serial clock cycle is added. This also delays the end of the current chip select signal (CSIHnTCSSx).

23.4.13 Error Detection

CSIH can detect five error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)
- Timeout error (in FIFO mode)
- Overflow error (in FIFO mode)

Check for parity, data consistency and timeout errors can be enabled/disabled individually.

If one of these errors is detected, the interrupt request CSIHnTIRE is generated and the corresponding flag is set.

(1) Data consistency checking

The purpose of data consistency checking is to ensure that the data physically sent as output signal is identical with the original data that was copied to the shift register.

Data consistency checking can be enabled/disabled by bit CSIHnCTL1.CSIHnDCS. It is not active if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When data consistency checking 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 CSIHnTSO are read back via the CSIHnTDCS signal into an own shift register.

After completion of the transmission, the sent data is compared with the original transmission data.

Mismatching is considered as a data consistency error.

When a data consistency error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnDCE is set.

Additionally, CISHnRX0W.CSIHnTDCE is set with the corresponding data.

The figure below is a block diagram of data consistency checking.

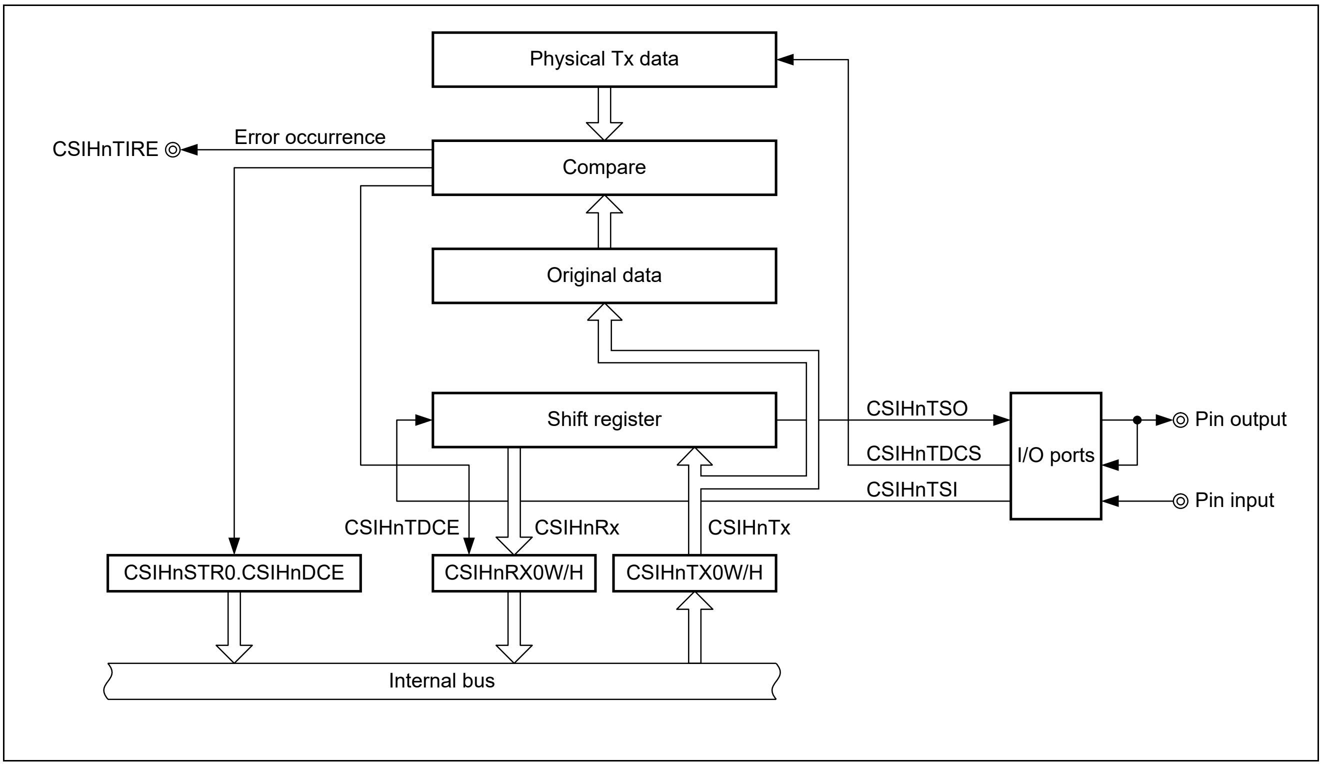


Figure 23.28 Block Diagram of Data Consistency Checking

(2) Parity check

Parity checks are often used to detect single bit errors during data transmission. 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.

When a parity error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnPE is set.

The following figure shows an example.

- Data length is 8 bits.
- The data transmitted is 05H and 35H.
- Data direction is LSB first.
- Parity type is odd.

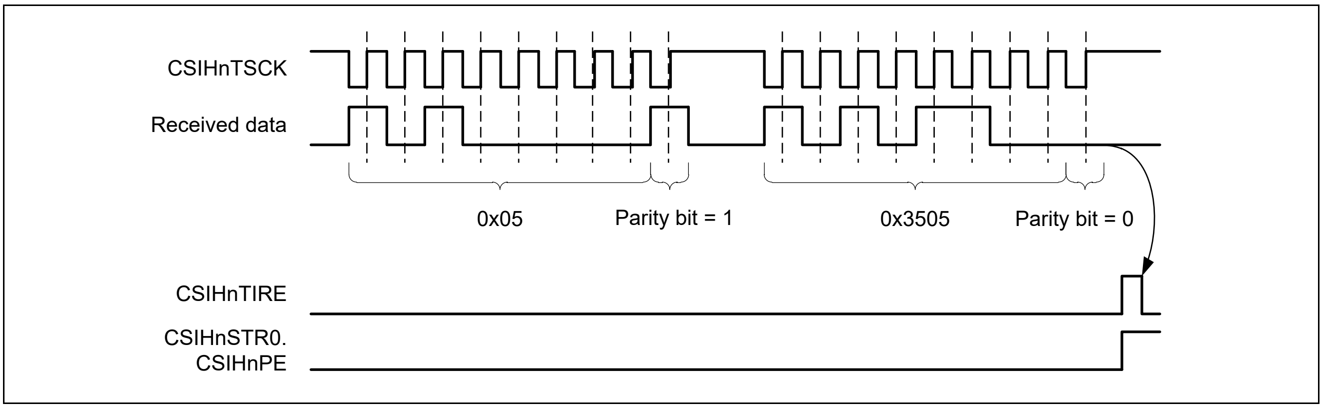


Figure 23.29 Parity Check Example

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 using the extended data length (EDL) function, the parity bit is added after the last data bit.

(3) Timeout error

Timeout error checks are only possible in slave FIFO mode.

A timeout error occurs if neither of the following occurs within a specific time:

- Reading reception data in the FIFO buffer
- Reception of data by the FIFO buffer from CSIHnTSI

The time is defined in CSIHnMCTL0.CSIHnTO[4:0] in multiples of 8 times the transmission clock CSIHnSCK. Timeout error occurs when the specified time is exceeded (When CSIHnMCTL0.CSIHnTO[4:0] is cleared to 00000B, the timeout time is not detected.).

A dedicated timeout counter measures the time between the last and the next read operation.

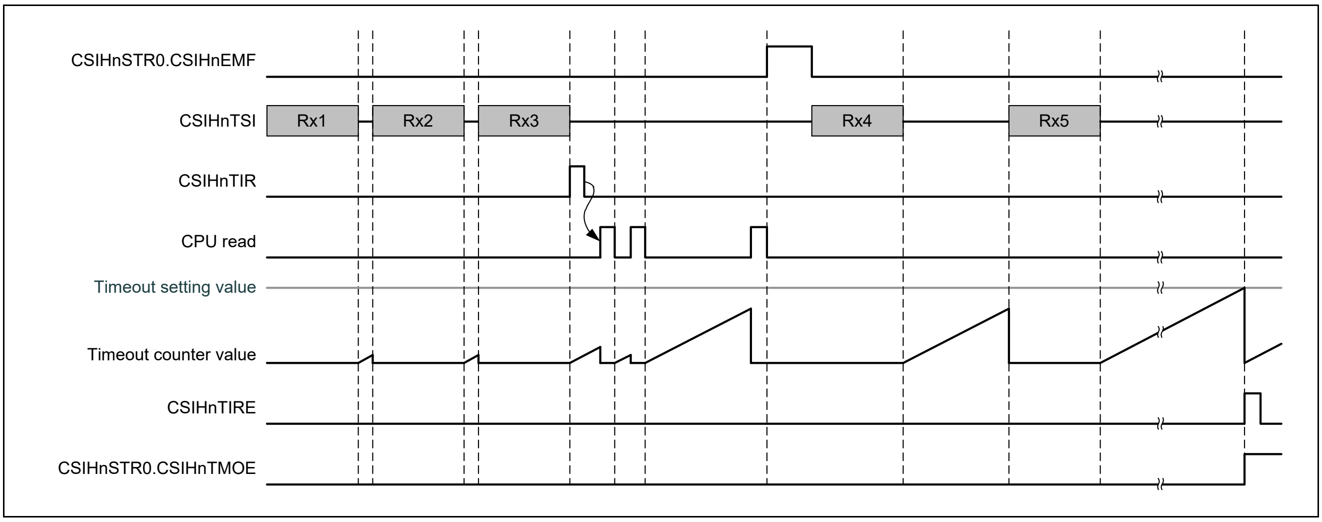


Figure 23.30 Timeout Error Check Functional Timing Chart

The timeout counter starts when:

- Reception ends.
- Reading data from the CPU ends. (If the buffer is empty, the counter does not start.)
- A timeout error is detected.

After a timeout error is detected, if the system is left as is, the timeout counter restarts.

If the time specified by the CSIHnMCTL0.CSIHnTO[4:0] bits is reached again, another CSIHnTIR interrupt is output.

The timeout counter continues counting as long as reception data is not read.

To stop the timeout counter, read all the reception data, or set CSIHnSTCR0.CSIHnPCT (to 1). However, the pointer is cleared in this case.

The timeout counter is reset when:

- Data is read.
- One new data item is received.
- A timeout error is detected.
- The CSIHnSTCR0.CSIHnPCT bit is set.

When a timeout error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnTMOE is set.

(4) Overflow error

Overflow errors can occur in FIFO mode. An overflow error occurs when transmission data is written to the CSIHnTX0W or CSIHnTX0H register while the FIFO buffer is full of transmission data and reception 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. When the CPU tries to write the 39th data, an overflow error happens.

This is illustrated in the following figure:

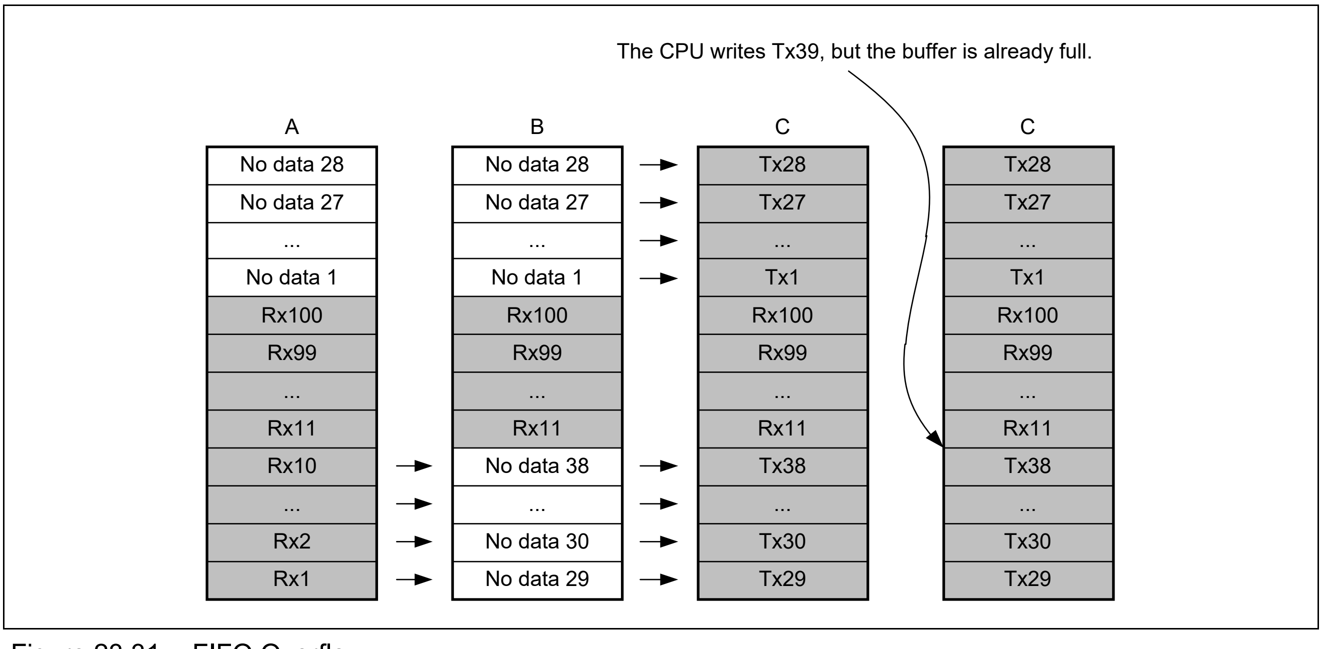


Figure 23.31 FIFO Overflow

The data after 39 are discarded. The following figure shows the associated timing.

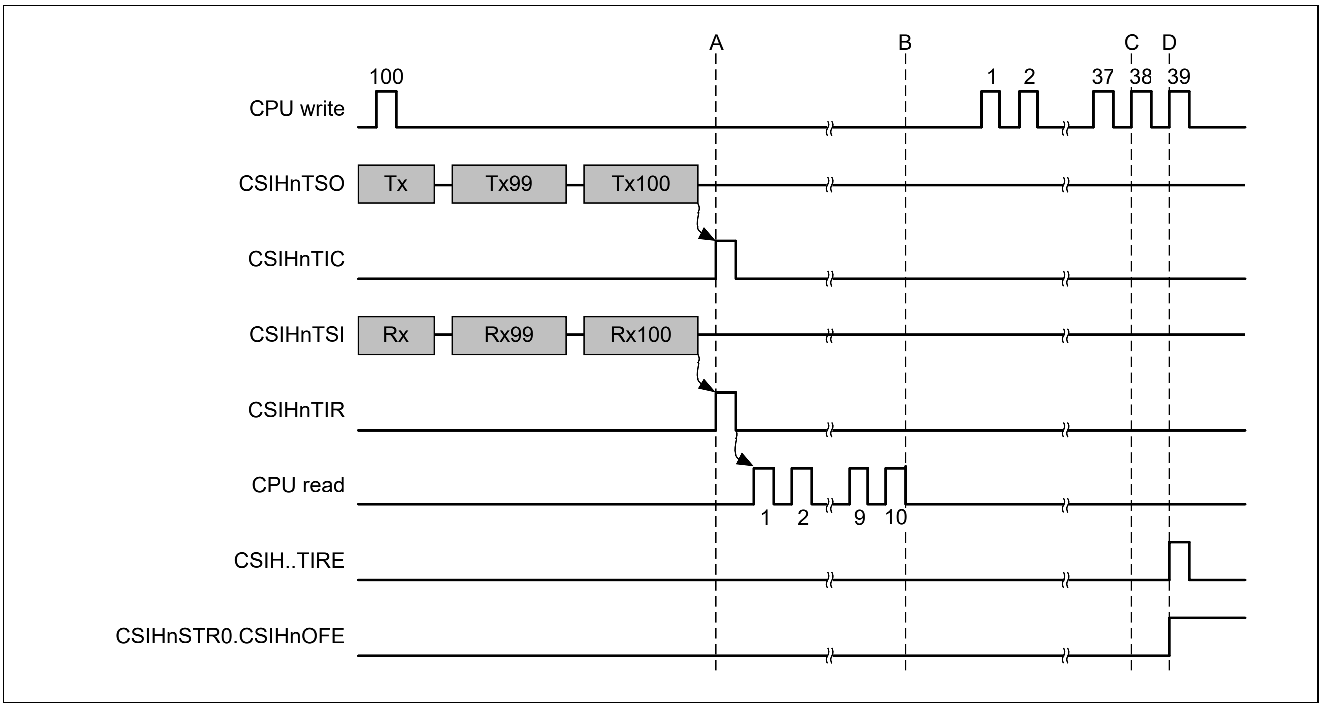


Figure 23.32 FIFO Overflow Timing

When an overflow error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnOFE is set.

(5) Overrun error

Overrun errors can occur in direct access mode, transmit-only buffer mode, and FIFO mode. They cannot occur in dual buffer mode.

(a) 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 data register CSIHnRX0. This happens when CSIHnRX0 was not read and therefore contains previous reception data.

In master mode, because the serial clock is stopped until the CPU reads reception data, overrun errors do not occur.

The following figure illustrates the function.

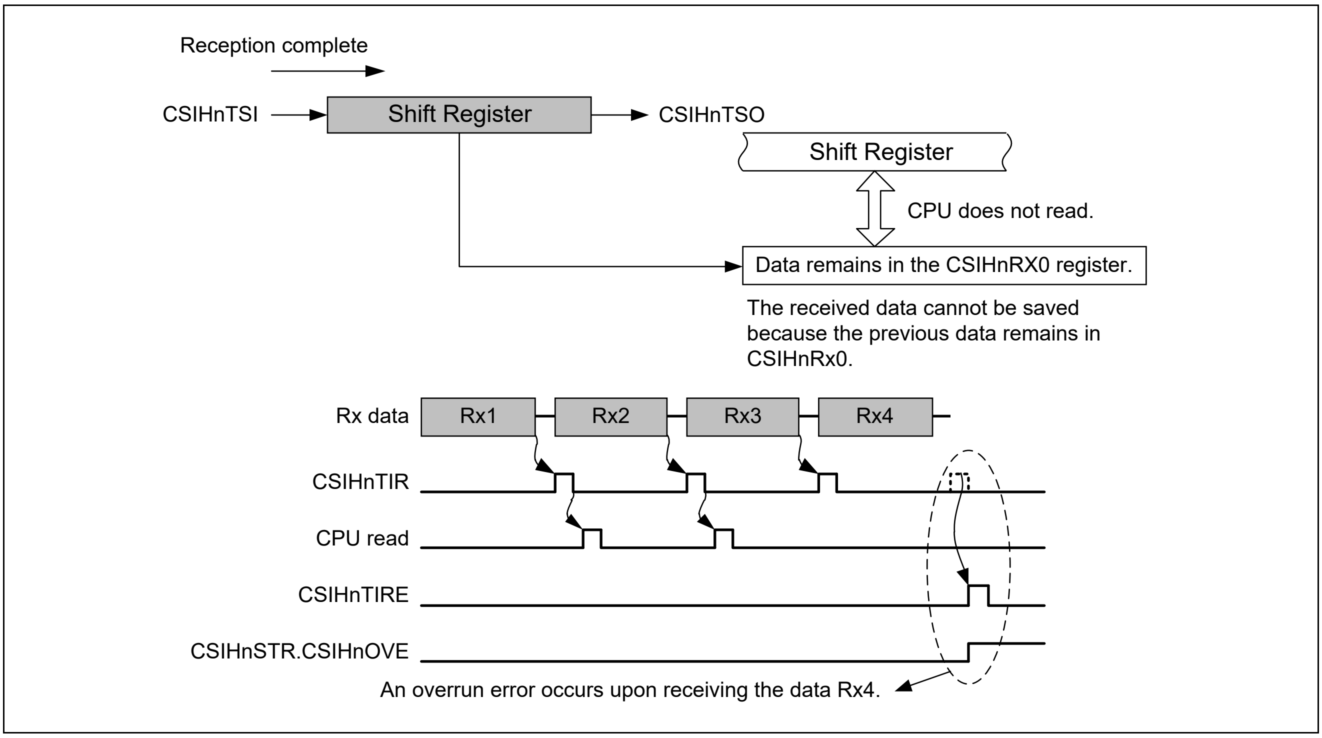


Figure 23.33 Overrun Error Detection in Direct Access and Transmit-Only Buffer Mode

(b) FIFO mode

In FIFO mode, an overrun error occurs if:

1. Because the FIFO buffer is full, new received data cannot be transferred from the shift register to the FIFO buffer.
2. No data. The CPU attempts to read reception data that does not exist.

Remark: If the CPU attempts to read reception data that does not exist in FIFO mode, an overrun error occurs even if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

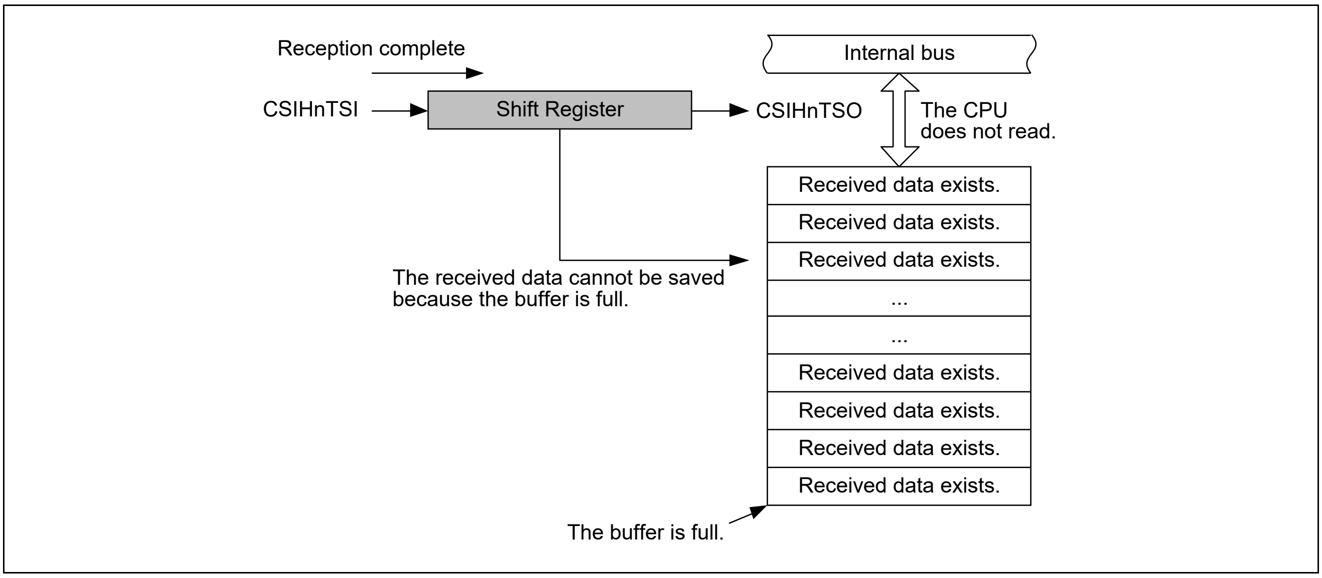


Figure 23.34 Overrun Error Detection in FIFO Mode (FIFO Full)

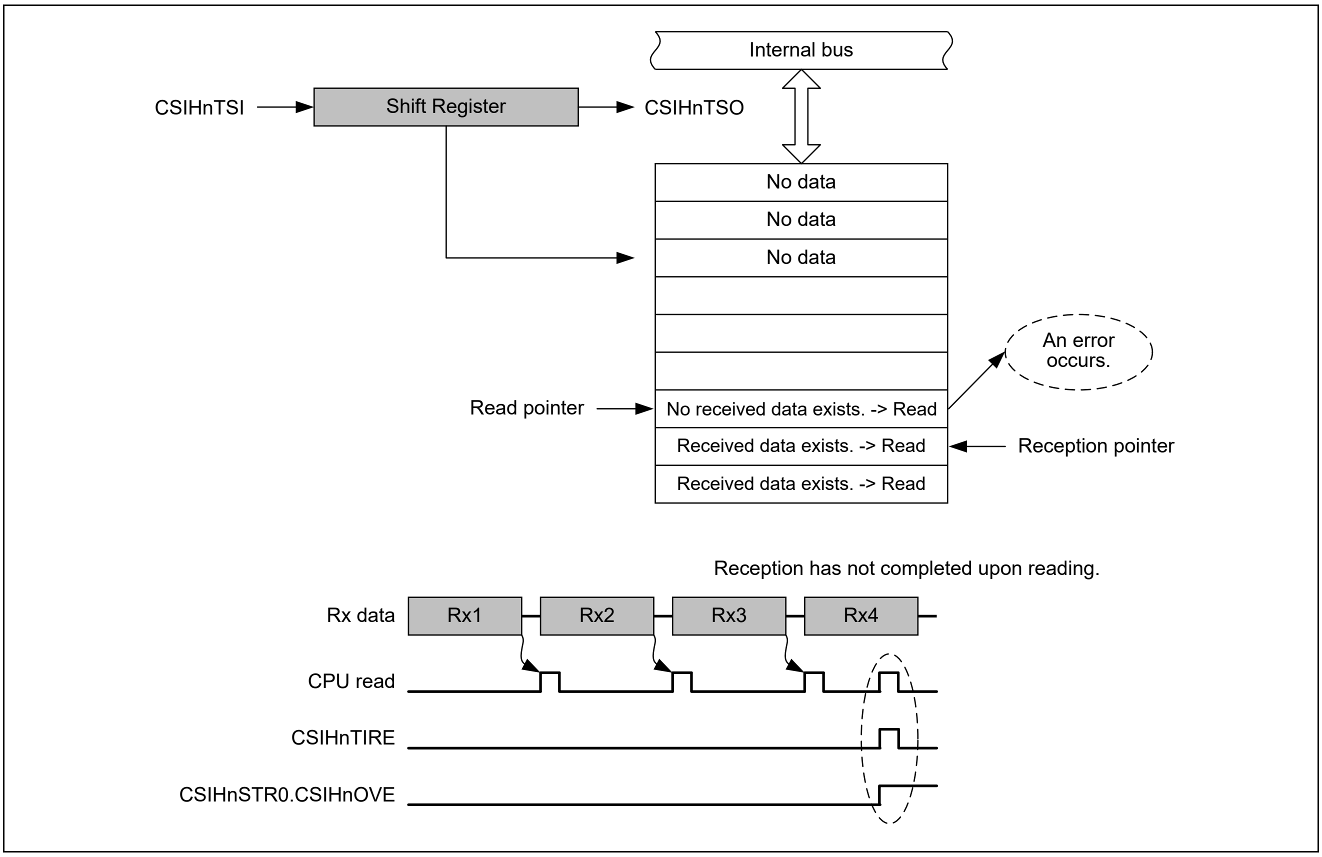


Figure 23.35 Overrun Error Detection in FIFO Mode (No Data)

When an overrun error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnOVE is set.
- CSIHnRX0W is written again with the received data.
- Communication continues (unless the CPU attempted to read data that did not exist).

23.4.14 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, the transmit and receive signals are internally connected, as shown in the figures below. The signals CSIHnTSCK,

CSIHnTSO, and CSIHnTSI are disconnected from the ports. In addition, the CSIHnTSO output level is fixed to low, and CSIHnTSCK becomes inactive according to the setting of CSIHnCFGx.CSIHnCKPx.

The CSIHnTSCK, CSIHnTSO, CSIHnTSI, and CSIHnTCSSx[1:0] signals are disconnected from ports. The CSIHnTSO signal is fixed to the low output level, and the CSIHnTSCK and CSIHnTCSSx[1:0] signals are set to the inactive level (the level specified by the CSIHnCFGx.CSIHnCKPx bit in the case of the CSIHnTSCK signal, and the level specified by the CSIHnCTL1.CSIHnCLS[1:0] bits in the case of the CSIHnTCSSx[1:0] signal).

To perform a self-test of the CSIH, CSIHnCTL1.CSIHnLBM is set to 1, and a normal transfer operation is executed. Next, whether the reception data and transmission data are the same is checked.

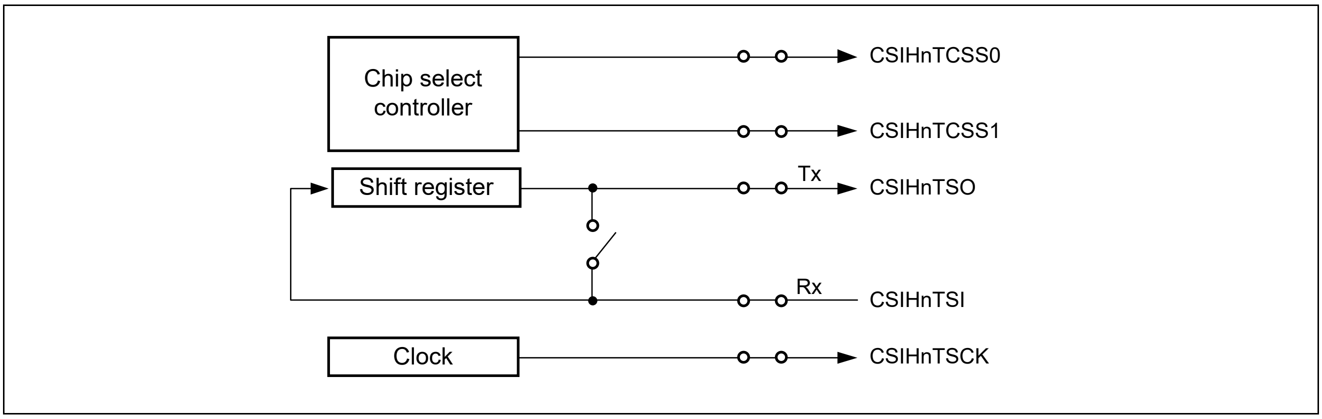


Figure 23.36 Normal Operation (CSIHnCTL1.CSIHnLBM = 0)

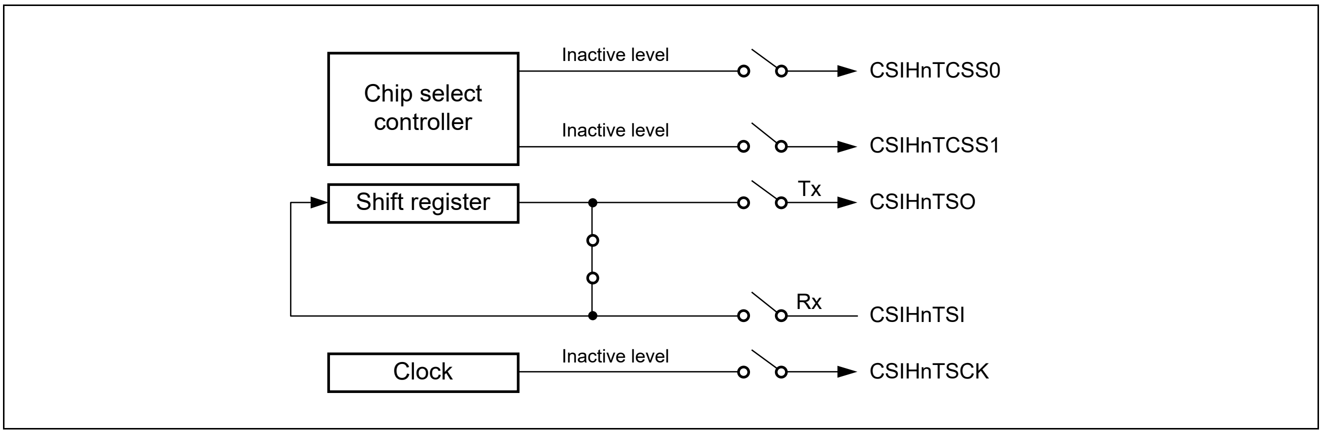


Figure 23.37 Loop-Back Operation (CSIHnCTL1.CSIHnLBM = 1)

23.5 Operating Procedures

The following examples and instructions are sorted according to the memory mode:

- Direct access
- Transmit-only buffer
- Dual buffer
- FIFO

23.5.1 Procedures in Direct Access Mode

(1) For transmission/reception in master mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- A CSIHnTIC interrupt is generated when transferring starts. (CSIHnCTL1.CSIHnCLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 0)

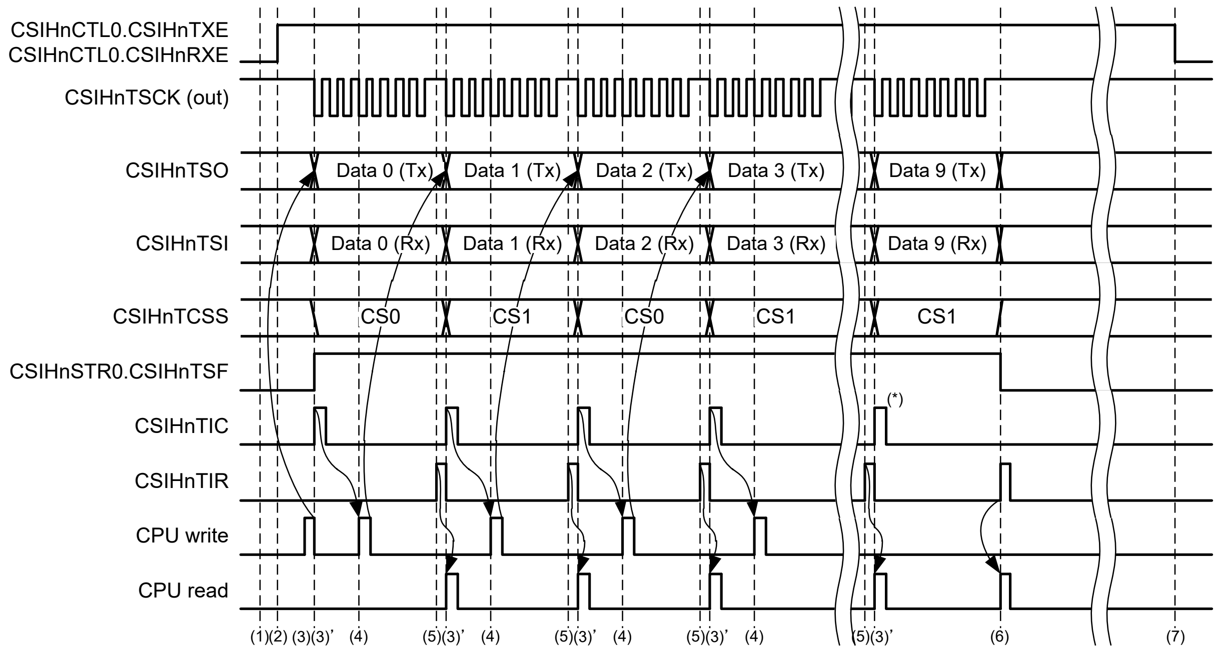


Figure 23.38 Direct Access Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0, CS1 are used.)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
  3. Write the first data to the transmission data register CSIHnTX0W. This write operation activates CS0, and transmission automatically starts.

  3'. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

  4. Write the second data to CSIHnTX0W. If necessary, it is possible to change the CS and make a different device the communication partner. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
  5. Each time data is received, a CSIHnTIR interrupt is generated.
     - CSIHnTIR indicates that the reception data register CSIHnRX0 must be read.
  6. If the CSIHnTIC interrupt indicated by (*) in the figure is the last one, it is not necessary to write to the transmission data register CSIHnTX0W based on the corresponding CSIHnTIC interrupt.
  7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of the CSIH.

(2) For reception in master mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0,CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- A CSIHnTIC interrupt is generated when transferring starts. (CSHICTL1.CSIHnSLIT = 1)
- Direct access mode (CSHICTL0.CSIHnMBS = 1)

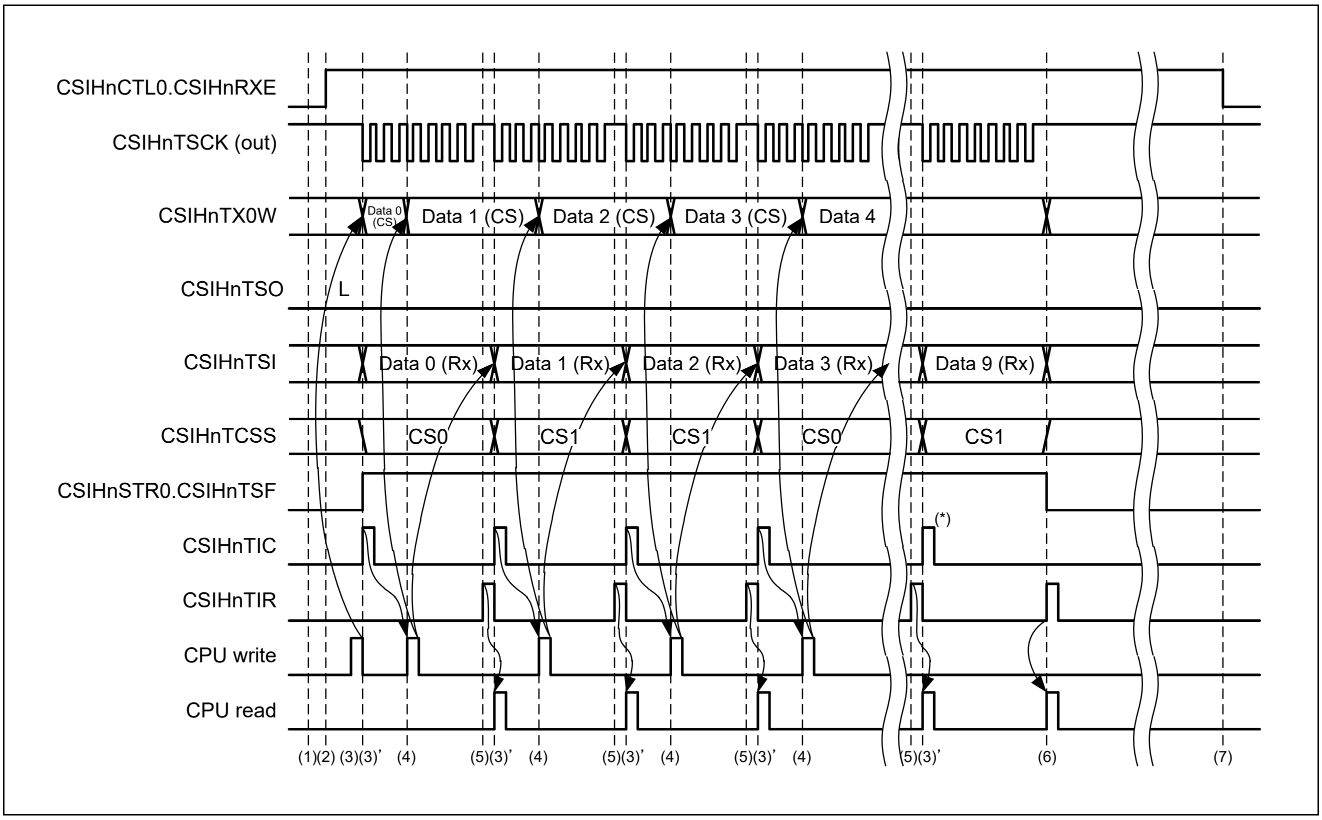


Figure 23.39 Direct Access Mode (for Reception in Master Mode, and when Job Mode is Disabled)

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0, CS1 are used.)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
  3. Write the transmission data to the transmission data register CSIHnTX0W for the CS data. This write operation activates CS0, and reception automatically starts.

  3'. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

  4. Write the second data to CSIHnTX0W. If necessary, it is possible to change the CS and make a different device the communication partner. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
  5. Each time data is received, a CSIHnTIR interrupt is generated.
     - CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
  6. If the CSIHnTIC interrupt indicated by (*) in the figure is the last one, it is not necessary to write to the transmission data register CSIHnTX0W based on the corresponding CSIHnTIC interrupt.
  7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of the CSIH.

(3) For transmission/reception in slave mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- CSIHnTIC interrupt generated at the transfer start timing (CSIHnCTL1.CSIHnSLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)

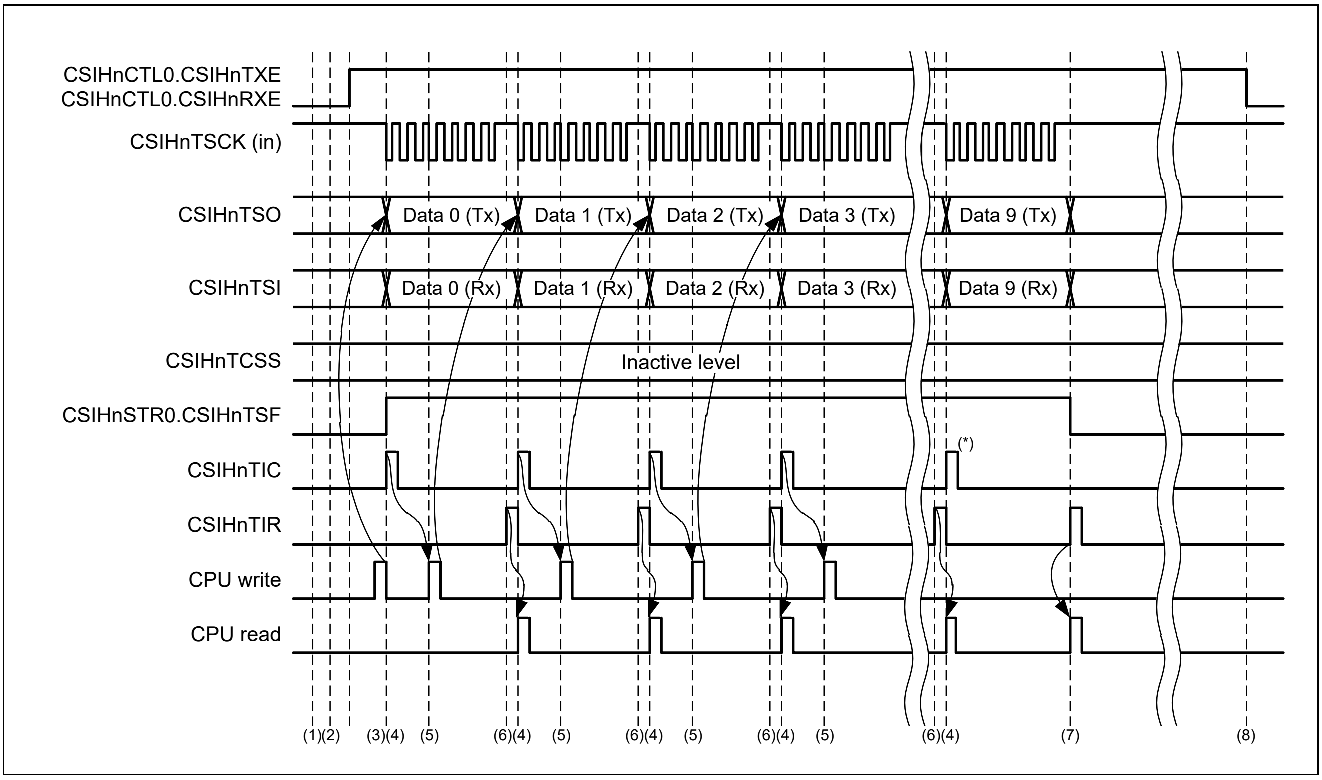


Figure 23.40 Direct Access Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnCFG0 (communication protocol)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
  3. Write the first data to the transmission data register CSIHnTX0W.
  4. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.
  5. Write the second data to CSIHnTX0W. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
  6. Each time data is received, a CSIHnTIR interrupt is generated.
     - CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
  7. If the CSIHnTIC interrupt indicated by (*) in the figure is the last one, it is not necessary to write to the transmission data register CSIHnTX0W based on the corresponding CSIHnTIC interrupt.
  8. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of the CSIH.

(4) For reception in slave mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- CSIHnTIC interrupt generated at the transfer start timing (CSIHnCTL1.CSIHnSLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)

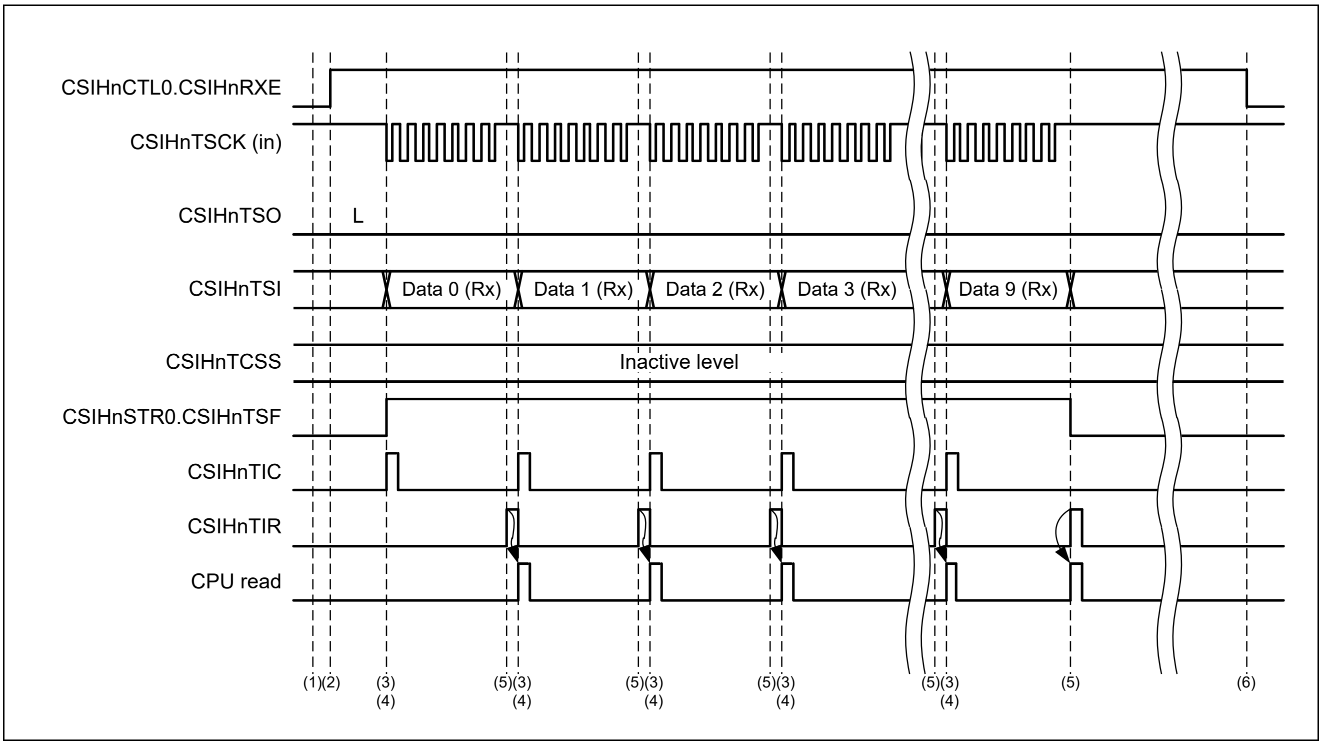


Figure 23.41 Direct Access Mode (for Reception in Slave Mode, and when Job Mode is Disabled)

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnCFG0 (communication protocol)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
  3. When a serial clock is supplied from the master, reception automatically starts.
  4. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK.
  5. Each time data is received, a CSIHnTIR interrupt is generated.
     - CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
  6. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of the CSIH.

(5) For transmission/reception in master mode, and when job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- CSIHnTIC interrupt generated at the transfer start timing (CSIHnCTL1.CSIHnSLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs that each transmit three data packets

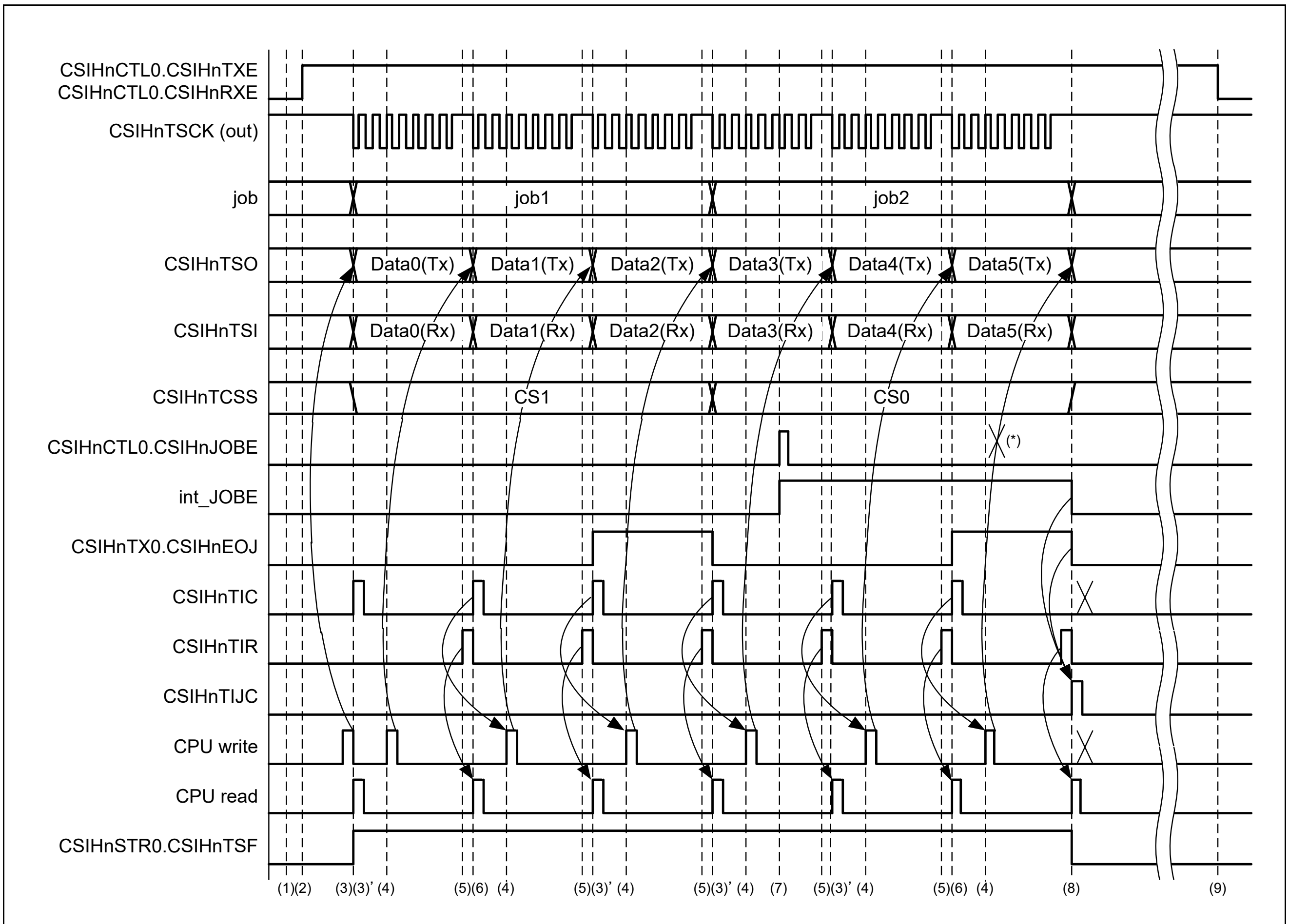


Figure 23.42 Direct Access Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)

Remark: The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

• Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 and CS1 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
3. Write the first transmission data packet to the transmission data register CSIHnTX0W.
   Transmission automatically starts when the first data can be used
   The CSIHnSTR0.CSIHnTSF flag indicates that communication is being performed.

3'. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

4. Write the second data to CSIHnTX0W. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
5. Each time data is received, a CSIHnTIR interrupt request is generated.
   - CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
6. If the CSIHnTX0W register transfer data is the last job data, CSIHnTX0W.CSIHnEOJ is set to 1.
7. By setting CSIHnCTL0.CSIHnJOBE to 1, communication is forcibly stopped when the current job (job 2) ends.
8. After communication is forcibly stopped, the interrupt request CSIHnTIC is replaced with CSIHnTIJC. CSIHnTIR is generated as usual.
   The interrupt request CSIHnTIJC indicates that communication was forcibly stopped when the current job ended.
   The interrupt request CSIHnTIC is not generated. Note that the usable transmission data in the CSIHnTX0 register (indicated by (*) in the figure) is not transmitted.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of CSIH.

(6) For reception in master mode, and when job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- CSIHnTIC interrupt generated at the transfer start timing (CSIHnCTL1.CSIHnSLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs that each transmit three data packets

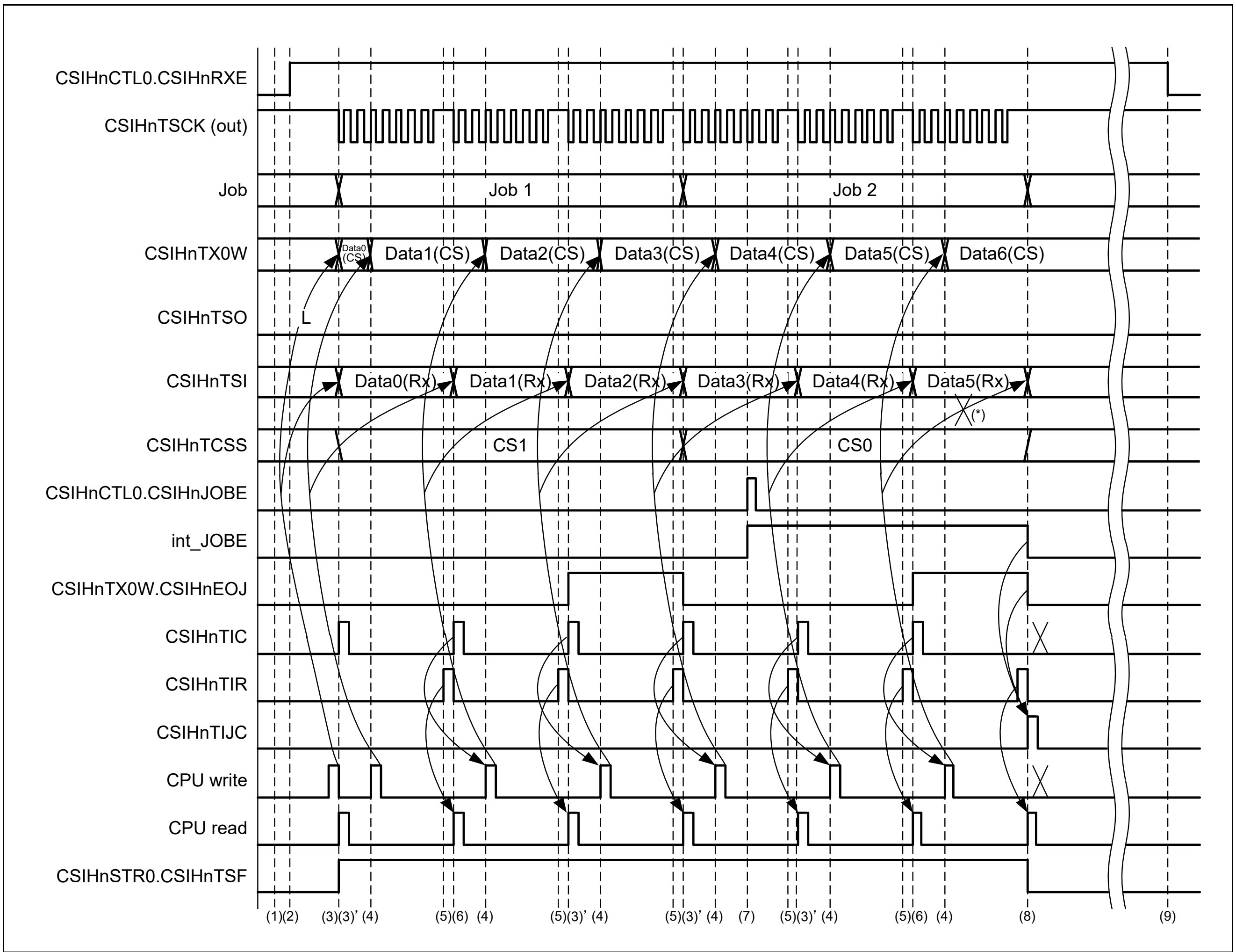


Figure 23.43 Direct Access Mode (for Reception in Master Mode, and when Job Mode is Enabled)

Remark: The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0 and CS1 are used.)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnMBS = 1 (direct access mode selected)
  3. Write the transmission data to the transmission data register CSIHnTX0W for reception.
     Reception automatically starts. In addition, the CSIHnSTR0.CSIHnTSF bit is set.

  3'. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

  4. Write the second data to CSIHnTX0H. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
  5. Each time data is received, a CSIHnTIR interrupt request is generated.
     - CSIHnTIR indicates that the reception data register CSIHnRX0 must be read.
  6. If the CSIHnTX0W register transfer data is the last job data, CSIHnTX0W.CSIHnEOJ is set to 1.
  7. By setting CSIHnCTL0.CSIHnJOBE to 1, communication is forcibly stopped when the current job (job 2) ends.
  8. When int_JOBE is set and the last job 2 data is received, the interrupt request CSIHnTIJC is generated instead of CSIHnTIC.
     CSIHnTIR is generated as usual.
     The interrupt request CSIHnTIJC indicates that reception was forcibly stopped when the current job ended.
     The interrupt request CSIHnTIC is not generated. Note that the data indicated by (*) in the figure is not transferred.
  9. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of CSIH.

23.5.2 Procedures in Transmit-Only Buffer Mode

This section provides examples where job mode is enabled or disabled.

(1) For transmission/reception in master mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0 , CSIHnMCTL0.CSIHnMMS[1:0] = 10B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

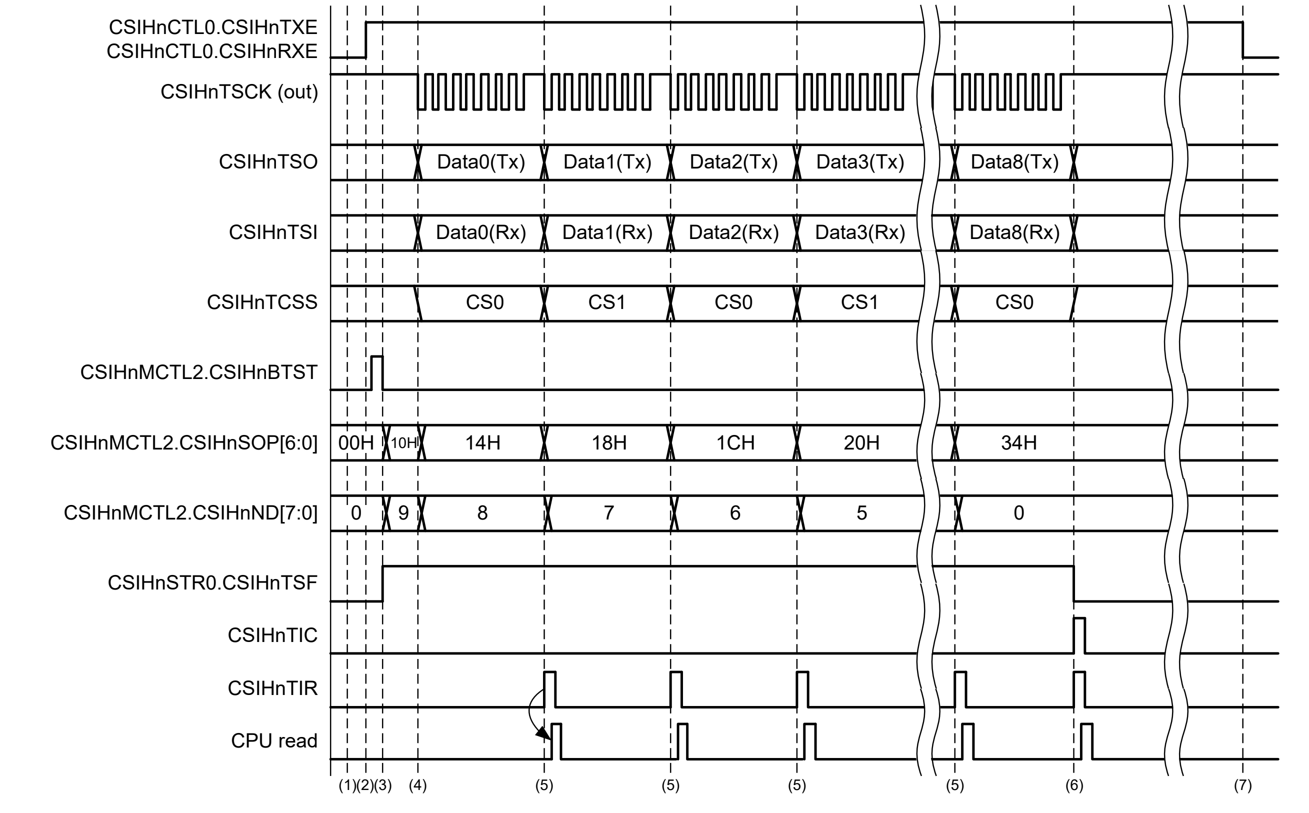


Figure 23.44 Transmit-Only Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)

Remark: The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

• Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0, CS1 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission/reception starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data item is transmitted.
5. When all the data are received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. When all the data are transmitted, a CSIHnTIC interrupt request is generated.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using CSIH.

(2) For reception in master mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

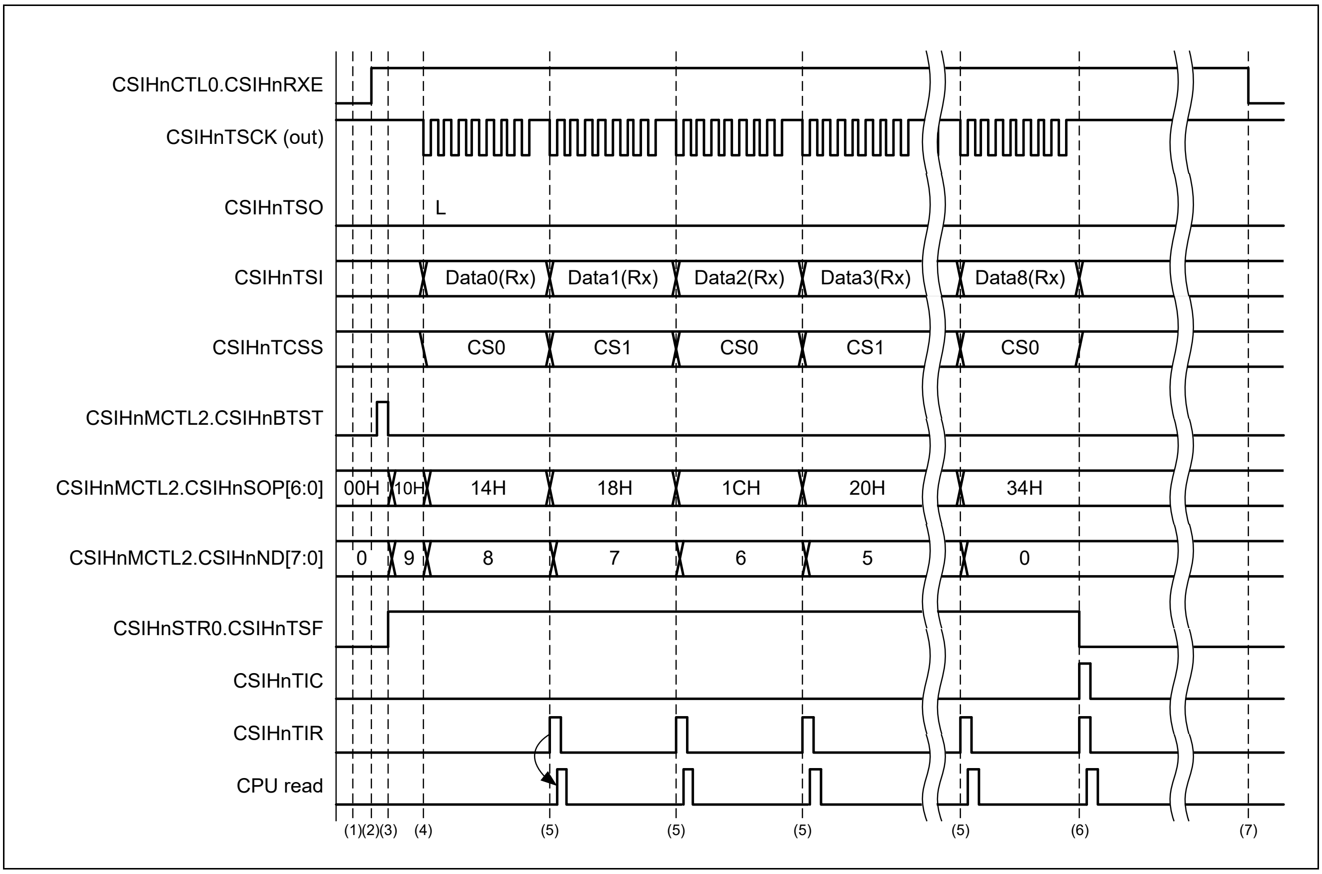


Figure 23.45 Transmit-Only Buffer Mode (for Reception in Master Mode, and when Job Mode is Disabled)

Remark: The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 10B (memory mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0 to CS1 are used.)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
     Communication is started by setting CSIHnMCTL2.CSIHnBTST.
  4. Reception starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data packet is transmitted.
  5. When all the data are received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
  6. When all the data are received, a CSIHnTIC interrupt request is generated.
  7. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
     In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using CSIH.

(3) For transmission/reception in slave mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

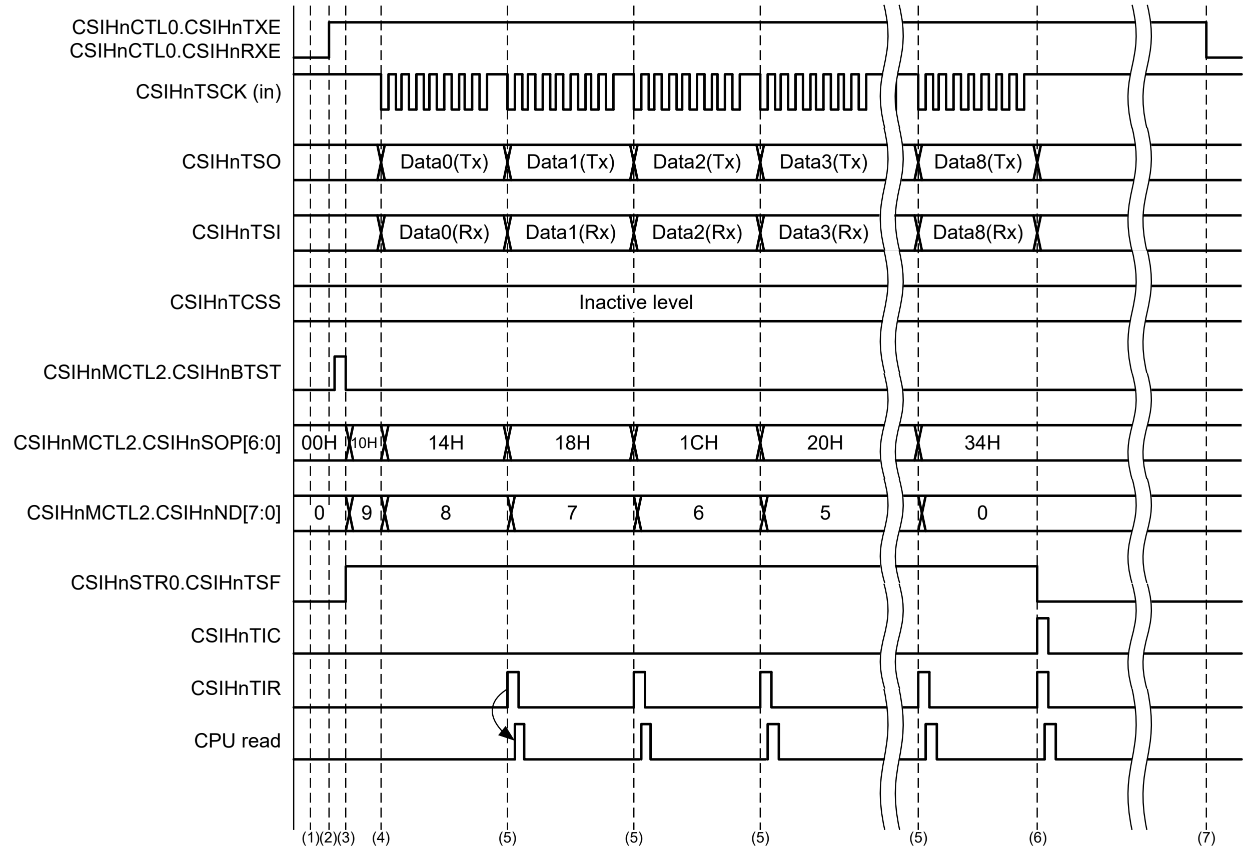


Figure 23.46 Transmit-Only Buffer Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)

Remark: The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 10B (memory mode)
     CSIHnCFG0 (communication protocol)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
     Communication is started by setting CSIHnMCTL2.CSIHnBTST.
  4. When a serial clock is supplied from the master, communication starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data packet is transmitted.
  5. Each time data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
  6. When all the data are received, a CSIHnTIC interrupt request is generated.
  7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using CSIH.

(4) For reception in slave mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

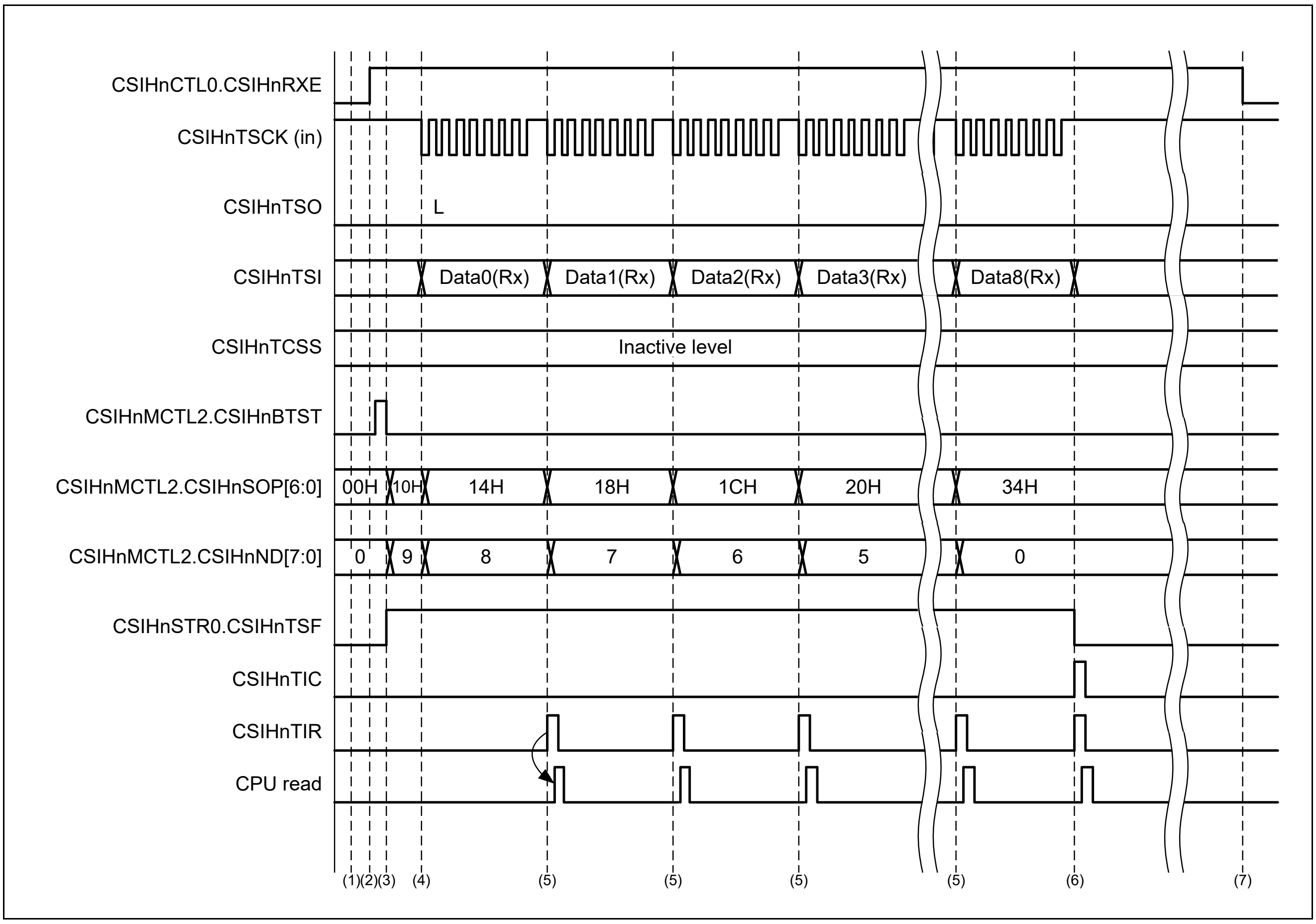


Figure 23.47 Transmit-Only Buffer Mode (for Reception in Slave Mode, and when Job Mode is Disabled)

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 10B (memory mode)
     CSIHnCFG0 (communication protocol)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
     Reception is started by setting CSIHnMCTL2.CSIHnBTST.
  4. When a serial clock is supplied from the master, reception starts.
     The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data packet is transmitted.
  5. Each time data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
  6. When all the data are received, a CSIHnTIC interrupt request is generated.
  7. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
     In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using CSIH.

(5) For transmission/reception in master mode, and when job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

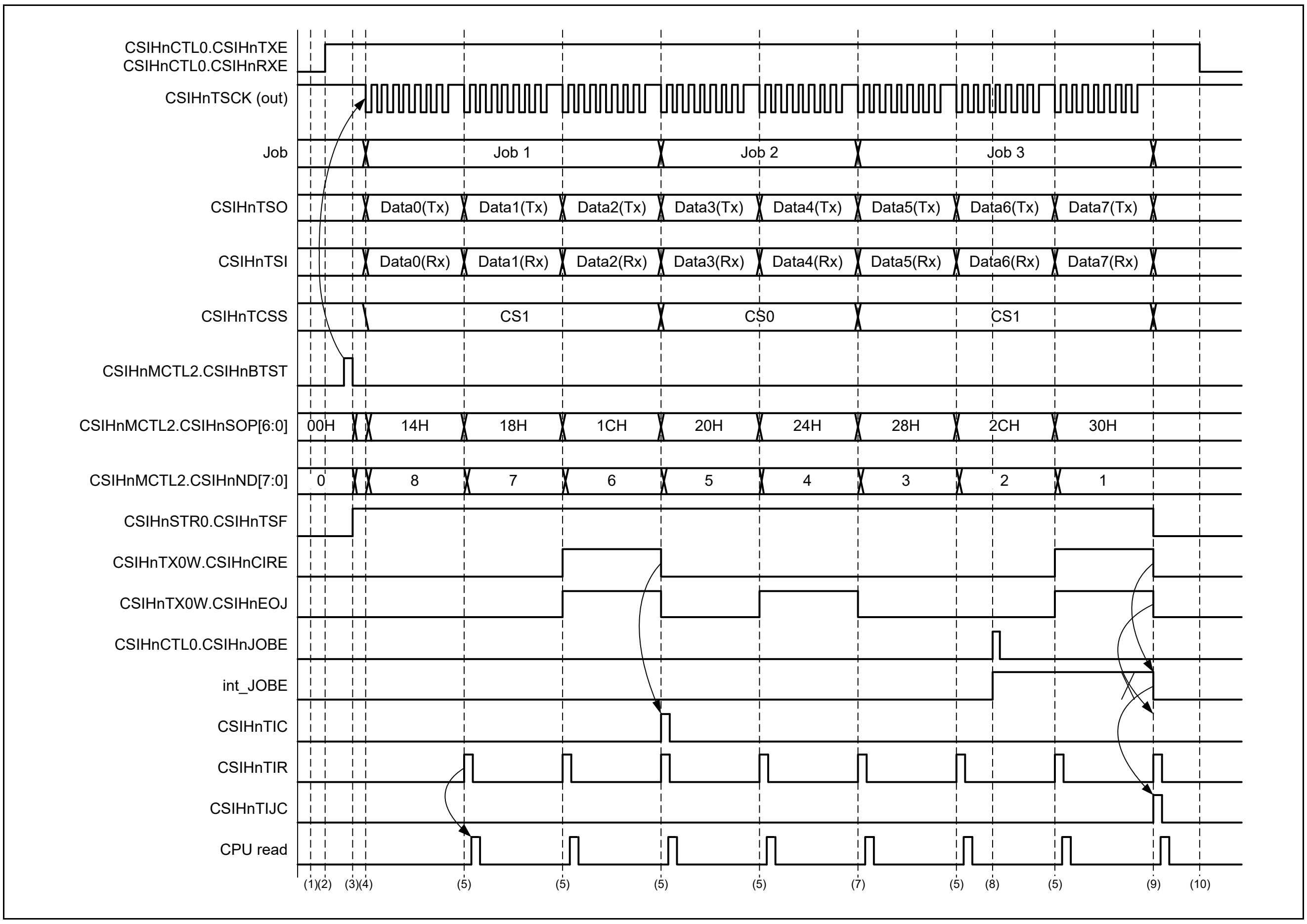


Figure 23.48 Transmit-Only Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)

Remarks 1. The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

2. The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

• Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 and CS1 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data item is transmitted.
5. Each time a data item is received, a CSIHnTIR interrupt request is generated.
   CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
6. CSIHnTIC is generated by setting CSIHnTX0W.CSIHnCIRE to 1.
   CSIHnTIC indicates that the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted.
7. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by clearing CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is not generated.
8. By setting CSIHnCTL0.CSIHnJOBE, communication is forcibly stopped when job 3 ends.
9. After communication is forcibly stopped, the interrupt requests CSIHnTIJC and CSIHnTIR are generated when job 3 ends.
   The interrupt request CSIHnTIJC indicates that communication was forcibly stopped when the current job ended.
   Because the interrupt request CSIHnTIJC is generated instead of the interrupt request CSIHnTIC, the interrupt request CSIHnTIC is not generated.
10. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(6) For reception in master mode, and when job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

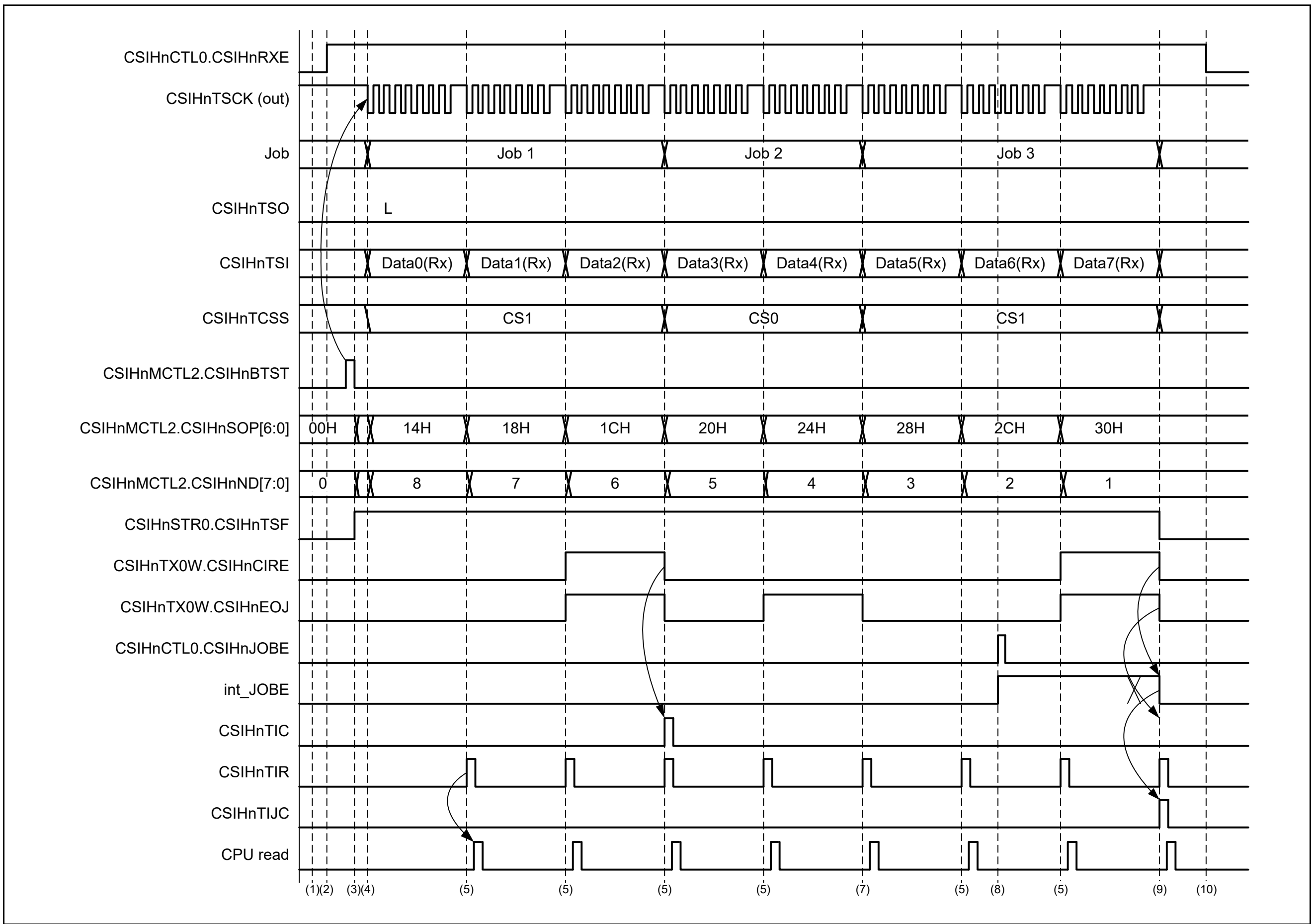


Figure 23.49 Transmit-Only Buffer Mode (for Reception in Master Mode, and when Job Mode is Enabled)

Remarks 1. The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

2. The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

• Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0, and CS1 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Reception starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data item is transmitted.
5. Each time a data item is received, a CSIHnTIR interrupt request is generated.
   CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
6. CSIHnTIC is generated by setting CSIHnTX0W.CSIHnCIRE to 1.
   CSIHnTIC indicates that the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted.
7. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by clearing CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is not generated.
8. By setting CSIHnCTL0.CSIHnJOBE, reception is forcibly stopped when job 3 ends.
9. After communication is forcibly stopped, the interrupt requests CSIHnTIJC and CSIHnTIR are generated when job 3 ends.
   The interrupt request CSIHnTIJC indicates that reception was forcibly stopped when the current job ended.
   Because the interrupt request CSIHnTIJC is generated instead of the interrupt request CSIHnTIC, the interrupt request CSIHnTIC is not generated.
10. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

23.5.3 Procedures in Dual Buffer Mode

(1) For transmission/reception in master mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01B)
- Number of data packets: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

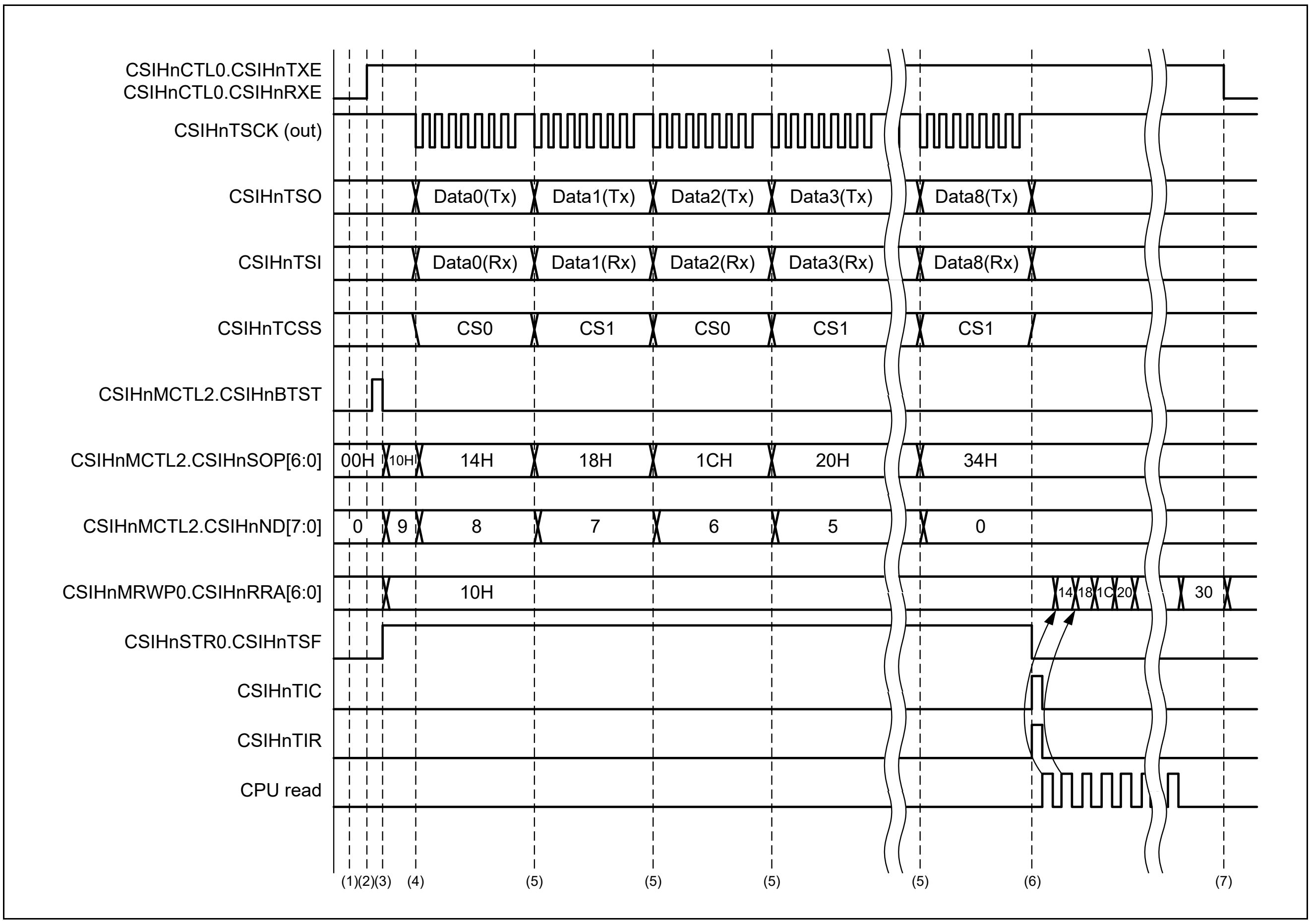


Figure 23.50 Dual Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)

Remark: The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 01B (memory mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0 and CS1 are used.)
     CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
     Communication is started by setting CSIHnMCTL2.CSIHnBTST.
  4. Transmission starts. Each time a data item is transmitted, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
  5. (4) is repeated until the last data is transmitted/received.
     The interrupt requests CSIHnTIC and CSIHnTIR are not generated.
  6. When all the communication ends, the interrupt requests CSIHnTIC and CSIHnTIR are generated.
     The CPU starts reading the received data from the reception buffer. The read start address is specified by the CSIHnMRWP0.CSIHnRRA[6:0] bits.
     The CSIHnMRWP0.CSIHnRRA[6:0] bits are incremented each time a data item is read.
  7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(2) For reception in master mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01B)
- Number of data packets: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

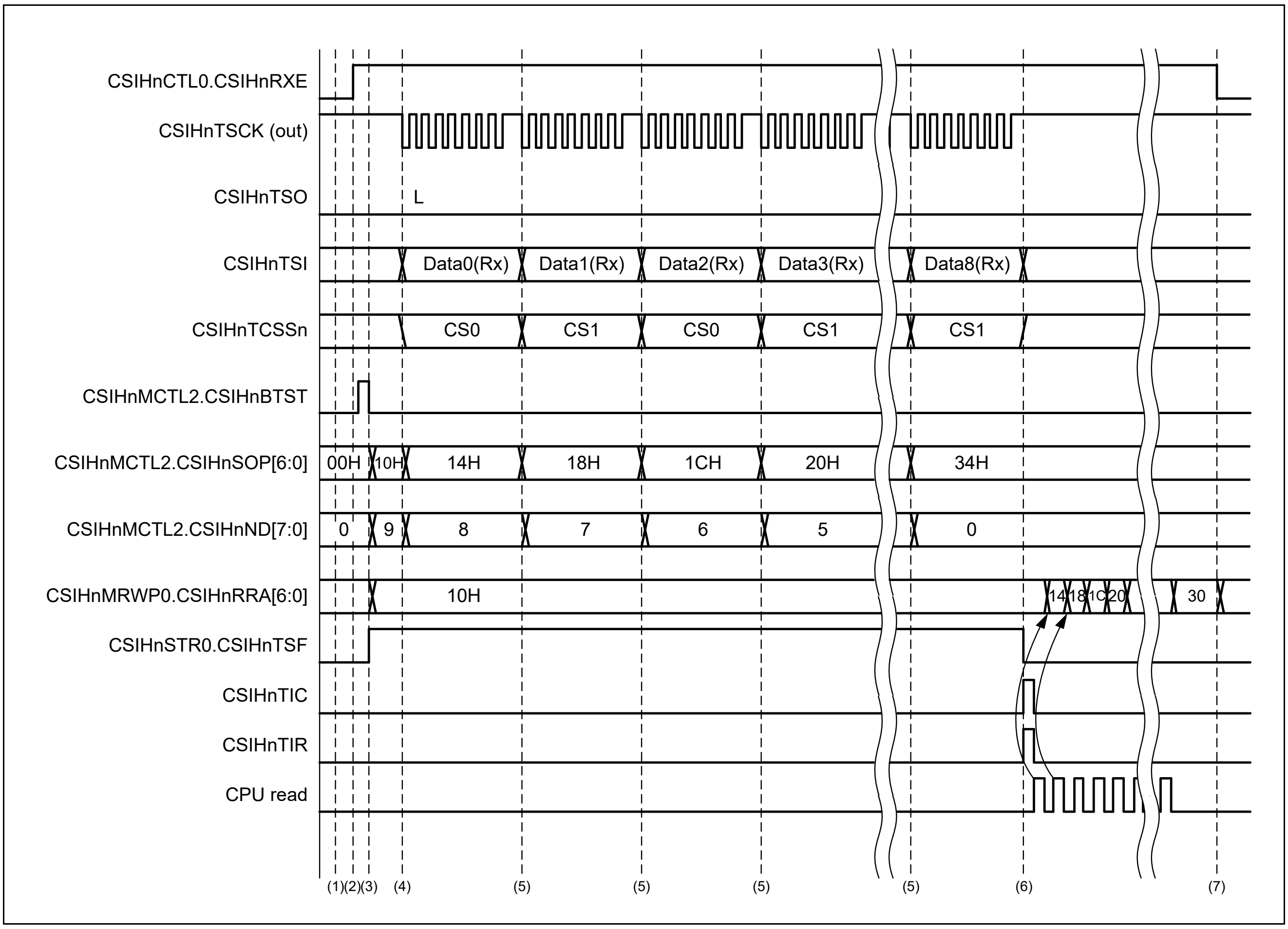


Figure 23.51 Dual Buffer Mode (for Reception in Master Mode, and when Job Mode is Disabled)

Remark: The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 01B (memory mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0 and CS1 are used.)
     CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
     Reception is started by setting CSIHnMCTL2.CSIHnBTST.
  4. Reception starts. Each time a data item is received, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
  5. (4) is repeated until the last data is received.
     The interrupt requests CSIHnTIC and CSIHnTIR are not generated.
  6. When all the reception ends, the interrupt requests CSIHnTIC and CSIHnTIR are generated.
     The CPU starts reading the received data from the reception buffer. The read start address is specified by the CSIHnMRWP0.CSIHnRRA[6:0] bits.
     The CSIHnMRWP0.CSIHnRRA[6:0] bits are incremented each time a data item is read.
  7. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
     In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(3) For transmission/reception in slave mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01B)
- Number of data packets: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

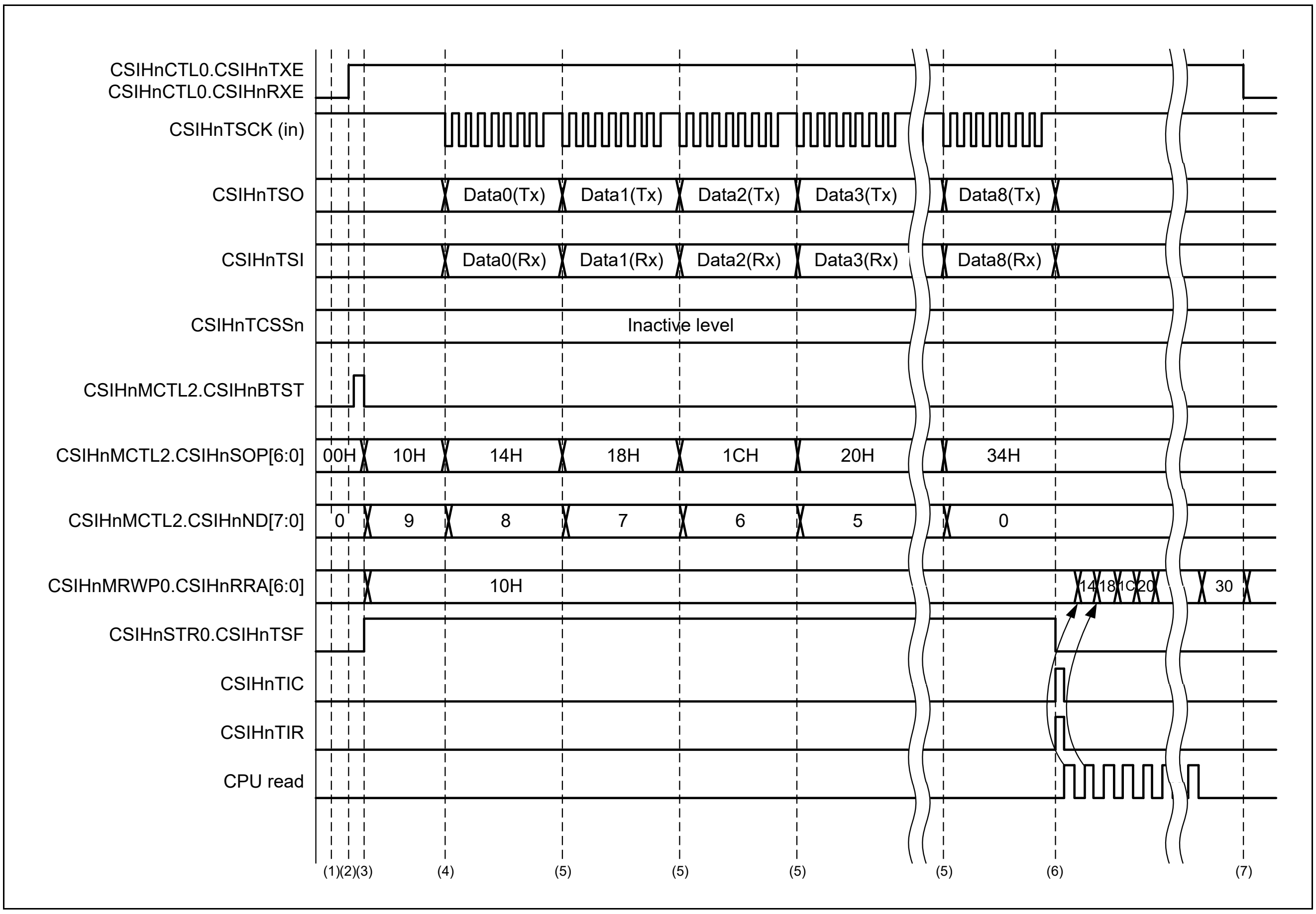


Figure 23.52 Dual Buffer Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)

Remark: The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 01B (memory mode)
     CSIHnCFG0 (communication protocol)
     CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
  2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
     Communication is started by setting CSIHnMCTL2.CSIHnBTST.
  4. Transmission starts. Each time a data item is transmitted, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
  5. (4) is repeated until the last data is transmitted/received.
     The interrupt requests CSIHnTIC and CSIHnTIR are not generated.
  6. When all the communication ends, the interrupt requests CSIHnTIC and CSIHnTIR are generated.
     The CPU starts reading the received data from the reception buffer. The read start address is specified by the CSIHnMRWP0.CSIHnRRA[6:0] bits.
     The CSIHnMRWP0.CSIHnRRA[6:0] bits are incremented each time a data item is read.
  7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(4) For reception in slave mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01B)
- Number of data packets: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09H)
- Transfer start address: 10H (CSIHnMCTL2.CSIHnSOP[6:0] = 10H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

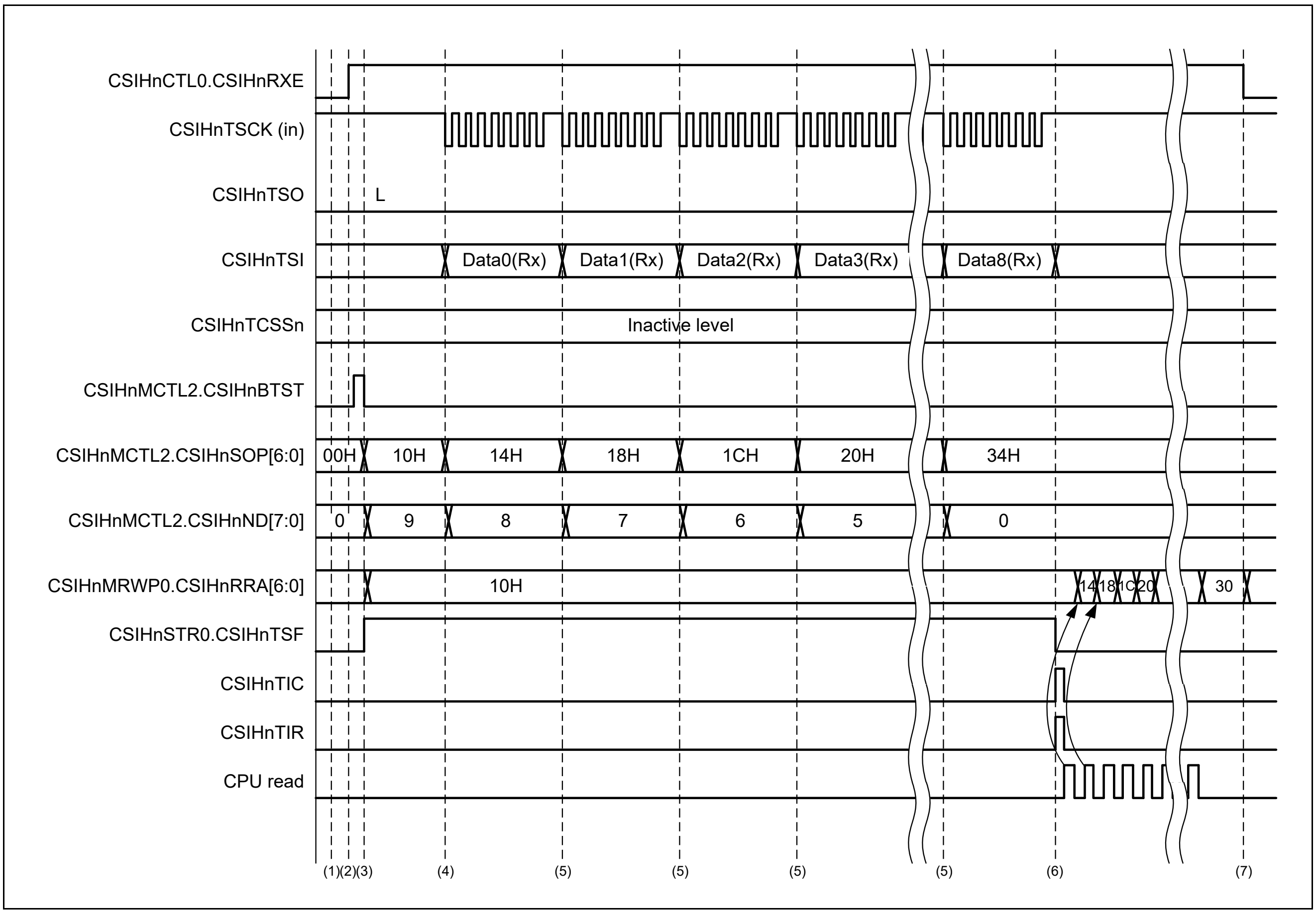


Figure 23.53 Dual Buffer Mode (for Reception in Slave Mode, and when Job Mode is Disabled)

Remark: The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

- Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01B (memory mode)
   CSIHnCFG0 (communication protocol)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Reception is started by setting CSIHnMCTL2.CSIHnBTST.
4. Reception starts. Each time a data item is received, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. (4) is repeated until the last data is received.
   The interrupt requests CSIHnTIC and CSIHnTIR are not generated.
6. When all the reception ends, the interrupt requests CSIHnTIC and CSIHnTIR are generated.
   The CPU starts reading the received data from the reception buffer. The read start address is specified by the CSIHnMRWP0.CSIHnRRA[6:0] bits.
   The CSIHnMRWP0.CSIHnRRA[6:0] bits are incremented each time a data item is read.
7. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
   In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(5) For transmission/reception in master mode, and when job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01B)
- Number of data packets: 12 (CSIHnMCTL2.CSIHnND[7:0] = 12H)
- Transfer start address: 00H (CSIHnMCTL2.CSIHnSOP[6:0] = 00H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

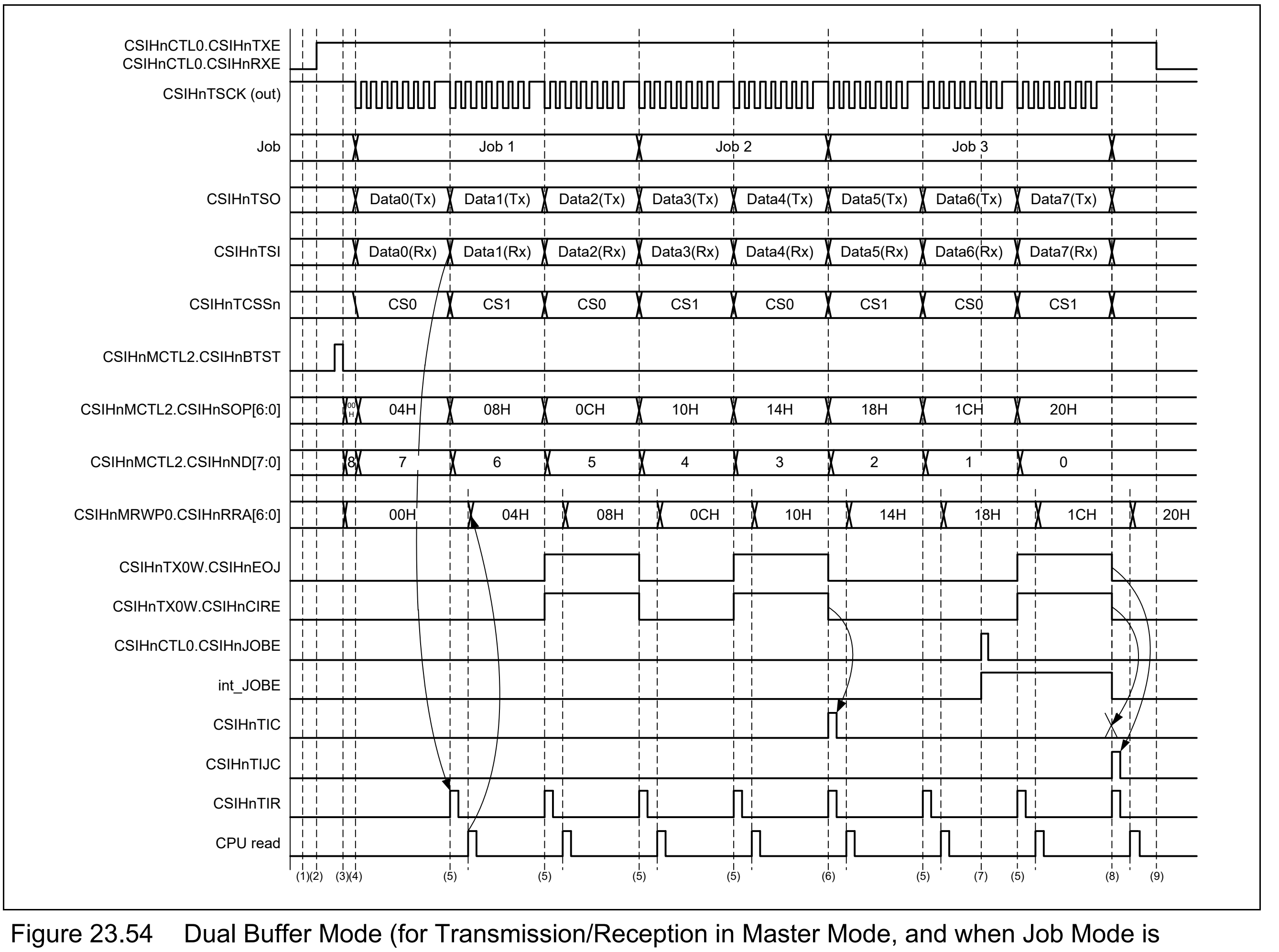


Figure 23.54 Dual Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)

Remarks 1. The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

2. The int_JOBE signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOBE bit.

• Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 and CS1 are used.)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission starts. Each time a data item is transmitted, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. When all the data are received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. CSIHnTIC is generated by setting CSIHnTX0W.CSIHnCIRE to 1.
   CSIHnTIC indicates that the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted.
7. By setting CSIHnCTL0.CSIHnJOBE to 1, communication is forcibly stopped when job 3 ends.
8. After communication is forcibly stopped, the interrupt requests CSIHnTIJC and CSIHnTIR are generated when job 3 ends.
   The interrupt request CSIHnTIJC indicates that communication was forcibly stopped when the current job ended.
   Because the interrupt request CSIHnTIJC is generated instead of the interrupt request CSIHnTIC, the interrupt request CSIHnTIC is not generated. Note that transfer data is not transmitted by the CSIHnTX0W register.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations.
   In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(6) For reception in master mode, and when job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01B)
- Number of data packets: 12 (CSIHnMCTL2.CSIHnND[7:0] = 12H)
- Transfer start address: 00H (CSIHnMCTL2.CSIHnSOP[6:0] = 00H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

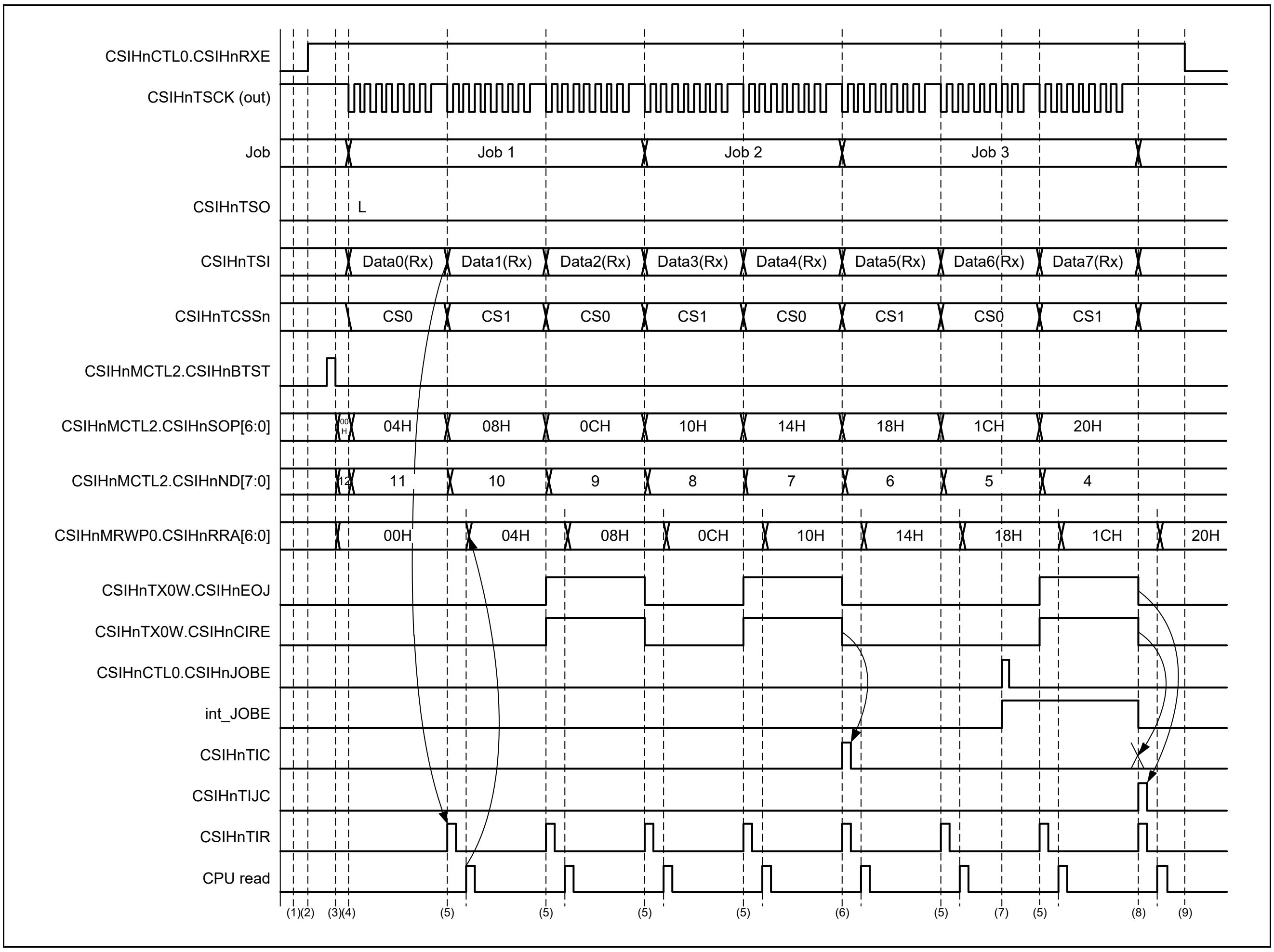


Figure 23.55 Dual Buffer Mode (for Reception in Master Mode, and when Job Mode is Enabled)

Remarks 1. The procedure for writing data to the buffer is not described here. The first data address is specified by CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

2. The int_JOBE signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOBE bit.

• Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 and CS1 are used.)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Reception starts. Each time a data item is received, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. Each time data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. CSIHnTIC is generated by setting CSIHnTX0W.CSIHnCIRE to 1.
   CSIHnTIC indicates that the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted.
7. By setting CSIHnCTL0.CSIHnJOBE to 1, reception is forcibly stopped when job 3 ends.
8. After reception is forcibly stopped, the interrupt requests CSIHnTIJC and CSIHnTIR are generated when job 3 ends.
   The interrupt request CSIHnTIJC indicates that reception was forcibly stopped when the current job ended.
   Because the interrupt request CSIHnTIJC is generated instead of the interrupt request CSIHnTIC, the interrupt request CSIHnTIC is not generated. Note that transfer data is not transmitted by the CSIHnTX0W register.
9. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
   In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

23.5.4 Procedures in FIFO Mode

(1) For transmission/reception in master mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

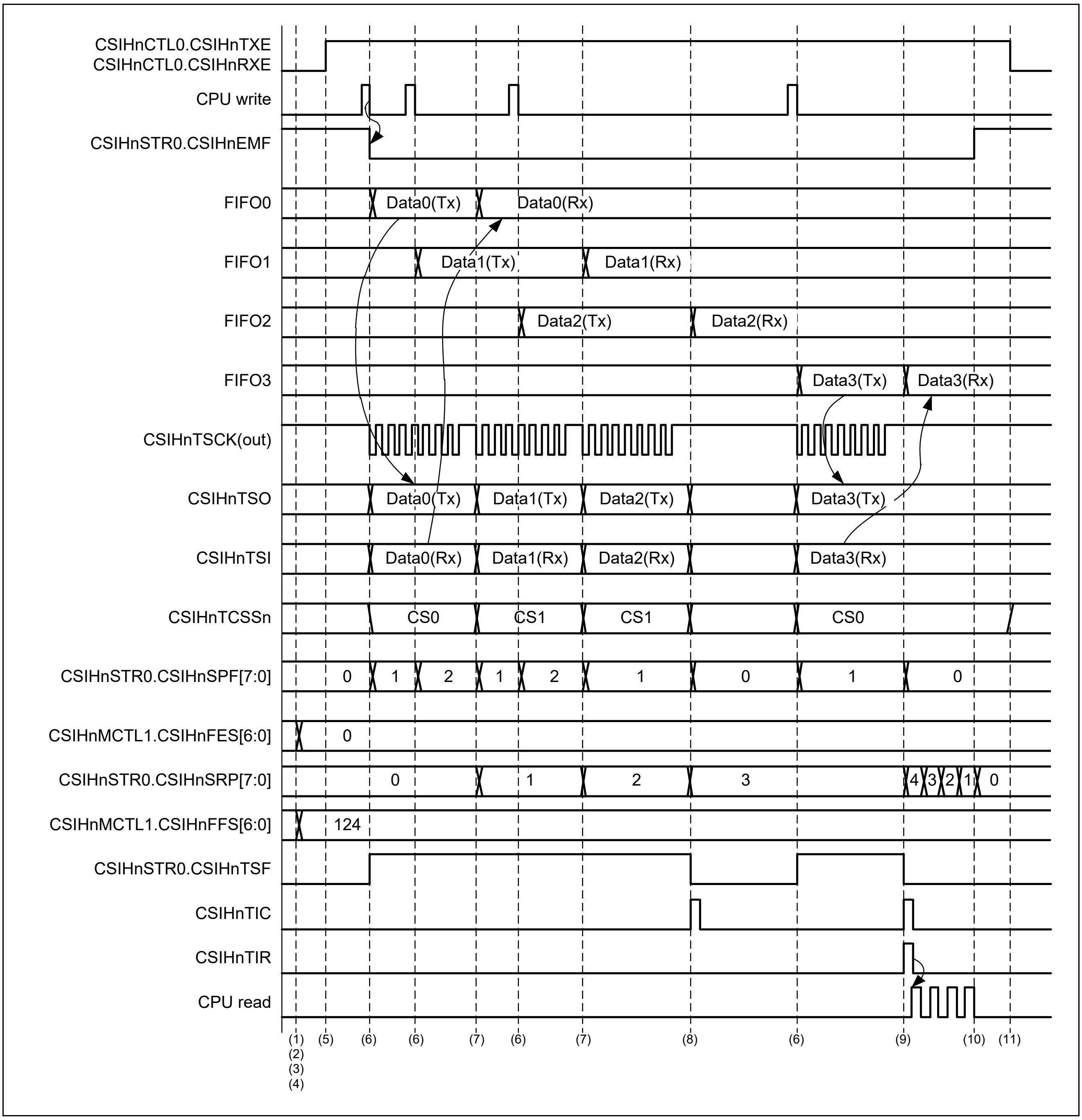


Figure 23.56 FIFO Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 00B (memory mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0 and CS1 are used.)
  2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
  3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00H.
  4. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
     Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
  5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  6. When transmission data is written to the transmission data register CSIHnTX0W, communication starts.
  7. Some of the communication finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
  8. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
  9. The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 - CSIHnSTR0.CSIHnSRP[7:0]) match.
     The interrupt request CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
     The CPU starts reading the received data stored in the reception buffer.
  10. The CPU finishes reading the received data. The CSIHnSTR0.CSIHnEMF bit is set because the FIFO buffer is empty.
  11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(2) For reception in master mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

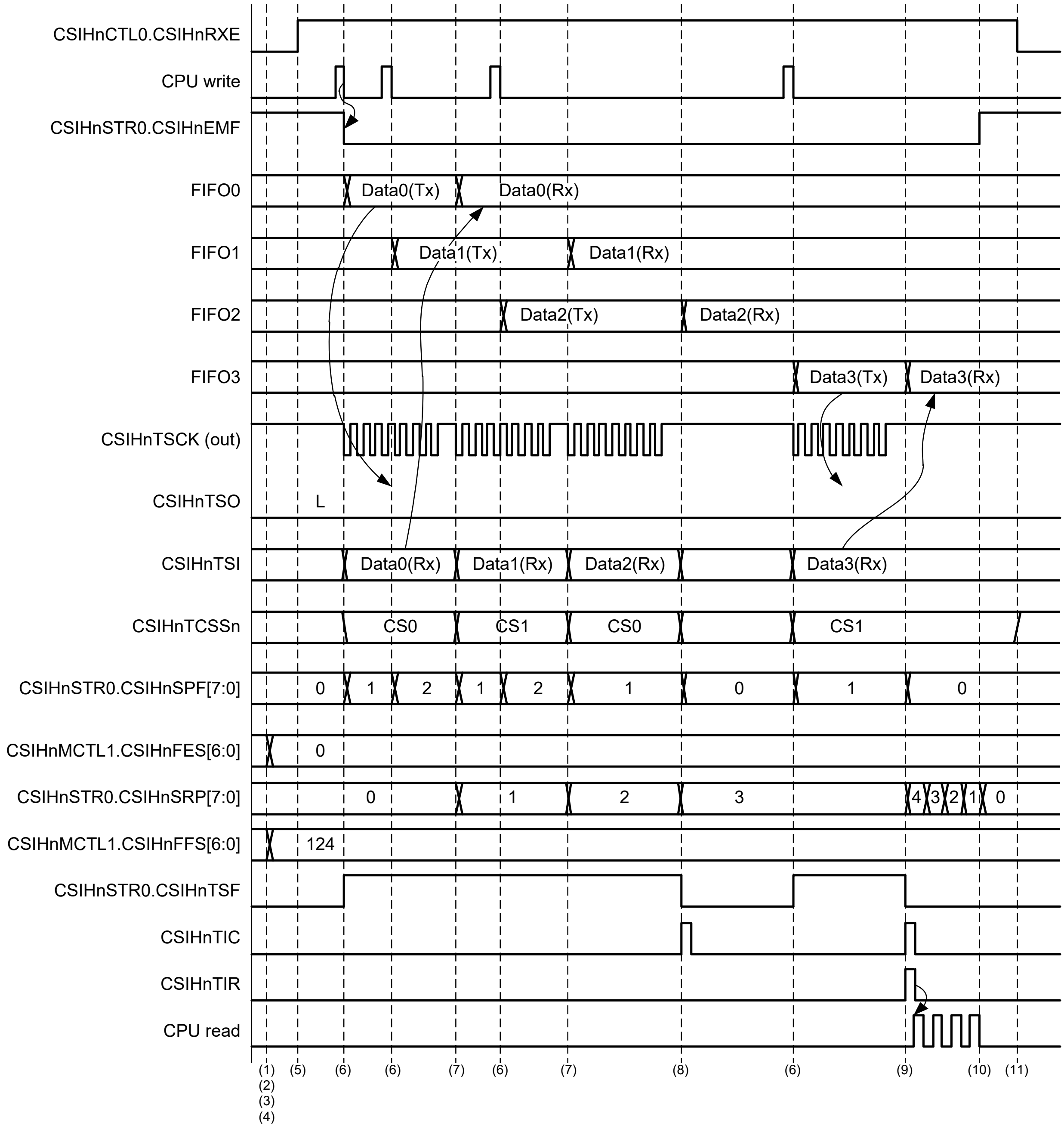


Figure 23.57 FIFO Mode (for Reception in Master Mode, and when Job Mode is Disabled)

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 00B (memory mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0 and CS1 are used.)
  2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
  3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00H.
  4. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
     Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
  5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  6. When transmission data is written to the transmission data register CSIHnTX0W, communication starts. (The transmission data is not used, but the chip select signal is enabled.)
  7. Some of the communication finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
  8. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
  9. The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 - CSIHnSTR0.CSIHnSRP[7:0]) match.
     The interrupt request CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
     The CPU starts reading the received data stored in the reception buffer.
  10. The CPU finishes reading the received data. The CSIHnSTR0.CSIHnEMF bit is set because the FIFO buffer is empty.
  11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable reception operations.
      In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(3) For transmission/reception in slave mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

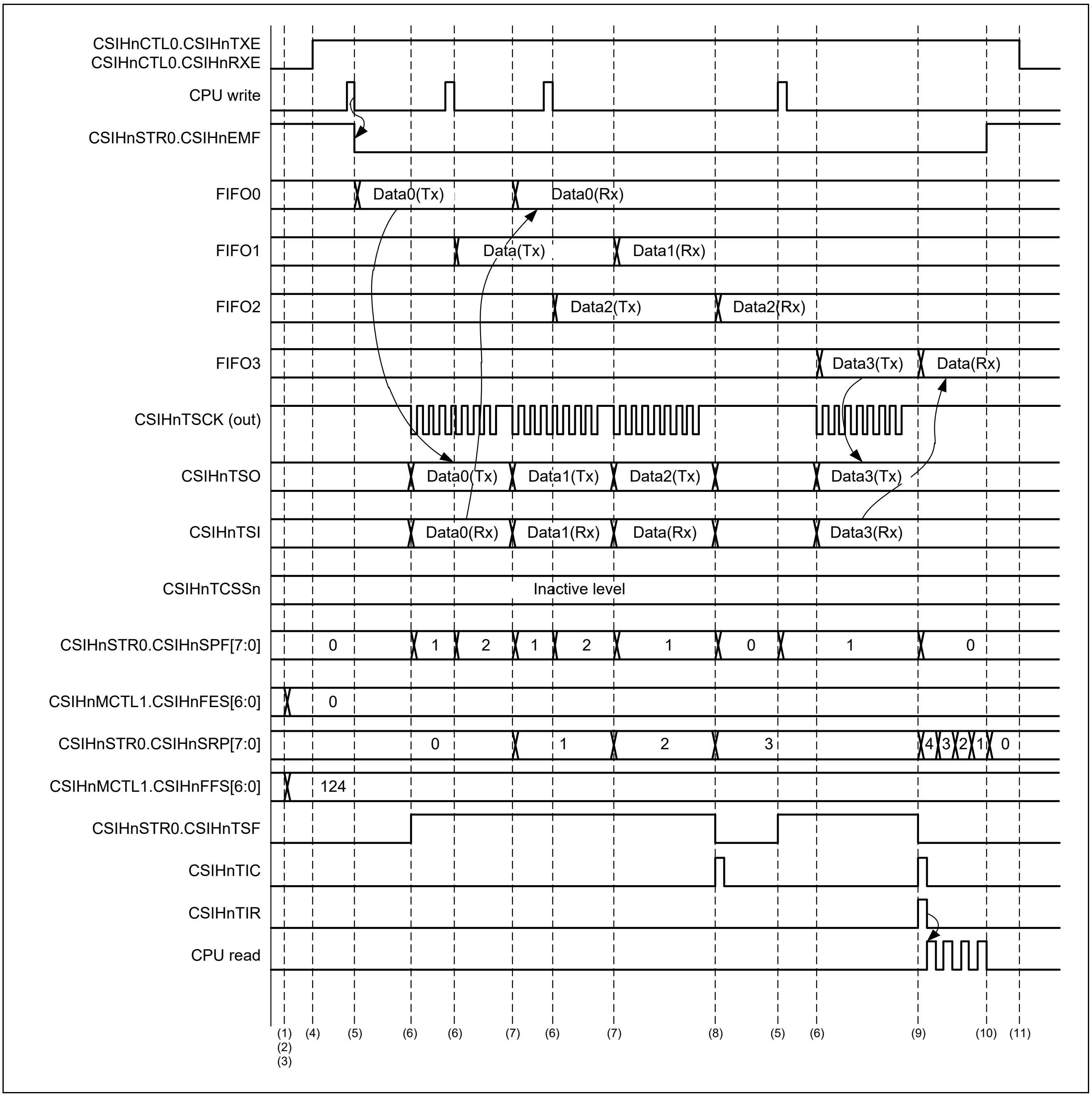


Figure 23.58 FIFO Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)

- Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 00B (memory mode)
   CSIHnCFG0 (communication protocol)
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
   Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00H.
3. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
   Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
4. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
5. Write the transfer data to the transmission data register CSIHnTX0W.
6. When a serial clock is supplied from the master, communication automatically starts.
7. Some of the communication finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
8. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
9. The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 - CSIHnSTR0.CSIHnSRP[7:0]) match.
   The interrupt request CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
   The CPU starts reading the received data stored in the reception buffer.
10. The CPU finishes reading the received data. The CSIHnSTR0.CSIHnEMF bit is set because the FIFO buffer is empty.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(4) For reception in slave mode, and when job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

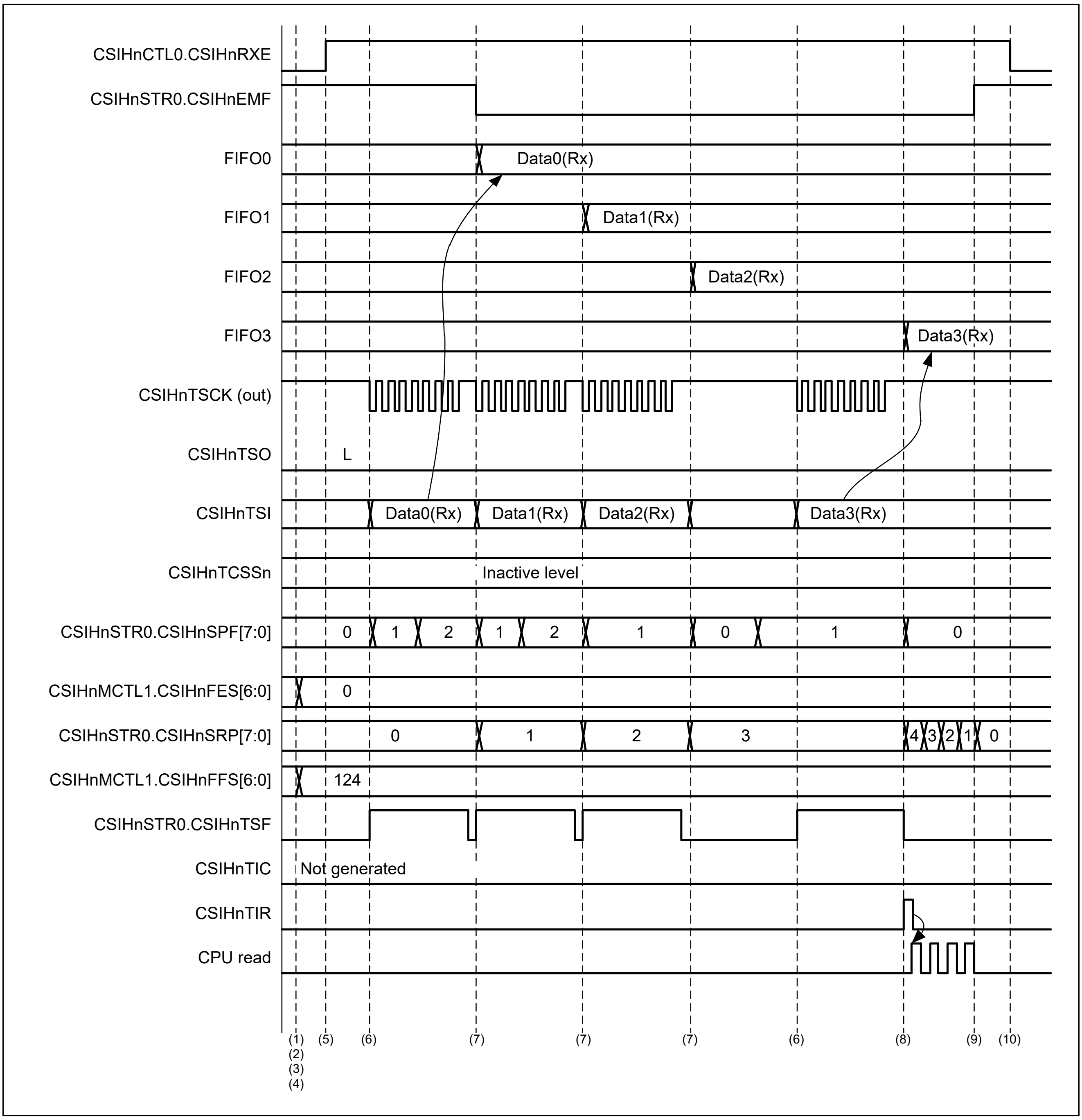


Figure 23.59 FIFO Mode (for Reception in Slave Mode, and when Job Mode is Disabled)

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 00B (memory mode)
     CSIHnCFG0 (communication protocol)
  2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
  3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00H.
  4. Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
  5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  6. When a serial clock is supplied from the master, reception automatically starts.
  7. Some of the communication finishes, but CSIHnTIC is not generated because the system is in reception mode.
  8. The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 - CSIHnSTR0.CSIHnSRP[7:0]) match.
     The CPU starts reading the received data stored in the reception buffer.
  9. The CPU finishes reading the received data. The CSIHnSTR0.CSIHnEMF bit is set because the FIFO buffer is empty.
  10. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
      In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(5) For transmission/reception in master mode, and when job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Job 1 = four data items, job 2 = three data items, and job 3 = five data items

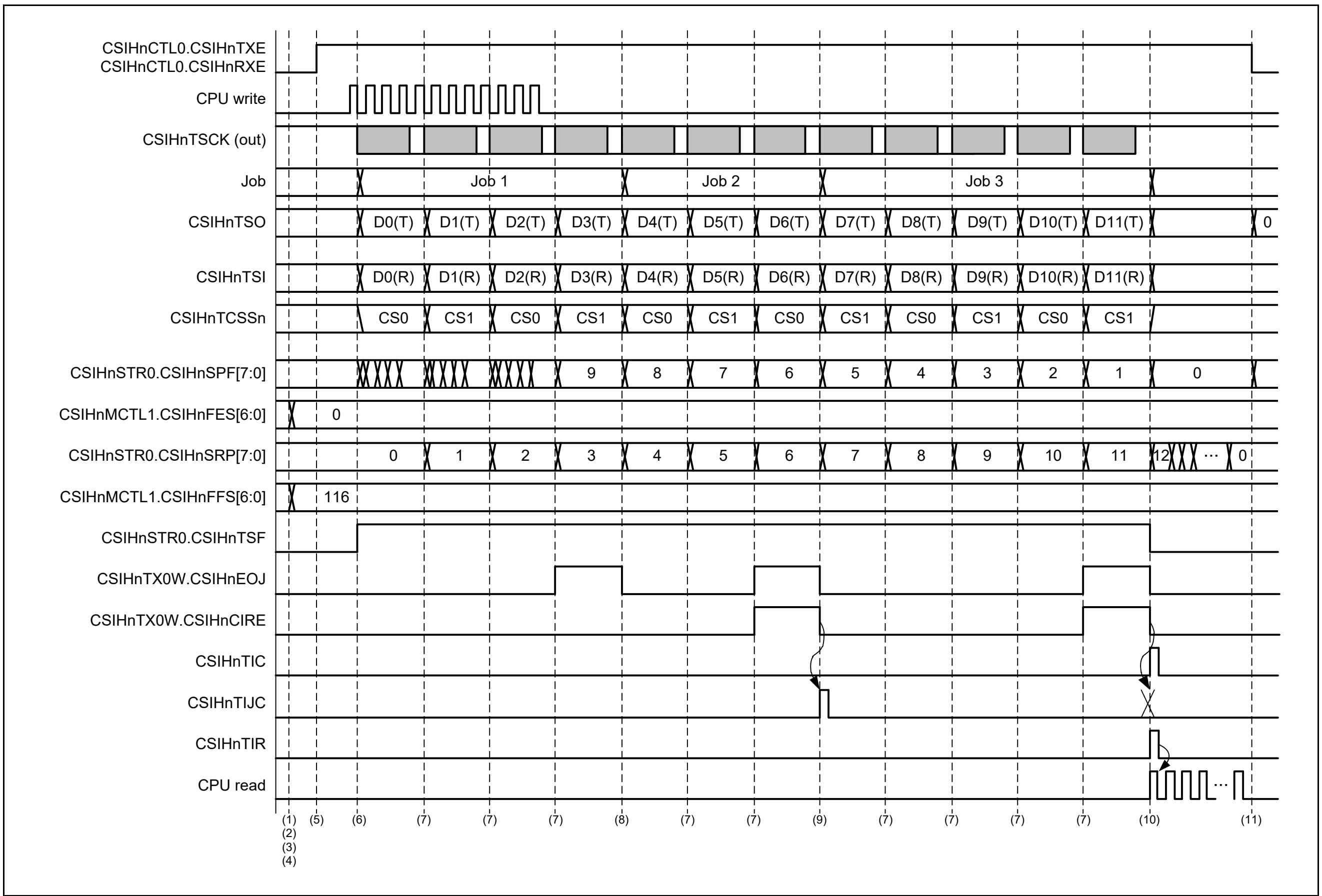


Figure 23.60 FIFO Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)

Remark: The int_JOBE signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOBE bit.

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 00B (memory mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0 and CS1 are used.)
  2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
  3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00H.
  4. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
     Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
  5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  6. When transmission data is written to the transmission data register CSIHnTX0W, communication starts.
  7. Some of the communication finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
  8. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by clearing CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is not generated.
  9. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by setting CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is generated.
  10. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
      Because CSIHnTIC was generated, CSIHnTIJC is not generated.
      The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 - CSIHnSTR0.CSIHnSRP[7:0]) match.
      The CPU starts reading the received data stored in the reception buffer.
  11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations.
      In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(6) For reception in master mode, and when job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Job 1 = four data items, job 2 = three data items, and job 3 = five data items

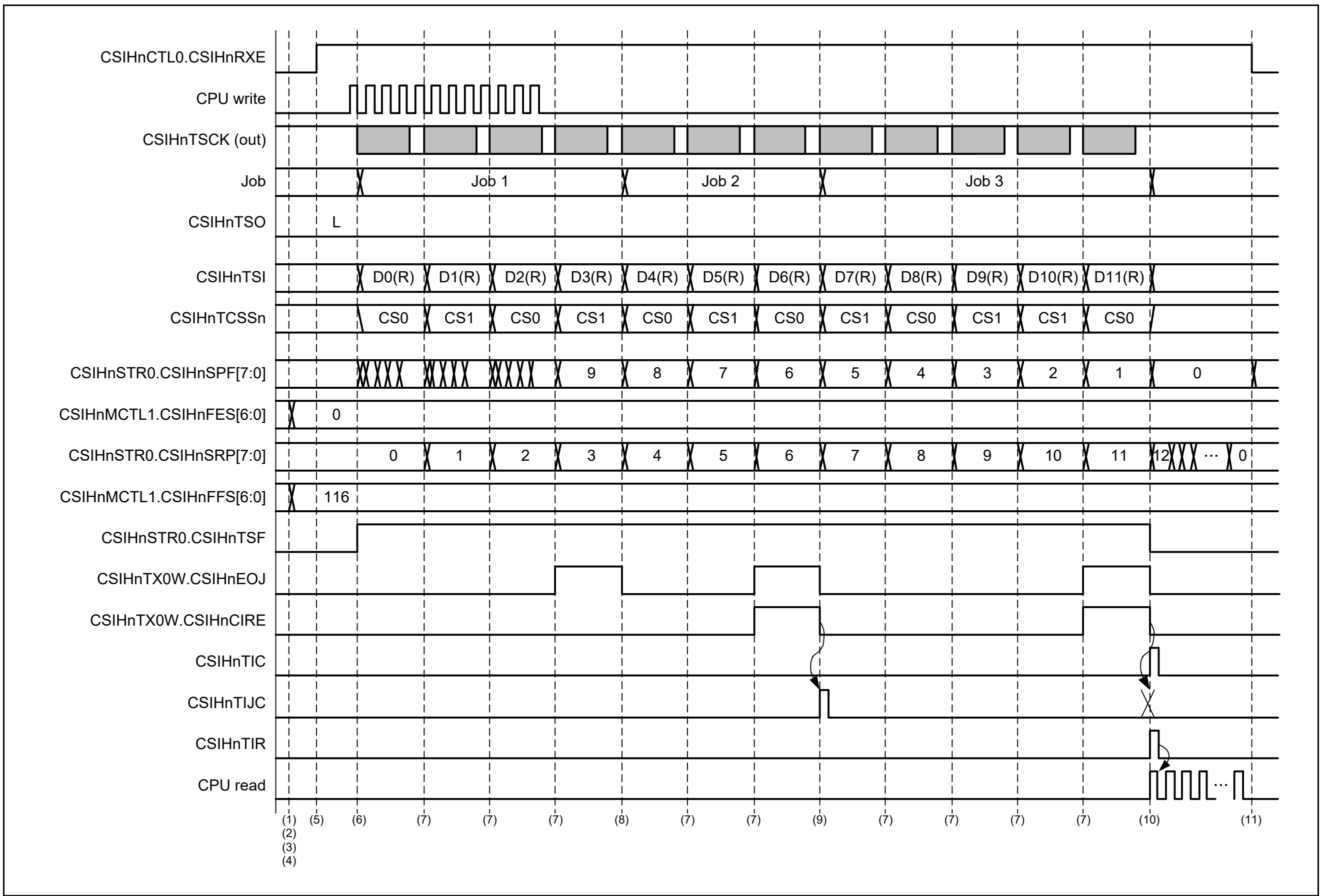


Figure 23.61 FIFO Mode (for Reception in Master Mode, and when Job Mode is Enabled)

Remark: The int_JOBE signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOBE bit.

- Procedure:
  1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
     CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
     CSIHnMCTL0.CSIHnMMS[1:0] = 00B (memory mode)
     CSIHnCFGx (communication protocol)
     (For this example, the chip select signals CS0 and CS1 are used.)
  2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
  3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00H.
  4. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
     Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
  5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
     CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
     CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
     CSIHnCTL0.CSIHnMBS = 0 (memory mode)
  6. When transmission data is written to the transmission data register CSIHnTX0W, communication starts. (The transmission data is not used, but the chip select signal is enabled.)
  7. Some of the reception finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
  8. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by clearing CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is not generated.
  9. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by setting CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is generated.
  10. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
      Because CSIHnTIC was generated, CSIHnTIJC is not generated.
      The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 - CSIHnSTR0.CSIHnSRP[7:0]) match.
      The CPU starts reading the received data stored in the reception buffer.
  11. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
      In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

24. I²C BUS (IICB)

This section describes the I²C bus (IICB).

24.1 Features of IICB

- Number of channels: R-IN32M4-CL3 products incorporate two channels of I²CB (IICBn).

Table 24.1 Channels of I²CB

IICB	
Number of channels	2
Names	IICB0, IICB1

- Index n: Throughout this section, the individual channels of the IICB are identified by the index "n" (n = 0, 1); for example, IICBnDAT for the IICBn data register.

- Interrupts and peripheral modules:
  The following interrupt requests from IICB can be used as triggers for interrupt service routines or hardware ISRs (where listed as such), for DMA transfer (by the general-purpose DMAC or real-time port DMAC), for capture by a timer (TAUJ2 or TAUD), and for updating the real-time port pins (RP00–RP37).

Table 24.2 IICBn Interrupts and Requests for Peripheral Modules

Interrupt Request Signal	Function	Connected to:
IICB0		
INTIICB0TIA	Data transmit/receive interrupt request signal	• Interrupt controller INTIICB0TIA • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-timer port trigger (RPTFR)
INTIICB0TIS	Status interrupt request signal	• Interrupt controller INTIICB0TIS
IICB1		
INTIICB1TIA	Data transmit/receive interrupt request signal	• Interrupt controller INTIICB1TIA • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-timer port trigger (RPTFR)
INTIICB1TIS	Status interrupt request signal	• Interrupt controller INTIICB1TIS

24.2 Functional Overview

- Operating mode: Standard mode (SCL clock frequency: Max. 100 kHz)
  Fast mode (SCL clock frequency: Max. 400 kHz)
- Transfer mode: Single transfer mode
  Continuous transfer mode
- Pin configuration: SCLn: Serial clock pin
  SDAn: Serial transmit/receive data pin
- Interrupt request signal: Data transmit/receive interrupt request signal (INTIICBnTIA)
  Status interrupt request signal (INTIICBnTIS)
- Communication data length: 8 bits
- Multimaster support: Multiple masters can control the bus simultaneously.
- SCLn level width: The high-level width and low-level width of the serial clock signal (SCLn) can be changed.
- Automatic detection: The start and stop conditions can be detected automatically.

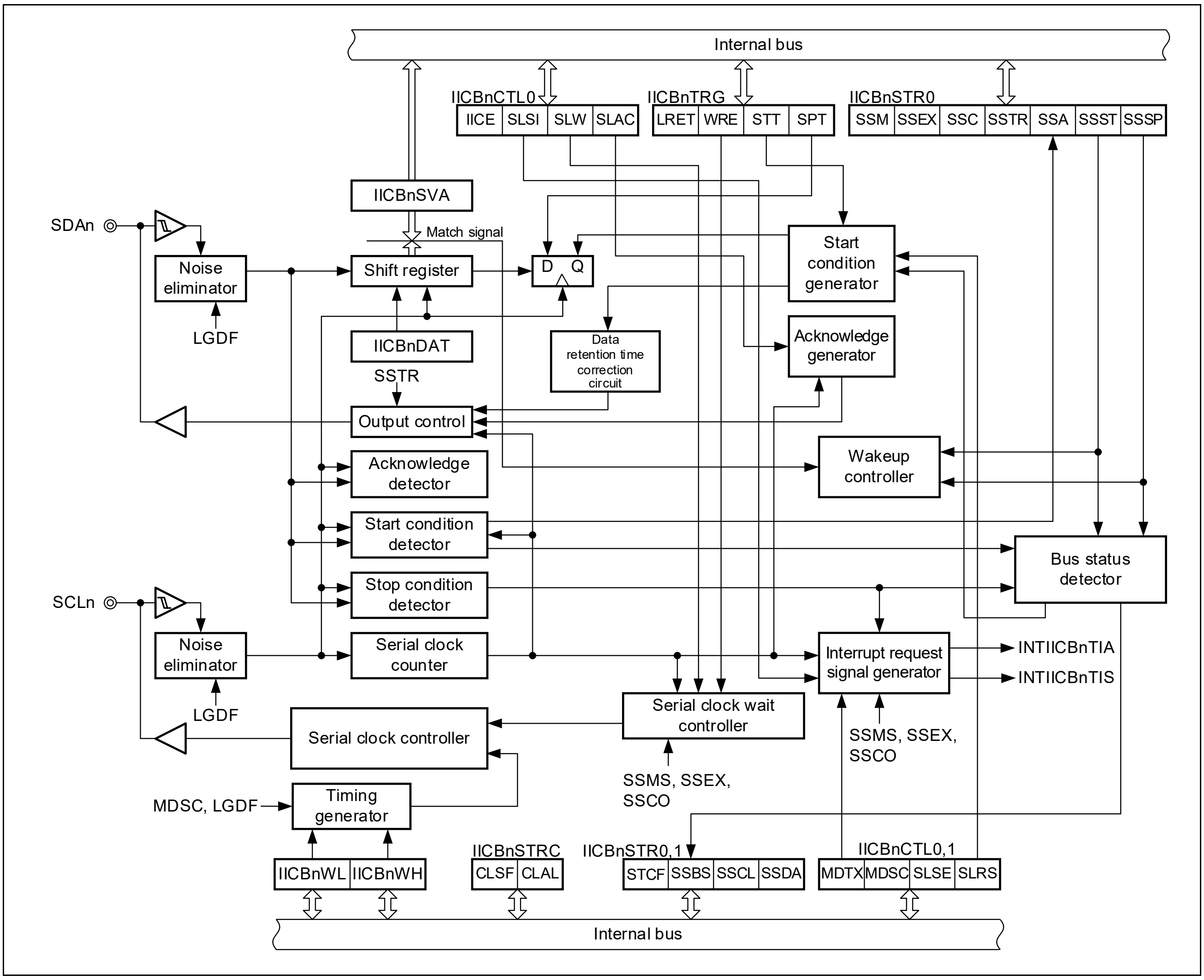


Figure 24.1 Block Diagram of IICBn

24.3 Registers

Caution: In this section, the description of operation when an extension code is received is omitted. For details about the extension code, refer to section 24.6.5, Extension Code.

I2Cn is controlled and operated by means of the following registers:

Table 24.3 I^{2}C Register

Register	Symbol	Address
IICB0 data register	IICB0DAT	4000 0500H
IICB0 slave address register	IICB0SVA	4000 0504H
IICB0 control register 0	IICB0CTL0	4000 0508H
IICB0 control register 1	IICB0CTL1	4000 0520H
IICB0 low level width setting register	IICB0WL	4000 0524H
IICB0 high-level width setting register	IICB0WH	4000 0528H
IICB0 trigger register	IICB0TRG	4000 050CH
IICB0 status register 0	IICB0STR0	4000 0510H
IICB0 status register 1	IICB0STR1	4000 0514H
IICB0 status clear register	IICB0STRC	4000 0518H
IICB1 data register	IICB1DAT	4000 0600H
IICB1 slave address register	IICB1SVA	4000 0604H
IICB1 control register 0	IICB1CTL0	4000 0608H
IICB1 control register 1	IICB1CTL1	4000 0620H
IICB1 low level width setting register	IICB1WL	4000 0624H
IICB1 high-level width setting register	IICB1WH	4000 0628H
IICB1 trigger register	IICB1TRG	4000 060CH
IICB1 status register 0	IICB1STR0	4000 0610H
IICB1 status register 1	IICB1STR1	4000 0614H
IICB1 status clear register	IICB1STRC	4000 0618H

(1) IICBn data register (IICBnDAT)

This register is used to transmit and receive transfer data.

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

This register is also initialized by changing the value of the IICBnCTL0.IICBnIICE bit from 1 to 0 or from 0 to 1.

Cautions 1. When the IICBn becomes a master in single transfer mode or continuous transfer mode, after the IICBnTRG.IICBnSTT bit has been set to 1, writing to the IICBnDAT register is allowed only once to transfer the address and communication direction.

2. When transferring data in single transfer mode, writing to the IICBnDAT register in communication state other than the wait state is prohibited.

3. When transferring data in continuous transfer mode, writing to the IICBnDAT register in response to an INTIICBnTIA interrupt request signal is only allowed once.

4. When executing transmission operations in continuous transfer mode, do not read the IICBnDAT register.

Similarly, when performing reception operations in continuous transfer mode, do not write to the IICBnDAT register.

	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnDAT	IICBn DAT7	IICBn DAT6	IICBn DAT5	IICBn DAT4	IICBn DAT3	IICBn DAT2	IICBn DAT1	IICBn DAT0	4000 0500H +100H × n	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 0	IICBnDAT7–IICBnDAT0	During reception, these bits hold the received data. During transmission, these bits write the transmit data. The prescribed procedure must be followed during access (read, write) to the IICBnDAT register. For the setting procedure, refer to section 24.9, Setting Procedure. The IICBn exits the wait state by performing access to the IICBnDAT register. • In single transfer mode - When write access to the IICBnDAT register is performed • In continuous transfer mode - When write access to the IICBnDAT register is performed - When read access to the IICBnDAT register is performed during a wait state for data transfer that is not triggered by NACK signal reception

Remark: n = 0, 1

(2) IICBn slave address register (IICBnSVA)

This register stores the slave address of the IICBn bus.

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

Caution: Write access to the IICBnSVA register is prohibited when the value of the IICBnCTL0.IICBnIICE bit is 1.

	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnSVA	IICBn SVA7	IICBn SVA6	IICBn SVA5	IICBn SVA4	IICBn SVA3	IICBn SVA2	IICBn SVA1	0	4000 0504H +100H × n	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0		

Bit Position	Bit Name	Function
7 to 1	IICBnSVA7– IICBnSVA1	Store the slave address of the IICBn bus. Address match/address mismatch is judged by comparing the received address and the IICBnSVA register. If the received address matches the IICBnSVA register, the IICBnSTR0.IICBnSSCO bit is set to 1.
0	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.

Remark: n = 0, 1

(3) IICBn control register 0 (IICBnCTL0)

This register is used to control the operations of the IICBn.

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

(1/3)

	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnCTL0	IICBn IICE	0	0	IICBn MDTX1	IICBn MDTX0	IICBn SLSI	IICBn SLWT	IICBn SLAC	4000 0508H +100H × n	00H
R/W	R/W	0	0	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7	IICBnIICE	Enables/disables operation of the IICBn. 0: Disables operation of IICBn. 1: Enables operation of IICBn. Synchronous reset of the following registers is executed when the value of the IICBnCTL0.IICBnIICE bit changes from 1 to 0, or the value of the IICBnCTL0.IICBnIICE bit changes from 0 to 1. • IICBnDAT and IICBnSTR0 registers When IICBnCTL0.IICBnIICE is 0, the SCLn and SDAn pins go into the high impedance state.
6, 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4	IICBnMDTX1	Specifies the transfer mode upon detection of expansion code in the slave. 0: Single transfer mode 1: Continuous transfer mode • Single transfer mode The IICBn enters a wait state after each transfer according to the setting of the IICBnCTL0.IICBnSLWT bit. • Continuous transfer mode The IICBn performs continuous communication without entering a wait state when the IICBnDAT register is read or written upon the output of the data transmit/receive interrupt request signal (INTIICBnTIA). For the operation in each mode, refer to section 24.6, Operation. **Caution: Changing the value of this bit is only allowed while IICBnCTL0.IICBnIICE is 0.**

Remark: n = 0, 1

(2/4)

Bit Position	Bit Name	Function
3	IICBnMDTX0	Specifies the transfer mode when the address matches between the master and slave. 0: Single transfer mode 1: Continuous transfer mode • Single transfer mode The IICBn enters a wait state after each transfer according to the setting of the IICBnCTL0.IICBnSLWT bit. • Continuous transfer mode The IICBn performs continuous communication without entering a wait state when the IICBnDAT register is read or written upon the output of the data transmit/receive interrupt request signal (INTIICBnTIA). For the operation in each mode, refer to section 24.6, Operation. **Caution: Changing the value of this bit is only allowed while IICBnCTL0.IICBnIICE is 0.**
2	IICBnSLSI	Enables/disables status interrupt request signal (INTIICBnTIS) output when a stop condition is detected. 0: Disables INTIICBnTIS signal output when a stop condition is detected. 1: Enables INTIICBnTIS signal output when a stop condition is detected. Set this bit to 1 when performing the following types of communication. - When the IICBn performs communication as a master while the communication reserve function is enabled - When the IICBn participates in communications as a slave - When the IICBn may lose in arbitration (when making the IICBn operate as a master in a multi-master environment))

Remark: n = 0, 1

(3/4)

Bit Position	Bit Name	Function
1	IICBnSLWT	Controls a wait and interrupt request output timing. 0: The IICBn enters the wait state and an interrupt request is output at the falling edge of the 8th clock during single transfer. 1: The IICBn enters the wait state and an interrupt request is output at the falling edge of the 9th clock during single transfer. The IICBnCTL0.IICBnSLWT bit controls wait state transition and interrupt request output at the following timing. - 8th and 9th clocks during data transfer For the conditions for transition to the wait state, refer to section 24.6.4, Entering and Exiting Wait State. During address transfer, the conditions for transiting to the wait state and for interrupt request output are as follows, regardless of the setting of the IICBnCTL0.IICBnSLWT bit. • In single transfer mode - Master: A data transmit/receive interrupt request signal (INTIICBnTIA) is output and the IICBn enters the wait state upon detection of the falling edge of the 9th clock. - Slave: When the address matches, the INTIICBnTIA signal is output and the IICBn enters the wait state upon detection of the falling edge of 9th clock. When the address does not match, the INTIICBnTIA signal is not output and the IICBn does not enter the wait state. • In continuous transfer mode In continuous transfer mode, transition to the wait state is not affected by the setting of the IICBnCTL0.IICBnSLWT bit. - Reception: The IICBn enters the wait state at the falling edge of the 8th clock. - Transmission: The IICBn enters the wait state at the falling edge of the 9th clock. **Caution: In single transfer mode, changing the value of this bit is only allowed while IICBnCTL0.IICBnIICE is 0 or during the wait period.**

Remark: n = 0, 1

(4/4)

Bit Position	Bit Name	Function
0	IICBnSLAC	Controls acknowledge signal output. 0: Disables acknowledge signal output. Master: The acknowledge signal is not output during data reception (SDAn = "H"). Slave: The acknowledge signal is not output during data transfer when an address match occurs (SDAn = "H"). 1: Enables acknowledge signal output. Master: The acknowledge signal is output during data reception (SDAn = "L"). Slave: The acknowledge signal is output during data transfer when an address match occurs (SDAn ="L"). When the IICBn is operating as a slave, in the case of an address match, an acknowledge signal is output during address transfer regardless of the value of the IICBnCTL0.IICBnSLAC bit (SDAn = "L"). Also, no acknowledge signal is output (SDAn = "H") while the IICBn is transmitting data or when it does not participate in communications.

Remark: n = 0, 1

(4) IICBn control register 1 (IICBnCTL1)

This register controls operation of IICBn.

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

Caution: Write access to the IICBnCTL1 register is prohibited when the value of the IICBnCTL0.IICBnIICE bit is 1.

(1/2)

	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnCTL1	IICBn MDSC	IICBn LGDF2	IICBn LGDF1	IICBn LGDF0	IICBn MDLB	0	IICBn SLSE	IICBn SLRS	4000 0520H +100H × n	00H
R/W	R/W	R/W	R/W	R/W	R/W	0	R/W	R/W		

Bit Position	Bit Name	Function
7	IICBnMDSC	Specifies the operation mode for the IICBn. 0: Standard mode (SCL clock frequency: up to 100 kHz) 1: Fast mode (SCL clock frequency: up to 400 kHz)
6 to 4	IICBnLGDF [2:0]	Specify the digital filter sampling frequency. Note that the digital filter can be used only in the fast mode. 000: Does not use digital filter. SCLn and SDAn are used without passing through the digital filter in the IICBn. The digital filter circuit operations are stopped. 101: Uses digital filter. SCLn and SDAn are used passing through the digital filter in the IICBn. Others: Setting prohibited
3	IICBnMDLB	Specifies the loop back mode. 0: Do not loop back. 1: Loop back. By setting the IICBnCTL1.IICBnMDLB bit, the output serial clock signal (SCLn) and serial transmit/receive data signal (SDAn) are looped back and used as the input serial clock signal (SCLn) and input serial transmit/receive data signal (SDAn). The output SCLn and SDAn immediately before output will be looped back. Note that both SCLn and SDAn are at the high level if the IICBnCTL1.IICBnMDLB bit is "1".
2	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
1	IICBnSLSE	Enables/disables start condition output in the initial communication state. 0: Disables start condition output in the initial communication state. 1: Enables start condition output in the initial communication state. If the IICBnCTL1.IICBnSLSE bit is set to 1, a start condition can be output by setting the IICBnTRG.IICBnSTT bit to 1 in the initial communication state (from when the IICBnCTL0.IICBnIICE bit is set to 1 until detection of a stop condition). The IICBnCTL1.IICBnSLSE bit is automatically cleared to 0 upon detection of a start condition (even without a 0 write operation). **Caution: Clear the IICBnCTL1.IICBnSLSE bit to 0 when participating in communications after other communications have started. When other communications are being performed, if the IICBnTRG.IICBnSTT bit has been set to 1 with the IICBnCTL1.IICBnSLSE bit set to 1, the other communications may be damaged.**
0	IICBnSLRS	Enables/disables the communication reserve function. 0: Enables communication reserve function. 1: Disables communication reserve function. Communication reserve function enabled state: If the IICBnCTL1.IICBnSLRS bit is cleared to 0 while the IICBn is not operating as a master, the communication reserve state can be set by setting the IICBnTRG.IICBnSTT bit to 1 while the bus is being used. Whether the communication reserve state is set can be confirmed by checking the IICBnSTR0.IICBnSSRS bit. Communication reserve function disabled state: If the IICBnTRG.IICBnSTT bit is set to 1 while the IICBn is not participating in communications as a master and the bus is being used, the value of the IICBnSTR0.IICBnSTCF becomes 1 and communication reservation is not done.

Remark: n = 0, 1

(5) IICBn low level width setting register (IICBnWL)

This register is used to set the low level width of the serial clock register (SCLn).

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

Caution: Write access to the IICBnWL register is prohibited when the value of the IICBnCTL0.IICBnIICE bit is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnWL	0	0	0	0	0	0	IICBn WL9	IICBn WL8	IICBn WL7	IICBn WL6	IICBn WL5	IICBn WL4	IICBn WL3	IICBn WL2	IICBn WL1	IICBn WL0	4000 0524H +100H × n	03FFH
R/W	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
15 to 10	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
9 to 0	IICBnWL[9:0]	Specify the t_{LOW} period (low level width of the SCLn clock) of the I^2C bus specification. The value of the IICBnWL register is used to determine the serial output timing of other I^2C bus specifications. For the serial output timing setting conditions, refer to Table 24.4, Conditions for Generating Serial Output Timing.

(6) IICBn high-level width setting register (IICBnWH)

This register is used to set the high level width of the serial clock signal (SCLn).

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

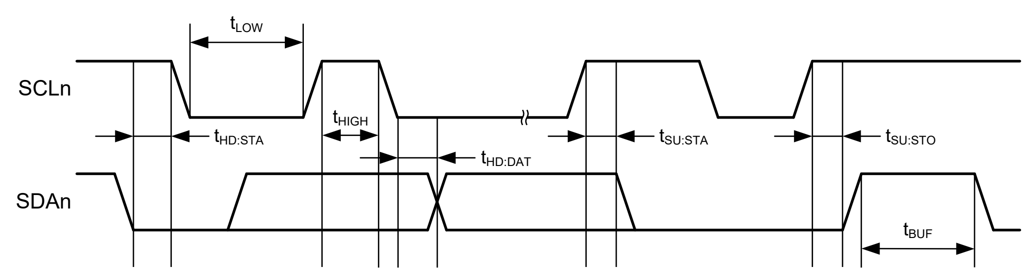
Caution: Write access to the IICBnWH register is prohibited when the value of the IICBnCTL0.IICBnIICE bit is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnWH	0	0	0	0	0	0	IICBn WH9	IICBn WH8	IICBn WH7	IICBn WH6	IICBn WH5	IICBn WH4	IICBn WH3	IICBn WH2	IICBn WH1	IICBn WH0	4000 0528H +100H × n	03FFH
R/W	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
15 to 10	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
9 to 0	IICBnWH[9:0]	Specify the t_{HIGH} period (high level width of the SCLn clock) of the I²C bus specification. The value of the IICBnWH register is used to determine the serial output timing of other I²C bus specifications. For the serial output timing setting conditions, refer to Table 24.4, Conditions for Generating Serial Output Timing.

Table 24.4 Conditions for Generating Serial Output Timing

Symbol	Description	Standard Mode	Fast Mode
$t_{HD:STA}$	Start condition hold time	IICB0WH / PCLK	IICB0WH / PCLK
t_{LOW}	SCL low-level width period	IICB0WL / PCLK	IICB0WL / PCLK
t_{HIGH}	SCL high-level width period	IICB0WH / PCLK	IICB0WH / PCLK
$t_{SU:STA}$	Start condition setup time	IICB0WL / PCLK	IICB0WH / PCLK
$t_{SU:STO}$	Stop condition setup time	IICB0WH / PCLK	IICB0WH / PCLK
t_{BUF}	Bus free time (interval between stop condition and start condition)	IICB0WL / PCLK	IICB0WL / PCLK
$t_{HD:DAT}$	Data hold time	IICB0WL[9:2] / PCLK	IICB0WL[9:2] / PCLK



(a) Setting transfer clock by using IICBnWL and IICBnWH registers

The various timings in compliance with the I^2C bus specifications can be set by setting the IICBnWL register and IICBnWH register.

[Setting transfer clock on master side]

Transfer clock (Hz) = PCLK / {(IICBnWL + IICBnWH) + PCLK (t_R + t_F)}

At this time, the optimal setting values of IICBnWL and IICBnWH are as follows.
(The fractional parts of all setting values are rounded up.)

- In the fast mode
  IICBnWL = (0.52/Transfer clock) × PCLK
  IICBnWH = (0.48/Transfer clock – t_R – t_F) × PCLK
- In the standard mode
  IICBnWL = (0.47/Transfer clock) × PCLK
  IICBnWH = (0.53/Transfer clock – t_R – t_F) × PCLK

Caution: The data hold time must be within 0.9 μs in the fast mode and within 3.45 μs in the standard mode.

Remark: The data hold time is determined by the IICBnWL register setting as follows:
Data hold time = IICBnWL.IICBnWL[9:2] / PCLK

[Setting IICBnWL and IICBnWH on slave side]
(The fractional parts of all setting values are rounded up.)

- In the fast mode
  IICBnWL = 1.3 μs × PCLK
  IICBnWH = (1.2 μs – t_R – t_F) × PCLK
- In the standard mode
  IICBnWL = 4.7 μs × PCLK
  IICBnWH = (5.3 μs – t_R – t_F) × PCLK

Remark:

IICBnWL	**: IICBn low-level width setting register**
IICBnWH	**: IICBn high-level width setting register**
t_F	**: SDAn and SCLn signal falling times**
t_R	**: SDAn and SCLn signal rising times**
PCLK	**: Frequency of the clock supplied to the IICBn**
f_{CLK}	**: SCL clock frequency**

(7) IICBn trigger register (IICBnTRG)

This register is used to set the IICBn trigger.

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

(1/5)

	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnTRG	0	0	0	0	IICBn LRET	IICBn WRET	IICBn STT	IICBn SPT	4000 050CH +100H × n	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit Position	Bit Name	Function
7 to 4	—	When writing to these bits, write 0. When read, 0 is returned.
3	IICBnLRET	Communication exit trigger bit 0: The read value is always 0, and writing 0 is ignored. 1: The IICBn exits the current communication and enters the wait state. This bit is automatically cleared to 0 following execution. The following occurs when IICBnTRG.IICBnLRET is 1. - SCLn and SDAn each go into high impedance (communication wait state). - Bits IICBnSSMS, IICBnSSDR, IICBnSSWT, IICBnSSEX, IICBnSSC0, IICBnSSTR, IICBnSSAC, IICBnSSRS, and IICBnSSST of the IICBnSTR0 register are cleared to 0. - When IICBnTRG.IICBnSTT = 1 (start condition output preparation) or IICBnTRG.IICBnSPT =1 (stop condition output preparation) has been set, output of a start condition or stop condition is stopped. The communication reserved state is released if the IICBn exits the communication in the communication reserved state. If it is necessary for the IICBn to operate a master again after this, the IICBnTRG.IICBnSTT bit must be set to 1 again. **Caution: If IICBnTRG.IICBnLRET is set to 1 during master operation (IICBnSTR0.IICBnSSMS = 1), the bus is released. Since serial clock output stops, problems occur during communication on the slave side.**

Remark: n = 0, 1

(2/5)

Bit Position	Bit Name	Function
2	IICBnWRET	This is the trigger bit for exiting the wait state. 0: Does not exit the wait state. 1: Exits the wait state and resumes communication. This bit is automatically cleared following execution. If the IICBn have exited the wait state by setting the IICBnTRG.IICBnWRET bit to 1 during the wait state triggered by the falling edge of the 9th clock, the IICBnSTR0.IICBnSSTR bit is cleared to 0 and SDAn goes into high impedance (this enables the external master to output a stop condition or start condition.) If the IICBn is not in the wait state (IICBnSTR0.IICBnSSWT = 0), setting this bit to 1 has no meaning. There are other conditions for exiting the wait state in addition to the setting of this bit. For details, refer to section 24.6.4, Entering and Exiting Wait State.
1	IICBnSTT	Start condition trigger bit 0: Does not output a start condition. 1: Outputs a start condition (This bit is automatically cleared to 0 after it has been set to 1. The IICBnTRG.IICBnSTT bit can be set to 1 under the following conditions: [1] IICBnSTR0.IICBnSSMS bit = Master state (1) • Single transfer mode - During wait state triggered by the falling edge of the 9th clock (both address transfer and data transfer) - During data reception, only after clearing the IICBnCTL0.IICBnSLAC bit to 0 to report the end of reception to the slave • Continuous transfer mode - During wait state triggered by the falling edge of the 9th clock of address transfer - During data transfer - During data reception, only after clearing the IICBnCTL0.IICBnSLAC bit to 0 to report the end of reception to the slave In the case of the wait period during the 9th clock, following wait cancellation, and in all other cases, upon detecting the falling edge of the 9th clock, SDAn and SCLn are set to the high level after the low-level width period of the SCLn clock, and then, when SDAn is set to the low level after waiting for the start condition setup time to elapse, a start condition is output. Next, SCLn is set to the low level after the start condition hold time has elapsed. For the individual time settings, see Table 24.4, Conditions for Generating Serial Output Timing.

Remark: n = 0, 1

(3/5)

Bit Position	Bit Name	Function
1	IICBnSTT	[2] Slave state or communication wait state (IICBnSTR0.IICBnSSMS = 0) • IICBnSTR0.IICBnSSBS bit = 0 (bus release state) After the bus free time elapses, a start condition is output when SDAn is changed from the high level to the low level while SCLn is high level. (At this time, SCLn outputs a high level signal.) Next, SCLn is set to the low level after the start condition hold time has elapsed. For the individual time settings, see Table 24.4, Conditions for Generating Serial Output Timing. • IICBnSTR0.IICBnSSBS bit = 1 (bus communication state) This status indicates that communication is performed on the bus while the IICBn is not operating as a master. - When communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0): A start condition is output after the bus has been released (the stop condition has been detected) and the bus free time has elapsed. However, even if the bus free time has not elapsed, upon detecting a start condition, SDAn is immediately set to the low level without waiting for the bus free time to elapse. For the individual time settings, see Table 24.4, Conditions for Generating Serial Output Timing. - When communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1): The IICBnSTR0.IICBnSTCF bit is set to 1 and a start condition is not output. **Caution: [2] shows the operations according to the value of the IICBnSTR0.IICBnSSBS bit when the IICBnTRG.IICBnSTT bit is 0.** **Even if the IICBnTRG.IICBnSTT bit is set to 1 after checking the value of the IICBnSTR0.IICBnSSBS bit through register read, the value of IICBnSTR0.IICBnSSBS may differ from its value when it was checked.**

Remark: n = 0, 1

(4/5)

Bit Position	Bit Name	Function
1	IICBnSTT	The output processing of the start condition is started by setting the IICBnTRG.IICBnSTT bit to 1, but upon detection of the following states, output processing of the start condition is stopped and the start condition is not output. - When 0 is written to the IICBnCTL0.IICBnIICE bit - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of arbitration loss - When 1 is written to the IICBnTRG.IICBnSPT bit after 1 is written to the IICBnTRG.IICBnSTT bit while the IICBn is operating as a master in continuous transfer mode - When 1 is written to the IICBnTRG.IICBnSTT and IICBnTRG.IICBnSPT bits during the same data transfer period while the IICBn is operating as a master in continuous transfer mode (In this case, writing 1 to the IICBnTRG.IICBnSTT bit is enabled.) **Cautions 1. When start in the initial communication state is enabled (IICBnCTL1.IICBnSLSE bit = 1), the start condition is output regardless of the bus status when the IICBnTRG.IICBnSTT bit is set to 1. If other communications are performed at that time, they may be damaged.** **2. Setting the IICBnTRG.IICBnSTT bit at the same time as the IICBnTRG.IICBnSPT bit is prohibited.**

Remark: n = 0, 1

(5/5)

Bit Position	Bit Name	Function
0	IICBnSPT	Stop condition trigger bit 0: Does not output a stop condition. 1: Outputs a stop condition (This bit is automatically cleared after it has been set to 1). The IICBnTRG.IICBnSPT bit can be set to 1 under the following conditions while the IICBn is performing communication as a master. • Single transfer mode - Wait state triggered by the falling edge of the 9th clock (both address transfer and data transfer) - During data reception, only after clearing the IICBnCTL0.IICBnSLAC bit to 0 to report the end of reception to the slave • Continuous transfer mode The IICBnTRG.IICBnSPT bit can be set to 1 in the following states. - During the wait state triggered by the falling edge of the 9th clock of address transfer - During data transfer - Detection of a NACK signal (IICBnSTR0.IICBnSSAC bit = 0) during the wait state triggered by the falling edge of the 9th clock for during data reception A stop condition can be output with the following procedure. (If the IICBn is in the wait state, after exiting the wait state) SCLn is released when SDAn has output a low level, and SCLn = high level, SDAn is low level are waited for. Then, following the lapse of the $t_{SU:STO}$ time, a stop condition is output by setting SDAn to high level. The output processing of the stop condition is started by setting the IICBnTRG.IICBnSPT bit to 1, but upon detection of the following states, output processing of the stop condition is stopped and the stop condition is not output. - When 0 is written to the IICBnCTL0.IICBnIICE bit - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - Upon detection of arbitration loss - When 1 is written to the IICBnTRG.IICBnSTT bit after IICBnTRG.IICBnSPT has been set to 1 while the IICBn is operating as a master in continuous transfer mode **Cautions 1. Setting the IICBnTRG.IICBnSPT bit to 1 is prohibited during slave operation (IICBnSTR0.IICBnSSMS bit = 0)** **2. Setting the IICBnTRG.IICBnSPT bit to 1 at the same time as the IICBnTRG.IICBnSTT bit is prohibited.**

Remark: n = 0, 1

(8) IICBn status register 0 (IICBnSTR0)

This register indicates the states of the IICBn and the bus.

- Access: This register is only readable in 16-bit units. However, when IICBnCTL0.IICBnIICE is 0, this register can also be written.

This register is initialized by any reset. This register is also initialized by changing the value of the IICBnCTL0.IICBnIICE bit from 1 to 0 or from 0 to 1.

(1/8)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnSTR0	IICBn SSMS	0	IICBn SSDR	IICBn SSWT	IICBn SSEX	IICBn SSCO	IICBn SSTR	IICBn SSAC	IICBn SSRS	IICBn SSBS	IICBn SSST	IICBn SSSP	0	0	IICBn STCF	IICBn ALDF	4000 0510H +100H × n	0000H
R/W	R	0	R	R	R	R	R	R	R	R	R	R	0	0	R	R		

Bit Position	Bit Name	Function
15	IICBnSSMS	Master state check flag 1: Indicates that the IICBn is operating as a master. Setting condition: Upon detection of a start condition after 1 is written to the IICBnTRG.IICBnSTT bit Clearing conditions: - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - Upon detection of arbitration loss If a setting condition coincides with a clearing condition, the clearing condition takes priority.
14	—	Reserved. This bit is read as 0.

Remark: n = 0, 1

(2/8)

Bit Position	Bit Name	Function
13	IICBnSSDR	IICBnDAT register status flag 1: Indicates that data in the IICBnDAT register remains unprocessed. During reception operation: Received data remains unread in the IICBnDAT register. During transmission operation: Data written to the IICBnDAT register has not been transferred to the shift register. Setting conditions: - When the IICBnDAT register is written during address transfer and data transfer while the IICBnSTR0.IICBnSSWT bit is 0 (Note that, even if the IICBnSTR0.IICBnSSWT bit is 0, the IICBnSSDR bit is not set to 1 if address data is written to the IICBnDAT register while the IICBn is operating as a master, because the address data is directly transferred to the shift register in this case.) - At the falling edge of the 9th clock after an address match with a slave - While IICBnCTL0.IICBnSLWT = 0 and single mode reception is being performed, at the falling edge of the 8th clock during data reception - At the falling edge of the 8th clock while in continuous transfer mode (reception), regardless of the IICBnCTL0.IICBnSLWT bit value - While IICBnCTL0.IICBnSLWT = 1, at the falling edge of the 9th clock during data reception Clearing conditions: • Clearing conditions given priority over setting conditions - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of arbitration loss - At the falling edge of the 9th clock during address transfer while the IICBn is operating as a master - At the falling edge of the 8th clock during data transmission while IICBnCTL0.IICBnSLWT = 0 and continuous transmission is being performed - At the falling edge of the 9th clock during data transmission while IICBnCTL0.IICBnSLWT = 1 and continuous transmission is being performed • Clearing condition for which setting conditions are given priority (while in continuous transfer mode (transmission) - When the IICBnDAT register is read while the shift register does not have any received data that must be transferred to the IICBnDAT register

Remark: n = 0, 1

(3/8)

Bit Position	Bit Name	Function
12	IICBnSSWT	Wait state flag 1: Indicates that the IICBn is in the wait state. Setting conditions: [In single transfer mode] <Common to master/slave> - During data transfer, upon detection of the falling edge of the 8th clock with IICBnCTL0.IICBnSLWT = 0 - During data transfer, upon detection of the falling edge of the 9th clock with IICBnCTL0.IICBnSLWT = 1 <Master> - When the IICBn becomes a master (IICBnSTR0.IICBnSSMS = 1) after 1 is written to the IICBnTRG.IICBnSTT bit, and the falling edge of the first SCLn is detected without the IICBnDAT register being written - Upon detection of the falling edge of the 9th clock during address transfer <Slave> - Upon detection of the falling edge of the 9th clock during address transfer when an address match occurred [In continuous transfer mode] <During data transfer period, common to master/slave> • During data transmission, when the data to be transmitted next has not been written - When IICBnCTL0.IICBnSLWT = 0, at the falling edge of the 8th clock during data transmission with IICBnSTR0.IICBnSSDR = 0 - When IICBnCTL0.IICBnSLWT = 1, at the falling edge of the 9th clock during data transmission with IICBnSTR0.IICBnSSDR = 0 • During data reception, when the previous received data has not been read - When IICBnCTL0.IICBnSLWT = 0, at the falling edge of the 8th clock during data reception with IICBnSTR0.IICBnSSDR = 1 - When IICBnCTL0.IICBnSLWT = 1, at the falling edge of the 9th clock during data reception with IICBnSTR0.IICBnSSDR = 1 - Upon NACK detection (However, only if 1 has not been written to IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT while the IICBn is operating as a master)

Remark: n = 0, 1

(4/8)

Bit Position	Bit Name	Function
12	IICBnSSWT	<During address transfer period, operating as master> - When the IICBn becomes a master (IICBnSTR0.IICBnSSMS = 1) after 1 is written to the IICBnTRG.IICBnSTT bit, and the first falling edge of SCLn is detected without the IICBnDAT register being written - Upon NACK detection (However, only if 1 has not been written to IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT) <During address transfer period, operating as slave> - Upon detection of the falling edge of the 9th clock while IICBnSTR0.IICBnSSTR bit is 0 during address transfer when an address match occurred - Upon NACK detection Clearing conditions: • Clearing conditions given priority over setting conditions - When 1 is written to the IICBnTRG.IICBnLRET bit - When 1 is written to the IICBnTRG.IICBnSTT bit while the IICBn is operating as a master in continuous transfer mode - When 1 is written to the IICBnTRG.IICBnSPT bit while the IICBn is operating as a master in continuous transfer mode - When the IICBnDAT register is written while the IICBn is performing transmission in continuous transfer mode - During the wait state triggered by the falling edge of the 8th clock, when the IICBnDAT register is read while reception is performed in continuous transfer mode - During the wait state triggered by the falling edge of the 9th clock, when the IICBnDAT register is read while the IICBn is performing reception in continuous transfer mode and an acknowledge signal ($\overline{\text{ACK}}$) has been received • Clearing conditions for which setting conditions are given priority - When 1 is written to the IICBnTRG.IICBnWRET bit - When 1 is written to the IICBnTRG.IICBnSTT bit while the IICBn is operating as a master in single transfer mode - When 1 is written to the IICBnTRG.IICBnSPT bit while the IICBn is operating as a master in single transfer mode - When the IICBnDAT register is written while the IICBn is performing reception in single transfer mode **Caution: If the IICBn exits the wait state that was triggered by the falling edge of the 9th clock by writing 1 to the IICBnTRG.IICBnWRET bit, the IICBnSTR0.IICBnSSTR bit is cleared to 0 and the bus is released (both SCLn and SDAn go into high impedance).**

Remark: n = 0, 1

(5/8)

Bit Position	Bit Name	Function
11	IICBnSSEX	Expansion code reception detection flag 1: Indicates that an expansion code has been received. Setting condition: Upon detection of the falling edge of the 8th clock while transferring received address data whose higher 4 bits are either 0000 or 1111 Clearing conditions: - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - Upon detection of a start condition **Caution: When the expansion codes match, the processing after the interrupt differs according to the ensuing data, and therefore is dependent on software processing.**
10	IICBnSSCO	Address match detection flag 1: Indicates that an address that matches the IICBnSVA register has been detected. Setting condition: Upon detection of the falling edge of the 8th clock while transferring a received address that matches the IICBnSVA register Clearing conditions: - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - Upon detection of a start condition

Remark: n = 0, 1

(6/8)

Bit Position	Bit Name	Function
9	IICBnSSTR	Transmission status detection flag 1: Indicates that data is being transmitted to the serial data bus. Setting conditions: <Master> - Upon detection of a start condition after 1 is written to the IICBnTRG.IICBnSTT bit <Slave> - Upon detection of the falling edge of the 8th clock following reception of 1 to R/$\overline{W}$ bit during address transfer when an address match occurred Clearing conditions: <Common to master/slave> - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - When 1 is written to the IICBnTRG.IICBnWRET bit during the wait state triggered by the falling edge of the 9th clock <Master> - Upon detection of the falling edge of the 8th clock following reception of 1 to R/$\overline{W}$ bit during address transfer - Upon detection of arbitration loss <Slave> - Upon detection of a start (restart) condition
8	IICBnSSAC	Acknowledge ($\overline{ACK}$) detection flag 1: Indicates that an acknowledge signal has been detected. Setting condition: Upon detection of the falling edge of SCLn when a low level has been received at the $\overline{ACK}$ bit during participation in communications Clearing conditions: - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of the rising edge of SCLn **Caution: The value of the IICBnSTR0.IICBnSSAC bit changes regardless of whether or not an interrupt has occurred.**
7	IICBnSSRS	Communication reserve state flag 0: Not communication reserve state 1: Communication reserve state Setting condition: When 1 is written to the IICBnTRG.IICBnSTT bit during bus communication while the IICBn is not operating as a master, in the communication reserve function enabled state (IICBnCTL1.IICBnSLRS = 0) Clearing conditions: - When 1 is written to the IICBnTRG.IICBnLRET bit - When IICBnSTR0.IICBnSSMS = 1

Remark: n = 0, 1

(7/8)

Bit Position	Bit Name	Function
6	IICBnSSBS	IICBn bus status flag 0: Bus released state (initial communication state when IICBnCTL1.IICBnSLSE = 1) 1: Bus communication state (initial communication state when IICBnCTL1.IICBnSLSE = 0) Setting condition: - Upon detection of a start condition - When 1 is written to the IICBnCTL0.IICBnIICE bit when IICBnCTL1.IICBnSLSE = 0 Clearing conditions: Upon detection of a stop condition **Remark: The IICBnSTR0.IICBnSSBS bit operates whether or not the IICBn is participating in communications.**
5	IICBnSSST	Start condition detection flag 1: Indicates that a start condition has been detected. Setting condition: Upon detection of a start condition Clearing conditions: - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - Upon detection of the rising edge of SCLn following the end of address transfer **Remark: The IICBnSTR0.IICBnSSST bit operates whether or not the IICBn is participating in communications.**
4	IICBnSSSP	Stop condition detection flag 1: Indicates that a stop condition has been detected. Setting condition: Upon detection of a stop condition Clearing conditions: Upon detection of the falling edge of the first SCLn following start condition detection **Remark: The IICBnSTR0.IICBnSSSP bit operates whether or not the IICBn is participating in communications.**
3, 2	—	Reserved. These bits are read as 0.

Remark: n = 0, 1

(8/8)

Bit Position	Bit Name	Function
1	IICBnSTCF	IICBnTRG.IICBnSTT bit clear flag 1: Indicates that the IICBnTRG.IICBnSTT bit has been cleared because start condition output failed. Setting condition: When 1 is written to the IICBnTRG.IICBnSTT bit during bus communication when the IICBn is not operating as a master, in the communication reserve function disabled state (IICBnCTL1.IICBnSLRS = 1) **Caution: Even if the bus is released in the external bus state, this bit is set to 1 when 1 is written to the IICBnTRG.IICBnSTT bit if the communication reserve function is disabled, unless the IICBn recognizes the bus release state (IICBnSTR0.IICBnSSBS = 1).** Clearing condition: When 1 is written to the IICBnSTRC.IICBnCLSF bit
0	IICBnALDF	Arbitration loss detection flag 1: Indicates that an arbitration loss has been detected. Setting condition: Upon detection of arbitration loss Clearing condition: When 1 is written to the IICBnSTRC.IICBnCLAF bit If a setting condition coincides with a clearing condition, the setting condition takes priority. Upon detection of arbitration loss, the IICBnSTR0.IICBnSSMS and IICBnSTR0.IICBnSSTR bits are cleared to 0. (SCLn and SDAn become high level and the bus is released.) **Caution: When the IICBnSTR0.IICBnALDF bit is set to 1 due to arbitration loss, the INTIICBnTIA or INTIICBnTIS interrupt request signal is output. After confirming that the IICBnSTR0.IICBnALDF bit has been set to 1 with an interrupt request signal, clear the IICBnSTR0.IICBnALDF bit with the IICBnSTRC.IICBnCLAF bit. If the value of the IICBnSTR0.IICBnALDF is not cleared and remains 1, the INTIICBnTIS interrupt request signal will be output at the interrupt timing, even during unrelated communication.**

Remark: n = 0, 1

(9) IICBn status register 1 (IICBnSTR1)

This register indicates the state of the serial bus.

- Access This register is only readable in 8-bit units.

Caution: The serial clock (SCLn) and serial transmit/receive data (SDAn) are also read from an external source in loopback mode (IICBnCTL1.IICBnMDLB = 1).

	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnSTR1	0	0	0	0	0	0	IICBn SSCL	IICBn SSDA	4000 0514H +100H × n	00H
R/W	0	0	0	0	0	0	R	R		

Bit Position	Bit Name	Function
7 to 2	—	Reserved. These bits are read as 0.
1	IICBnSSCL	Indicates the level of the SCLn pin (input). 0: Low level 1: High level
0	IICBnSSDA	Indicates the level of the SDAn pin (input). 0: Low level 1: High level

Remark: n = 0, 1

(10) IICBn status clear register (IICBnSTRC)

This register clears the IICBnSTCF and IICBnALDF bits of the IICBnSTR0 register.

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

	7	6	5	4	3	2	1	0	Address	Initial Value
IICBnSTRC	0	0	0	0	0	0	IICBn CLSF	IICBn CLAF	4000 0518H +100H × n	00H
R/W	0	0	0	0	0	0	R/W	R/W		

Bit Position	Bit Name	Function
7 to 2	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
1	IICBnCLSF	Clears the IICBnSTR0.IICBnSTCF bit. 1: Clears the IICBnSTR0.IICBnSTCF bit. **Remark: If the IICBnSTRC.IICBnCLSF bit is read after setting data, 0 is returned.**
0	IICBnCLAF	Clears the IICBnSTR0.IICBnALDF bit. 1: Clears the IICBnSTR0.IICBnALDF bit. **Caution: If writing 1 to the IICBnSTRC.IICBnCLAF bit and the setting condition of the IICBnSTR0.IICBnALDF bit occur at the same time, the setting condition of the IICBnSTR0.IICBnALDF takes priority.** **Remark: If the IICBnSTRC.IICBnCLAF bit is read after data setting, 0 is returned.**

Remark: n = 0, 1

24.4 IIC Bus Mode Functions

24.4.1 Pin Configuration

The serial clock pin (SCLn) and serial data bus pin (SDAn) are configured as follows.

SCLn: This pin is used for serial clock input and output.
This pin is an N-channel open-drain output for both master and slave devices.

SDAn: This pin is used for serial data input and output.
This pin is an N-channel open-drain output for both master and slave devices.

Because the outputs of the serial clock line and serial data bus line are N-channel open-drain outputs, an external pull-up resistor must be connected to these lines.

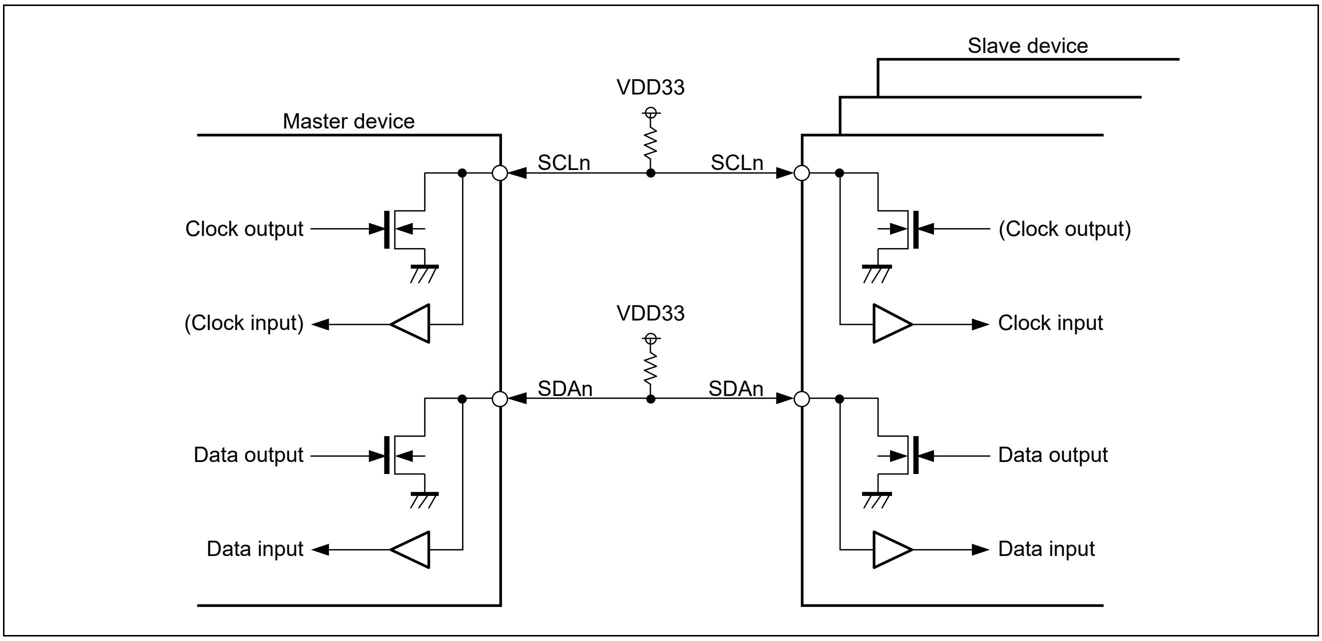


Figure 24.2 Pin Configuration Diagram

24.5 IIC Bus Definition

This section describes the IIC bus's serial data communication format and the signals used by the IIC bus.

Figure 24.3 shows the transfer timing for the "start condition", "address", "transfer direction specification", "data", and "stop condition", which are output onto the IIC bus's serial data bus.

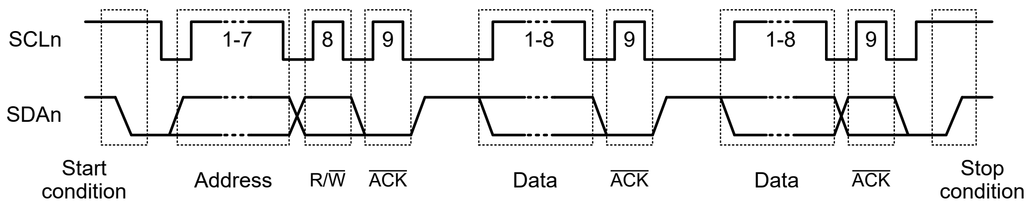


Figure 24.3 IIC bus Serial Data Transfer Timing

The start condition, slave address, and stop condition are output by the master device.

$\overline{ACK}$ can be output by either the master or slave device. (Normally, it is output by the device that receives 8-bit data.)

The serial clock signal (SCLn) is continuously output by the master device. In the slave device, the low-level period of the SCLn signal can be extended to insert a wait.

24.5.1 Start Condition

The start condition is met if the SDAn signal level changes from high to low while the SCLn signal is high. The start condition is output when the master device starts serial data transfer to a slave device. When the IICBn is in the slave mode, it detects the start condition.

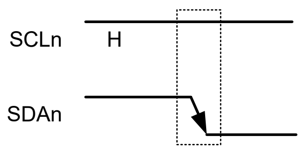


Figure 24.4 Start Condition

24.5.2 Addresses

The 7 bits of data following the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines.
Therefore, each slave device connected via the bus lines must have a unique address.

The slave device checks whether the 7-bit data matches its own address. If they match, that slave device is selected as the communication destination and communicates with the master device until the master device outputs another start condition or a stop condition.

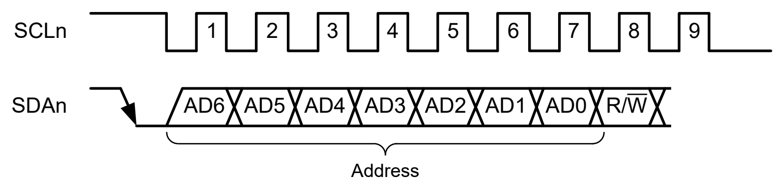


Figure 24.5 Address

24.5.3 Extension Code

When the higher-order 4 bits of the address are 0000 or 1111, these bits are called extension code. Table 24.5 lists the bit definitions of extension code.

Table 24.5 Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	×	CBUS address
0000 010	×	Address reserved for different bus format
0000 011	×	Reserved for future use
0000 1xx	×	HS mode master code Note
1111 0xx	×	10-bit slave address specification
1111 1xx	×	Reserved for future use

Note: The HS mode cannot be used for IICB.

24.5.4 Transfer Direction Specification

After the 7-bit address data, the master device transmits 1 bit that specifies the transfer direction.

If this transfer direction specification bit is 0, it indicates that the master device transmits data to a slave device. If this bit is 1, it indicates that the master device receives data from a slave device.

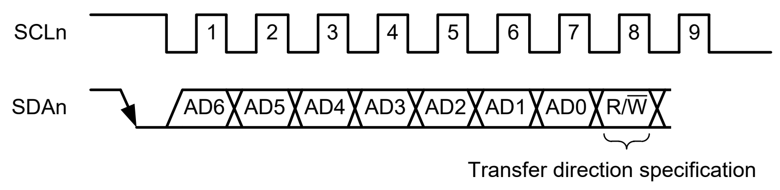


Figure 24.6 Transfer Direction Specification

24.5.5 Acknowledge ($\overline{\text{ACK}}$)

The 1-bit data after the transfer direction bit (R/$\overline{\text{W}}$) and the 1-bit data after the 8-bit data during address transfer are defined as an acknowledge signal ($\overline{\text{ACK}}$). $\overline{\text{ACK}}$ is used to check the serial data status of the transmitting and receiving devices.

The receiving device returns $\overline{\text{ACK}}$ after receiving 8-bit data.

The transmitting device normally receives $\overline{\text{ACK}}$ after transmitting 8-bit data. If the transmitting device receives $\overline{\text{ACK}}$ from the receiving device, it continues processing assuming that the transmitted data is normally received.

If the master device is the receiving device and receives the final data, it does not return $\overline{\text{ACK}}$ and outputs a stop condition. If the slave device is the receiving device and does not return $\overline{\text{ACK}}$, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return $\overline{\text{ACK}}$ may be caused by the following factors.

(1) The transmitted data has not been received normally.

(2) The final data has been received.

(3) The receiving device (slave) does not exist for the specified address.

$\overline{\text{ACK}}$ is output when the SDAn line of the receiving device changes to low level at the 9th clock (normal reception).

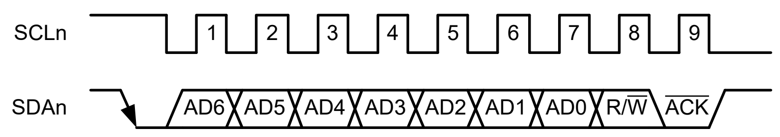


Figure 24.7 Acknowledge ($\overline{\text{ACK}}$)

24.5.6 Data

The bits other than the nine bits following the start condition (seven address bits, an R/$\overline{W}$ bit, and the acknowledge bit ($\overline{ACK}$)) and the acknowledge bits are defined as data.

If a 10-bit address is specified using an extension code, the 8-bit data that is transferred after the address is used as the second address.

24.5.7 Stop Condition

A stop condition is met if the SDAn signal level changes from low to high while the SCLn signal is high.

The stop condition is output when serial data transfer from the master device to the slave device has been completed.

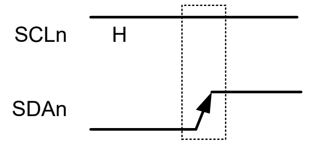


Figure 24.8 Stop Condition

24.5.8 Wait State

A wait state is used to report to the communication destination that the IICBn (master or slave) is preparing to transmit or receive data.

The wait state is reported to the communication destination by setting the SCLn signal to low. The next data transfer cannot start until both the master and slave devices exit the wait state.

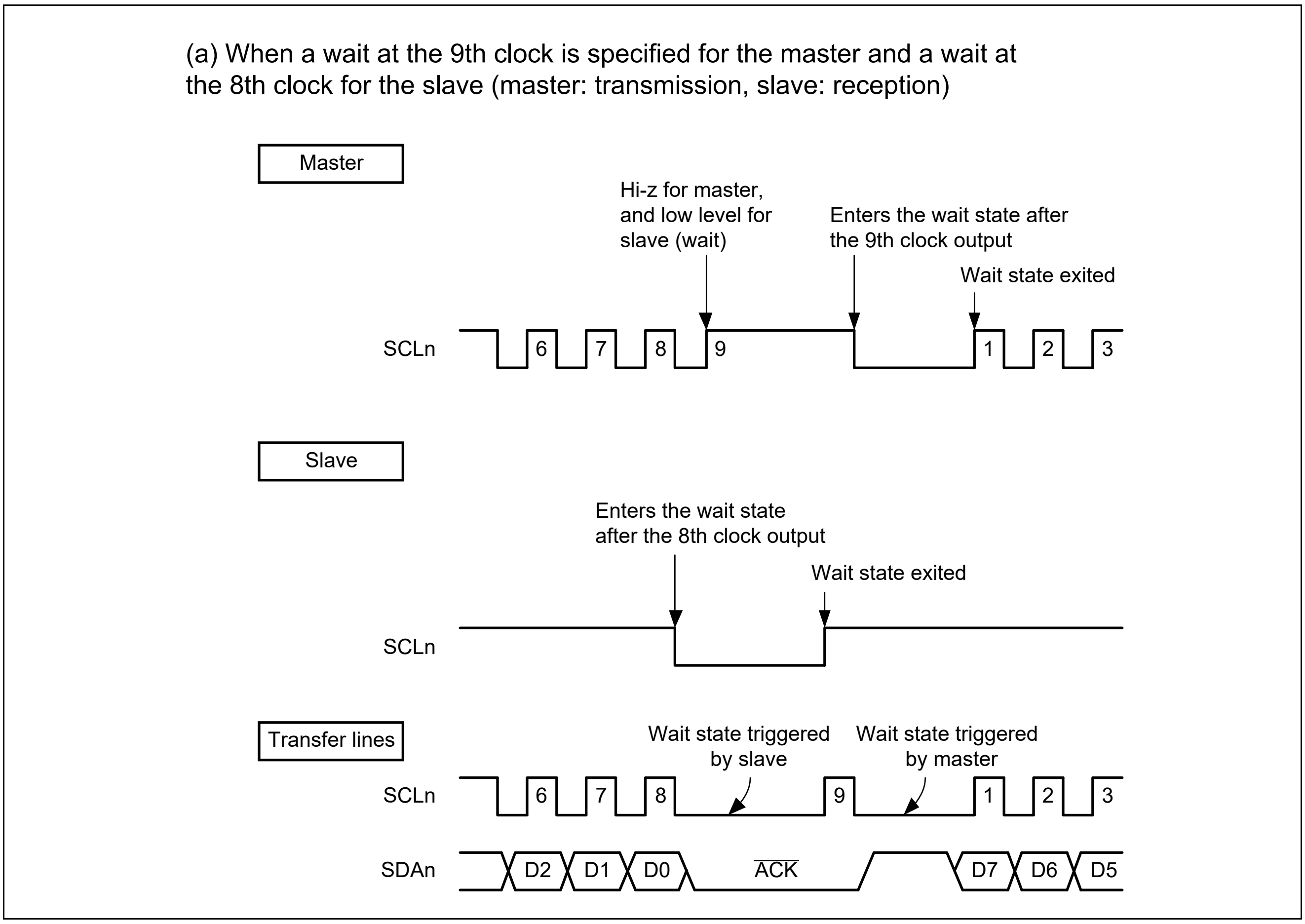


Figure 24.9 Wait State (1/2)

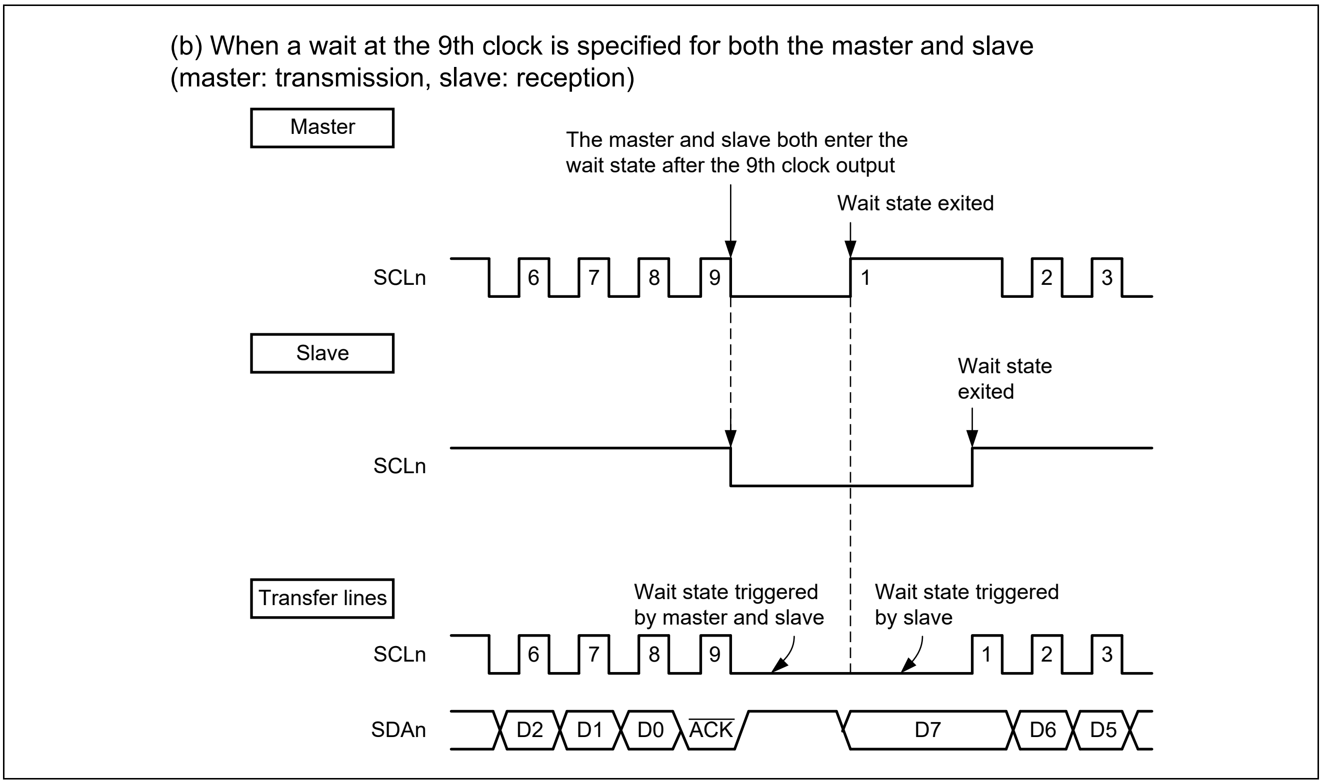


Figure 24.9 Wait State (2/2)

24.5.9 Arbitration

When several master devices simultaneously output a start condition, communication with the master devices continues until the data differs, while adjusting the clocks. An example where two masters simultaneously output a start condition and arbitration is conducted is described below.

This example assumes that one master outputs the SDAn line high (master 1) and the other master outputs the SDAn line low (master 2) while the SCLn line is low.

In this case, the communication with master 2 is prioritized, and communication is not authorized for master 1.

This kind of operation is called arbitration, and the state in which communication is not authorized is called arbitration loss. The master that lost arbitration releases the bus by setting both the SCLn and SDAn line to high impedance.

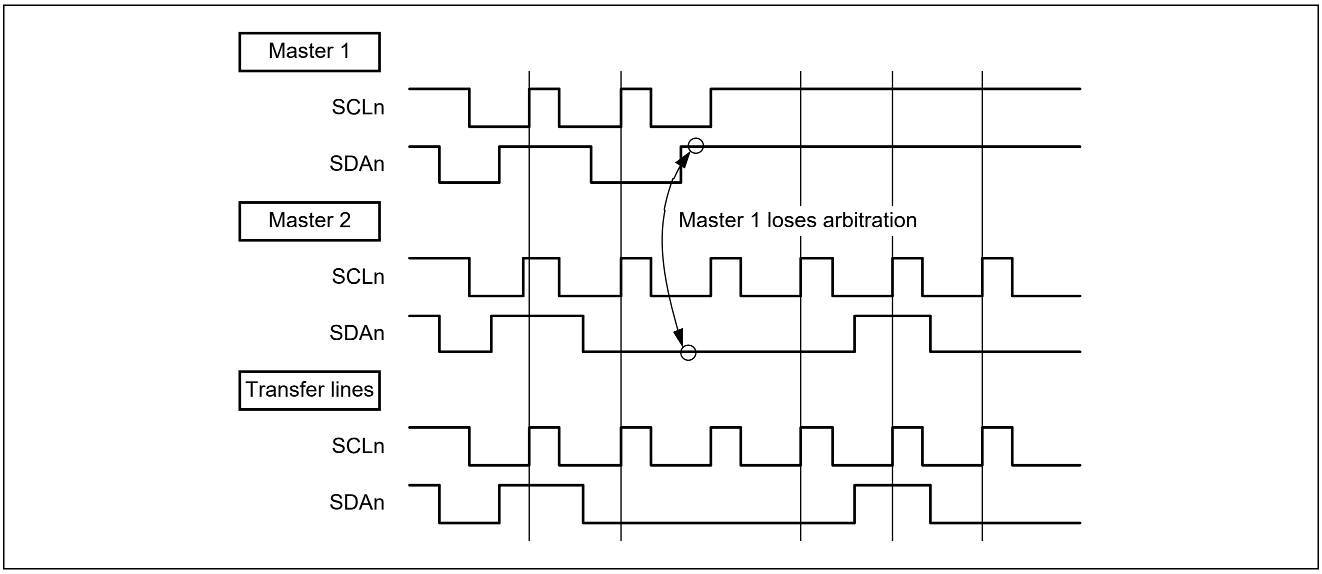


Figure 24.10 Arbitration Timing Example

24.6 Operation

The IICBn supports two transfer modes, single transfer mode and continuous transfer mode.

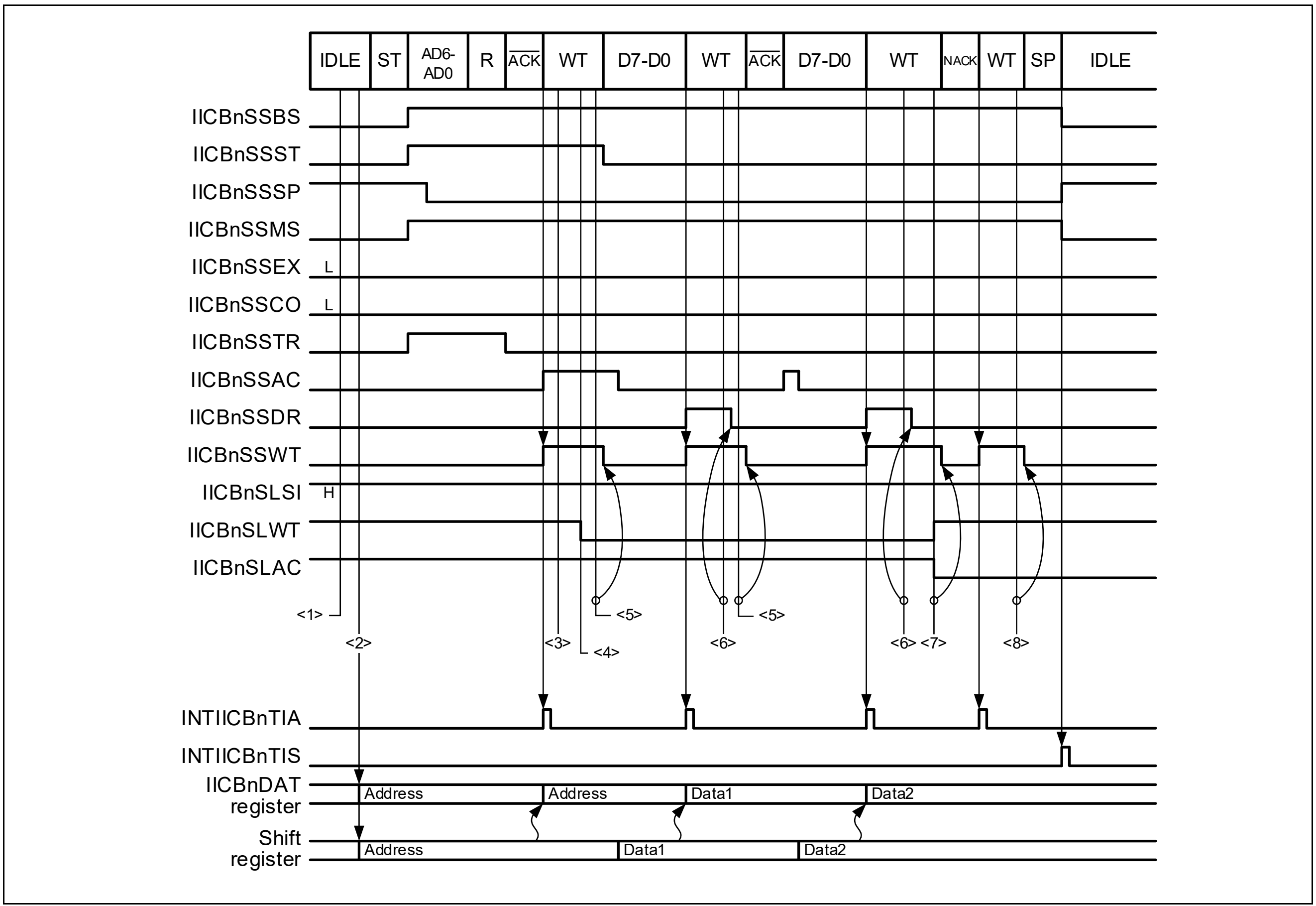
The transfer mode when the addresses of the master device and the slave device match is selected with the IICBnCTL0.IICBnMDTX0 bit, and the transfer mode when the extension code is detected by the slave device is selected with the IICBnCTL0.IICBnMDTX1 bit.

24.6.1 Single Transfer Mode

In single transfer mode, a data transmit/receive interrupt request signal (INTIICBnTIA) is output at the timing specified using the IICBnCTL0.IICBnSLWT bit to make the IICBn enter the wait state, and transmit/receive data processing is performed during this wait state.

The various processing operations are described below.

(1) Example of communications in single transfer mode (master reception)



<1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).

<2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.

<3> Acknowledge result check
Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit using the INTIICBnTIA interrupt.

<4> Wait timing setting
During data reception, clear the IICBnCTL0.IICBnSLWT bit (to 0) so that the IICBn enters the wait state at the falling edge of the 8th clock.

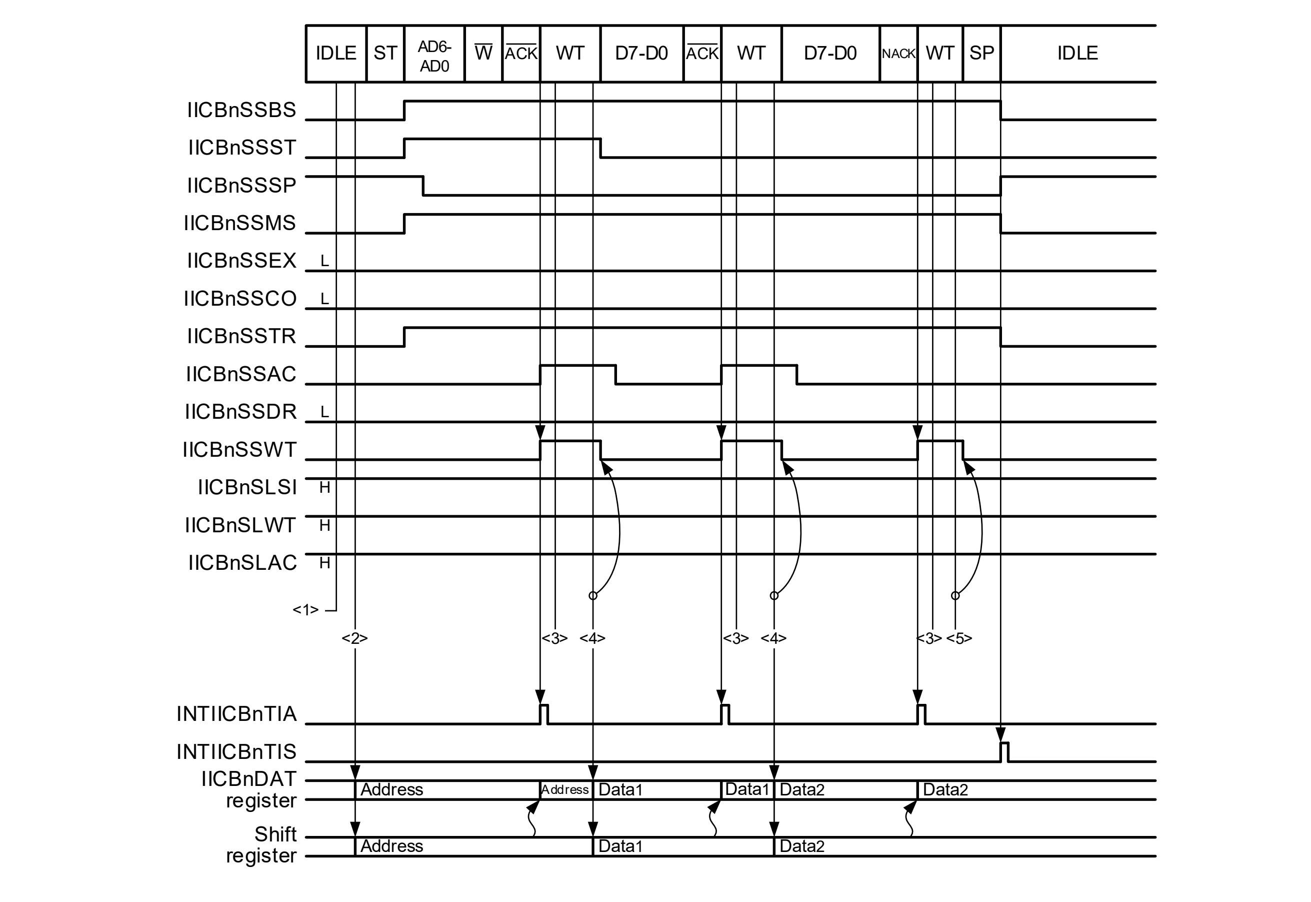
<5> Data reception
Exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1) to start reception.

<6> Receive data load
Read the receive data from the IICBnDAT register using the INTIICBnTIA interrupt.

<7> Data reception completion processing
- Set the IICBnCTL0.IICBnSLWT bit to 1 and the IICBnCTL0.IICBnSLAC bit to 0.
- Next, exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1). The end of the data is notified to the transmitting device without outputting ACK.

<8> Stop condition output
Set the IICBnTRG.IICBnSPT bit (to 1).

(2) Example of communications in single transfer mode (master transmission)



<1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).

<2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.

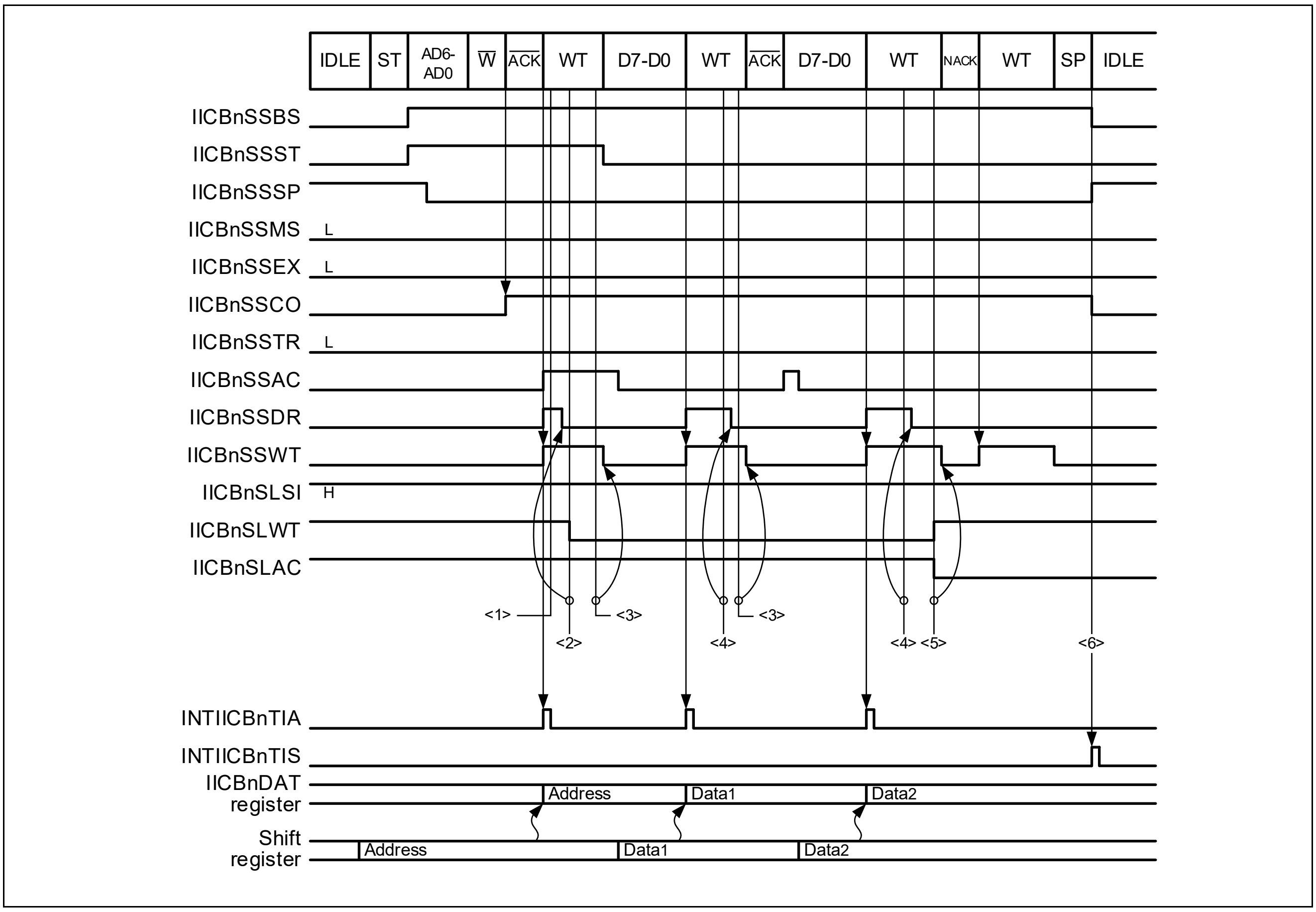
<3> Acknowledge result check
Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit using the INTIICBnTIA interrupt.

<4> Data transmission
Exit the wait state by setting the transmit data into the IICBnDAT register to start transmission.

<5> Stop condition output
Set the IICBnTRG.IICBnSPT bit (to 1).

Remark: During data transmission, set the IICBnCTL0.IICBnSLWT bit (to 1) so that the IICBn enters the wait state at the falling edge of the 9th clock.

(3) Example of communications in single transfer mode (slave reception)



<1> Operation mode check in slave mode
- Check the operation mode using the INTIICBnTIA interrupt.
- Check the address transfer, address match, and reception operation with the IICBnSTR0.IICBnSSST, IICBnSTR0.IICBnSSCO, and IICBnSTR0.IICBnSSTR bits.
- Read the IICBnDAT register (empty read).

<2> Wait timing setting
During data reception, clear the IICBnCTL0.IICBnSLWT bit (to 0) so that the IICBn enters the wait state at the falling edge of the 8th clock.

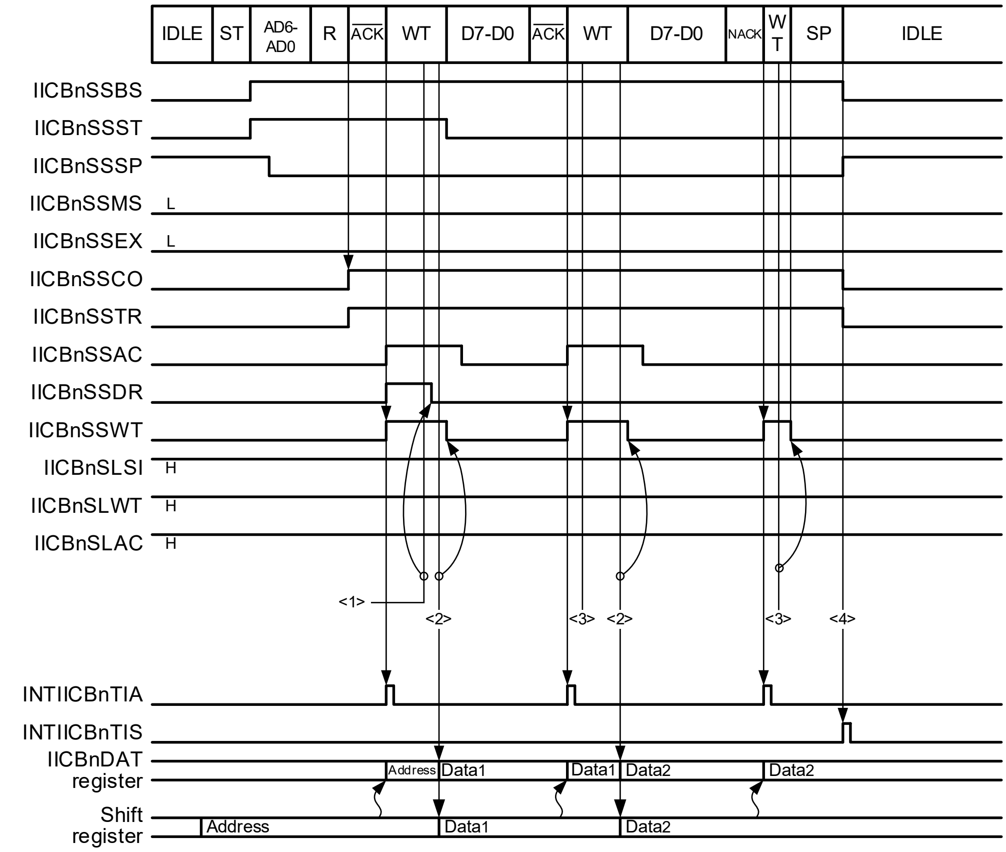
<3> Data reception
Exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1) to start reception.

<4> Receive data load
Read the receive data from the IICBnDAT register using the INTIICBnTIA interrupt.

<5> Data reception completion processing
- Set the IICBnCTL0.IICBnSLAC bit to 0.
- Next, exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1). The end of the data is notified to the transmitting device without outputting $\overline{\text{ACK}}$.

<6> Stop condition detection
Detect the stop condition using the INTIICBnTIS interrupt.

(4) Example of communications in single transfer mode (slave transmission)



<1> Operation mode check in slave mode

- Check the operation mode using the INTIICBnTIA interrupt.
- Check the address transfer, address match, and reception operation with the IICBnSTR0.IICBnSSST, IICBnSTR0.IICBnSSCO, and IICBnSTR0.IICBnSSTR bits.
- Read the IICBnDAT register (empty read).

<2> Data transmission

Exit the wait state by setting the transmit data into the IICBnDAT register to start transmission.

<3> Acknowledge result check

Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit using the INTIICBnTIA interrupt. If $\overline{ACK}$ is not output, the transmission is judged to have been completed, and the IICBn exits the wait state by setting the IICBnTRG.IICBnWRET bit (to 1).

<4> Stop condition detection

Detect the stop condition using the INTIICBnTIS interrupt.

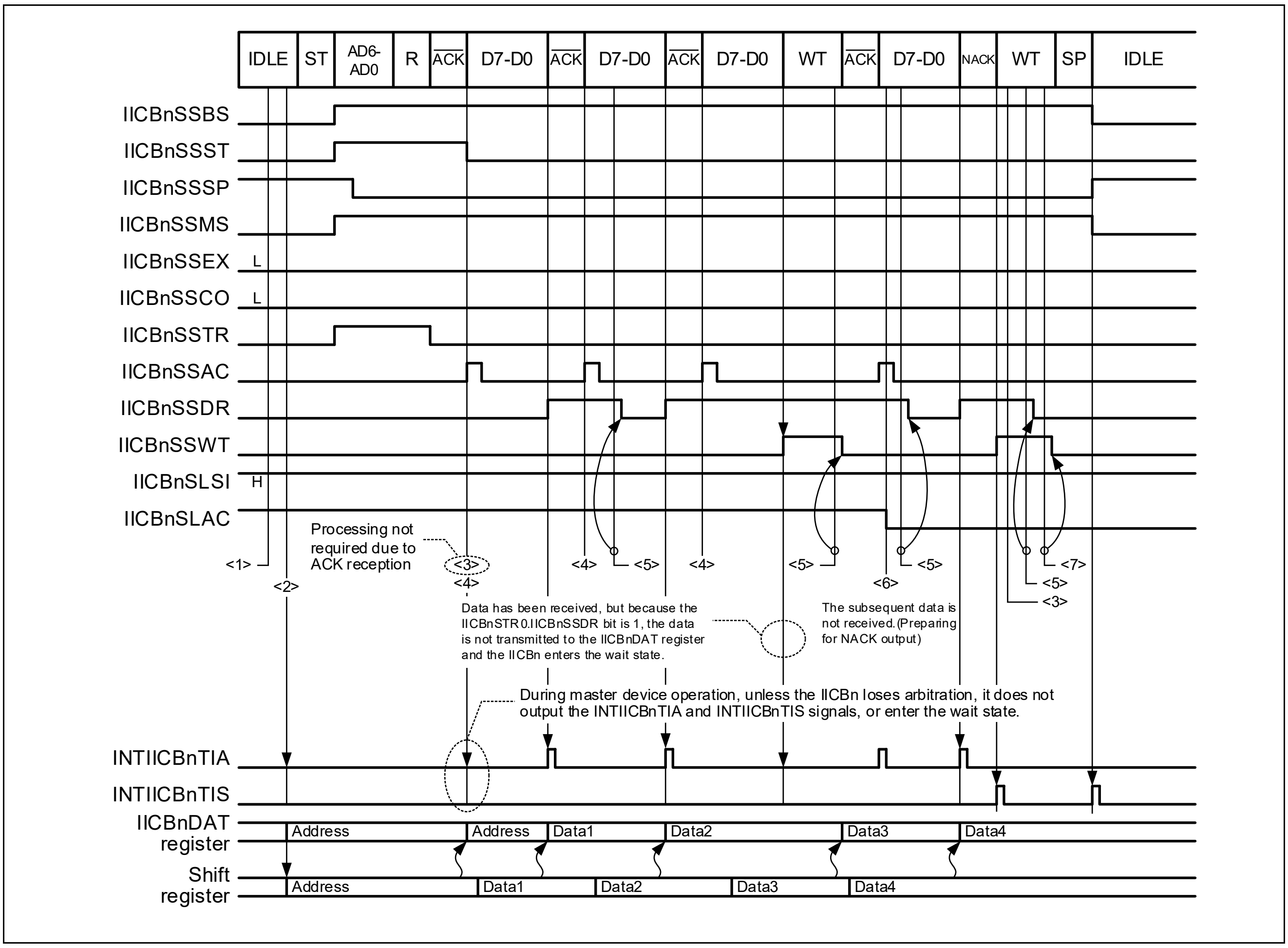
Remark: Since the bus enters the wait state on the falling edge of the 9th clock cycle in data transmission, set the IICBnCTL0.IICBnSLWT bit to 1.

24.6.2 Continuous Transfer Mode

Continuous transfer mode allows continuous communication without entering the wait state by reading/writing data from/to the IICBnDAT register each time the data transmit/receive interrupt request signal (INTIICBnTIA) is output.

The processing operations are described below.

(1) Example of communications in continuous transfer mode (master reception)



<1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).

<2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.

<3> Acknowledge result check
The INTIICBnTIS interrupt occurs only if the slave device does not return $\overline{ACK}$. Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit.

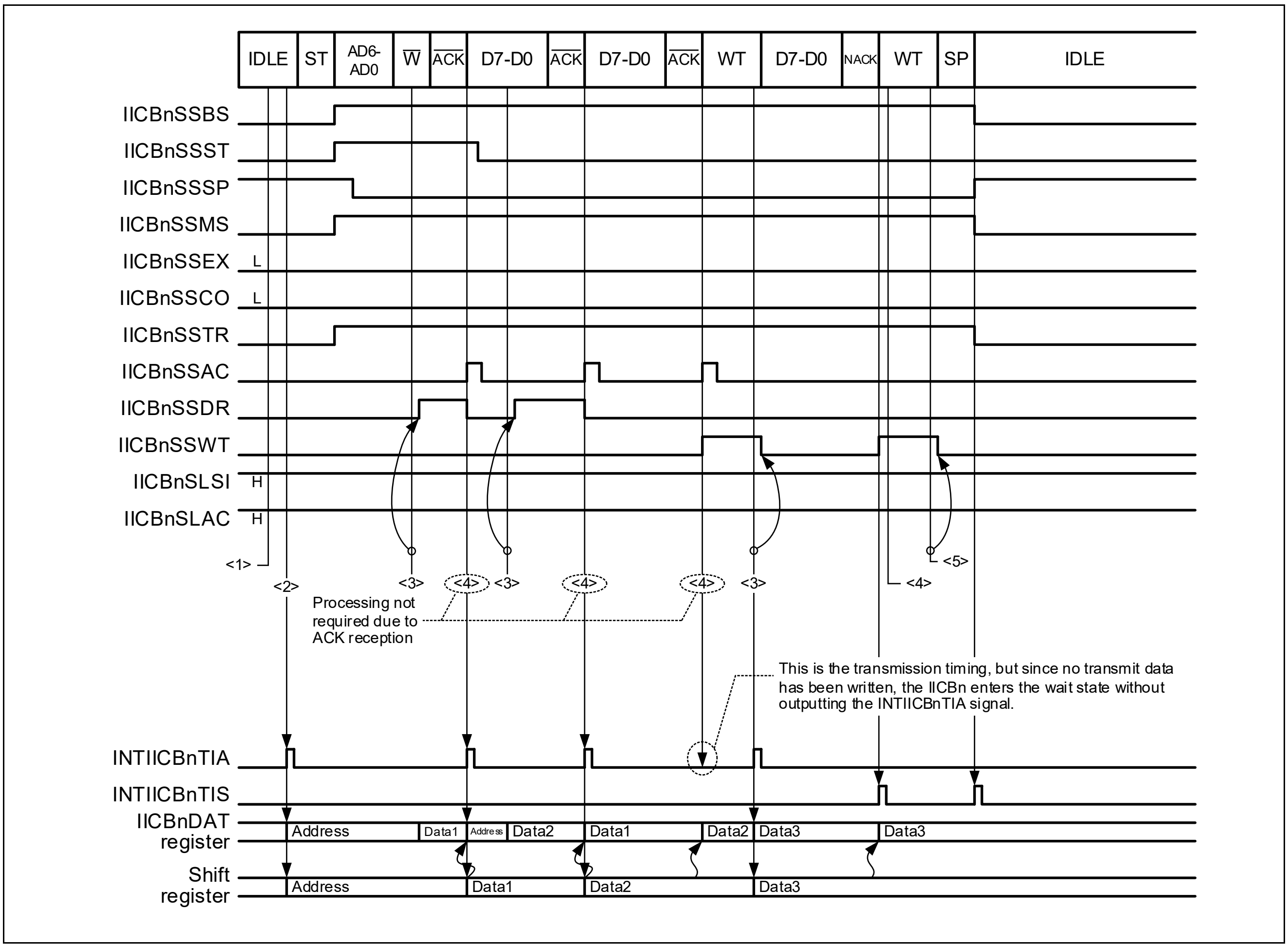
<4> Acknowledge result check
If there is no unread data in the IICBnDAT register by the time reception starts, the IICBn starts reception without entering the wait state.

<5> Receive data load
Read the receive data from the IICBnDAT register using the INTIICBnTIA interrupt.

<6> Data reception completion processing
By clearing the IICBnCTL0.IICBnSLAC bit (to 0) before reading the receive data immediately preceding the final receive data, the next $\overline{ACK}$ is not output and the end of the data is notified to the transmitting device.

<7> Stop condition output
Set the IICBnTRG.IICBnSPT bit (to 1).

(2) Example of communications in continuous transfer mode (master transmission)



<1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).

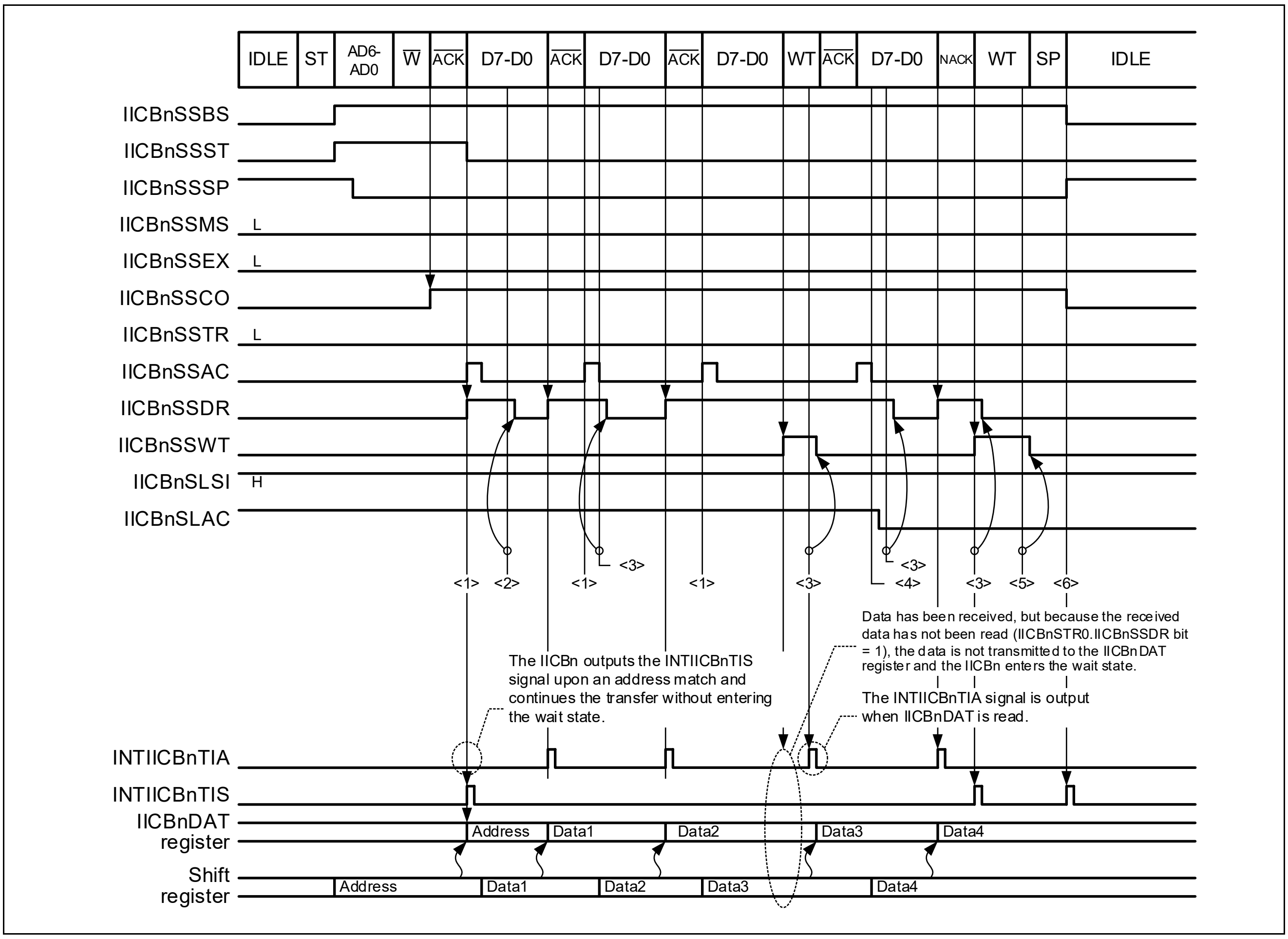
<2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.

<3> Data transmission
Set the transmit data to the IICBnDAT register using the INTIICBnTIA interrupt.

<4> Acknowledge result check
The INTIICBnTIS interrupt occurs only if the slave device does not return $\overline{ACK}$. Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit.

<5> Stop condition output
Set the IICBnTRG.IICBnSPT bit (to 1).

(3) Example of communications in continuous transfer mode (slave reception)



<1> Data reception
If there is no unread data in the IICBnDAT register by the time reception starts, the IICBn starts reception without entering the wait state.

<2> Operation mode check in slave mode
- Check the operation mode using the INTIICBnTIS interrupt.
- Check the address transfer, address match, and reception operation with the IICBnSTR0.IICBnSSST, IICBnSTR0.IICBnSSCO, and IICBnSTR0.IICBnSSTR bits.
- Read the IICBnDAT register (empty read).

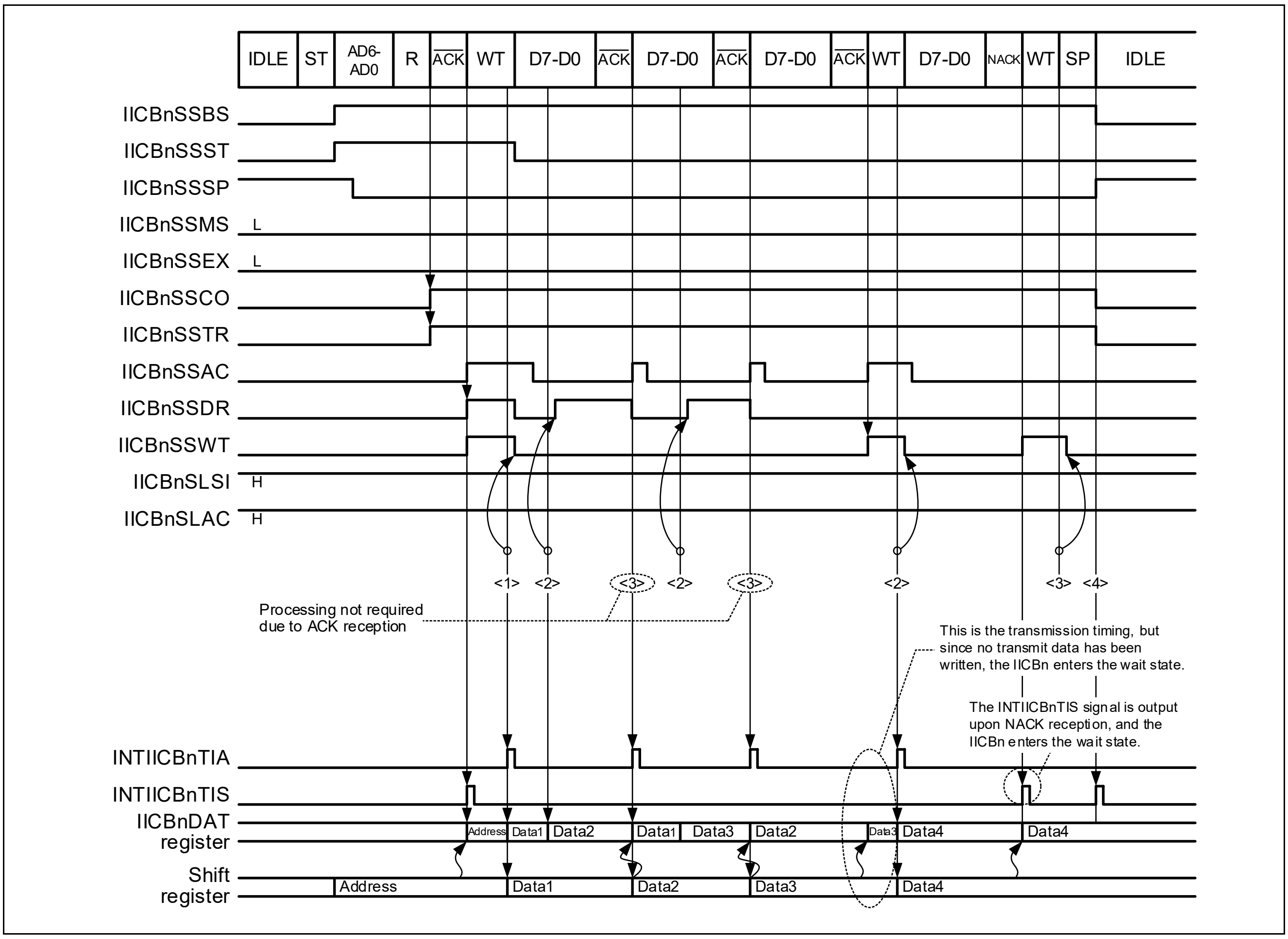
<3> Receive data load
Read the receive data from the IICBnDAT register using the INTIICBnTIA interrupt.

<4> Data reception completion processing <1>
By clearing the IICBnCTL0.IICBnSLAC bit (to 0) before reading the receive data immediately preceding the final receive data, the next $\overline{ACK}$ is not output and the end of the data is notified to the transmitting device.

<5> Data reception completion processing <2>
The INTIICBnTIS interrupt occurs only if the slave device does not return $\overline{ACK}$.
Exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1).

<6> Stop condition detection
Detect the stop condition using the INTIICBnTIS interrupt.

(4) Example of communications in continuous transfer mode (slave transmission)



<1> Operation mode check in slave mode

- Check the operation mode using the INTIICBnTIS interrupt.
- Check the address transfer, address match, and reception operation with the IICBnSTR0.IICBnSSST, IICBnSTR0.IICBnSSCO, and IICBnSTR0.IICBnSSTR bits.
- After reading (empty read) the IICBnDAT register, set the first transmit data to the IICBnDAT register.

<2> Data transmission

Set the transmit data to the IICBnDAT register using the INTIICBnTIA interrupt.

<3> Acknowledge result check

The INTIICBnTIS interrupt occurs only if the slave device does not return $\overline{ACK}$. Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit.

If $\overline{ACK}$ is not output, the transmission is judged to have been completed, and the IICBn exits the wait state by setting the IICBnTRG.IICBnWRET bit (to 1).

<4> Stop condition detection

Detect the stop condition using the INTIICBnTIS interrupt.

24.6.3 Arbitration

When the IICBn operates as the master device and loses arbitration, it enters the slave standby state by setting both SCLn and SDAn to high level upon detection of the arbitration loss, and then the IICBnSTR0.IICBnALDF bit is set (to 1) each time the status interrupt request signal (INTIICBnTIS) is output.

(1) Status upon occurrence of arbitration

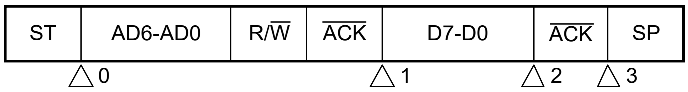
The statuses upon occurrence of arbitration during master device operation (IICBnSTR0.IICBnSSMS bit = 1) are listed below.

(1) Address transmission
(2) R/$\overline{\text{W}}$ bit transmission of address transfer
(3) Extension code transmission
(4) R/$\overline{\text{W}}$ bit transmission of extension code transfer
(5) Data transmission
(6) $\overline{\text{ACK}}$ bit transmission after data reception
(7) Start condition detection during address transfer or data transfer
(8) Stop condition detection during address transfer or data transfer
(9) The SDAn signal is low when the IICBn is attempting to output a restart condition
(10) The SDAn signal is low when the IICBn is attempting to output a stop condition
(11) The falling edge of the SCLn signal is detected when the IICBn is attempting to output a restart condition

24.6.4 Entering and Exiting Wait State

The IICBn enters the wait state at the following timings.

Table 24.6 Wait State Transit Timings



Timing	Description	Refer to:
Δ0	Upon detection of the first falling edge of the SCLn, following detection of start condition as the master device	(1) "Wait state at falling edge of first SCLn after IICBn became master"
Δ1	Upon detection of the falling edge of the 9th SCLn during address transfer after the start condition	(2) "Wait state upon completion of address transfer"
Δ2	Upon detection of the falling edge of the 8th SCLn during data transfer	(3) "Wait state upon detection of the falling edge of the 8th SCLn during data transfer"
Δ3	Upon detection of the falling edge of the 9th SCLn during data transfer	(4) "Wait state upon detection of the falling edge of the 9th SCLn during data transfer"

Remark: **ST : Start condition**
AD6–AD0 : Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7–D0 : Data
SP : Stop condition

The method to exit the wait state differs according to the wait state.

Exit the wait state by applying the appropriate method for each of the four wait states as described below.

(1) Wait state at falling edge of first SCLn after IICBn became master

Δ0 indicates the wait state when the data to be transferred has not been written (to the IICBnDAT register) when the falling edge of the first SCLn after the IICBn became the master is detected, after 1 was written to the IICBnTRG.IICBnSTT bit.

(a) Wait state transit condition

The IICBn enters the wait state if data is not written to the IICBnDAT register in the period from when the IICBnTRG.IICBnSTT bit becomes 1 until the Δ0 timing, upon detection of the first falling edge of SCLn after the IICBn became master, after 1 was written to the IICBnTRG.IICBnSTT bit.

However, the valid times to write data to the IICBnDAT register (without entering the wait state) after 1 was written to the IICBnTRG.IICBnSTT bit differ depending on whether the communication reservation function is enabled. The valid times to write to the IICBnDAT register for each of these cases are shown in Figure 24.11.

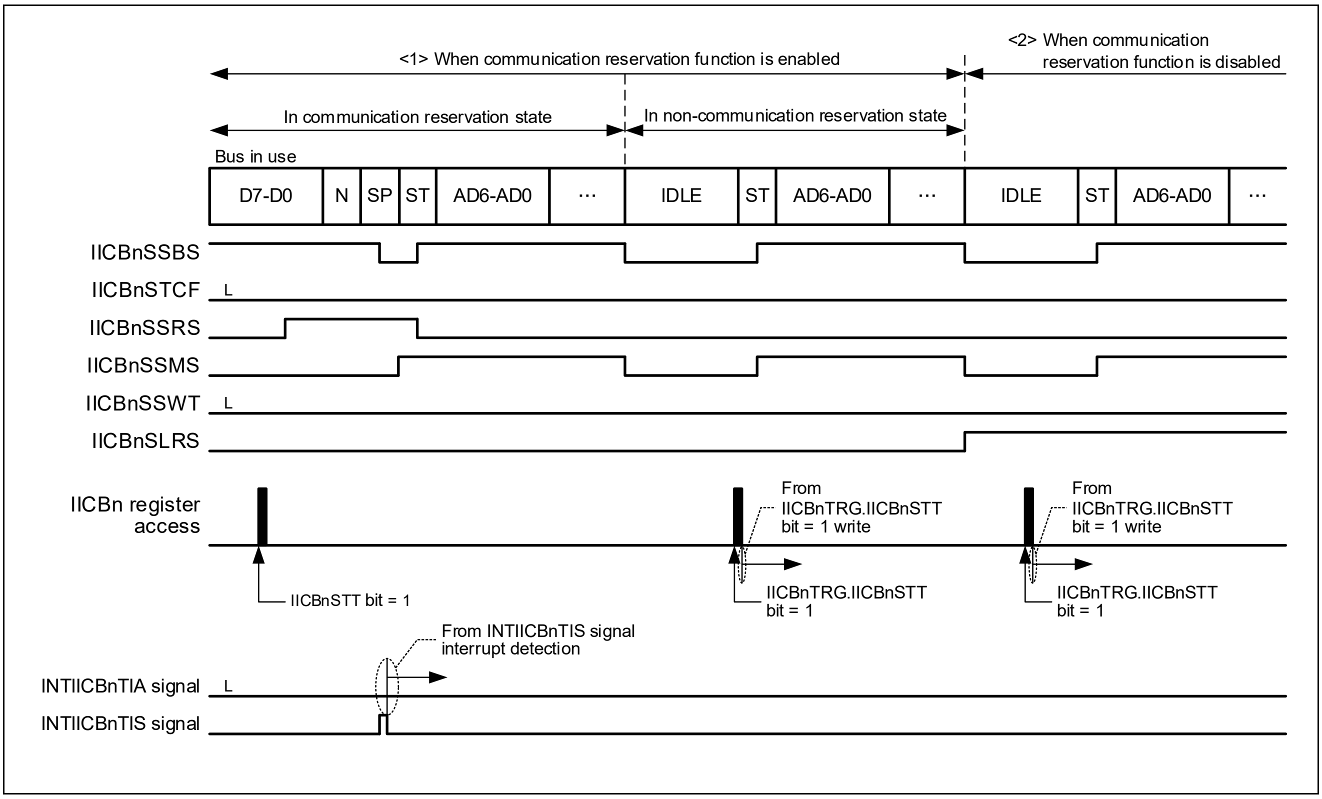


Figure 24.11 Valid Times to Write to IICBnDAT Register

Caution: The communication reservation function is disabled (<2> in the above figure) while the IICBnSTR0.IICBnSTCF bit is 0.
When the IICBnSTR0.IICBnSTCF bit becomes 1, setting from IICBnSTR0.IICBnSTCF bit = 1 write is required again.

(b) Wait state exit conditions

Exit the wait state by writing to the IICBnDAT register.

(2) Wait state upon completion of address transfer

Δ1 indicates the wait state entered upon completion of address transfer.

(a) Wait state transit condition

<Single transfer mode>

In single transfer mode, the IICBn always enters the wait state while it operates as the master.

While the IICBn operates as a slave, it enters the wait state upon an address match, or upon extension code detection while the IICBnCTL0.IICBnSLWT bit is 1.

<Continuous transfer mode>

In continuous transfer mode, the IICBn enters the wait state in the following cases.

- Upon detection of NACK
- When the IICBn operates as the master and transmits data, if the data to be transferred next has not been written
- When the IICBn operates as a slave, if the received data has not been read, or during transmission

(b) Wait state exit conditions

<Single transfer mode>

Exit the wait state by writing to the IICBnDAT register during transmission, or by writing 1 to the IICBnTRG.IICBnWRET bit during reception. When the IICBn operates as the master and the IICBnSTR0.IICBnSSAC bit is 0 or the IICBn is at the transmission side, the wait state can be released by writing 1 to the IICBnTRG.IICBnSTT or the IICBnTRG.IICBnSPT bit.

<Continuous transfer mode>

Exit the wait state by writing to the IICBnDAT register during transmission, or by reading the IICBnDAT register during reception. When the IICBn operates as the master and the IICBnSTR0.IICBnSSAC bit is 0, the wait state can be released by writing 1 to the IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT bit.

(3) Wait state upon detection of the falling edge of the 8th SCLn during data transfer

Δ2 indicates the wait state entered upon detection of the falling edge of the 8th SCLn during data transfer.

(a) Wait state transit condition

<Single transfer mode>

When the IICBn participates in communications and the IICBnCTL0.IICBnSLWT bit is 0, the IICBn enters the wait state if the falling edge of the 8th SCLn is detected.

<Continuous transfer mode>

When the IICBn participates in communications and the IICBnSTR0.IICBnSSTR bit is 0, the IICBn enters the wait state if processing of the previous data (read from the IICBnDAT register) has not completed and 1 has not been written to the IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT bit before the falling edge of the 8th SCLn.

(b) Wait state exit conditions

<Single transfer mode>

Exit the wait state by writing to the IICBnDAT register during transmission, or by writing 1 to the IICBnTRG.IICBnWRET bit during reception.

<Continuous transfer mode>

Exit the wait state by reading the IICBnDAT register.

(4) Wait state upon detection of the falling edge of the 9th SCLn during data transfer

Δ3 indicates the wait state entered upon detection of the falling edge of the 9th SCLn during data transfer.

During continuous transfer mode, the IICBn enters the wait state upon NACK reception.

(a) Wait state transit condition

<Single transfer mode>

When the IICBn participates in communications and the IICBnCTL0.IICBnSLWT bit is 1, the IICBn enters the wait state if the falling edge of the 9th SCLn is detected.

<Continuous transfer mode>

When the IICBn participates in communications, it enters the wait state in the following three cases during data transmission:

- Upon reception of NACK by $\overline{\text{ACK}}$ bit while the IICBnCTL0.IICBnSLWT bit is 1
- When transmit data is not written to the data register during transmission
- When the previous received data is not read during reception

(b) Wait state exit conditions

The wait state exit conditions are listed for each transfer mode in Table 24.7.

Table 24.7 Wait State Exit Conditions

Master/Slave	Transfer Mode	Transfer Direction	IICBnSTR0. IICBnSSAC Bit	Exit Conditions
Master	Single transfer mode	Reception	0	IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
			1	IICBnTRG.IICBnWRET = 1
		Transmission	0	IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
			1	Write to IICBnDAT register or IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
	Continuous transfer mode	Reception	0	IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
			1	Read from IICBnDAT register [Note 1]
		Transmission	0	IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
			1	Write to IICBnDAT register [Note 2]
Slave	Single transfer mode	Reception	—	IICBnTRG.IICBnWRET bit = 1
		Transmission	0	IICBnTRG.IICBnWRET bit = 1
			1	Write to IICBnDAT register [Note 1]
	Continuous transfer mode	Reception	0	IICBnTRG.IICBnWRET bit = 1
		Transmission	0	IICBnTRG.IICBnWRET bit = 1
			1	Write to IICBnDAT register

Notes 1. Condition for exiting the wait state that was entered when no transmit data has been written to the data register

2. Condition for exiting the wait state that was entered when the received data has not been read

24.6.5 Extension Code

The processing when the extension code is received differs according to the data after the extension code and thus must be executed through the user's software.

Therefore, the operation differs from that during normal slave address reception. These differences are described below.

(1) When the upper 4 bits of the received address are 0000 or 1111, the extension code reception flag (IICBnSTR0.IICBnSSEX bit) is set to 1 to indicate that an extension code has been received. The status interrupt request signal (INTIICBnTIS) is output at the falling edge of the 8th clock, and the IICBn enters the wait state (IICBnTRG.IICBnSSWT = 1). The IICBnSTR0.IICBnSSDR and IICBnSTR0.IICBnSSTR bits are then set (to 1).

(2) During address transfer, the acknowledge output can be controlled by setting the IICBnCTL0.IICBnSLAC bit. (Note that an acknowledge is always output upon an address match, regardless of the setting of this bit, during address transfer for normal slave address reception.)

(3) The method for exiting the wait state entered upon extension code detection depends on the setting of the IICBnCTL0.IICBnMDTX1 bit as follows.

<When IICBnCTL0.IICBnMDTX1 bit is 0>

During transmission while the IICBnCTL0.IICBnSLWT bit is 0, exit the wait state by writing to the IICBnDAT register. During transmission while the IICBnCTL0.IICBnSLWT bit is 1, or during reception, exit the wait state by writing 1 to the IICBnTRG.IICBnWRET bit.

<When IICBnCTL0.IICBnMDTX1 bit is 1>

During transmission, exit the wait state by writing to the IICBnDAT register, and, during reception, exit the wait state by reading from the IICBnDAT register.

(4) At the falling edge of the 9th clock, if the IICBnCTL0.IICBnSLWT bit is 1, the interrupt request signal (INTIICBnTIA) is output and the IICBn enters the wait state (IICBnTRG.IICBnSSWT = 1). If the IICBnCTL0.IICBnSLWT bit is 0, the interrupt request signal (INTIICBnTIA) is not output and the IICBn does not enter the wait state.

(5) If the IICBn receives an extension code, it participates in communications even if the addresses do not match.

For example, to avoid operating the IICBn as a slave device after receiving an extension code, set the IICBnTRG.IICBnLRET bit to 1. The IICBn enters the standby state for the next communication.

24.7 Interrupt Request Signals

**Caution: In this section, the operation when an extension code is received is omitted.
For details about the extension code, refer to section 24.6.5, Extension Code.**

The IICBn has two interrupt request signals, the data transmit/receive interrupt request signal (INTIICBnTIA) and the status interrupt request signal (INTIICBnTIS).

Both signals are pulses of one PCLK clock width. The interrupt request signal output timing differs according to the transfer mode set using the IICBnCTL0.IICBnMDTX1 and IICBnCTL0.IICBnMDTX0 bits. The interrupt request signals are explained below for each transfer mode.

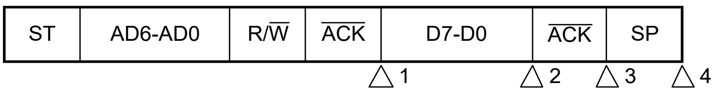
To perform transfer with an address match between the master device and the slave device, select single transfer mode or continuous transfer mode with the IICBnCTL0.IICBnMDTX0 bit, and to perform transfer with extension code detection by the slave, select single transfer mode or continuous transfer mode using the IICBnCTL0.IICBnMDTX1 bit.

24.7.1 Single Transfer Mode

The interrupt request signal timing in single transfer mode is described in Table 24.8 below.

During single transfer mode, for the INTIICBnTIA and INTIICBnTIS interrupt request signals, whether to output an interrupt is judged based on the IICBn state when the falling edge of SCLn is detected during the bus cycle. Note, however, that whether to output an interrupt is judged based on the IICBn state when a stop condition is detected at the Δ4 timing.

Table 24.8 Interrupt Request Signal Output Timing (Single Transfer Mode)



Output Timing	Description	Refer to:
Δ1	Upon detection of the falling edge of the 9th SCLn during address transfer	24.7.1 (1)
Δ2	Upon detection of the falling edge of the 8th SCLn during data transfer	24.7.1 (2)
Δ3	Upon detection of the falling edge of the 9th SCLn during data transfer	24.7.1 (2)
Δ4	Upon detection of a stop condition	24.7.1 (3)

Remark:
- **ST : Start condition**
- **AD6–AD0 : Address**
- **R/$\overline{W}$: Transfer direction specification**
- **$\overline{ACK}$: Acknowledge**
- **D7–D0 : Data**
- **SP : Stop condition**

(1) Interrupt request signal output conditions and output interrupt request signals during address transfer

Δ1 in Table 24.8 indicates the interrupt request signal output timing during an address transfer.

Table 24.9 indicates the interrupt request signal output condition and the interrupt request signal that is output (INTIICBnTIA or INTIICBnTIS) at the timing of Δ1.

Table 24.9 Interrupt Request Signal Output Conditions and Interrupt Request Signals Output during Address Transfer (Single Transfer Mode)

IICBn SSMS	IICBn ALDF	IICBn SLWT	IICBn SSCO	Δ1		Remark
				Interrupt	Wait	
1	0	X	X	INTIICBnTIA	Wait	—
1	1	X	X	This state does not exist.		—
0	0	X	0	INTIICBnTIS Note	—	After restart, non-participation in communications
0	0	X	1	IICBITAn	Wait	—
0	1	X	0	INTIICBnTIS	—	After arbitration loss, non-participation in communications
0	1	X	1	INTIICBnTIA	Wait	—

Note: In case of an address match or extension code detection, before the restart condition

Remark: X: don’t care

(2) Interrupt request signal output conditions and interrupt request signals output during data transfer

Δ2 and Δ3 in Table 24.8 indicate the interrupt request signal output timing during a data transfer. The interrupt request signal output timing of Δ2 or Δ3 is determined according to the setting of the IICBnCTL0.IICBnSLWT bit. Table 24.10 indicates the interrupt request signal output condition and the interrupt request signal that is output (INTIICBnTIA or INTIICBnTIS) at the timing of Δ2 and Δ3.

Table 24.10 Interrupt Request Signal Output Conditions and Interrupt Request Signals Output during Data Transfer (Single Transfer Mode)

IICBn SSMS	IICBn ALDF	IICBn SLWT	IICBn SSCO	Δ2		Δ3		Remark
				Interrupt	Wait	Interrupt	Wait	
1	0	0	X	INTIICBnTIA	Wait	—	—	—
1	0	1	X	—	—	INTIICBnTIA	Wait	—
1	1	X	X	This state does not exist.				—
0	0	X	0	—	—	—	—	Non-participation in communications
0	0	0	1	INTIICBnTIA	Wait	—	—	—
0	0	1	1	—	—	INTIICBnTIA	Wait	—
0	1	0	0	IICBITSn	—	—	—	Non-participation in communications after arbitration loss
0	1	1	0	—	—	INTIICBnTIS	—	Non-participation in communications after arbitration loss
0	1	0	1	INTIICBnTIA	Wait	—	—	—
0	1	1	1	—	—	INTIICBnTIA	Wait	—

Remark: X: don’t care

(3) Interrupt request signal output upon stop condition detection

Δ4 in Table 24.8 indicates the interrupt request signal output timing upon detection of a stop condition.

Interrupt request signal output is controlled according to the IICBnCTL0.IICBnSLSI bit. If a stop condition is detected while the IICBnCTL0.IICBnSLSI bit is 1, the status interrupt request signal (INTIICBnTIS) is output.

24.7.2 Continuous Transfer Mode

(1) Data transmit/receive interrupt request signal (INTIICBnTIA)

The conditions for outputting an INTIICBnTIA signal in continuous transfer mode are described below.

- Interrupt request signal output condition during reception

  When receive data is saved from the shift register to the IICBnDAT register (timing <1> in Figure 24.12)

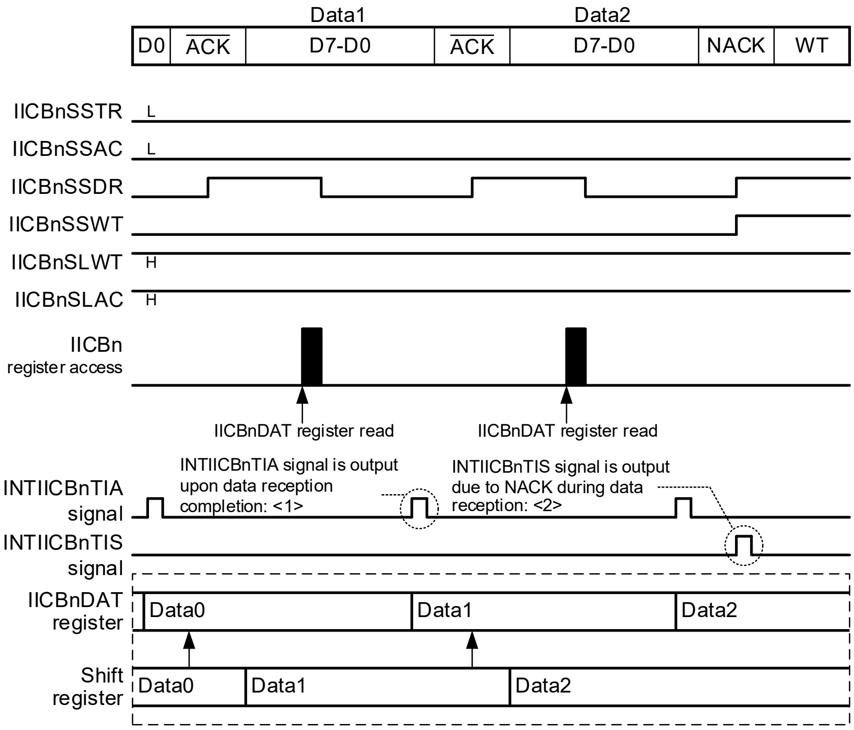


Figure 24.12 INTIICBnTIA Signal Output Timing (Reception in Continuous Transfer Mode)

- Interrupt request signal output condition during transmission

  When data is written to the IICBnDAT register while there is no transmit data in the shift register and IICBnDAT register (timing <2> in Figure 24.13).

  When data is saved from the IICBnDAT register to the shift register (timing <1> in Figure 24.13).

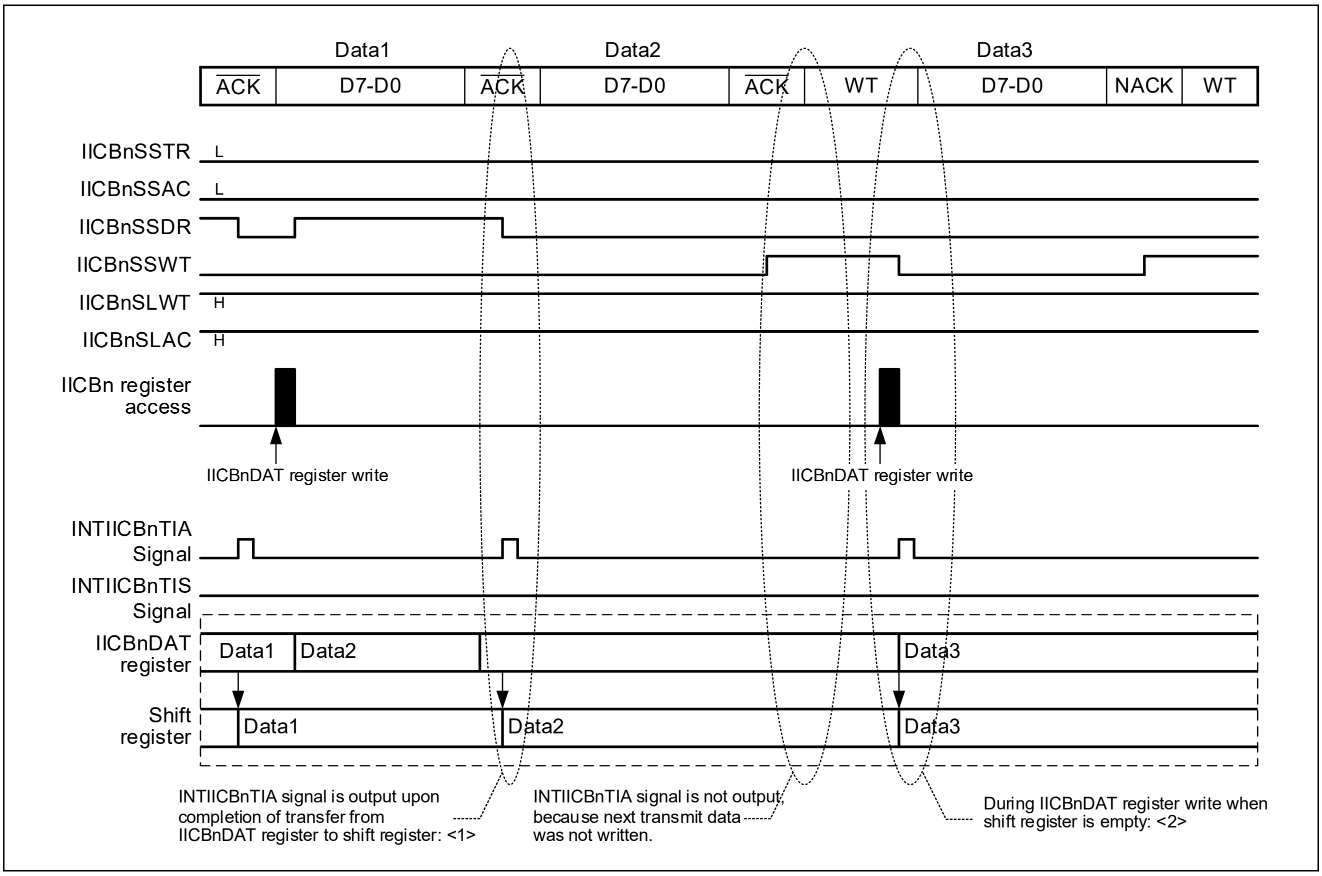
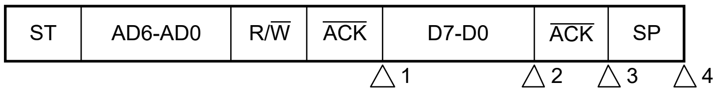


Figure 24.13 INTIICBnTIA Signal Output Timing (Transmission in Continuous Transfer Mode)

(2) Status interrupt request signal (INTIICBnTIS)

The INTIICBnTIS signal output timing in continuous transfer mode is the same as that in single transfer mode.

Table 24.11 INTIICBnTIS Signal Output Timing



Output Timing	Description	Refer to:
Δ1	Upon detection of the falling edge of the 9th SCLn during address transfer after the start condition	24.7.2 (a)
Δ2	Upon detection of the falling edge of the 8th SCLn during data transfer	24.7.2 (b)
Δ3	Upon detection of the falling edge of the 9th SCLn during data transfer	24.7.2 (b)
Δ4	Upon detection of a stop condition	24.7.2 (c)

Remark:
ST : Start condition
AD6–AD0 : Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7–D0 : Data
SP : Stop condition

(a) INTIICBnTIS signal output conditions during address transfer

Δ1 in Table 24.11 indicates the INTIICBnTIS signal output timing during address transfer. Table 24.12 indicates the INTIICBnTIS signal output conditions at the Δ1 timing.

Table 24.12 INTIICBnTIS Signal Output Conditions during Address Transfer (Continuous Transfer Mode)

IICBn SSMS	IICBn SSCO	IICBn ALDF	Transfer direction	IICBn SSDR	IICBn SSAC	Δ1	
						Interrupt	Wait
1	X	0	Transmission	0	1	—	Wait
1	X	0	Transmission	0	0	INTIICBnTIS	Wait
1	X	0	Transmission	1	1	—	—
1	X	0	Transmission	1	0	INTIICBnTIS	Wait
1	X	0	Reception	0	1	—	—
1	X	0	Reception	0	0	INTIICBnTIS	Wait
1	X	0	Reception	1	1	INTIICBnTIS during IICBnDAT read Note 1	Wait
1	X	0	Reception	1	0	INTIICBnTIS during IICBnDAT read	Wait
1	X	1	X	X	X	This state does not exist.	
0	0	0	X	X	X	INTIICBnTIS Note 2	—
0	0	1	X	X	X	INTIICBnTIS	—
0	1	X	Transmission	X	1	INTIICBnTIS	Wait
0	1	X	Reception	0	1	INTIICBnTIS	—
0	1	X	Reception	1	1	INTIICBnTIS during IICBnDAT read	Wait

Notes 1. Upon restarting without reading IICBnDAT after the reception ends

2. Upon an address match before restart condition

Caution: For Δ1, the IICBnSTR0.IICBnSSAC bit is always 0.

Remark: X: don't care

(b) INTIICBnTIS signal output conditions during data transfer

Δ2 and Δ3 in Table 24.11 indicate the INTIICBnTIS signal output timings during data transfer. Table 24.13 indicates the INTIICBnTIS signal output conditions at the Δ2 and Δ3 timings.

Table 24.13 INTIICBnTIS Signal Output Conditions during Data Transfer (Continuous Transfer Mode)

IICBn SSMS	IICBn SSCO	IICBn SLWT	IICBn ALDF	Transfer direction	IICBn SSDR	IICBn SSAC	IICBnSTT or IICBnSPT	Δ2		Δ3	
								Interrupt	wait	Interrupt	wait
1	X	0	X	Transmission	0	1	Note 1	—	—	—	wait
1	X	0	X	Transmission	0	0	Note 1	—	—	INTIICBnTIS	wait
1	X	0	X	Transmission	1	1	Note 1	—	—	—	—
1	X	0	X	Transmission	1	0	Note 1	—	—	INTIICBnTIS	wait
1	X	0	X	Reception	0	1	Note 1	—	—	—	—
1	X	0	X	Reception	0	0	Note 1	—	—	INTIICBnTIS	wait
1	X	0	X	Reception	1	1	Note 1	—	—	—	—
1	X	0	X	Reception	1	0	Note 1	—	—	INTIICBnTIS after IICBnDAT read	wait
1	X	X	X	X	X	0	Note 2	—	—	INTIICBnTIS	—
1	X	X	X	X	X	1	Note 2	—	—	—	—
0	0	X	0	X	X	X	X	—	—	—	—
0	0	0	1	Reception	X	X	X	INTIICBn TIS	—	—	—
0	0	1	1	Transmission	X	X	X	—	—	INTIICBnTIS	—
0	1	0	X	Transmission	0	1	Note 1	—	—	—	wait
0	1	0	X	Transmission	0	0	Note 1	—	—	INTIICBnTIS	wait
0	1	0	X	Transmission	1	1	Note 1	—	—	—	—
0	1	0	X	Transmission	1	0	Note 1	—	—	INTIICBnTIS	wait
0	1	0	X	Reception	0	1	Note 1	—	—	—	—
0	1	0	X	Reception	0	0	Note 1	—	—	INTIICBnTIS	wait
0	1	0	X	Reception	1	1	Note 1	—	—	—	—
0	1	0	X	Reception	1	0	Note 1	—	—	INTIICBnTIS during IICBnDAT read	wait

Notes 1. When 1 has not been written to the IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT bit

2. When 1 has been written to the IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT bit

Remark: X: don't care

(c) INTIICBnTIS signal output upon detection of stop condition

Δ4 in Table 24.11 indicates the INTIICBnTIS signal output timing upon detection of a stop condition. INTIICBnTIS signal output is controlled according to the IICBnCTL0.IICBnSLSI bit.

If a stop condition is detected while the IICBnCTL0.IICBnSLSI bit is 1, the INTIICBnTIS signal is output.

24.8 Interrupt Outputs and States

This section describes the states of the IICBnSTR0 register during interrupt output by communication flow.

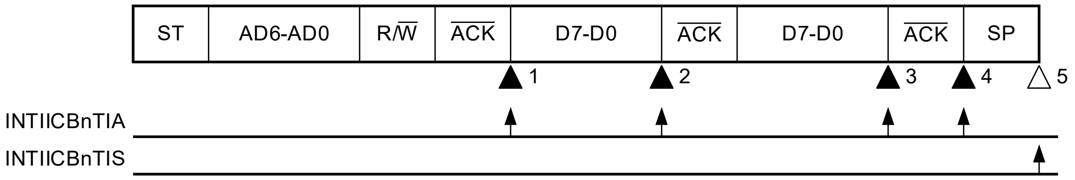
The meanings of the symbols used in the figures are as follows.

ST	: Start condition
AD6–AD0	: Address
R, $\overline{\text{W}}$, R/$\overline{\text{W}}$	: Transfer direction specification
$\overline{\text{ACK}}$	: Acknowledge
NACK	: Not acknowledge
D7–D0	: Data
SP	: Stop condition

24.8.1 Single Transfer Mode (Master Device Operation)

(1) Start – Address – Data – Data – Stop (normal transmission/reception)

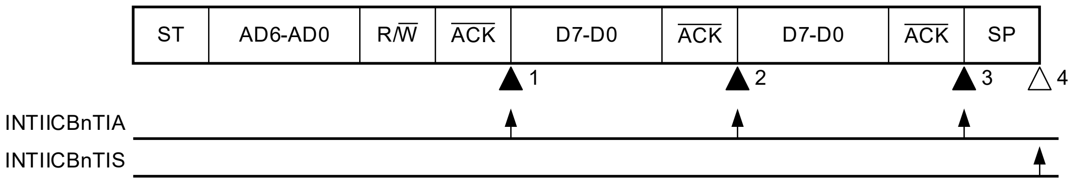
(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-0100X1 0110--00B
▲2: IICBnSTR0 register = 1-0100X0 0100--00B
▲3: IICBnSTR0 register = 1-0100X0 0100--00B (IICBnCTL0.IICBnSLWT bit= 1)
▲4: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don’t care**

(b) When IICBnCTL0.IICBnSLWT bit is 1

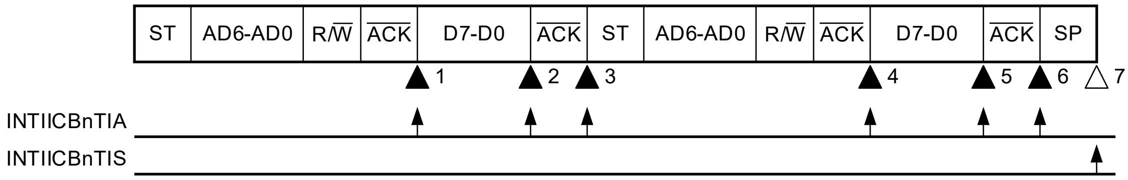


▲1: IICBnSTR0 register = 1-0100X1 0110--00B
▲2: IICBnSTR0 register = 1-0100X1 0100--00B
▲3: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit= 1)
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don’t care**

(2) Start – Address – Data – Start – Address – Data – Stop (restart)

(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-0100X1 0110--00B

▲2: IICBnSTR0 register = 1-0100X0 0100--00B (IICBnCTL0.IICBnSLWT bit=1)

▲3: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSTT bit= 1, IICBnCTL0.IICBnSLWT bit= 0)

▲4: IICBnSTR0 register = 1-0100X1 0110--00B

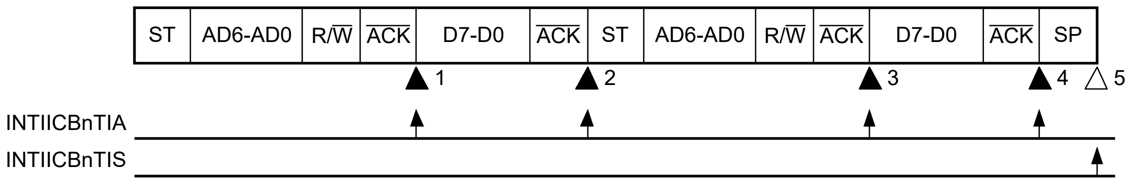
▲5: IICBnSTR0 register = 1-0100X0 0100--00B (IICBnCTL0.IICBnSLWT bit= 1)

▲6: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit= 1)

△7: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-0100X1 0110--00B

▲2: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSTT bit = 1)

▲3: IICBnSTR0 register = 1-0100X1 0110--00B

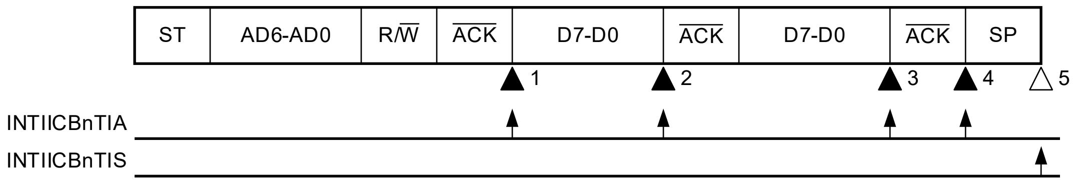
▲4: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)

△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(3) Start – Code – Data – Data – Stop (extension code transmission)

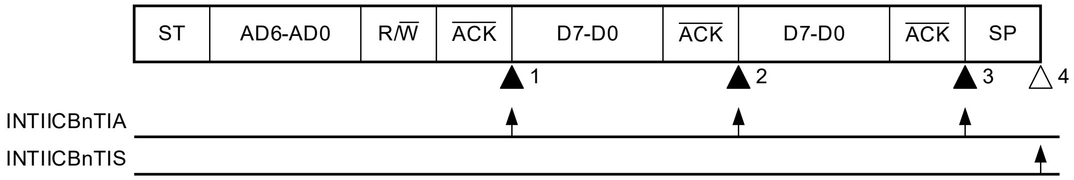
(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-0110X1 0110--00B
▲2: IICBnSTR0 register = 1-0110X0 0100--00B
▲3: IICBnSTR0 register = 1-0110X0 0100--00B (IICBnCTL0.IICBnSLWT bit= 1)
▲4: IICBnSTR0 register = 1-0110XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1



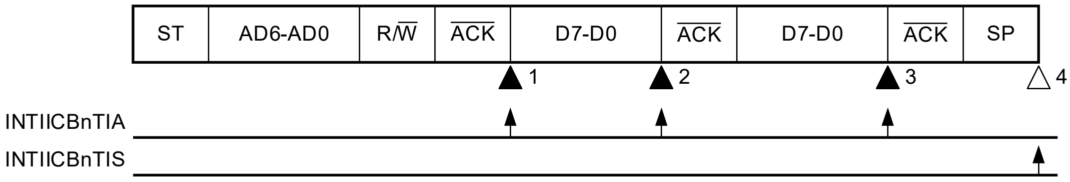
▲1: IICBnSTR0 register = 1-0110X1 0110--00B
▲2: IICBnSTR0 register = 1-0110X1 0100--00B
▲3: IICBnSTR0 register = 1-0110XX 0100--00B (IICBnTRG.IICBnSPT bit= 1)
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

24.8.2 Single Transfer Mode (Slave Device Operation: during Slave Address Reception (IICBnSTR0.IICBnSSC0 bit = 1))

(1) Start – Address – Data – Data – Stop

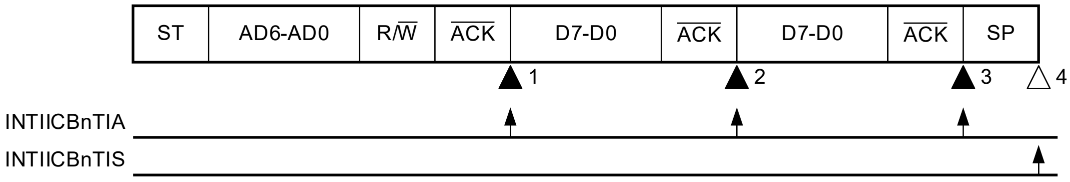
(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-0101X1 0110--00B
▲2: IICBnSTR0 register = 0-0101X0 0100--00B
▲3: IICBnSTR0 register = 0-0101X0 0100--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1

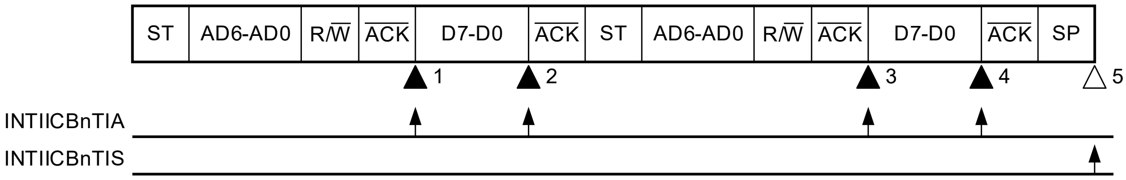


▲1: IICBnSTR0 register = 0-0101X1 0110--00B
▲2: IICBnSTR0 register = 0-0101X1 0100--00B
▲3: IICBnSTR0 register = 0-0101XX 0100--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(2) Start – Address – Data – Start – Address – Data – Stop

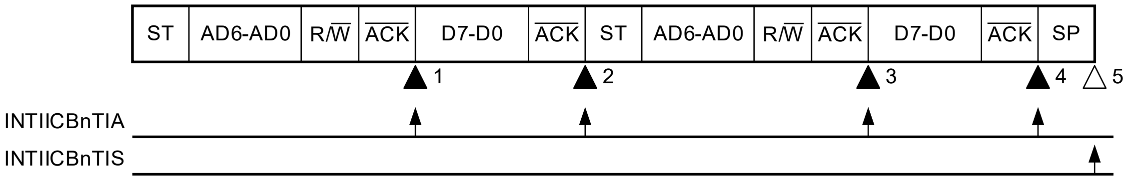
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)



▲1: IICBnSTR0 register = 0-0101X1 0110--00B
▲2: IICBnSTR0 register = 0-0101X0 0100--00B
▲3: IICBnSTR0 register = 0-0101X1 0110--00B
▲4: IICBnSTR0 register = 0-0101X0 0100--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address match)

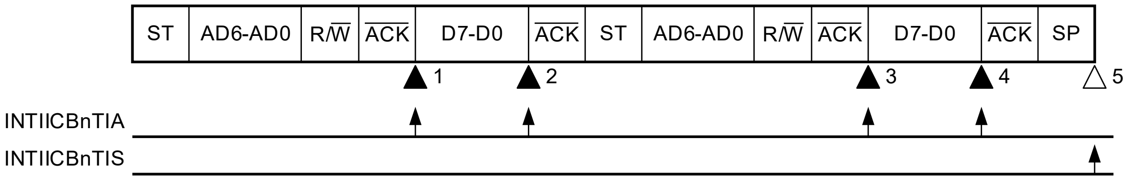


▲1: IICBnSTR0 register = 0-0101X1 0110--00B
▲2: IICBnSTR0 register = 0-0101XX 0100--00B
▲3: IICBnSTR0 register = 0-0101X1 0110--00B
▲4: IICBnSTR0 register = 0-0101XX 0100--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(3) Start – Address – Data – Start – Code – Data – Stop

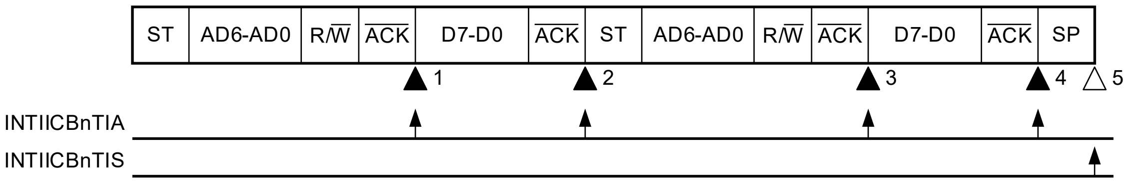
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-0101X1 0110--00B
▲2: IICBnSTR0 register = 0-0101X0 0100--00B
▲3: IICBnSTR0 register = 0-0110X1 0110--00B
▲4: IICBnSTR0 register = 0-0110X0 0100--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, extension code reception)

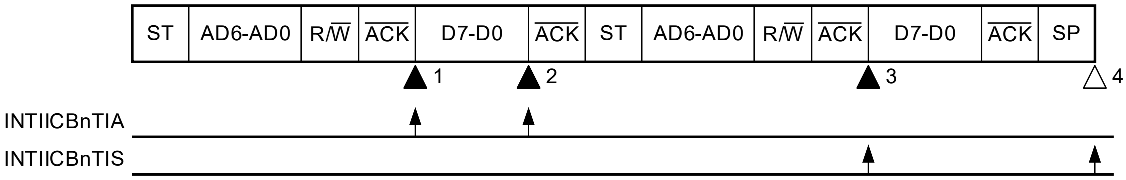


▲1: IICBnSTR0 register = 0-0101X1 0110--00B
▲2: IICBnSTR0 register = 0-0101XX 0100--00B
▲3: IICBnSTR0 register = 0-0110X1 0110--00B
▲4: IICBnSTR0 register = 0-0110XX 0100--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(4) Start – Address – Data – Start – Address – Data – Stop

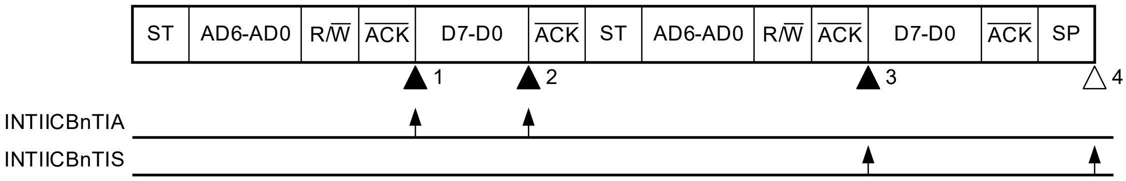
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-0101X1 0110--00B
▲2: IICBnSTR0 register = 0-0101X0 0100--00B
▲3: IICBnSTR0 register = 0-0000X0 0110--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-0101X1 0110--00B
▲2: IICBnSTR0 register = 0-0101X0 0100--00B
▲3: IICBnSTR0 register = 0-0000X0 0110--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

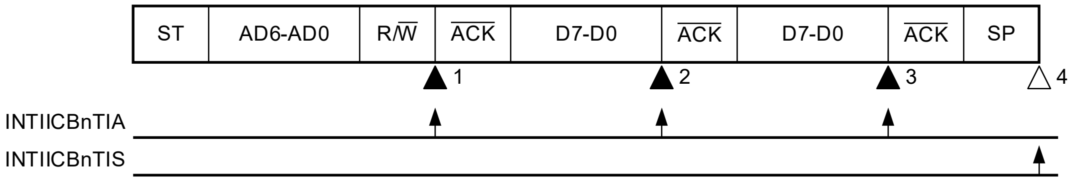
Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

24.8.3 Single Transfer Mode (Slave Device Operation: during Extension Code Reception (IICBnSTR0.IICBnSSEX bit = 1))

The IICBn always participates in communications when it receives an extension code.

(1) Start – Code – Data – Data – Stop

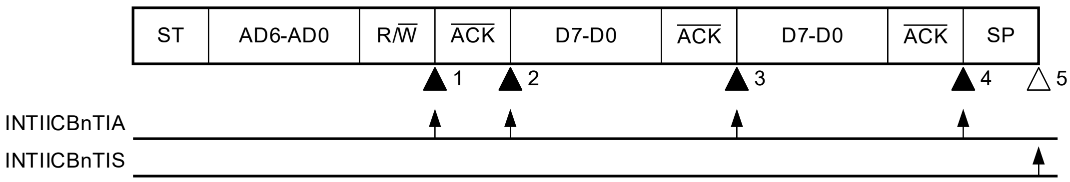
(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-0110X0 0110--00B
▲2: IICBnSTR0 register = 0-0110X0 0100--00B
▲3: IICBnSTR0 register = 0-0110X0 0100--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1

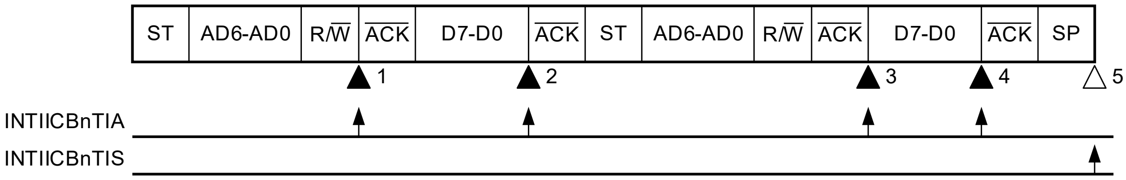


▲1: IICBnSTR0 register = 0-0110X0 0110--00B
▲2: IICBnSTR0 register = 0-0110X1 0110--00B
▲3: IICBnSTR0 register = 0-0110X0 0100--00B
▲4: IICBnSTR0 register = 0-0110XX 0100--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(2) Start – Code – Data – Start – Address – Data – Stop

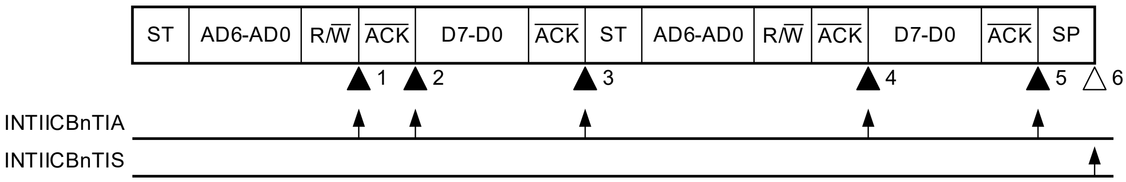
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)



▲1: IICBnSTR0 register = 0-0110X0 0110--00B
▲2: IICBnSTR0 register = 0-0110X0 0100--00B
▲3: IICBnSTR0 register = 0-0101X1 0110--00B
▲4: IICBnSTR0 register = 0-0101X0 0100--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address match)

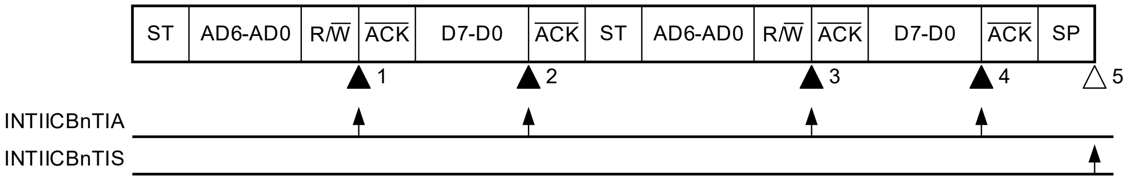


▲1: IICBnSTR0 register = 0-0110X0 0110--00B
▲2: IICBnSTR0 register = 0-0110X1 0110--00B
▲3: IICBnSTR0 register = 0-0110X0 0100--00B
▲4: IICBnSTR0 register = 0-0101X1 0110--00B
▲5: IICBnSTR0 register = 0-0101XX 0100--00B
△6: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(3) Start – Code – Data – Start – Code – Data – Stop

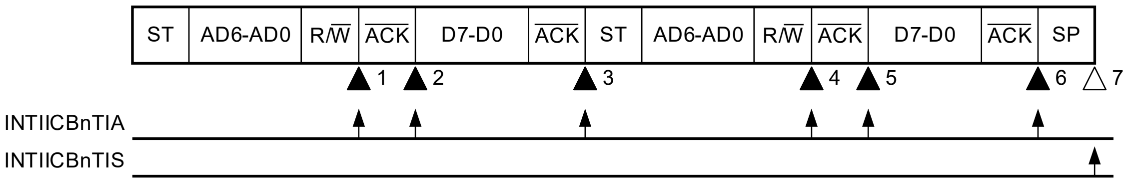
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-0110X0 0110--00B
▲2: IICBnSTR0 register = 0-0110X0 0100--00B
▲3: IICBnSTR0 register = 0-0110X0 0110--00B
▲4: IICBnSTR0 register = 0-0110X0 0100--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don’t care

(b) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, extension code reception)

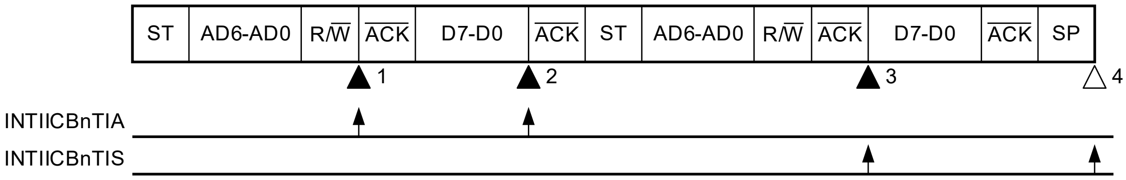


▲1: IICBnSTR0 register = 0-0110X0 0110--00B
▲2: IICBnSTR0 register = 0-0110X1 0110--00B
▲3: IICBnSTR0 register = 0-0110XX 0100--00B
▲4: IICBnSTR0 register = 0-0110X0 0110--00B
▲5: IICBnSTR0 register = 0-0110X1 0110--00B
▲6: IICBnSTR0 register = 0-0110XX 0100--00B
△7: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don’t care

(4) Start – Code – Data – Start – Address – Data – Stop

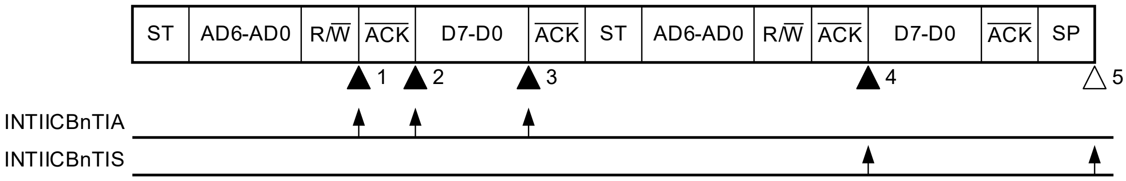
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-0110X0 0110--00B
▲2: IICBnSTR0 register = 0-0110X0 0100--00B
▲3: IICBnSTR0 register = 0-0000X0 0110--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address mismatch (extension code mismatch))

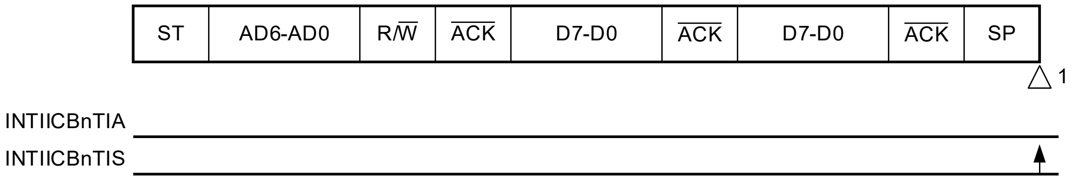


▲1: IICBnSTR0 register = 0-0110X0 0110--00B
▲2: IICBnSTR0 register = 0-0110X1 0110--00B
▲3: IICBnSTR0 register = 0-0000X0 0100--00B
▲4: IICBnSTR0 register = 0-0000X0 0110--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

24.8.4 Single Transfer Mode (Non-Participation in Communications)

(1) Start – Code – Data – Data – Stop



△1: IICBnSTR0 register = 0-000000 0001--00B

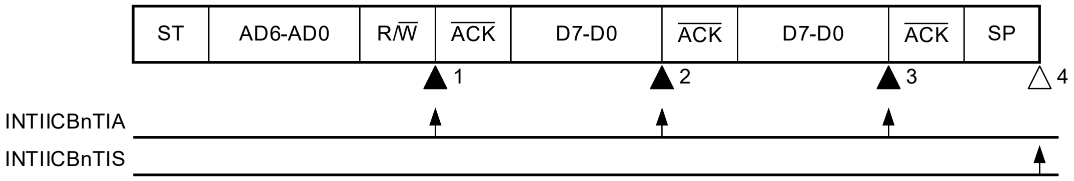
Remark: △ **Output only when IICBnCTL0.IICBnSLSI = 1**
- **Undefined**

24.8.5 Single Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1): Operation as Slave after Arbitration Loss)

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each INTIICBnTIS interrupt occurrence to confirm the arbitration result.

(1) Address match after arbitration loss

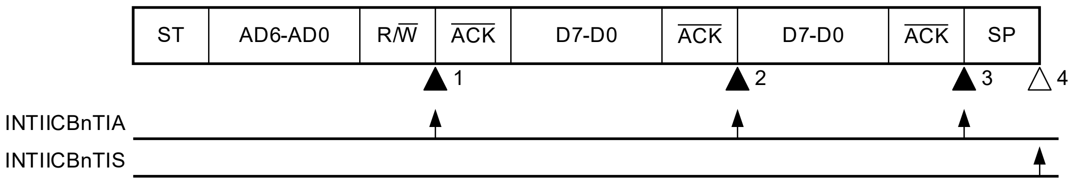
(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-0101X1 0110--01B (IICBnSTRC.IICBnCLAF bit= 1)
▲2: IICBnSTR0 register = 0-0101X0 0100--00B
▲3: IICBnSTR0 register = 0-0101X0 0100--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1

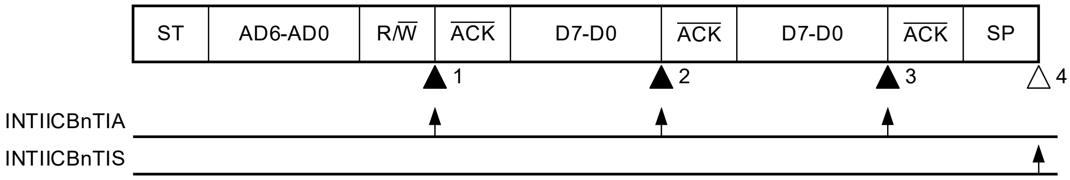


▲1: IICBnSTR0 register = 0-0101X1 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)
▲2: IICBnSTR0 register = 0-0101X1 0100--00B
▲3: IICBnSTR0 register = 0-0101XX 0100--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(2) Upon extension code detection after arbitration loss

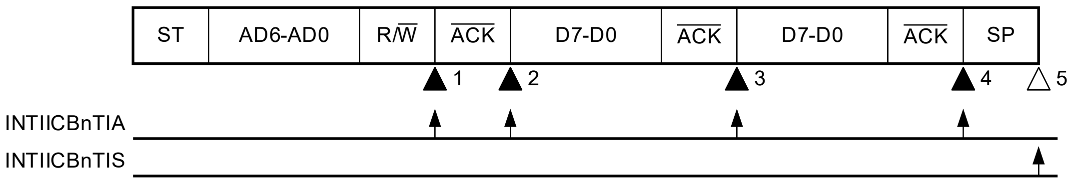
(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-0110X0 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)
▲2: IICBnSTR0 register = 0-0110X0 0100--00B
▲3: IICBnSTR0 register = 0-0110X0 0100--00B
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: **▲ Always output**
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(b) When IICBnCTL0.IICBnSLWT bit is 1



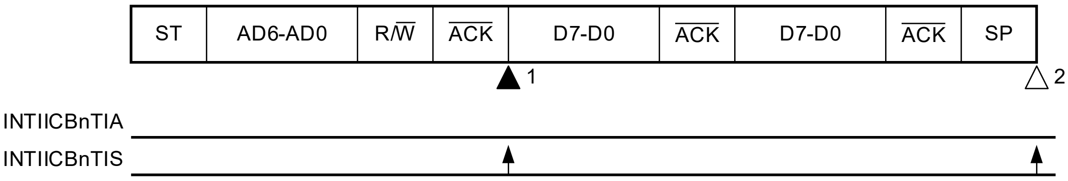
▲1: IICBnSTR0 register = 0-0110X0 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)
▲2: IICBnSTR0 register = 0-0110X1 0110--00B
▲3: IICBnSTR0 register = 0-0110X0 0100--00B
▲4: IICBnSTR0 register = 0-0110XX 0100--00B
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: **▲ Always output**
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

24.8.6 Single Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1): Non-Participation in Communications after Arbitration Loss)

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each INTIICBnTIS interrupt occurrence to confirm the arbitration result.

(1) Arbitration loss during transmission of slave address



▲1: IICBnSTR0 register = 0-0000X1 0110--01B (IICBnSTRC.IICBnCLAF bit= 1)

△2: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**

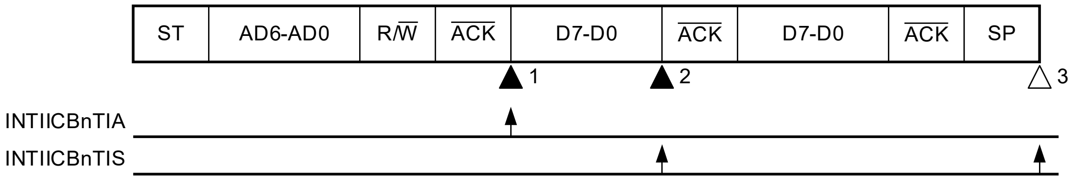
△ **Output only when IICBnCTL0.IICBnSLSI = 1**

\- **Undefined**

X **don't care**

(2) Arbitration loss during data transfer

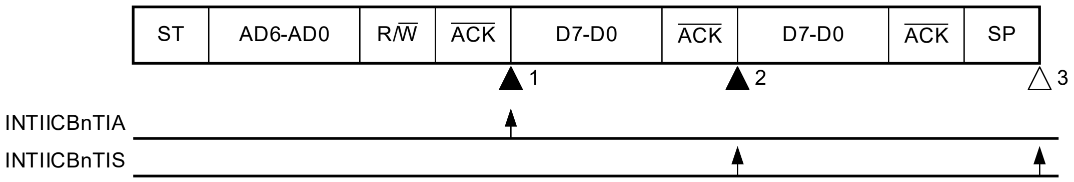
(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-1000X1 0110--00B
▲2: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
△3: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don’t care

(b) When IICBnCTL0.IICBnSLWT bit is 1

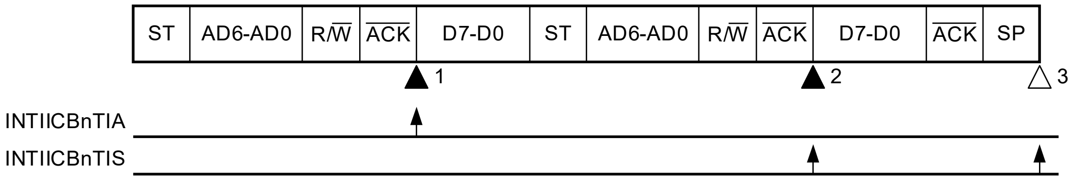


▲1: IICBnSTR0 register = 1-1000X1 0110--00B
▲2: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
△3: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don’t care

(3) Arbitration loss for the restart condition during data transfer

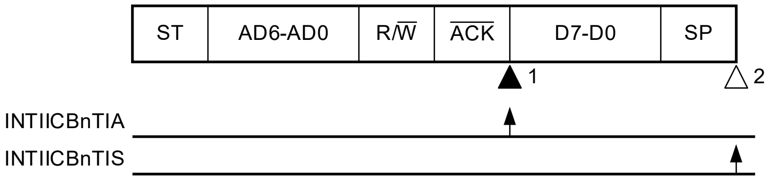
(a) When IICBnCTL0.IICBnSLWT bit is 1 (extension code mismatch, address mismatch)



▲1: IICBnSTR0 register = 1-1000X1 0110--00B
▲2: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
△3: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(4) Arbitration loss for the stop condition during data transfer

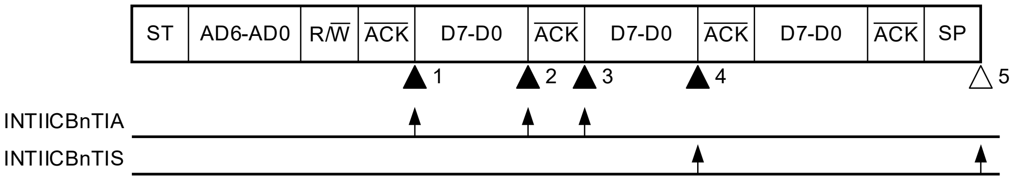


▲1: IICBnSTR0 register = 1-1000X1 0110--00B
△2: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output regardless of the setting of IICBnCTL0.IICBnSLSI bit**
\- **Undefined**
X **don't care**

(5) Arbitration loss because the SDAn signal is low level when attempting to output restart condition

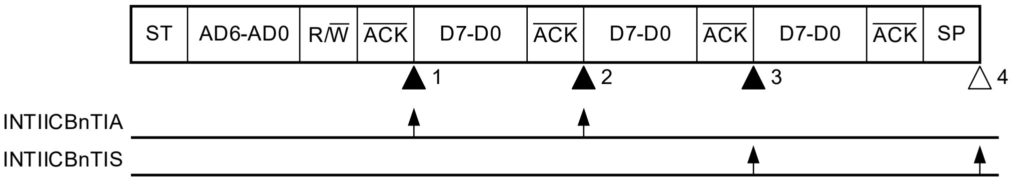
(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-1000X1 0110--00B
▲2: IICBnSTR0 register = 1-1000X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 1)
▲3: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnCTL0.IICBnSLWT bit = 0, IICBnTRG.IICBnSTT bit = 1)
▲4: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1

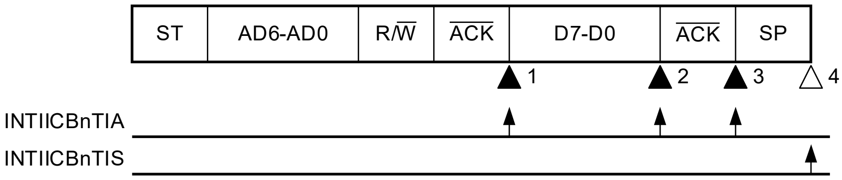


▲1: IICBnSTR0 register = 1-1000X1 0110--00B
▲2: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnCTL0.IICBnSLWT bit = 0, IICBnTRG.IICBnSTT bit = 1)
▲3: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(6) Arbitration loss for the stop condition when attempting to output restart condition

(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

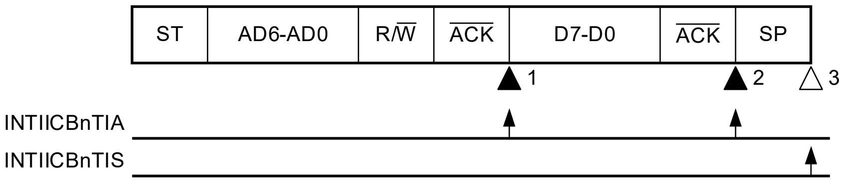
▲2: IICBnSTR0 register = 1-1000X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 0)

▲3: IICBnSTR0 register = 1-0000XX 0100--00B (IICBnTRG.IICBnSTT bit = 1)

△4: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
△ Output regardless of the setting of IICBnCTL0.IICBnSLSI bit
- Undefined
X don't care

(b) When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

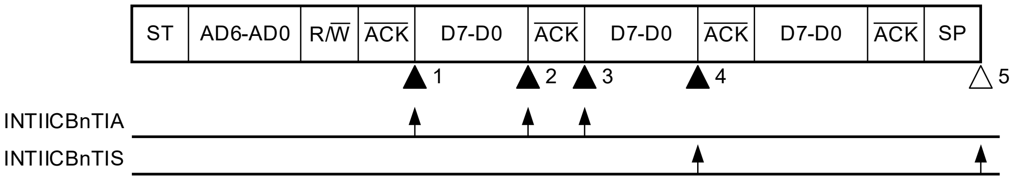
▲2: IICBnSTR0 register = 1-0000XX 0100--00B (IICBnTRG.IICBnSTT bit = 1)

△3: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
△ Output regardless of the setting of IICBnCTL0.IICBnSLSI bit
- Undefined
X don't care

(7) Arbitration loss because the SDAn signal is low level when attempting to output stop condition

(a) When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 1-1000X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 1)

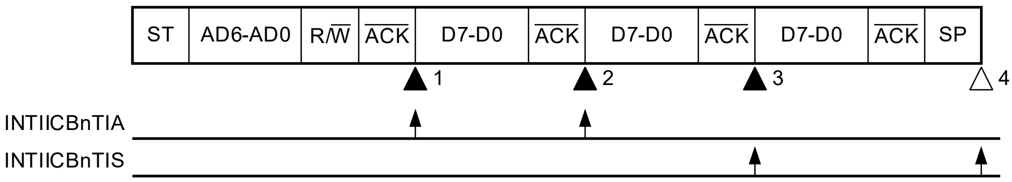
▲3: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnCTL0.IICBnSLWT bit = 0, ICBnTRG.IICBnSPT bit = 1)

▲4: IICBnSTR0 register = 0-0000XX 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

△5: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

(b) When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnTRG.IICBnSPT bit= 1)

▲3: IICBnSTR0 register = 0-0000XX 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

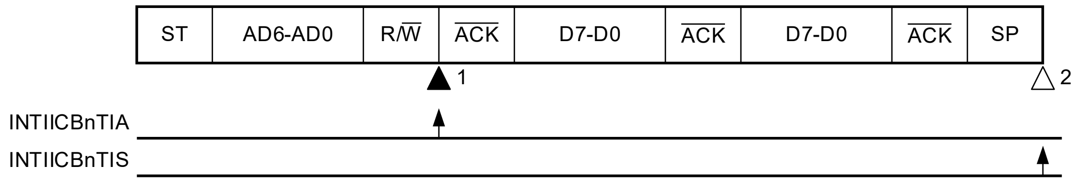
△4: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

24.8.7 Single Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1): Non-Participation in Communications after Arbitration Loss (during Extension Code Transfer))

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each INTIICBnTIS interrupt occurrence to confirm the arbitration result.

(1) Arbitration loss during extension code transfer

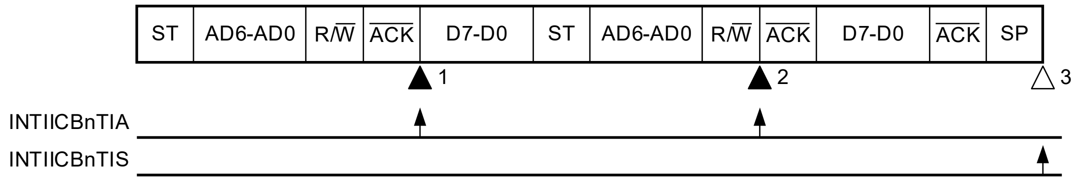


▲1: IICBnSTR0 register = 0-1100X0 0110--01B (IICBnSTRC.IICBnCLAF bit = 1, IICBnTRG.IICBnLRET bit = 1)

△2: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(2) Arbitration loss for the restart condition during data transfer (extension code match)



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 0-1100X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1, IICBnTRG.IICBnLRET bit = 1)

△3: IICBnSTR0 register = 0-000000 0001--01B

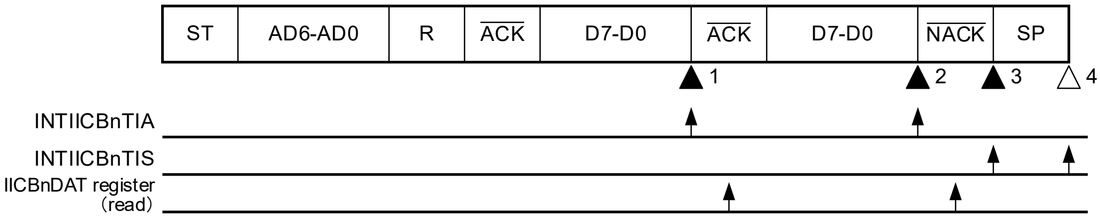
Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

24.8.8 Continuous Transfer Mode (Master Device Operation (Reception))

Remark: The interrupts enclosed in brackets [ ] do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start – Address – Data – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0

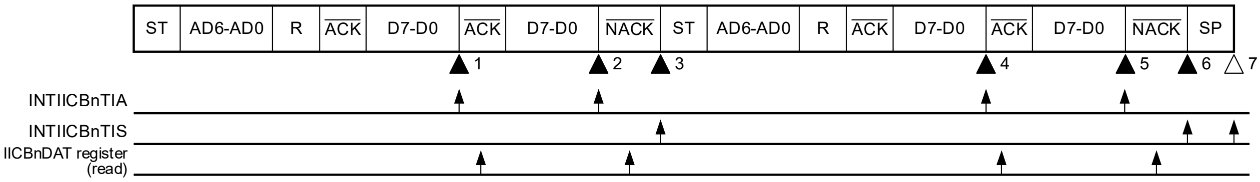


[▲1: IICBnSTR0 register = 1-100000 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲2: IICBnSTR0 register = 1-100000 0100--00B]
IICBnDAT register read
→ IICBnSTR0 register = 1-000000 0100--00B
▲3: IICBnSTR0 register = 1-010000 0100--00B
→ IICBnTRG.IICBnSPT bit = 1
△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(2) Start – Address – Data × 2 – Start – Address – Data × 2 – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0

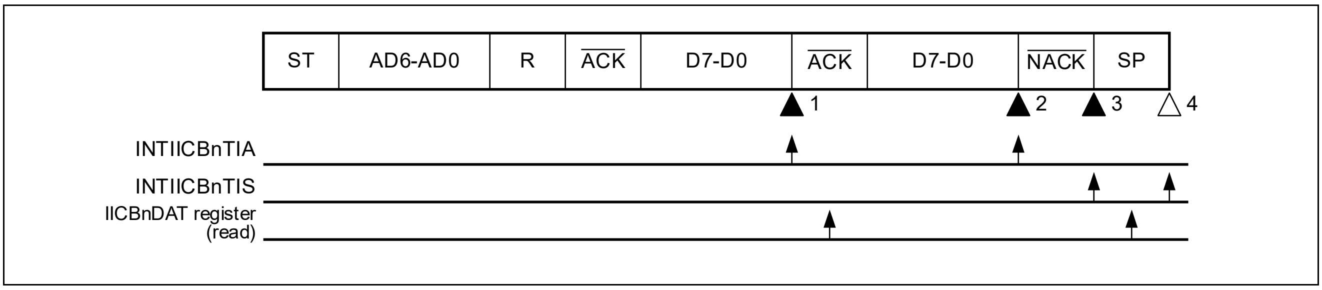


[▲1: IICBnSTR0 register = 1-100001 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲2: IICBnSTR0 register = 1-100000 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
→ IICBnSTR0 register = 1-010000 0100--00B
▲3: IICBnSTR0 register = 1-010000 0100--00B
→ IICBnTRG.IICBnSTT bit = 1
[▲4: IICBnSTR0 register = 1-100000 0100--00B]
IICBnDAT register read
[▲5: IICBnSTR0 register = 1-100000 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
→ IICBnSTR0 register = 1-000000 0100--00B
▲6: IICBnSTR0 register = 1-010000 0100--00B
→ IICBnTRG.IICBnSTT bit = 1
△7: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(3) Start – Code – Data – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0



[▲1: IICBnSTR0 register = 1-101001 0100--00B]
IICBnDAT register read
→ IICBnSTR0 register = 1-0010001 0100--00B
[▲2: IICBnSTR0 register = 1-101000 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
→ IICBnSTR0 register = 1-011000 0100--00B
▲3: IICBnSTR0 register = 1-01000 0100--00B
→ IICBnTRG.IICBnSPT bit = 1
△4: IICBnSTR0 register = 0-000000 0001--00B

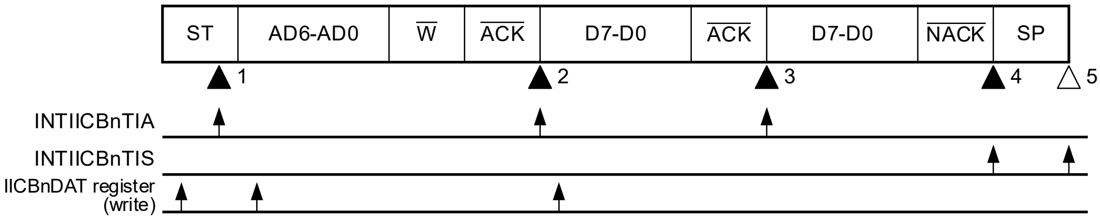
Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

24.8.9 Continuous Transfer Mode (Master Device Operation (Transmission))

Remark: The interrupts enclosed in brackets [ ] do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start – Address – Data – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1



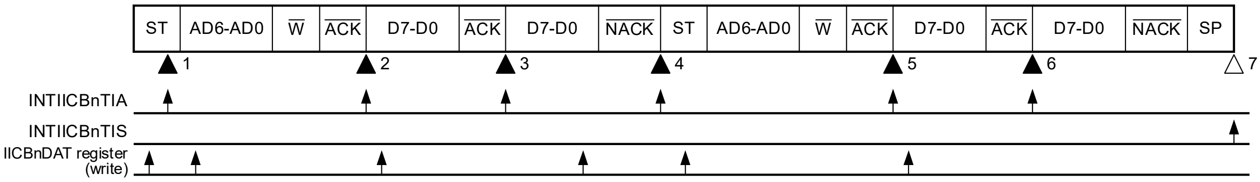
IICBnDAT register write (address)
[▲1: IICBnSTR0 register = X-0000X0 0X0X--00B]
IICBnDAT register write
[▲2: IICBnSTR0 register = 1-000011 0110--00B]
IICBnDAT register write
[▲3: IICBnSTR0 register = 1-000011 0100--00B]
▲4: IICBnSTR0 register = 1-010010 0100--00B
IICBnTRG.IICBnSPT bit= 1
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark:

▲	**Always output**
△	**Output only when IICBnCTL0.IICBnSLSI = 1**
-	**Undefined**
X	**don't care**

(2) Start – Address – Data × 2 – Start – Address – Data × 2 – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1

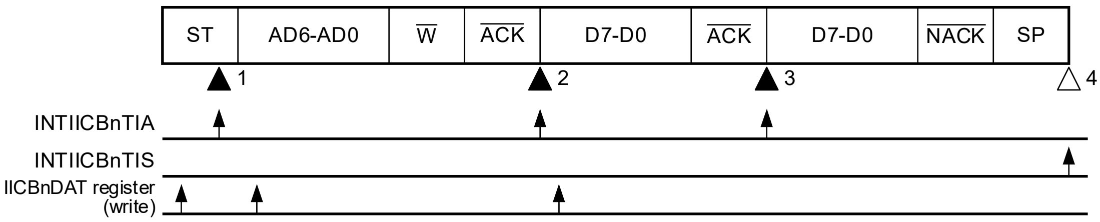


IICBnDAT register write (address)
[▲1: IICBnSTR0 register = X-0000X0 0X0X--00B]
IICBnDAT register write
[▲2: IICBnSTR0 register = 1-000011 0110--00B]
IICBnDAT register write
[▲3: IICBnSTR0 register = 1-000011 0100--00B]
IICBnTRG.IICBnSTT bit= 1
IICBnDAT register write (address)
[▲4: IICBnSTR0 register = 1-000010 010X--00B]
IICBnDAT register write
[▲5: IICBnSTR0 register = 1-000011 0110--00B]
IICBnDAT register write
[▲6: IICBnSTR0 register = 1-000011 0110--00B]
IICBnTRG.IICBnSPT bit = 1
IICBnDAT register write
△7: IICBnSTR0 register = 0-000000 0001--00B

Remark: **▲ Always output**
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don’t care

(3) Start – Code – Data – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1



IICBnDAT register write (address)
[▲1: IICBnSTR0 register = X-0000X0 0X0X--00B]
IICBnDAT register write
[▲2: IICBnSTR0 register = 1-000011 0110--00B]
IICBnDAT register write
[▲3: IICBnSTR0 register = 1-000011 0100--00B]
IICBnTRG.IICBnSPT bit= 1
△4: IICBnSTR0 register = 0-000000 0001--00B

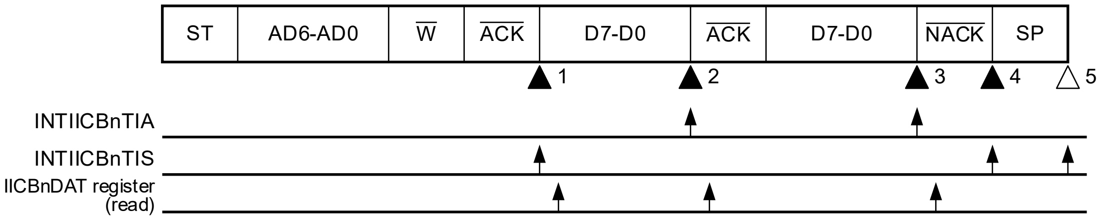
Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

24.8.10 Continuous Transfer Mode (Slave Device Operation (Reception): during Slave Address Reception (IICBnSTR0.IICBnSSC0 bit = 1))

Remark: The interrupts enclosed in brackets [ ] do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start – Address – Data – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0

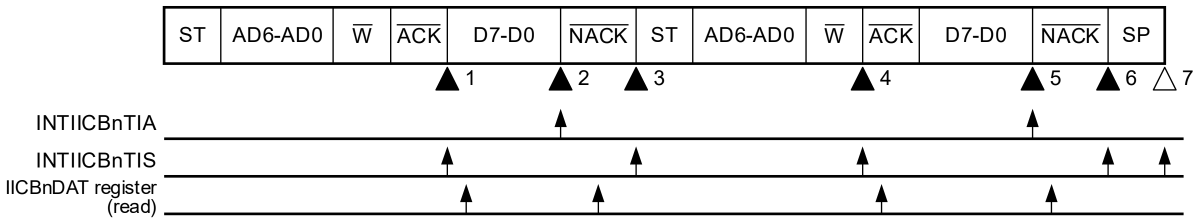


[▲1: IICBnSTR0 register = 0-100101 0110--00B]
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-100100 0100--00B]
IICBnDAT register read
→ IICBnSTR0 register = 0-000100 0100--00B
[▲3: IICBnSTR0 register = 0-100100 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
→ IICBnSTR0 register = 0-000100 0100-00B
▲4: IICBnSTR0 register = 0-010100 0100-00B
IICBnTRG.IICBnWRET bit = 1
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(2) Start – Address – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)

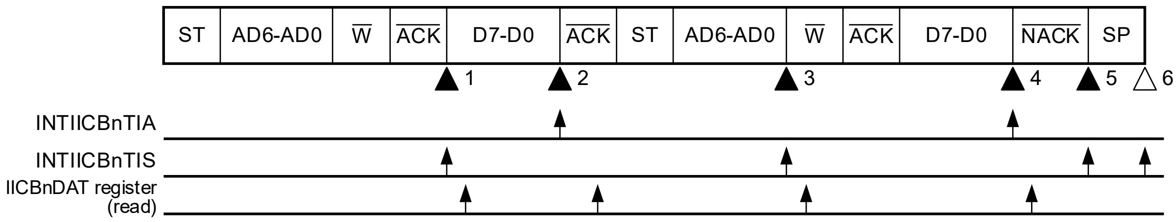


[▲1: IICBnSTR0 register = 0-110101 0110--00B]
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-100101 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
▲3: IICBnSTR0 register = 0-110101 0110--00B
IICBnTRG.IICBnWRET bit = 1
[▲4: IICBnSTR0 register = 0-100100 0110--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
→ IICBnSTR0 register = 0-000100 0110--00B
[▲5: IICBnSTR0 register = 0-100100 0100--00B]
▲6: IICBnSTR0 register = 0-010100 0100--00B
IICBnTRG.IICBnWRET bit = 1
△7: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(3) Start – Address – Data – Start – Code – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)

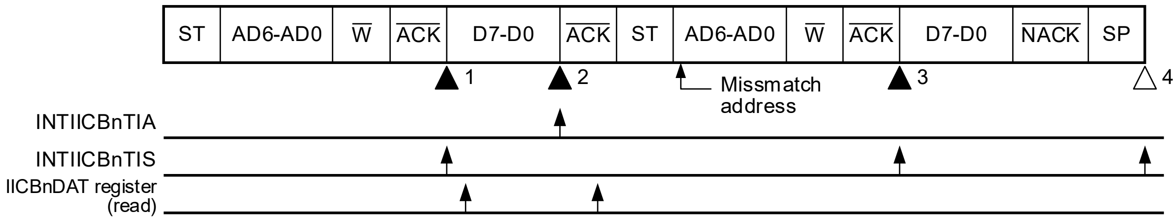


[▲1: IICBnSTR0 register = 0-100101 0110--00B]
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-100100 0100--00B]
IICBnDAT register read
[▲3: IICBnSTR0 register = 0-100100 0110--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲4: IICBnSTR0 register = 0-100100 0110--00B]
IICBnDAT register read
▲5: IICBnSTR0 register = 0-111000 0100--00B
IICBnTRG.IICBnWRET bit = 1
△6: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(4) Start – Address – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



[▲1: IICBnSTR0 register = 0-000101 0110--00B]
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-100100 0100--00B]
IICBnDAT register read
[▲3: IICBnSTR0 register = 0-000000 0110--00B]
△4: IICBnSTR0 register = 0-000000 0001--00B

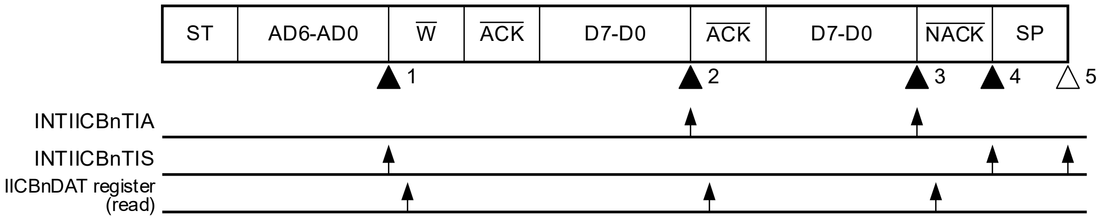
Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

24.8.11 Continuous Transfer Mode (Slave Device Operation (Reception): during Extension Code Reception (IICBnSTR0.IICBnSSEX bit = 1))

Remark: The interrupts enclosed in brackets [ ] do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start – Code – Data – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0

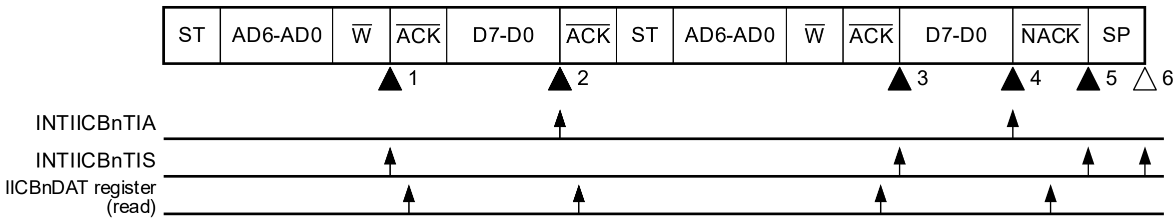


[▲1: IICBnSTR0 register= 0-101000 0110--00B]
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-101001 0110--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲3: IICBnSTR0 register = 0-10001 0100--00B]
IICBnDAT register read
▲4: IICBnSTR0 register = 0-111000 0100--00B
IICBnTRG.IICBnWRET bit = 1
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(2) Start – Code – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)

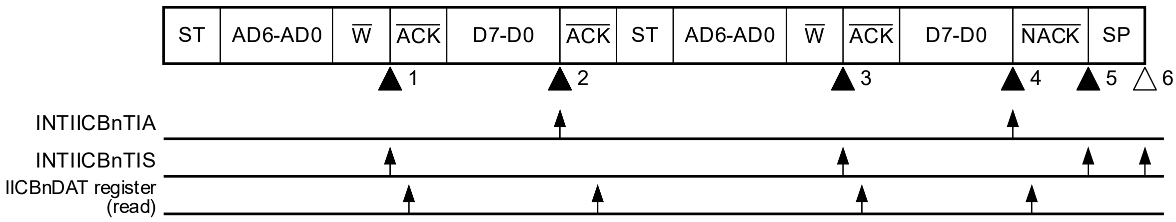


[▲1: IICBnSTR0 register = 0-101000 0110--00B]
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-011000 0110--00B]
IICBnDAT register read
[▲3: IICBnSTR0 register = 0-111001 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲4: IICBnSTR0 register = 0-010100 0110--00B]
IICBnDAT register read
▲5: IICBnSTR0 register = 0-110100 0100--00B
IICBnTRG.IICBnWRET bit = 1
△6: IICBnSTR0 register = 0-000000 0001--00B

Remark: **▲ Always output**
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(3) Start – Code – Data – Start – Code – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)

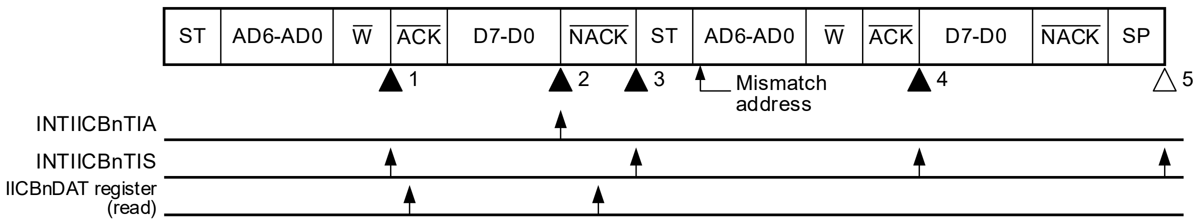


[▲1: IICBnSTR0 register = 0-101000 0110--00B]
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-011001 0110--00B]
IICBnDAT register read
[▲3: IICBnSTR0 register = 0-101000 0110--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲4: IICBnSTR0 register = 0-101001 0110--00B]
IICBnDAT register read
▲5: IICBnSTR0 register = 0-011000 0100--00B
IICBnTRG.IICBnWRET bit = 1
△6: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(4) Start – Code – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



[▲1: IICBnSTR0 register = 0-101000 0110--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-101001 0110--00B]
IICBnCTL0.IICBnSLAC bit = 0
▲3: IICBnSTR0 register = 0-010000 0100--00B
IICBnTRG.IICBnWRET bit = 1
[▲4: IICBnSTR0 register = 0-000000 0110--00B]
△5: IICBnSTR0 register = 0-000000 0001--00B

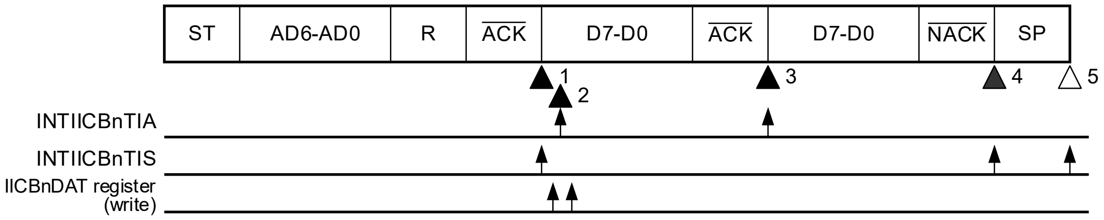
Remark: **▲ Always output**
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don’t care

24.8.12 Continuous Transfer Mode (Slave Device Operation (Transmission): during Slave Address Reception (IICBnSTR0.IICBnSSC0 bit = 1))

Remark: The interrupts enclosed in brackets [ ] do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start – Address – Data – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 0-110111 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-00011X 0100--00B]

IICBnDAT register write

→ IICBnSTR0 register = 0-100011X 0100--00B

▲3: IICBnSTR0 register = 0-000111 0100--00B

▲4: IICBnSTR0 register = 0-010110 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Remark ▲ Always output

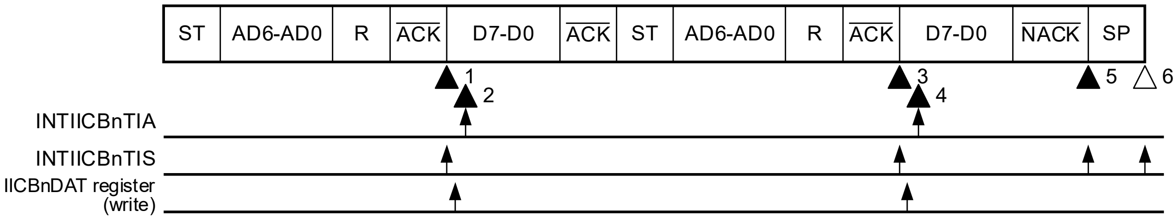
: △ Output only when IICBnCTL0.IICBnSLSI = 1

- Undefined

X don’t care

(2) Start – Address – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address match)

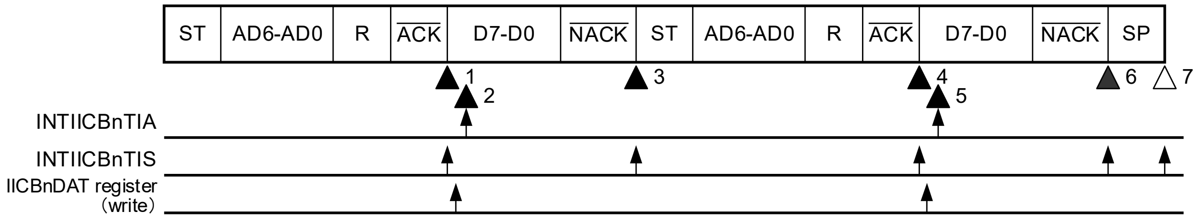


▲1: IICBnSTR0 register = 0-010111 0110--00B
IICBnDAT register write
[▲2: IICBnSTR0 register = 0-00111X 01X0--00B]
▲3: IICBnSTR0 register = 0-010111 0110--00B
IICBnDAT register write
[▲4: IICBnSTR0 register = 0-100101 01X0--00B]
▲5: IICBnSTR0 register = 0-110100 0100--00B
△6: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don’t care

(3) Start – Address – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-110111 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-100111 0100--00B]

▲3: IICBnSTR0 register = 0-111010 0110--00B

▲4: IICBnSTR0 register = 0-111010 0110--00B

IICBnDAT register write

[▲5: IICBnSTR0 register = 0-111011 0110--00B]

▲6: IICBnSTR0 register = 0-111010 0100--00B

△7: IICBnSTR0 register = 0-000000 0001--00B

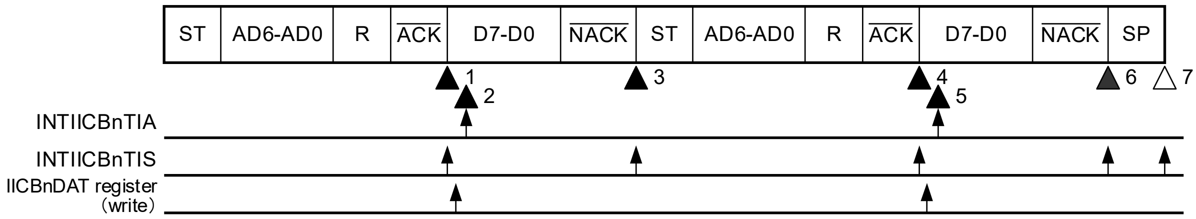
Remark: ▲ **Always output**

△ **Output only when IICBnCTL0.IICBnSLSI = 1**

\- **Undefined**

(4) Start – Address – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-110111 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-100111 0100--00B]

▲3: IICBnSTR0 register = 0-000010 0100--00B

▲4: IICBnSTR0 register = 0-000011 0110--00B

IICBnDAT register write

[▲5: IICBnSTR0 register = 0-00001X 0100--00B]

▲6: IICBnSTR0 register = 0-000010 0100--00B

△7: IICBnSTR0 register = 0-000000 0001--00B

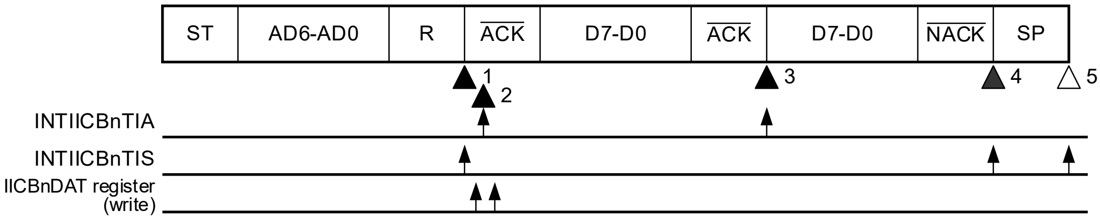
Remark:	**▲**	**Always output**
	△	**Output only when IICBnCTL0.IICBnSLSI = 1**
	-	**Undefined**
	X	**don't care**

24.8.13 Continuous Transfer Mode (Slave Device Operation (Transmission): during Extension Code Reception (IICBnSTR0.IICBnSSEX bit = 1))

Remark: The interrupts enclosed in brackets [ ] do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start – Code – Data – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1

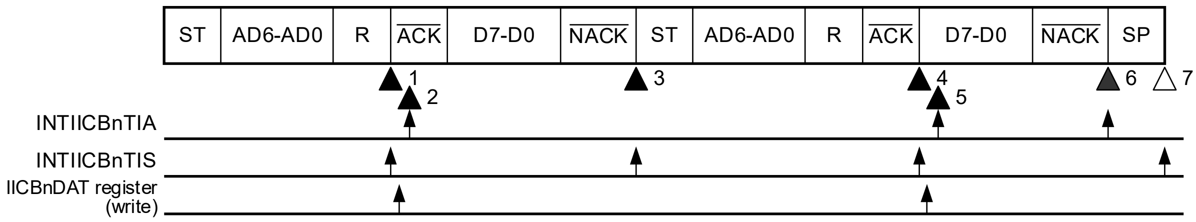


▲1: IICBnSTR0 register = 0-011010 0110--00B
IICBnDAT register write
[▲2: IICBnSTR0 register = 0-011011 0110--00B]
IICBnDAT register write
[▲3: IICBnSTR0 register = 0-011011 0100--00B]
▲4: IICBnSTR0 register = 0-111010 0100--00B
△5: IICBnSTR0 register = 0-000010 0001--00B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(2) Start – Code – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address match)

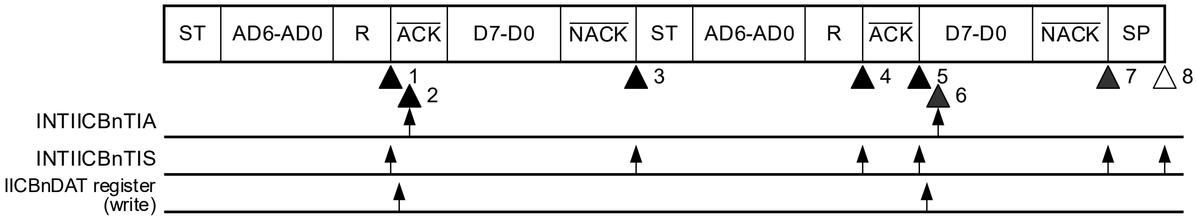


▲1: IICBnSTR0 register = 0-011000 0110--00B
IICBnDAT register write
[▲2: IICBnSTR0 register = 0-011001 0110--00B]
▲3: IICBnSTR0 register = 0-011000 0100--00B
▲4: IICBnSTR0 register = 0-010101 0110--00B
IICBnDAT register write
[▲5: IICBnSTR0 register = 0-010101 0110--00B]
▲6: IICBnSTR0 register = 0-010100 0100--00B
△7: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(3) Start – Code – Data – Start – Code – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, extension code reception)

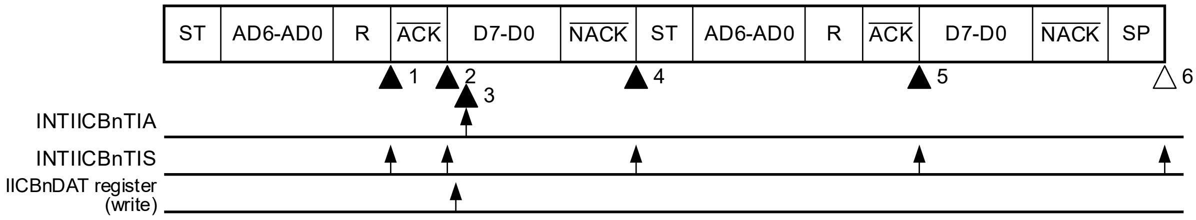


▲1: IICBnSTR0 register = 0-011000 0110--00B
IICBnDAT register write
[▲2: IICBnSTR0 register = 0-011001 0110--00B]
▲3: IICBnSTR0 register = 0-011000 0100--00B
▲4: IICBnSTR0 register = 0-011000 0110--00B
▲5: IICBnSTR0 register = 0-011001 0110--00B
IICBnDAT register write
[▲6: IICBnSTR0 register = 0-011001 0110--00B]
▲7: IICBnSTR0 register = 0-011000 0100--00B
△8: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(4) Start – Code – Data – Start – Address – Data – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address mismatch (extension code mismatch))

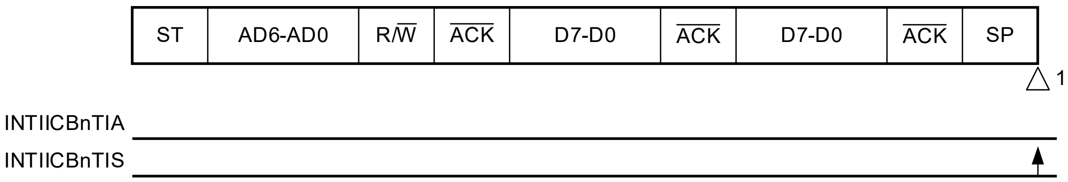


▲1: IICBnSTR0 register = 0-011000 0110--00B
▲2: IICBnSTR0 register = 0-011001 0110--00B
IICBnDAT register write
[▲3: IICBnSTR0 register = 0-011010 0100--00B]
▲4: IICBnSTR0 register = 0-000000 0100--00B
▲5: IICBnSTR0 register = 0-000000 0110--00B
△6: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

24.8.14 Continuous Transfer Mode (Non-Participation in Communications)

(1) Start – Code – Data – Data – Stop



△1: IICBnSTR0 register = 0-0000X0 0001--00B

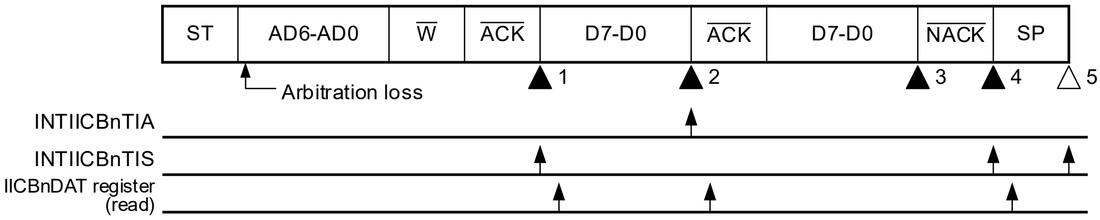
Remark: △ **Output only when IICBnCTL0.IICBnSLSI = 1**
- **Undefined**

24.8.15 Continuous Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1) (when address was transferred during reception): Operation as Slave after Arbitration Loss)

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each INTIICBnTIS interrupt occurrence to confirm the arbitration result.

(1) Address match after arbitration loss

(a) During reception, when IICBnCTL0.IICBnSLWT bit is 0

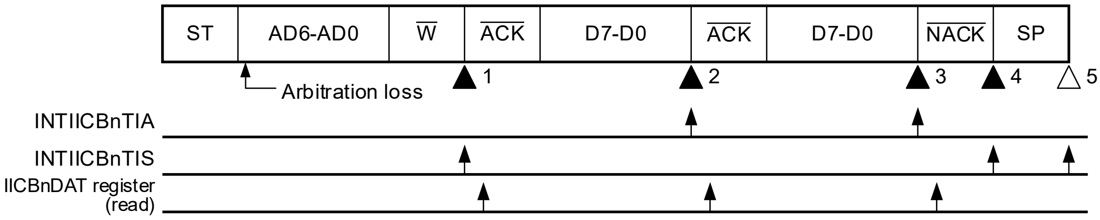


[▲1: IICBnSTR0 register = 0-100101 0110--01B]
IICBnSTRC.IICBnCLAF bit = 1
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-100101 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲3: IICBnSTR0 register = 0-100100 0100--00B]
IICBnDAT register read
▲4: IICBnSTR0 register = 0-010100 0100--00B
IICBnTRG.IICBnWRET bit = 1
△5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**

(2) Upon extension code detection after arbitration loss

(a) During reception, when IICBnCTL0.IICBnSLWT bit is 0



[▲1: IICBnSTR0 register = 0-101000 0110--01B]
IICBnSTRC.IICBnCLAF bit = 1
IICBnDAT register read
[▲2: IICBnSTR0 register = 0-101000 0110--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
[▲3: IICBnSTR0 register = 0-101000 0100--00B]
IICBnDAT register read
▲4: IICBnSTR0 register = 0-011000 0100--00B]
IICBnTRG.IICBnWRET bit = 1
△5: IICBnSTR0 register = 0-000000 0001--00B

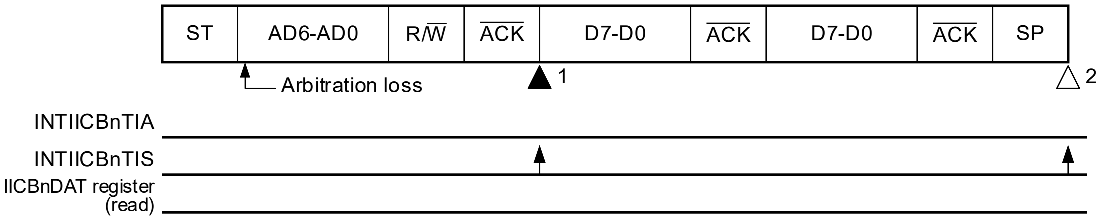
Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

24.8.16 Continuous Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1) (when address was transferred during reception): Non-Participation in Communications after Arbitration Loss)

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each INTIICBnTIS interrupt occurrence to confirm the arbitration result.

(1) Arbitration loss during slave address transmission

(a) During reception, when IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-000001 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)

△2: IICBnSTR0 register = 0-000000 0001--00B

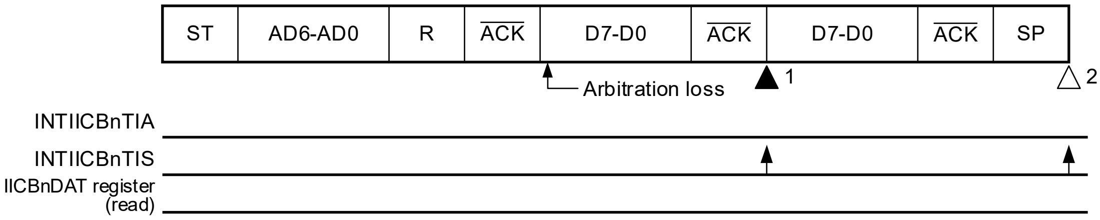
Remark: ▲ **Always output**

△ **Output only when IICBnCTL0.IICBnSLSI = 1**

\- **Undefined**

(2) Arbitration loss during data transfer

(a) During reception, when IICBnCTL0.IICBnSLWT bit is 1

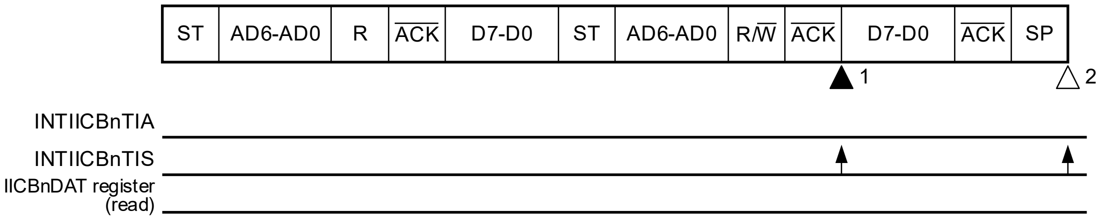


[▲1: IICBnSTR0 register = 0-000000 0100--01B]
IICBnSTRC.IICBnCLAF bit = 1
△2: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
- **Undefined**

(3) Arbitration loss for the restart condition during data transfer

(a) During reception, when IICBnCTL0.IICBnSLWT bit is 1 (extension code mismatch, address mismatch)

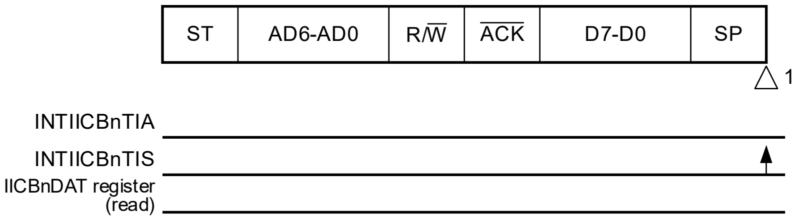


[▲1: IICBnSTR0 register = 0-000001 0100--01B]
IICBnSTRC.IICBnCLAF bit = 1
△2: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(4) Arbitration loss for the stop condition during data transfer

(a) During reception, when IICBnCTL0.IICBnSLWT bit is 1

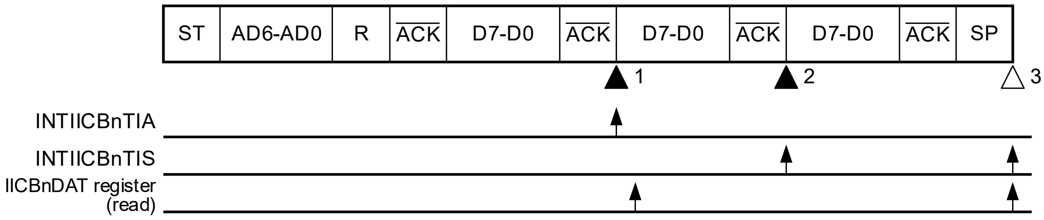


△1: IICBnSTR0 register = 0-000000 0001--01B
IICBnSTRC.IICBnCLAF bit = 1

Remark: △ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(5) Arbitration loss because the SDAn signal is low level when attempting to output restart condition

(a) When IICBnCTL0.IICBnSLWT bit is 1



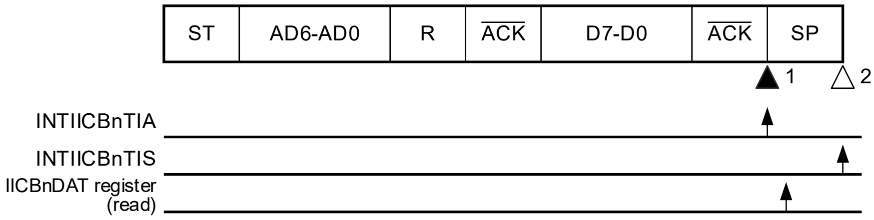
[▲1: IICBnSTR0 register = 1-1000XX 0100--00B]
IICBnDAT register read
IICBnTRG.IICBnSTT bit = 1
▲2: IICBnSTR0 register = 0-000000 0100--01B
IICBnSTRC.IICBnCLAF bit = 1
△3: IICBnSTR0 register = 0-000000 0001--00B

Remark:

▲	**Always output**
△	**Output only when IICBnCTL0.IICBnSLSI = 1**
-	**Undefined**
X	**don’t care**

(6) Arbitration loss for the stop condition when attempting to output restart condition

(a) When IICBnCTL0.IICBnSLWT bit is 1

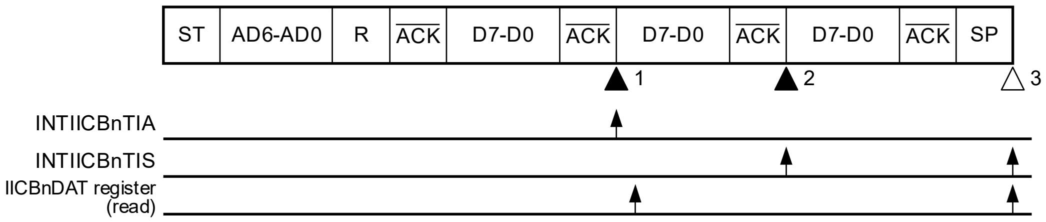


[▲1: IICBnSTR0 register = 1-000001 0100--00B]
IICBnDAT register read
IICBnTRG.IICBnSTT bit = 1
△2: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(7) Arbitration loss because the SDAn signal is low level when attempting to output stop condition

(a) When IICBnCTL0.IICBnSLWT bit is 1



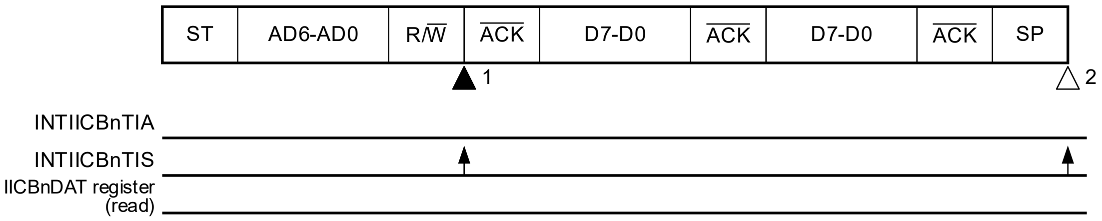
[▲1: IICBnSTR0 register = 1-1000XX 0100--00B]
IICBnDAT register read
IICBnTRG.IICBnSPT bit = 1
[▲2: IICBnSTR0 register = 0-0000XX 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
△3: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

24.8.17 Continuous Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1) (when address was transferred during reception): Non-Participation in Communications after Arbitration Loss (during Extension Code Transfer))

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each INTIICBnTIS interrupt occurrence to confirm the arbitration result.

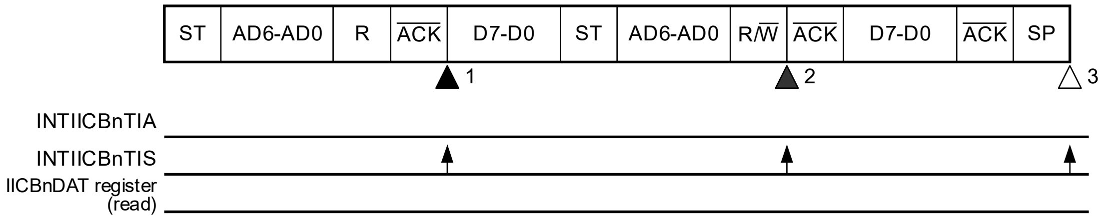
(1) Arbitration loss during extension code transfer



[▲1: IICBnSTR0register = 0-1000X0 0110--01B]
IICBnSTRC.IICBnCLAF bit = 1
IICBnTRG.IICBnLRET bit = 1
△2: IICBnSTR0 register = 0-000000 0001--01B

Remark:
▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(2) Arbitration loss for the restart condition during data transfer (extension code match)



▲1: IICBnSTR0 register = 1-0000X1 0110--00B
▲2: IICBnSTR0 register = 0-0100X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1, IICBnTRG.IICBnLRET bit = 1)
△3: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ **Always output**
△ **Output only when IICBnCTL0.IICBnSLSI = 1**
\- **Undefined**
X **don't care**

24.9 Setting Procedure

24.9.1 Single Master Environment

(1) Master operation setting procedure during single transfer mode

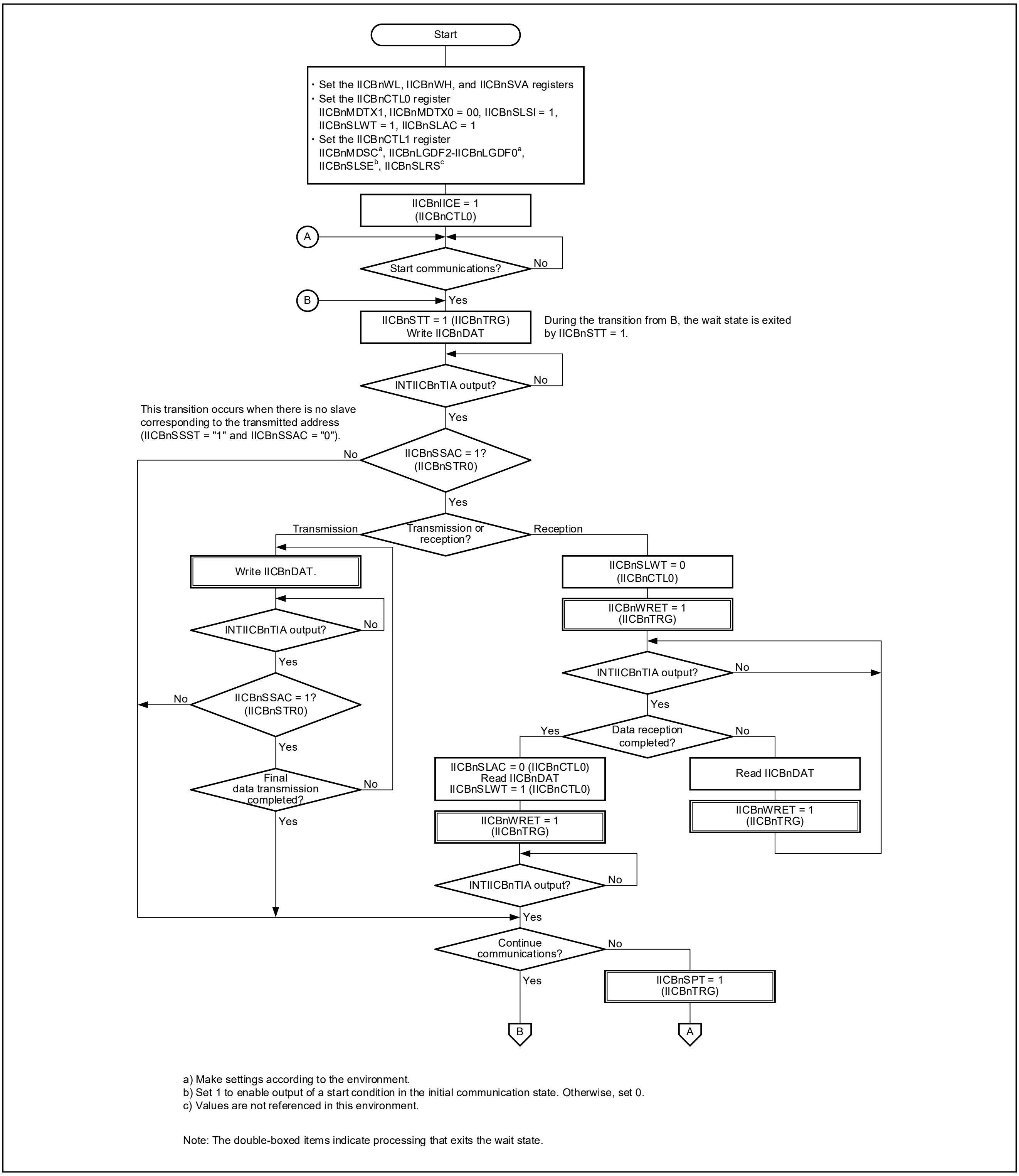
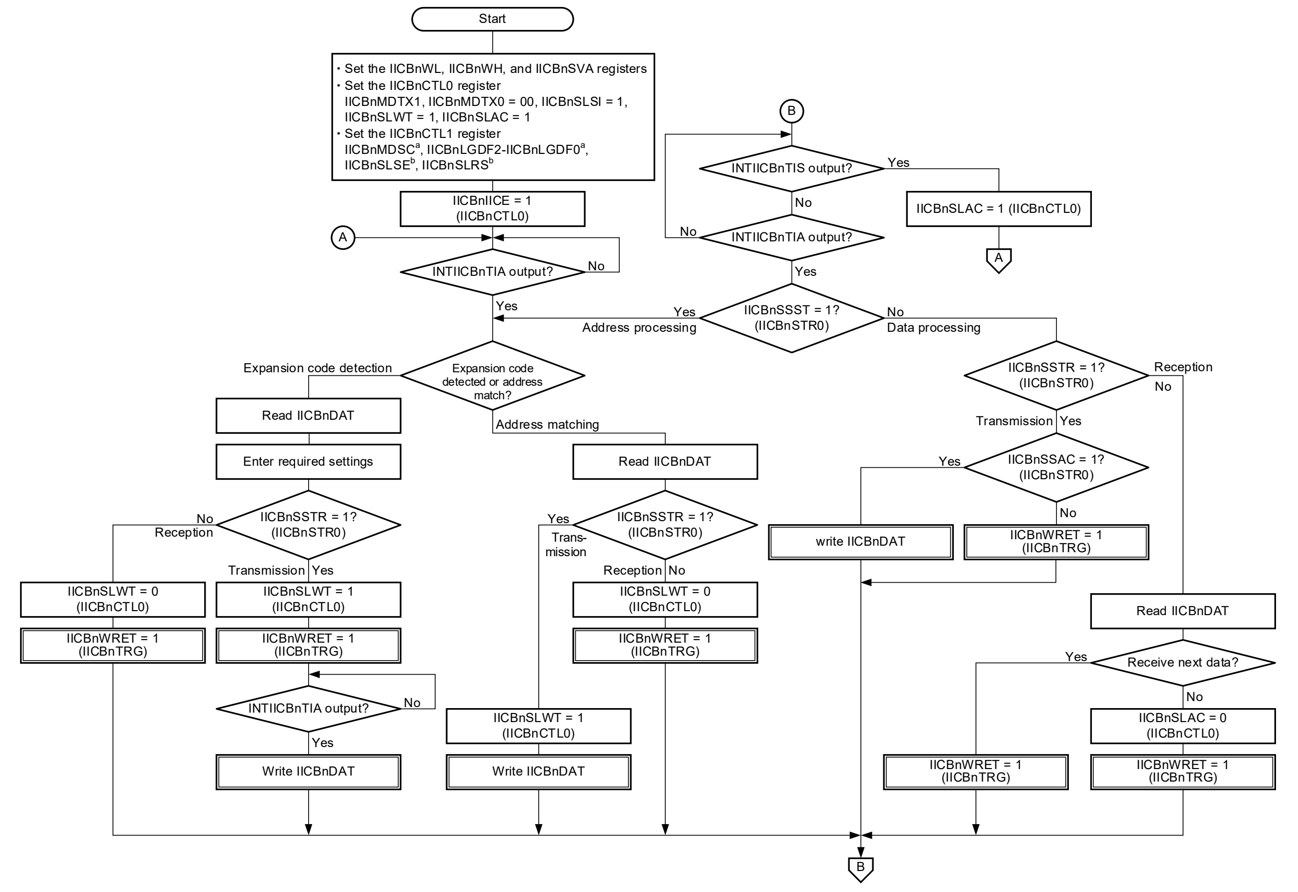


Figure 24.14 Master Operation Setting Procedure during Single Transfer Mode (Single Master Environment)

(2) Slave operation setting procedure during single transfer mode



a) Make settings according to the environment.
b) Values are not referenced in this environment.

Note: The double-boxed items indicate processing that exits the wait state.

Figure 24.15 Slave Operation Setting Procedure during Single Transfer Mode (Single Master Environment)

(3) Master operation setting procedure during continuous transfer mode

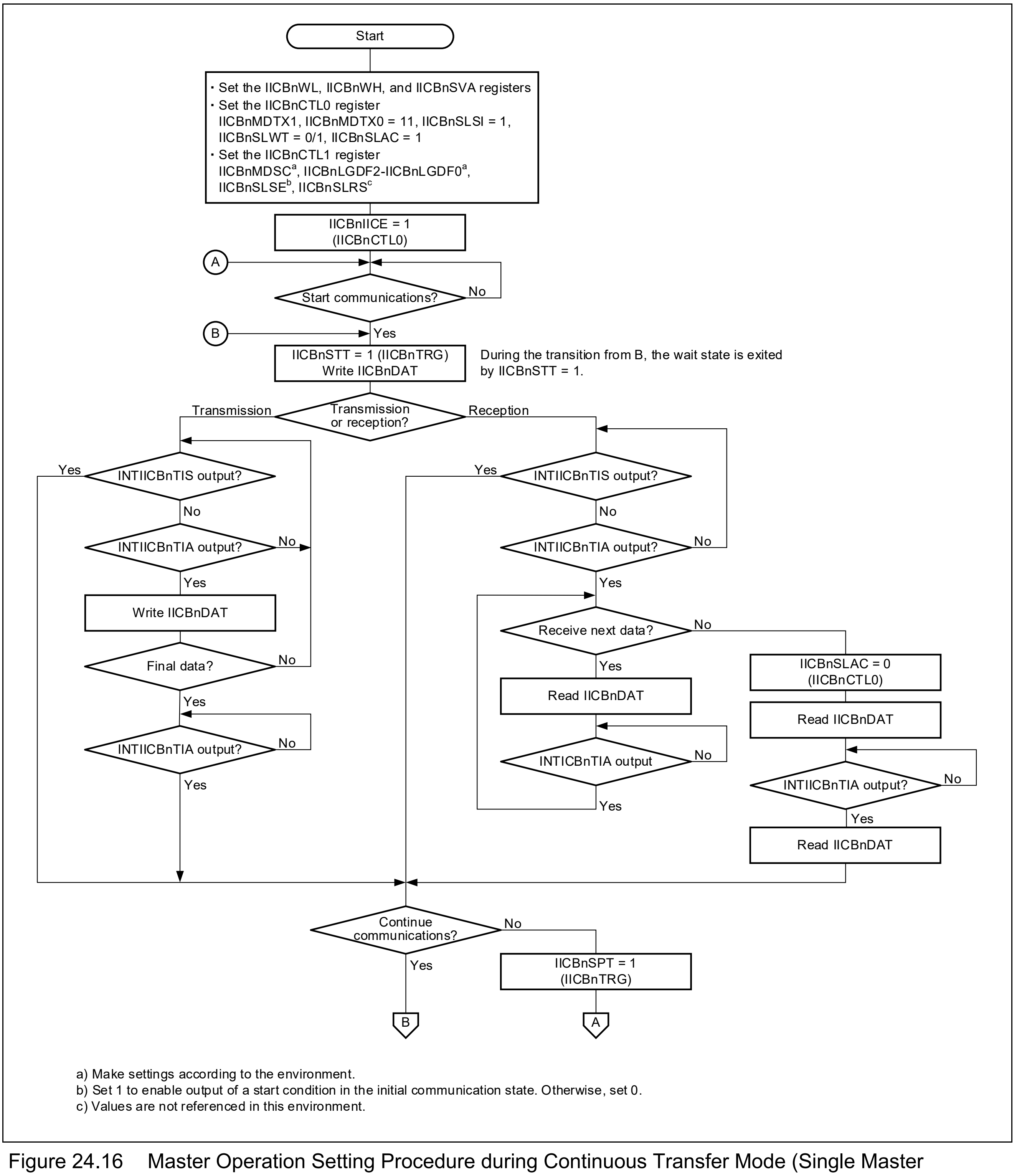


Figure 24.16 Master Operation Setting Procedure during Continuous Transfer Mode (Single Master Environment)

(4) Slave operation setting procedure during continuous transfer mode

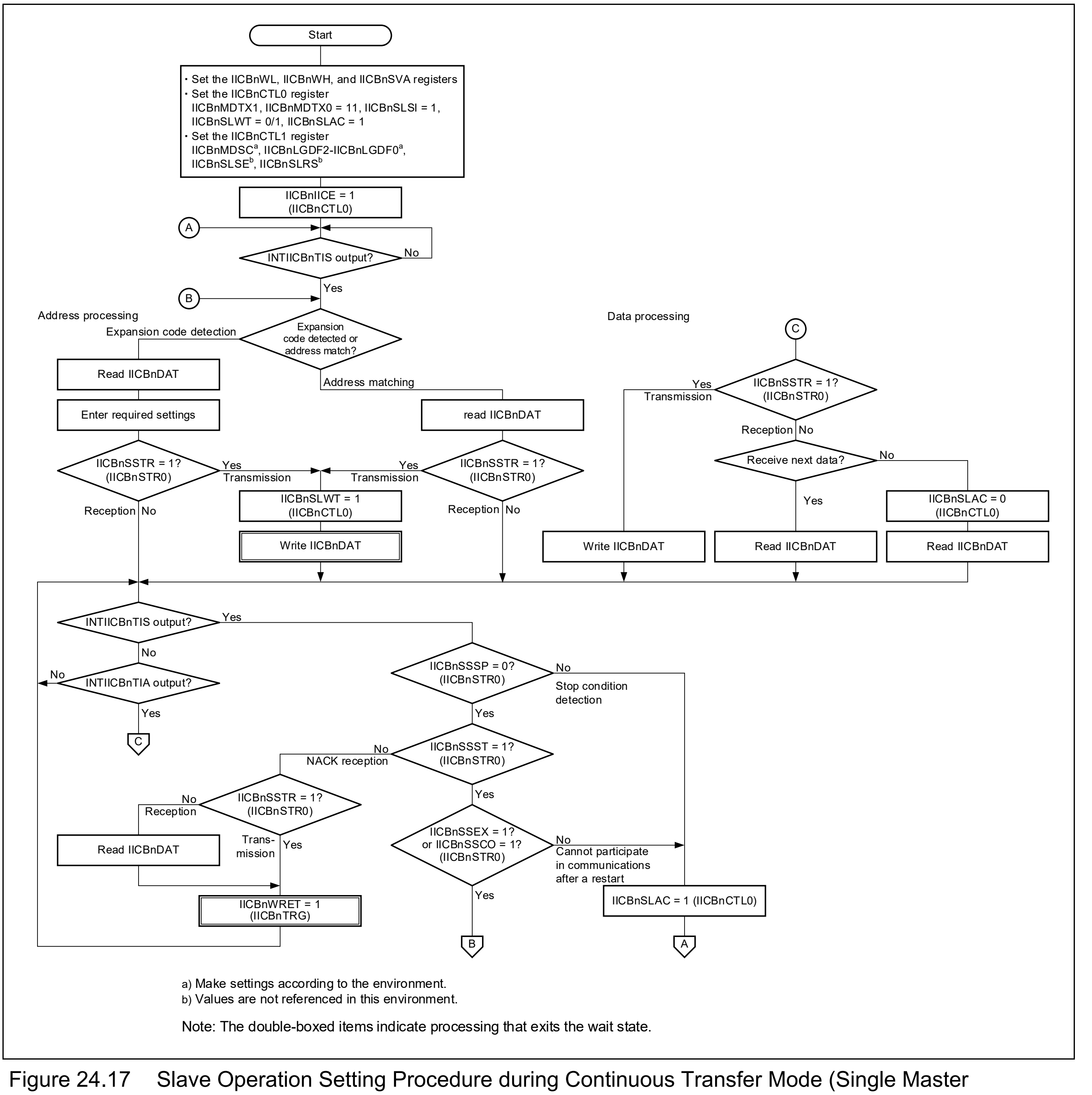


Figure 24.17 Slave Operation Setting Procedure during Continuous Transfer Mode (Single Master Environment)

24.9.2 Multi-Master Environment

(1) Single transfer mode setting procedure when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0)

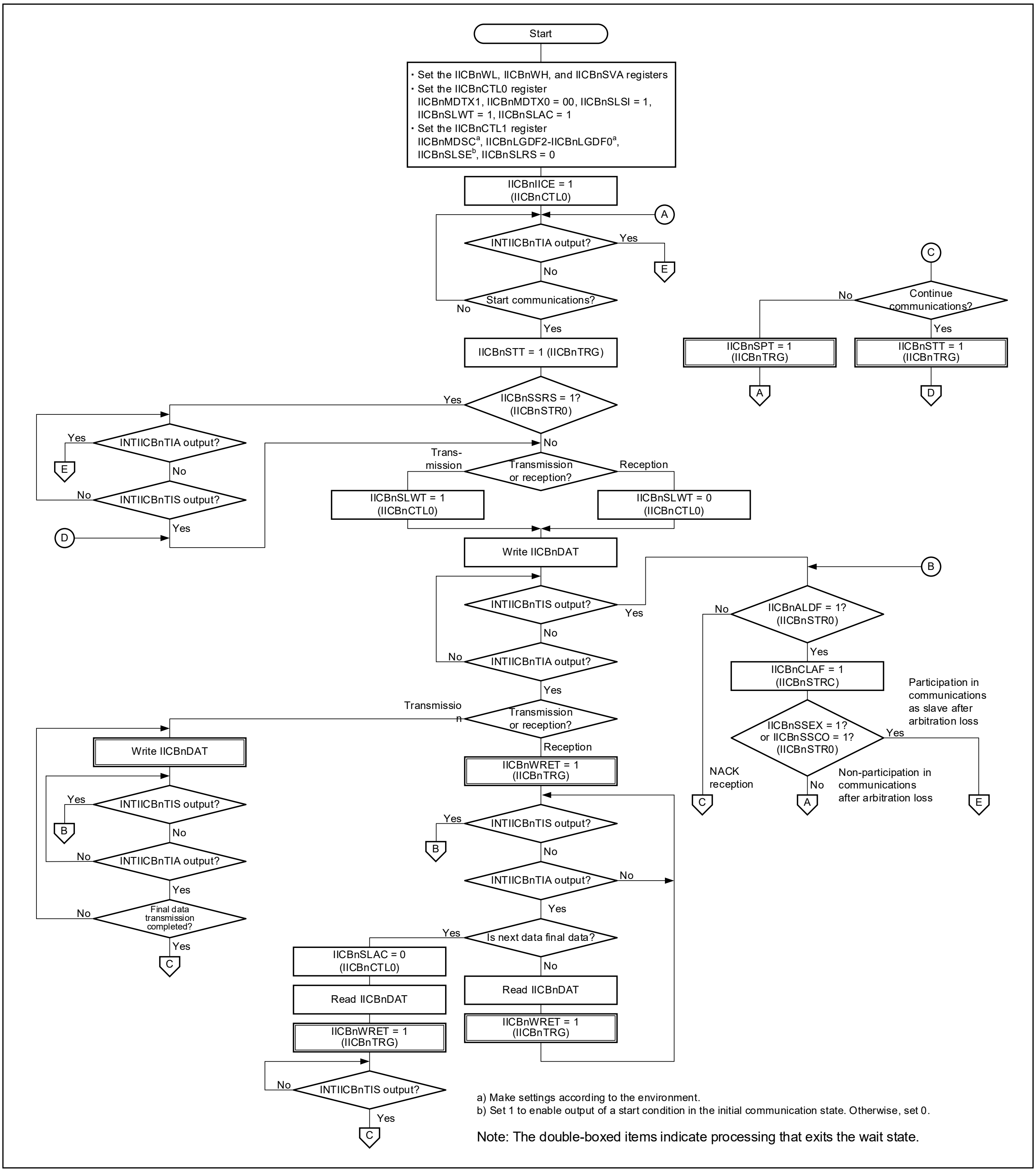


Figure 24.18 Single Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (IICBnCTL1.IICBnSLRS bit = 0) (Multi-Master Environment) (1/2)

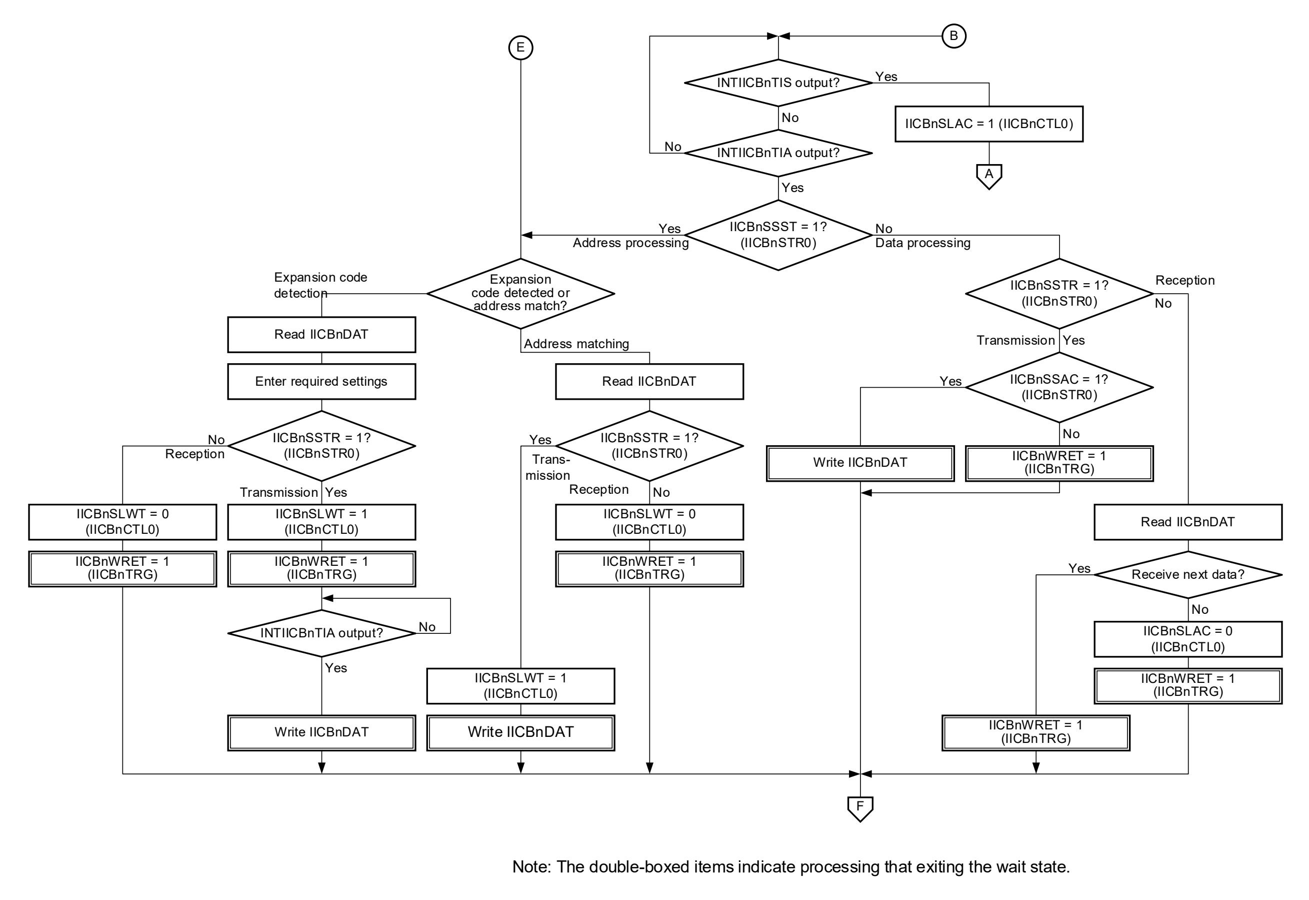
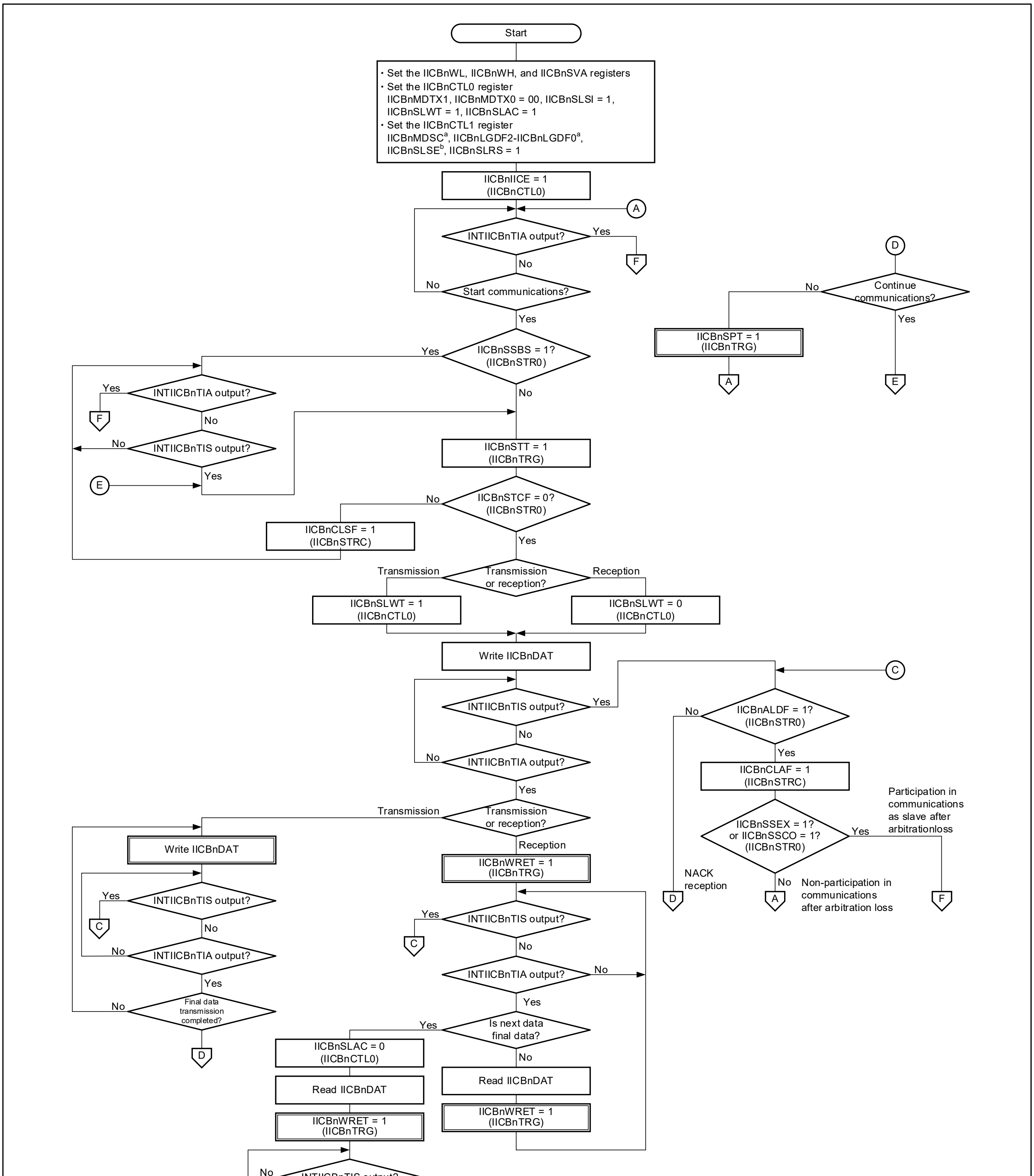


Figure 24.18 Single Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (IICBnCTL1.IICBnSLRS bit = 0) (Multi-Master Environment) (2/2)

(2) Single transfer mode setting procedure when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1)



a) Make settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.

Note: The double-boxed items indicate processing that exits the wait state.

Figure 24.19 Single Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (1/2)

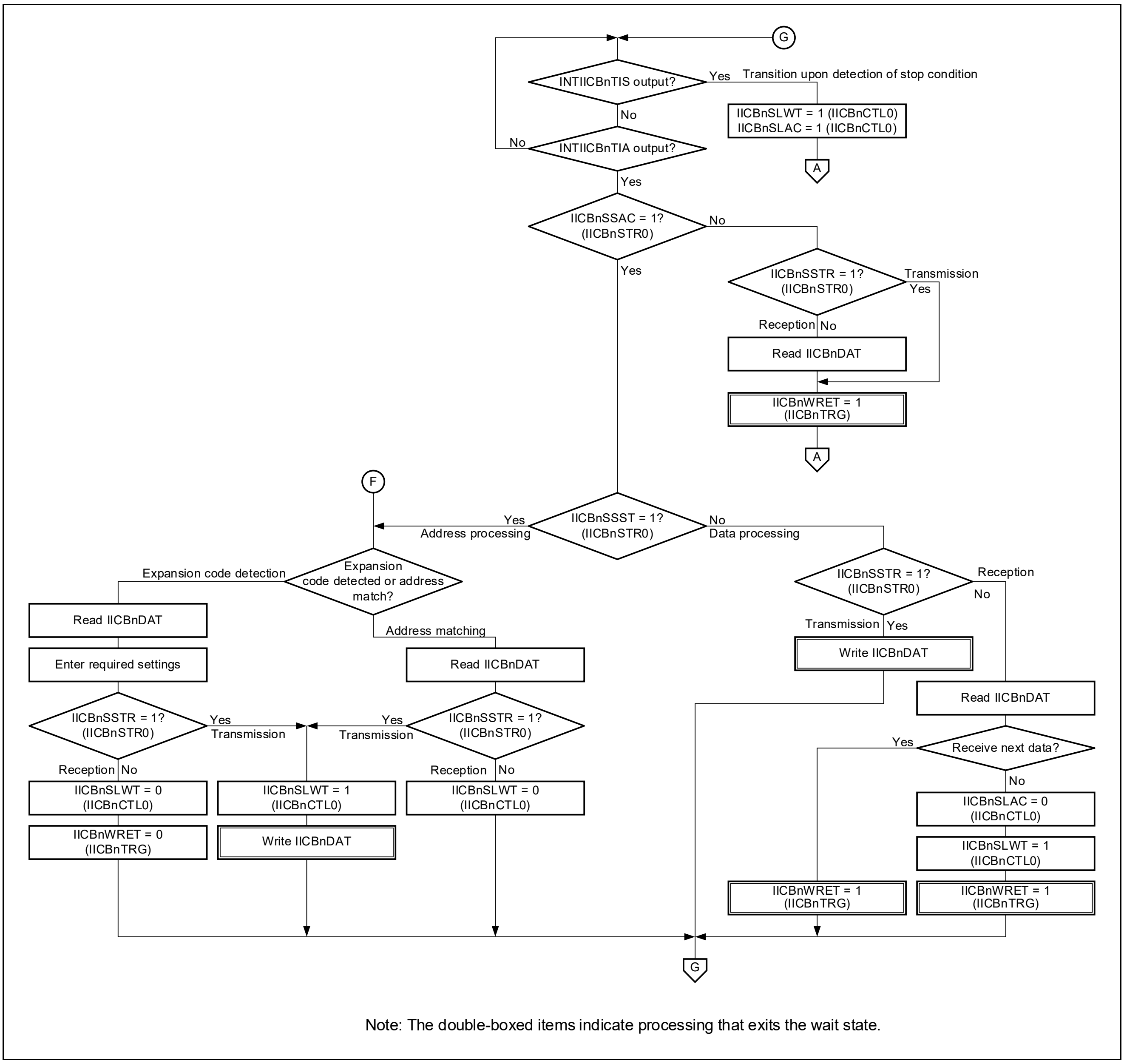
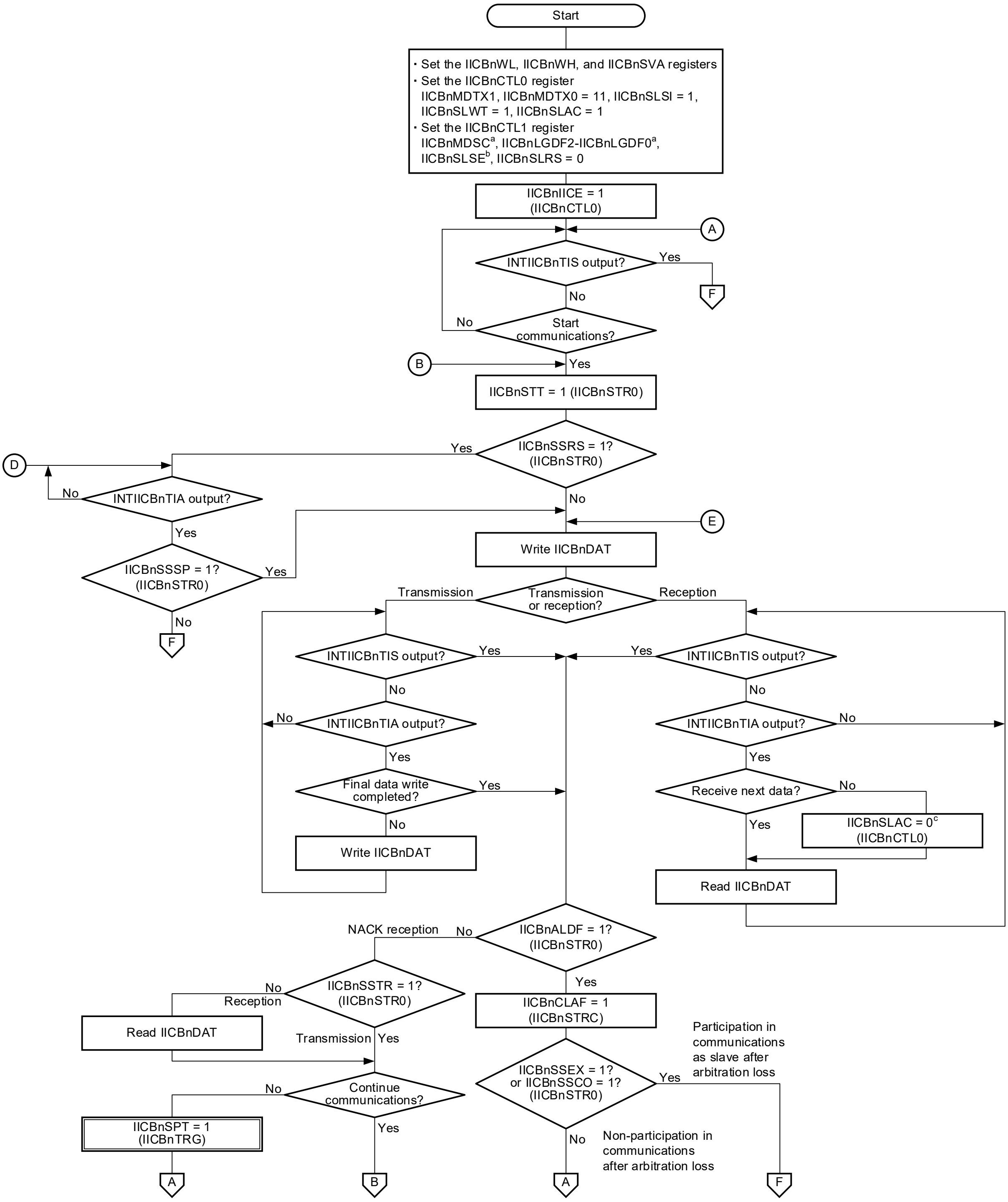


Figure 24.19 Single Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (2/2)

(3) Continuous transfer mode setting procedure when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0)



a) Make settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.
c) The IICBnCTL0.IICBnSLAC bit must be cleared to 0 before the 8th clock of SCLn.

Note: The double-boxed items indicate processing that exits the wait state.

Figure 24.20 Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (IICBnCTL1.IICBnSLRS bit = 0) (Multi-Master Environment) (1/2)

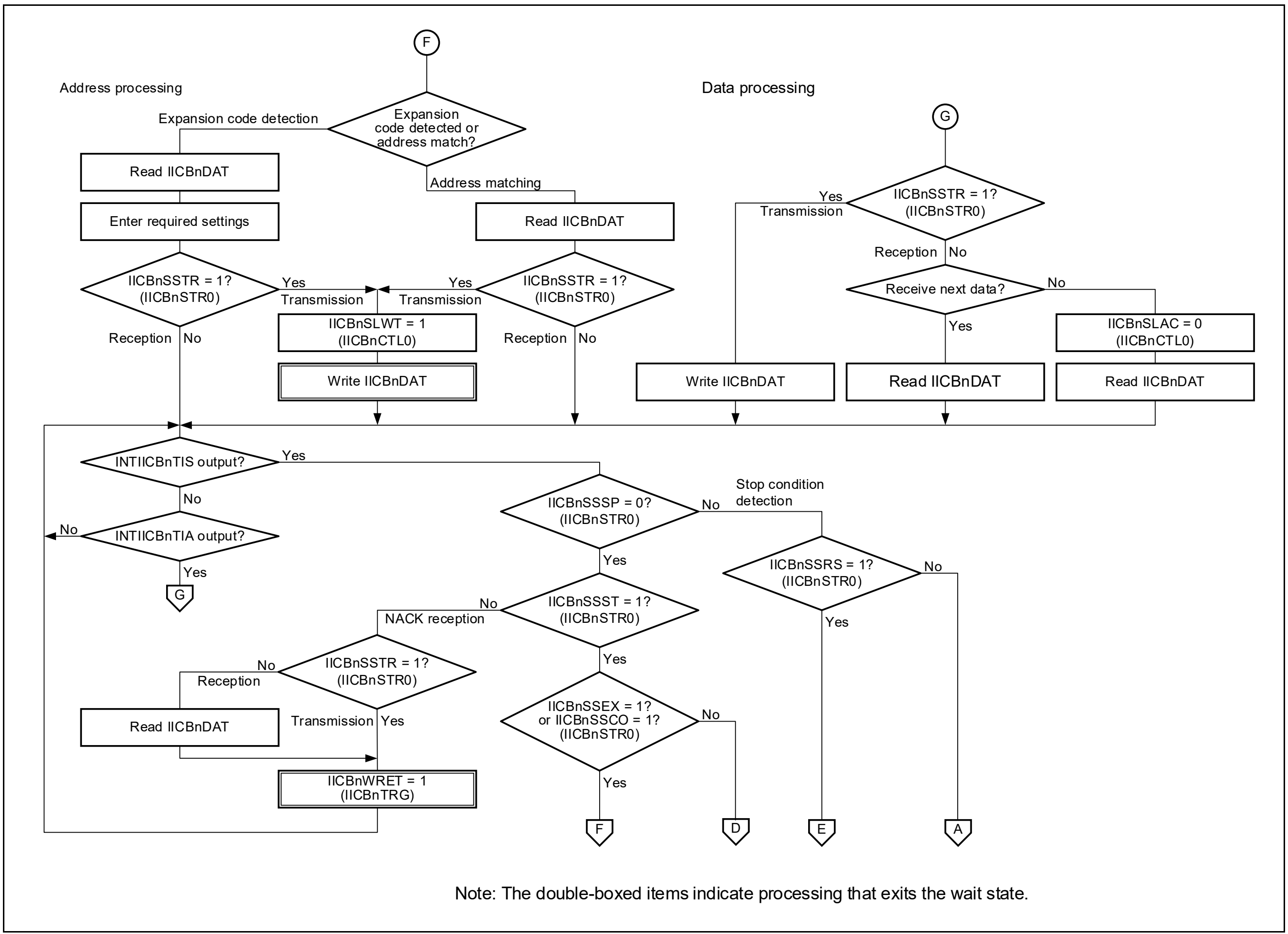
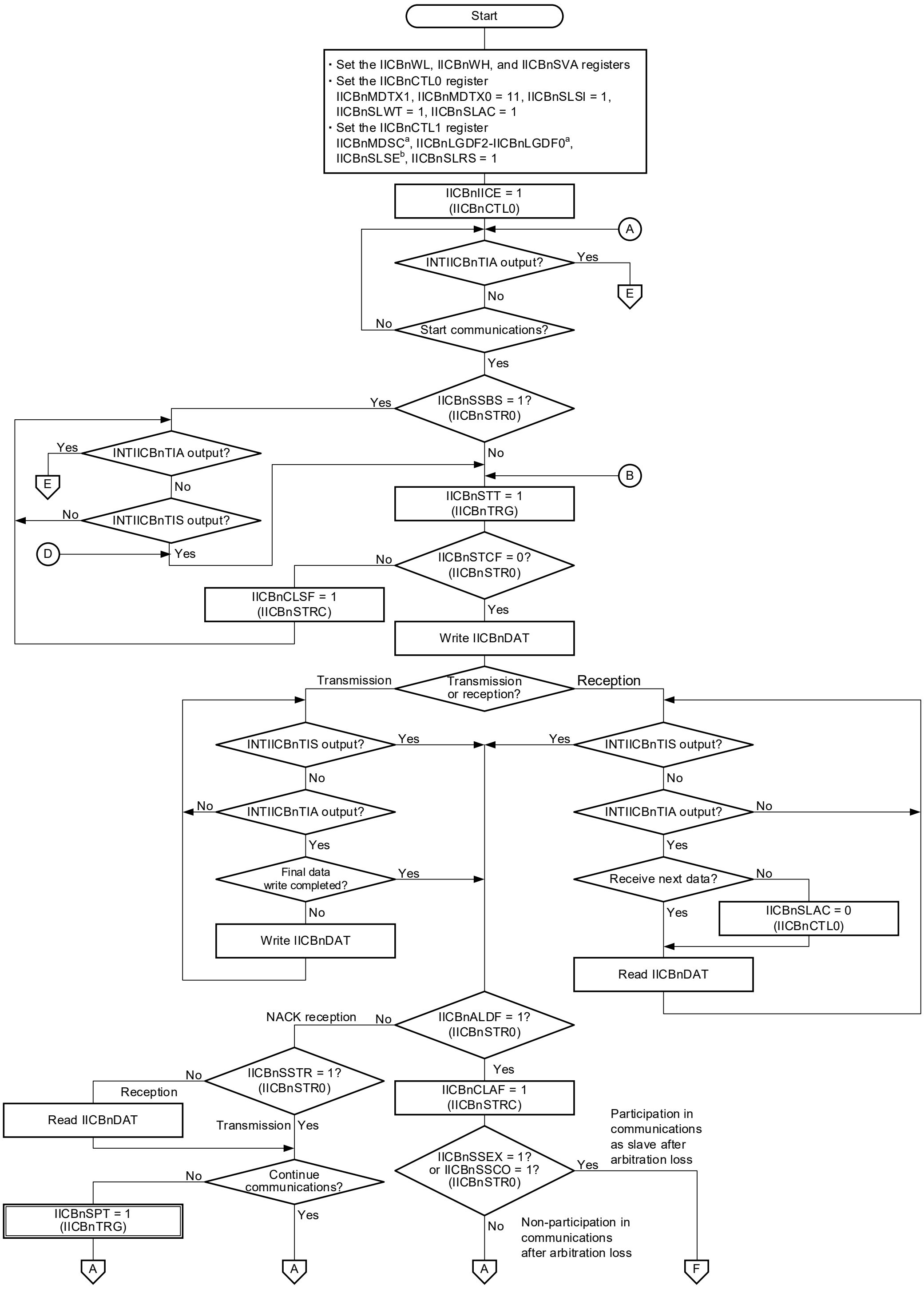


Figure 24.20 Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (IICBnCTL1.IICBnSLRS bit = 0) (Multi-Master Environment) (2/2)

(4) Continuous transfer mode setting procedure when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1)



a) Make settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.

Note: The double-boxed items indicate processing that exits the wait state.

Figure 24.21 Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (1/2)

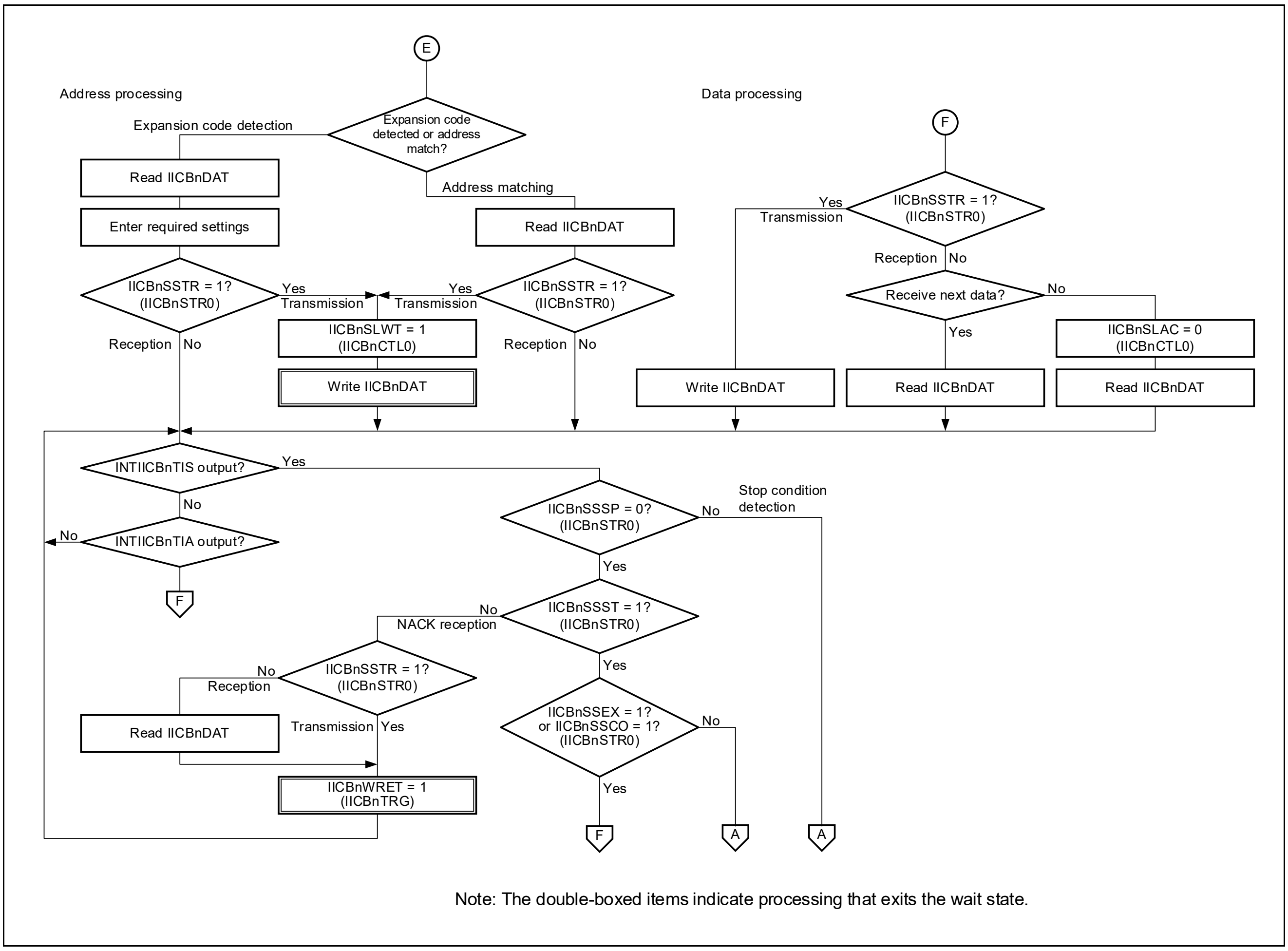


Figure 24.21 Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (2/2)

25. CAN Controller (FCN)

This section explains the CAN (Controller Area Network) controllers that comply with the CAN protocol as standardized in ISO 11898.

25.1 Features of FCN

This product has the following number of channels of the CAN controller.

Table 25.1 Channels of FCN

FCN	
Number of channels	2
Name	FCN0, FCN1

- Meaning of "n":
  Throughput this section, the individual channels of the CAN controller are identified by the index "n" (n = 0, 1); for example, FCNnGMCLCTL for the FCNn control register.

Table 25.2 Message Buffers of FCN Channels

Channels	Number m of Message Buffers
FCN0	64
FCN1	64

- Meaning of "m":
  Throughout this section, the FCN message buffer registers are identified by "m" (m = 000–063); for example, FCNnMmDAT4B for message data byte 4 of message buffer register m for FCN channel n.

- Interrupts and peripheral modules:

  The following interrupt requests from FCN can be used as triggers for interrupt service routines or hardware ISRs (where listed as such), for DMA transfer (by the general-purpose DMAC or real-time port DMAC), for capture by a timer (TAUJ2 or TAUD), and for updating the real-time port pins (RP00–RP37).

Table 25.3 FCNn Interrupts and Requests for Peripheral Modules

FCNn Interrupt Signal	Function	Connected To
FCN0		
INTC0ERR	FCN0 error detection	• Interrupt controller INTFCN0ERR
INTC0REC	FCN0 reception completion	• Interrupt controller INTFCN0REC • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTC0TRX	FCN0 transmission completion	• Interrupt controller INTFCN0TRX • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTC0WUP	FCN0 sleep wake-up/transmission abortion	• Interrupt controller INTFCN0WUP • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
FCN1		
INTC1ERR	FCN1 error detection	• Interrupt controller INTFCN1ERR
INTC1REC	FCN1 reception completion	• Interrupt controller INTFCN1REC • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTC1TRX	FCN1 transmission completion	• Interrupt controller INTFCN1TRX • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)
INTC1WUP	FCN1 sleep wake-up/transmission abortion	• Interrupt controller INTFCN1WUP • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR/TMDTFR) • Real-time port trigger (RPTFR)

- I/O signals:
  The I/O signals of the CAN controllers are listed in the Table 25.4.

Table 25.4 FCN I/O Signals

FCNn Signals	Function	Connected To
FCN0		
CRXD0	FCN0 CAN bus reception input	Port 53 (CRXD0)
CTXD0	FCN0 CAN bus transmission output	Port 54 (CTXD0)
FCN1		
CRXD1	FCN1 CAN bus receive input	Port 55 (CRXD1)
CTXD1	FCN1 CAN bus transmit output	Port 56 (CTXD1)

25.2 Features

- Compliant with ISO 11898
- Standard frame and extended frame transmission/reception
- Transfer rate: up to 1 Mbps (If FCN clock input ≥ 16 MHz)
- 64 message buffers per channel
- Receive/transmit history list function (can be set individually for each message buffer)
- Automatic block transmission
- Multi-buffer reception blocking
- Mask setting of 8 patterns is possible for each channel, applicable for data and remote frames
- Data bit time, communication baud rate and sample point can be controlled FCN by FCN module bit-rate prescaler register (FCNnCMBRPRS) and bit rate register (FCNnCMBTCTL)
  - For example, the following sample-point can be configured: 66.7%, 70.0%, 75.0%, 80.0%, 81.3%, 85.0%, 87.5%
  - Baud rates in the range of 10 kbps up to 1 Mbps can be configured
- Enhanced features:
  - Each message buffer can be configured to operate as a transmit or a receive message buffer
  - A transmission request can be aborted by clearing the transmission request flag of the concerned message buffer. Support for transmission abort interrupts upon successful abortion.
  - Automatic block transmission operation mode (ABT)
  - Timestamping for FCN channels 0 to 2 in collaboration with timers capture channels
  - Centrally managed global data update bit monitor registers allow checking of all data update bits from one location.

25.2.1 Overview of Functions

Table 25.5 lists an overview of the CAN controller functions.

Table 25.5 Overview of Functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Up to 1 Mbps (minimum FCN clock input = 16 MHz)
Data storage	Storing messages in the FCN RAM
Number of messages	• 64/128 message buffers per channel • Each message buffer can be set to be either a transmit message buffer or a receive message buffer.
Message reception	• Unique ID can be set to each message buffer. • Mask setting of 8 patterns is possible for each channel, applicable for data and remote frames • A receive completion interrupt is generated each time a message is received and stored in a message buffer (receive completion interrupts can be enabled/disabled for each message buffer) • Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer reception blocking). • Receive history list function (can be set individually for each message buffer) • Centrally managed global data update bit monitor registers
Message transmission	• Unique ID can be set to each message buffer. • Receive completion interrupts can be enabled/disabled for each message buffer • Transmit Abort interrupt and Transmit Completion flag for each message buffer (only one transmission of any buffer can be aborted at a time) • Message buffer numbers 0 to 15/31 specified as the transmit message buffers can be used for automatic block transfer. The message transmission interval is programmable (using the automatic block transmission ("ABT") function). • Transmission history list function (can be set individually for each message buffer)
Remote frame processing	• Remote frame processing by transmit message buffer • Remote frame processing by receive message buffer, when applying one of the 8 masks
Timestamping	• Timestamping can be set for reception of messages when a 32-bit timer is used in combination. • Timestamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected).
Diagnosis	• Readable error counters • "Valid protocol operation flag" for verification of bus connections • Receive-only mode • Single-shot mode • CAN protocol error identification • Self-test mode
Release from bus-off state	• Forced release from bus-off possible by software. • No automatic release from bus-off (software must send recovery request).
Power save mode	• CAN sleep mode (can be woken up by CAN bus) • CAN stop mode (cannot be woken up by CAN bus)

25.2.2 Configuration

The CAN controller is composed of the following four blocks.

- APB interface
  This functional block provides an APB interface and a means of transmitting and receiving messages between the FCN module and the host CPU.
- MCM (Message Control Module)
  This functional block controls access to the CAN protocol layer and to the FCN RAM within the FCN module.
- CAN protocol layer
  This functional block is involved in the operation of the CAN protocol and its related settings
- CAN RAM
  This is the CAN memory functional block, which is used to store message IDs, message data, etc.

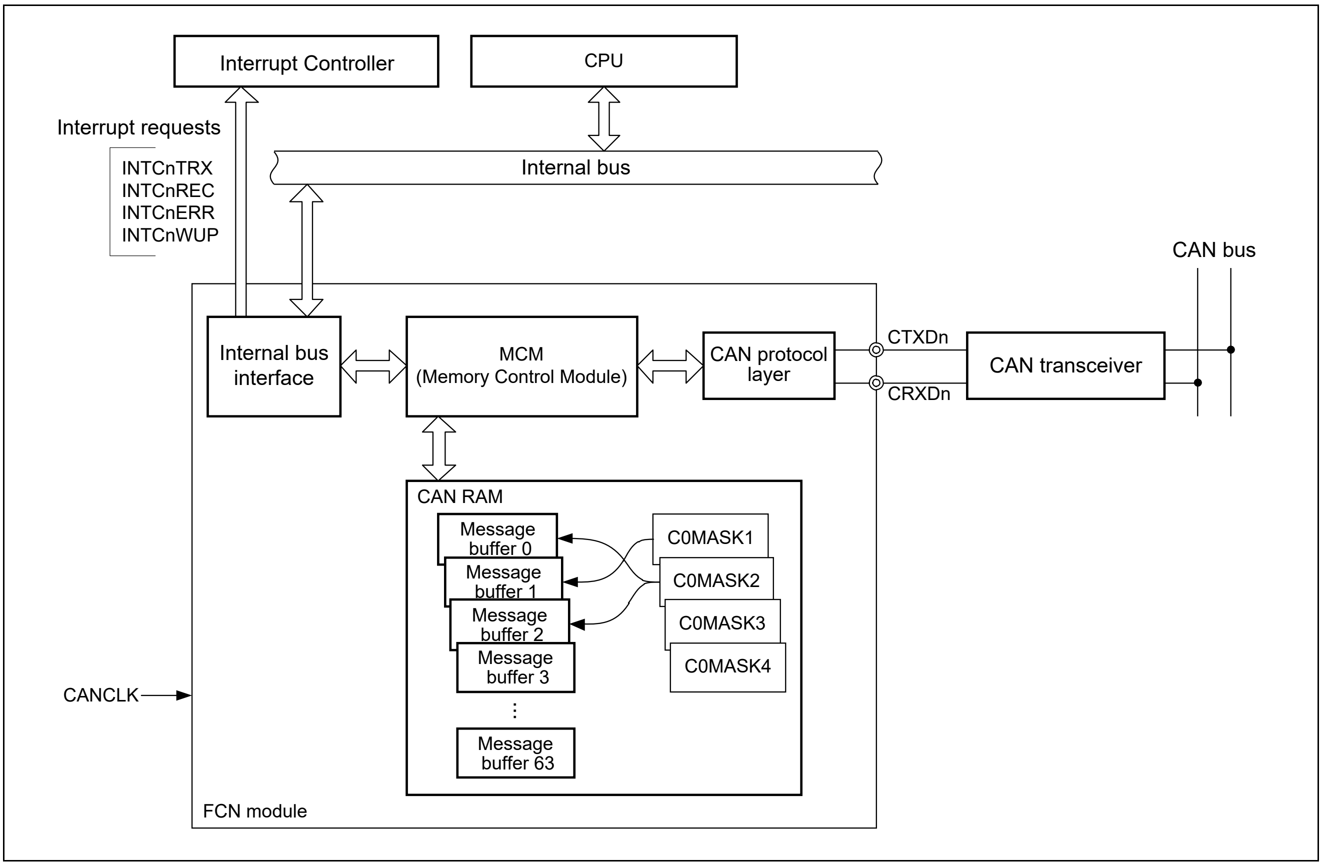


Figure 25.1 Block Diagram of the CAN Controller

25.3 Internal Registers of FCN

25.3.1 CAN Controller Configuration

Table 25.6 List of FCN Registers

(1/2)

Item	Register Name
FCNn global registers	FCNn global control register (FCNnGMCLCTL)
	FCNn global clock selection register (FCNnGMCSPRE)
	FCNn global automatic block transmission control register (FCNnGMABCTL)
	FCNn global automatic block transmission delay setting register (FCNnGMADCTL)
	FCNn global data update bit monitor registers (FCNnDNBMRX0–FCNnDNBMRX1)
FCNn module registers	FCNn module mask 1 registers (FCNnCMMKCTL01H, FCNnCMMKCTL02H, FCNnCMMKCTL01W)
	FCNn module mask 2 registers (FCNnCMMKCTL03H, FCNnCMMKCTL04H, FCNnCMMKCTL03W)
	FCNn module mask3 registers (FCNnCMMKCTL05H, FCNnCMMKCTL06H, FCNnCMMKCTL05W)
	FCNn module mask 4 registers (FCNnCMMKCTL07H, FCNnCMMKCTL08H, FCNnCMMKCTL07W)
	FCNn module mask 5 registers (FCNnCMMKCTL09H, FCNnCMMKCTL10H, FCNnCMMKCTL09W)
	FCNn module mask 6 registers (FCNnCMMKCTL11H, FCNnCMMKCTL12H, FCNnCMMKCTL11W)
	FCNn module mask 7 registers (FCNnCMMKCTL13H, FCNnCMMKCTL14H, FCNnCMMKCTL13W)
	FCNn module mask 8 registers (FCNnCMMKCTL15H, FCNnCMMKCTL16H, FCNnCMMKCTL15W)
	FCNn module control register (FCNnCMCLCTL)
	FCNn module last error information register (FCNnCMLCSTR)
	FCNn module information register (FCNnCMINSTR)
	FCNn module error counter register (FCNnCMERCNT)
	FCNn module interrupt enable register (FCNnCMIECTL)
	FCNn module interrupt status register (FCNnCMISCTL)
	FCNn module bit rate prescaler and FCN clock selector register (FCNnCMBRPRS)
	FCNn module bit rate register (FCNnCMBTCTL)
	FCNn module last in-pointer register (FCNnCMLISTR)
	FCNn module receive history list register (FCNnCMRGRX)
	FCNn module last out-pointer register (FCNnCMLOSTR)
	FCNn module transmit history list register (FCNnCMTGTX)
	FCNn module timestamp register (FCNnCMTSCTL)

Table 25.6 List of FCN Registers

(2/2)

Item	Register Name
FCN message buffer registers	FCNn message data byte 0 to 3 registers m (FCNnMmDAT0W, FCNnMmDAT0H, FCNnMmDAT2H, FCNnMmDAT0B, FCNnMmDAT1B, FCNnMmDAT2B, FCNnMmDAT3B)
	FCNn message data byte 4 to 7 registers m (FCNnMmDAT4W, FCNnMmDAT4H, FCNnMmDAT6H, FCNnMmDAT4B, FCNnMmDAT5B, FCNnMmDAT6B, FCNnMmDAT7B)
	FCNn message data length register m (FCNnMmDTLGB)
	FCNn message configuration register m (FCNnMmSTRB)
	FCNn message ID registers m (FCNnMmMID0H, FCNnMmMID1H, FCNnMmMID0W)
	FCNn message control register m (FCNnMmCTL)

25.3.2 CAN Controller Registers Overview

(1) FCNn Global and Module Registers

Table 25.7 FCN0 Global and Module Registers

(1/2)

Address Offset	Register Name	Symbol	R/W	Access Bit	After Reset
4002 0008H	FCN0 global clock selection register	FCN0GMCSPRE	R/W	8	0FH
4002 0020H	FCN0 global automatic block transmission delay setting register	FCN0GMADCTL	R/W	8	00H
4002 8000H	FCN0 global control register	FCN0GMCLCTL	R/W	16	00X0H Note1
4002 8018H	FCN0 global automatic block transmission control register	FCN0GMABCTL	R/W	16	0000H
4003 00C0H	FCN0 global data update bit monitor register 0	FCN0DNBMRX0	R	32	Note2
4003 00D0H	FCN0 global data update bit monitor register 1	FCN0DNBMRX1	R	32	Note2
4002 8300H	FCN0 module mask 1 register	FCN0CMMKCTL01H	R/W	16	Note2
4002 8308H		FCN0CMMKCTL02H			
4003 0300H		FCN0CMMKCTL01W		32	
4002 8310H	FCN0 module mask 2 register	FCN0CMMKCTL03H	R/W	16	Note2
4002 8318H		FCN0CMMKCTL04H			
4003 0310H		FCN0CMMKCTL03W		32	
4002 8320H	FCN0 module mask 3 register	FCN0CMMKCTL05H	R/W	16	Note2
4002 8328H		FCN0CMMKCTL06H			
4003 0320H		FCN0CMMKCTL05W		32	
4002 8330H	FCN0 module mask 4 register	FCN0CMMKCTL07H	R/W	16	Note2
4002 8338H		FCN0CMMKCTL08H			
4003 0330H		FCN0CMMKCTL07W		32	

Notes 1. The initial value depends on FCNnGMCLCTL.FCNnGMCLECCF, which indicates whether an error has been detected when reading from the message buffer RAM. Refer to the detailed description of the FCNnGMCLCTL register.

2. The value after a reset is 0000H or 00000000H.

Table 25.7 FCN0 Global and Module Registers

(2/2)

Address Offset	Register Name	Symbol	R/W	Access Bit	After Reset
4002 8340H	FCN0 module mask 5 register	FCN0CMMKCTL09H	R/W	16	Note
4002 8348H		FCN0CMMKCTL10H			
4003 0340H		FCN0CMMKCTL09W		32	
4002 8350H	FCN0 module mask 6 register	FCN0CMMKCTL11H	R/W	16	Note
4002 8358H		FCN0CMMKCTL12H			
4003 0350H		FCN0CMMKCTL11W		32	
4002 8360H	FCN0 module mask 7 register	FCN0CMMKCTL13H	R/W	16	Note
4002 8368H		FCN0CMMKCTL14H			
4003 0360H		FCN0CMMKCTL13W		32	
4002 8370H	FCN0 module mask 8 register	FCN0CMMKCTL15H	R/W	16	Note
4002 8378H		FCN0CMMKCTL16H			
4003 0370H		FCN0CMMKCTL15W		32	
4002 0248H	FCN0 module last error information register	FCN0CMLCSTR	R/W	8	00H
4002 024CH	FCN0 module information register	FCN0CMINSTR	R	8	00H
4002 0268H	FCN0 module bit-rate prescaler register	FCN0CMBRPRS	R/W	8	FFH
4002 0278H	FCN0 module last receive pointer register	FCN0CMLISTR	R	8	Undefined
4002 0288H	FCN0 module last transmit pointer register	FCN0CMLOSTR	R	8	Undefined
4002 8240H	FCN0 module control register	FCN0CMCLCTL	R/W	16	0000H
4002 8250H	FCN0 module error counter register	FCN0CMERCNT	R	16	0000H
4002 8258H	FCN0 module interrupt enable register	FCN0CMIECTL	R/W	16	0000H
4002 8260H	FCN0 module interrupt status register	FCN0CMISCTL	R/W	16	0000H
4002 8270H	FCN0 module bit-rate register	FCN0CMBTCTL	R/W	16	370FH
4002 8280H	FCN0 module receive history list register	FCN0CMRGRX	R/W	16	xx02H
4002 8290H	FCN0 module transmit history list register	FCN0CMTGTX	R/W	16	xx02H
4002 8298H	FCN0 module timestamp register	FCN0CMTSCTL	R/W	16	0000H

Note: The value after a reset is 0000H or 00000000H.

Table 25.8 FCN1 Global and Module Registers

(1/2)

Address Offset	Register Name	Symbol	R/W	Access Bit	After Reset
4004 0008H	FCN1 global clock selection register	FCN1GMCSPRE	R/W	8	0FH
4004 0020H	FCN1 global automatic block transmission delay setting register	FCN1GMADCTL	R/W	8	00H
4004 8000H	FCN1 global control register	FCN1GMCLCTL	R/W	16	00X0H Note1
4004 8018H	FCN1 global automatic block transmission control register	FCN1GMABCTL	R/W	16	0000H
4005 00C0H	FCN1 global data update bit monitor register 0	FCN1DNBMRX0	R	32	Note2
4005 00D0H	FCN1 global data update bit monitor register 1	FCN1DNBMRX1	R	32	Note2
4004 8300H	FCN1 module mask 1 register	FCN1CMMKCTL01H	R/W	16	Note2
4004 8308H		FCN1CMMKCTL02H			
4005 0300H		FCN1CMMKCTL01W		32	
4004 8310H	FCN1 module mask 2 register	FCN1CMMKCTL03H	R/W	16	Note2
4004 8318H		FCN1CMMKCTL04H			
4005 0310H		FCN1CMMKCTL03W		32	
4004 8320H	FCN1 module mask 3 register	FCN1CMMKCTL05H	R/W	16	Note2
4004 8328H		FCN1CMMKCTL06H			
4005 0320H		FCN1CMMKCTL05W		32	
4004 8330H	FCN1 module mask 4 register	FCN1CMMKCTL07H	R/W	16	Note2
4004 8338H		FCN1CMMKCTL08H			
4005 0330H		FCN1CMMKCTL07W		32	

Notes 1. The initial value depends on FCNnGMCLCTL.FCNnGMCLECCF, which indicates whether an error has been detected when reading from the message buffer RAM. Refer to the detailed description of the FCNnGMCLCTL register.

2. The value after a reset is 0000H or 00000000H.

Table 25.8 FCN1 Global and Module Registers

(2/2)

Address Offset	Register Name	Symbol	R/W	Access Bit	After Reset
4004 8340H	FCN1 module mask 5 register	FCN1CMMKCTL09H	R/W	16	Note
4004 8348H		FCN1CMMKCTL10H			
4005 0340H		FCN1CMMKCTL09W		32	
4004 8350H	FCN1 module mask 6 register	FCN1CMMKCTL11H	R/W	16	Note
4004 8358H		FCN1CMMKCTL12H			
4005 0350H		FCN1CMMKCTL11W		32	
4004 8360H	FCN1 module mask 7 register	FCN1CMMKCTL13H	R/W	16	Note
4004 8368H		FCN1CMMKCTL14H			
4005 0360H		FCN1CMMKCTL13W		32	
4004 8370H	FCN1 module mask 8 register	FCN1CMMKCTL15H	R/W	16	Note
4004 8378H		FCN1CMMKCTL16H			
4005 0370H		FCN1CMMKCTL15W		32	
4004 0248H	FCN1 module last error information register	FCN1CMLCSTR	R/W	8	00H
4004 024CH	FCN1 module information register	FCN1CMINSTR	R	8	00H
4004 0268H	FCN1 module bit-rate prescaler register	FCN1CMBRPRS	R/W	8	FFH
4004 0278H	FCN1 module last receive pointer register	FCN1CMLISTR	R	8	Undefined
4004 0288H	FCN1 module last transmit pointer register	FCN1CMLOSTR	R	8	Undefined
4004 8240H	FCN1 module control register	FCN1CMCLCTL	R/W	16	0000H
4004 8250H	FCN1 module error counter register	FCN1CMERCNT	R	16	0000H
4004 8258H	FCN1 module interrupt enable register	FCN1CMIECTL	R/W	16	0000H
4004 8260H	FCN1 module interrupt status register	FCN1CMISCTL	R/W	16	0000H
4004 8270H	FCN1 module bit-rate register	FCN1CMBTCTL	R/W	16	370FH
4004 8280H	FCN1 module receive history list register	FCN1CMRGRX	R/W	16	xx02H
4004 8290H	FCN1 module transmit history list register	FCN1CMTGTX	R/W	16	xx02H
4004 8298H	FCN1 module timestamp register	FCN1CMTSCTL	R/W	16	0000H

Note: The value after a reset is 0000H or 00000000H.

25.3.3 Bit Configuration of Registers

The addresses of registers in the CAN controller are defined as offsets from the FCNn base addresses.

Channels	Base Addresses
FCN0	4002 0000H
FCN1	4004 0000H

Table 25.9 Bit Configuration of FCN Global Registers

Address Offset	Symbol	Bit 7/15/31/23	Bit 6/14/30/22	Bit 5/13/29/21	Bit 4/12/28/20	Bit 3/11/27/19	Bit 2/10/26/18	Bit 1/9/25/17	Bit 0/8/24/16
0 8000H	FCNnGMCLCTL (W)	0	0	FCNnGM CLCLMB		0	0	0	FCNnGMC LCLOM
		0	0	0	FCNnGM CLSESR	0	0	FCNnGM CLSEDE	FCNnGMC LSEOM
	FCNnGMCLCTL (R)	0	0	FCNnGM CLECCF	FCNnGM CLSORF	0	0	FCNnGM CLESDE	FCNnGMC LPWOM
		FCN0GM CLSSMO	0	0	0	0	0	0	0
0 0008H	FCNnGMCSPRE	0	0	0	0	FCNnGMCSPRSC[3:0]			
0 8018H	FCNnGMABCTL (W)	0	0	0	0	0	0	0	FCNnGM ABCLAT
		0	0	0	0	0	0	FCNnGM ABSEAC	FCNnGM ABSEAT
	FCNnGMABCTL (R)	0	0	0	0	0	0	FCNnGM ABCLRF	FCNnGM ABABTT
		0	0	0	0	0	0	0	0
0 0020H	FCNnGMADCTL	0	0	0	0	FCNnGMADSSAD[3:0]			
1 00C0H	FCNnDNBMRX0 (R)	FCNnDNBMSSDN[7:0]							
		FCNnDNBMSSDN[15:8]							
		FCNnDNBMSSDN[23:16]							
		FCNnDNBMSSDN[31:24]							
1 00D0H	FCNnDNBMRX1 (R)	FCNnDNBMSSDN[39:32]							
		FCNnDNBMSSDN[47:40]							
		FCNnDNBMSSDN[55:48]							
		FCNnDNBMSSDN[63:56]							

Table 25.10 Bit Configuration of FCN Module Mask Control 16-Bit Registers

Address Offset	Symbol	Bit 15	Bit 14	Bit 13	Bits 12 to 0
0 8300H	FCNnCMMKCTL01H	FCNnCMMKSSID[15:0]			
0 8308H	FCNnCMMKCTL02H	0	0	0	FCNnCMMKSSID[28:16]
0 8310H	FCNnCMMKCTL03H	FCNnCMMKSSID[15:0]			
0 8318H	FCNnCMMKCTL04H	0	0	0	FCNnCMMKSSID[28:16]
0 8320H	FCNnCMMKCTL05H	FCNnCMMKSSID[15:0]			
0 8328H	FCNnCMMKCTL06H	0	0	0	FCNnCMMKSSID[28:16]
0 8330H	FCNnCMMKCTL07H	FCNnCMMKSSID[15:0]			
0 8338H	FCNnCMMKCTL08H	0	0	0	FCNnCMMKSSID[28:16]
0 8340H	FCNnCMMKCTL09H	FCNnCMMKSSID[15:0]			
0 8348H	FCNnCMMKCTL10H	0	0	0	FCNnCMMKSSID[28:16]
0 8350H	FCNnCMMKCTL11H	FCNnCMMKSSID[15:0]			
0 8358H	FCNnCMMKCTL12H	0	0	0	FCNnCMMKSSID[28:16]
0 8360H	FCNnCMMKCTL13H	FCNnCMMKSSID[15:0]			
0 8368H	FCNnCMMKCTL14H	0	0	0	FCNnCMMKSSID[28:16]
0 8370H	FCNnCMMKCTL15H	FCNnCMMKSSID[15:0]			
0 8378H	FCNnCMMKCTL16H	0	0	0	FCNnCMMKSSID[28:16]

Table 25.11 Bit Configuration of FCN Module Mask Control 32-Bit Registers

Address Offset	Symbol	Bit 31	Bit 30	Bit 29	Bits 28 to 0
1 0300H	FCNnCMMKCTL01W	0	0	0	FCNnCMMKSSID[28:0]
1 0310H	FCNnCMMKCTL03W	0	0	0	FCNnCMMKSSID[28:0]
1 0320H	FCNnCMMKCTL05W	0	0	0	FCNnCMMKSSID[28:0]
1 0330H	FCNnCMMKCTL07W	0	0	0	FCNnCMMKSSID[28:0]
1 0340H	FCNnCMMKCTL09W	0	0	0	FCNnCMMKSSID[28:0]
1 0350H	FCNnCMMKCTL11W	0	0	0	FCNnCMMKSSID[28:0]
1 0360H	FCNnCMMKCTL13W	0	0	0	FCNnCMMKSSID[28:0]
1 0370H	FCNnCMMKCTL15W	0	0	0	FCNnCMMKSSID[28:0]

Table 25.12 Bit Configuration of FCN Module Registers

Address Offset	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
0 8240H	FCNnCMCLCTL (W)	0	FCNnCM CLCLAL	FCNnCM CLCLVL	FCNnCMCLCLPS [1:0]		FCNnCMCLCLOP[2:0]		
		FCNnCM CLSERC	FCNnCM CLSEAL	0	FCNnCMCLSEPS[1:0]		FCNnCMCLSEOP[2:0]		
	FCNnCMCLCTL (R)	FCNnCM CLERCF	FCNnCM CLALBF	FCNnCM CLVALF	FCNnCMCLMDPF[1:0]		FCNnCMCLMDOF[2:0]		
		0	0	0	0	0	0	FCNnCM CLSSRS	FCNnCM CLSSTS
0 00248H	FCNnCMLCSTR (W)	0	0	0	0	0	0	0	0
	FCNnCMLCSTR (R)	0	0	0	0	0	FCN0CMLCSSLC[2:0]		
0 024CH	FCNnCMINSTR	0	0	0	FCNnCMI NBOFF	FCNnCMINSSTE[1:0]		FCNnCMINSSRE[1:0]	
0 8250H	FCNnCMERCNT	FCNnCMERTECF[7:0]							
		FCNnCM ERRPSF	FCNnCMERRECF[6:0]						
0 8258H	FCNnCMIECTL (W)	0	FCNnCMIECLIE[6:0]						
		0	FCNnCMIESEIE[6:0]						
	FCNnCMIECTL (R)	0	FCNnCMIEINTF[6:0]						
		0	0	0	0	0	0	0	0
0 8260H	FCNnCMISCTL (W)	0	FCNnCMISCLTS[6:0]						
		0	0	0	0	0	0	0	0
	FCNnCMISCTL (R)	0	FCNnCMISITSF[6:0]						
		0	0	0	0	0	0	0	0
0 0268H	FCNnCMBRPRS	FCNnCMBRPRS[7:0]							
0 8270H	FCNnCMBTCTL	0	0	0	0	FCNnCMBTS1LG[3:0]			
		0	0	FCNnCMBTJWLG[1:0]		0	FCNnCMBTS2LG[2:0]		
0 0278H	FCNnCMLISTR	FCNnCMLISSLR[7:0]							
0 8280H	FCNnCMRGRX (W)	0	0	0	0	0	0	0	FCNnCM RGCLRV
		0	0	0	0	0	0	0	0
	FCNnCMRGRX (R)	0	0	0	0	0	0	FCNnCM RGSSPM	FCNnCM RGRVFF
		FCNnCMRDSSPT[7:0]							
0 0288H	FCNnCMLOSTR	FCNnCMLOSSLT[7:0]							
0 8290H	FCNnCMTGTX (W)	0	0	0	0	0	0	0	FCNnCM TGCLTV
		0	0	0	0	0	0	0	0
	FCNnCMTGTX (R)	0	0	0	0	0	0	FCNnCM TGSSPM	FCNnCM TGTVFF
		FCNnCMTGSSPT[7:0]							
0 8298H	FCNnCMTSCTL (W)	0	0	0	0	0	FCNnCM TSCLLK	FCNnCM TSCLSL	FCNnCM TSCLTS
		0	0	0	0	0	FCNnCM TSSELK	FCNnCM TSSESL	FCNnCM TSSETS
	FCNnCMTSCTL (R)	0	0	0	0	0	FCNnCM TSLOKE	FCNnCM TSSELE	FCNnCM TSTSGE
		0	0	0	0	0	0	0	0

Table 25.13 Bit Configuration of FCN Message Buffer Registers

(1/2)

Address Offset	Symbol	Bit 7/15/31/23	Bit 6/14/30/22	Bit 5/13/29/21	Bit 4/12/28/20	Bit 3/11/27/19	Bit 2/10/26/18	Bit 1/9/25/17	Bit 0/8/24/16
1 1000H + m × 40H	FCNnMmDAT0W	FCNnMmSSD[07:00]							
		FCNnMmSSD[17:10]							
		FCNnMmSSD[27:00]							
		FCNnMmSSD[37:30]							
0 9000H + m × 40H	FCNnMmDAT0H	FCNnMmSSD[07:00]							
		FCNnMmSSD[17:10]							
0 1000H + m × 40H	FCNnMmDAT0B	FCNnMmSSD[07:00]							
0 1004H + m × 40H	FCNnMmDAT1B	FCNnMmSSD[17:10]							
0 9008H + m × 40H	FCNnMmDAT2H	FCNnMmSSD[27:20]							
		FCNnMmSSD[37:30]							
0 1008H + m × 40H	FCNnMmDAT2B	FCNnMmSSD[27:20]							
0 100CH + m × 40H	FCNnMmDAT3B	FCNnMmSSD[37:30]							
1 1010H + m × 40H	FCNnMmDAT4W	FCNnMmSSD[47:40]							
		FCNnMmSSD[57:50]							
		FCNnMmSSD[67:60]							
		FCNnMmSSD[77:70]							
0 9010H + m × 40H	FCNnMmDAT4H	FCNnMmSSD[47:40]							
		FCNnMmSSD[57:50]							
0 1010H + m × 40H	FCNnMmDAT4B	FCNnMmSSD[47:40]							
0 1014H + m × 40H	FCNnMmDAT5B	FCNnMmSSD[57:50]							
0 9018H + m × 40H	FCNnMmDAT6H	FCNnMmSSD[67:60]							
		FCNnMmSSD[77:70]							
0 1018H + m × 40H	FCNnMmDAT6B	FCNnMmSSD[67:60]							
0 101CH + m × 40H	FCNnMmDAT7B	FCNnMmSSD[77:70]							

Table 25.13 Bit Configuration of FCN Message Buffers

(2/2)

Address Offset	Symbol	Bit 7/15/31/23	Bit 6/14/30/22	Bit 5/13/29/21	Bit 4/12/28/20	Bit 3/11/27/19	Bit 2/10/26/18	Bit 1/9/25/17	Bit 0/8/24/16
0 1020H + m × 40H	FCNnMmDTLGB	0				FCNnMmDTLG[3:0]			
0 1024H + m × 40H	FCNnMmSTRB	FCNnMm SSOW	FCNnMmSSMT[3:0]				FCNnMm SSRT	0	FCNnMm SSAM
0 9028H + m × 40H	FCNnMmMID0H	FCNnMmSSID[7:0]							
		FCNnMmSSID[15:8]							
0 9030H + m × 40H	FCNnMmMID1H	FCNnMmSSID[23:16]							
		FCNnMm SSIE	0	0	FCNnMmSSID[28:24]				
1 1028H + m × 40H	FCNnMmMID0W	FCNnMmSSID[7:0]							
		FCNnMmSSID[15:8]							
		FCNnMmSSID[23:16]							
		FCNnMm SSIE	0	0	FCNnMmSSID[28:24]				
0 9038H + m × 40H	FCNnMmCTL (W)	0	FCNnMm CLNH	0	FCNnMm CLMW	FCNnMm CLIE	FCNnMm CLDN	FCNnMm CLTR	FCNnMm CLRY
		0	FCNnMm SENH	0	0	FCNnMm SEIE	0	FCNnMm SETR	FCNnMm SERY
	FCNnMmCTL (R)	0	FCNnMm NHMF	0	FCNnMm MOWF	FCNnMmI ENF	FCNnMm DTNF	FCNnMm TRQF	FCNnMm RDYF
		0	0	FCNnMm MUCF	0	0	0	FCNnMm TCPF	0

25.4 Setting or Clearing of Bits

The FCN control registers include registers whose bits can be set or cleared via the CPU and via the CAN controller. These register bits cannot be changed directly by the CPU bit-band access. Instead a special bit-set/bit-clear mechanism is used.

All registers where bit manipulation operations are prohibited are organized in such a way that all bits allowed for changing by the CPU are located in the lower byte (RWx in the register layout below), while in the upper byte either no or read-only information is located (ROx in the register layout below).

The registers can be read in the usual way of acquiring all 16 data bits in their current setting and as described in the register description.

When writing 16-bit data to the register address, the following mechanism is used to set or clear the 8 lower-order bits.

(1) Clearing Bits

Each of the 8 lower-order data bits (CLx in the register layout below) indicates whether the corresponding register bit RWx should be

- cleared, i.e. set to 0: if CLx = 1, the corresponding RWx is cleared to 0
- remain unchanged: if CLx = 0, the corresponding RWx does not change

(2) Setting Bits

Each of the upper 8 data bits (SEx in the register layout below) indicate whether the corresponding register bit should be

- set, i.e. set to 1: if SEx = 1, the corresponding RWx is set to 1
- remain unchanged: if SEx = 0, the corresponding RWx does not change

Register layout for read access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RO7	RO6	RO5	RO4	RO3	RO2	RO1	RO0	RW7	RW6	RW5	RW4	RW3	RW2	RW1	RW0
changing by the CPU not possible								bits for CPU manipulation via SE7-SE0 and CL7-CL0							

Register layout for write access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	CL7	CL6	CL5	CL4	CL3	CL2	CL1	CL0
SEx = 1 sets the corresponding RW7-RW0								CLx = 1 clears the corresponding RW7-RW0							

The following table denotes the operations applied to the RWx bits:

Table 25.14 Bit Set/Clear Operation

CLx	SEx	Operation on RWx
0	0	RWx: Not changed
0	1	RWx: Set (1)
1	0	RWx: Cleared (0)
1	1	RWx: Not changed

Example The following shows an example.
Changing the register with the content 1883H as follows:
- Bit 3 shall be set to 1: SE3 = 1
- Bit 1 shall be cleared (0): CL1 = 1

Register read before bit manipulations:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	1
may hold any value, here 18H								RW7 to RW0: 83H							

Register write access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
SE3 = 1:08H								CL1 = 1:02H							

Register read after bit manipulations:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	1
may hold any value, here 18H								RW7 to RW0: 89H							

25.5 Control Registers

25.5.1 FCN Global Registers

(1) FCNn Global Control Register (FCNnGMCLCTL)

This register is used to control the operation of the FCN module.

- Access: This register can be read or written in 16-bit units.
- Address: <FCNn_base> + 0 8000H
- Initial Value: 00x0H[Note]
  The register is initialized by any reset.

Note: Software reset starts automatically after hardware reset.
So the initial value is:
- If an error is not detected after software reset, then it is 0000H.
- If an error is not detected on software reset, then it is 0010H.
- If an error is detected after software reset, then it is 0020H.
- If an error is detected on software reset, then it is 0030H.

(a) When FCNnGMCLCTL is read

(1/2)

15	14	13	12	11	10	9	8
FCNnGM CLSSMO	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	FCNnGM CLECCF	FCNnGM CLSORF	0	0	FCNnGM CLESDE	FCNnGM CLPWOM

Bit Position	Bit Name	Description
15	FCNnGMCLSSMO	Enables access to the FCN message buffer register and transmit/receive history registers 0: Write access and read access to the FCN message buffer register and the transmit/receive history list registers are disabled. 1: Write access and read access to the FCN message buffer register and the transmit/receive history list registers are enabled. **Cautions 1. While the FCNnGMCLCTL.FCNnGMCLSSMO is cleared to 0, software access to FCN message buffer registers (i.e. all FCNnMm registers) and registers related to transmit history or receive history (FCNnCMLOSTR, FCNnCMTGTX, FCNnCMLISTR, FCNnCMRGRX) is disabled.** **2. FCNnGMCLCTL.FCNnGMCLSSMO is read-only. Even if 1 is written while it is 0, its value does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.** **Remark: FCNnGMCLCTL.FCNnGMCLSSMO is cleared to 0 when the FCN module enters FCN sleep mode or FCN stop mode, or when the FCNnGMCLCTL.FCNnGMCLPWOM is cleared to 0.** **FCNnGMCLSSMO is set to 1 when the FCN sleep mode or FCN stop mode is released, or when the FCNnGMCLCTL.FCNnGMCLPWOM is set to 1.**

(2/2)

Bit Position	Bit Name	Description
5	FCNnGMCLECCF	Message buffer RAM read error detect bit 0: Indicates that no error was detected when the message buffer RAM was read. 1: Indicates that an error was detected when the message buffer RAM was read.
4	FCNnGMCLSORF	Software reset execution status bit 0: No software reset 1: Software reset is ongoing. **Remarks 1. While a software reset is ongoing (FCNnGMCLCTL.FCNnGMCLSORF is set to 1), it is impossible to set FCNnGMCLCTL.FCNnGMCLPWOM and FCNnGMCLCTL.EFSD.** **It is possible to set start of a software reset by FCNnGMCLCTL.FCNnGMCLSESR = 1 while FCNnGMCLCTL.FCNnGMCLPWOM bit is cleared to 0.** **2. When FCNnGMCLCTL.FCNnGMCLSORF is set to 1, the initialization of the message buffer RAM starts.** **3. When FCNnGMCLCTL.FCNnGMCLSORF already set to 1 is set to 1 again, software reset processing does not restart, but continues.** **4. After releasing hardware reset, FCNnGMCLCTL.FCNnGMCLSORF is automatically set to 1 and initialization of the message buffer RAM starts.** **5. Clearing FCNnGMCLCTL.FCNnGMCLPWOM (0) and setting FCNnGMCLCTL.FCNnGMCLSORF (1) cannot proceed at the same time.** **6. If a hardware reset occurs while FCNnGMCLCTL.FCNnGMCLSORF = 1, then software reset processing is stopped (aborted), and a hardware reset starts.**
1	FCNnGMCLESDE	Bit enabling forced shutdown 0: Forced shutdown is disabled while FCNnGMCLCTL.FCNnGMCLPWOM = 0. 1: Forced shutdown is enabled while FCNnGMCLCTL.FCNnGMCLPWOM = 0. **Caution: To request a forced shutdown, FCNnGMCLCTL.FCNnGMCLPWOM must be cleared to 0 immediately for access after FCNnGMCLCTL.FCNnGMCLESDE has been set to 1. If any access to another register (including reading the FCNnGMCLCTL register) is executed without clearing FCNnGMCLPWOM immediately after FCNnGMCLESDE has been set to 1, FCNnGMCLESDE is forcibly cleared to 0, and the forced shutdown request is disabled.**
0	FCNnGMCLPWOM	Global operation mode bit 0: The FCN module is disabled. 1: The FCN module is enabled. **Caution: FCNnGMCLCTL.FCNnGMCLPWOM can only be cleared in the initialization mode or immediately after FCNnGMCLCTL.FCNnGMCLESDE has been set (forced shutdown).**

(b) When FCNnGMCLCTL is written

15	14	13	12	11	10	9	8
0	0	0	FCNnGM CLSESR	0	0	FCNnGM CLSESD	FCNnGM CLSEOM

7	6	5	4	3	2	1	0
0	0	FCNnGM CLCLMB	0	0	0	0	FCNnGM CLCLOM

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>12</td><td>FCNnGMCLSESR</td><td>Software reset start
0: No change
1: Start software reset.</td></tr>
<tr><td>9</td><td>FCNnGMCLSESD</td><td>FCNnGMCLSESD bit setting
0: No change in the FCNnGMCLESDE bit.
1: The FCNnGMCLESDE bit set to 1.</td></tr>
<tr><td>8,
0</td><td>FCNnGMCLSEOM,
FCNnGMCLCLOM</td><td>FCNnGMCLPWOM bit setting
<table>
<tr><th>FCNnGMCLSEOM</th><th>FCNnGMCLCLOM</th><th>FCNnGMCLPWOM Bit Setting</th></tr>
<tr><td>0</td><td>1</td><td>FCNnGMCLCTL.FCNnGMCLPWOM bit cleared to 0.</td></tr>
<tr><td>1</td><td>0</td><td>FCNnGMCLCTL.FCNnGMCLPWOM bit set to 1.</td></tr>
<tr><td colspan="2">Other than above</td><td>No change in the FCNnGMCLCTL.FCNnGMCLPWOM bit.</td></tr>
</table>
Caution: The FCNnGMCLCTL.FCNnGMCLPWOM and FCNnGMCLCTL.FCNnGMCLESDE bits must be separately.</td></tr>
<tr><td>1</td><td>FCNnGMCLCLMB</td><td>FCNnGMCLCTL.FCNnGMCLECCF bit clear
0: No change in the FCNnGMCLCTL.FCNnGMCLECCF bit.
1: The FCNnGMCLCTL.FCNnGMCLECCF bit cleared to 0.</td></tr>
</table>

(2) FCNn Global Clock Selection Register (FCNnGMCSPRE)

This register is used to select the FCN module system clock.

- Access This register can be read or written in 8-bit units.
- Address <FCNn_base> + 0008H
- Initial Value 0FH. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	FCNnGMCSPRSC[3:0]			

Bit Position	Bit Name	Description
3 to 0	FCNnGMCSPRSC[3:0]	FCN module system clock (f_{CANMOD})

FCNnGMCSPRSC[3:0]	FCN Module System Clock (f_{CANMOD})
0000B	f_{CAN} / 1
0001B	f_{CAN} / 2
0010B	f_{CAN} / 3
0011B	f_{CAN} / 4
0100B	f_{CAN} / 5
0101B	f_{CAN} / 6
0110B	f_{CAN} / 7
0111B	f_{CAN} / 8
1000B	f_{CAN} / 9
1001B	f_{CAN} / 10
1010B	f_{CAN} / 11
1011B	f_{CAN} / 12
1100B	f_{CAN} / 13
1101B	f_{CAN} / 14
1110B	f_{CAN} / 15
1111B	f_{CAN} / 16 (default value)

Remark: f_{CAN} = clock supplied to FCN on system level (clock generation, distribution and selection).

(3) FCNn Global Automatic Block Transmission Control Register (FCNnGMABCTL)

This register is used to control the automatic block transmission (ABT) operation.

- Access This register can be read or written in 16-bit units.
- Address <FCNn_base> + 0 8018H
- Initial Value 0000H. The register is initialized by any reset.

(a) When FCNnGMABCTL is read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnGM ABCLRF	FCNnGM ABABTT

Bit Position	Bit Name	Description
1	FCNnGMABCLRF	Automatic block transmission engine clear status bit. 0: Clearing the automatic transmission engine is completed. 1: The automatic transmission engine is being cleared. **Remark: FCNnGMABCLRF must be set to 1 while FCNnGMABABTT is cleared to 0.** **Correct operation is not guaranteed if FCNnGMABCLRF is set to 1 while FCNnGMABABTT = 1.**
0	FCNnGMABABTT	Automatic block transmission status bit 0: Automatic block transmission is stopped. 1: Automatic block transmission is in progress.

(b) When FCNnGMABCTL is written

15	14	13	12	11	10	9	8
0	0	0	0	0	0	FCNnGM ABSEAC	FCNnGM ABSEAT

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnGM ABCLAT

Remark: When the automatic block transmission engine is cleared by setting FCNnGMABCTL.FCNnGMABSEAC to 1, FCNnGMABCLRF is automatically set, and cleared to 0 at the same time as requested processing for clearing is completed.

Cautions 1. Before changing the normal operation mode with ABT to the initialization mode, be sure to set the FCNnGMABCTL register to the default value (0000H) and confirm the FCNnGMABCTL register has been initialized to the default value (0000H).

2. Do not start automatic block transmission in the initialization mode. If automatic block transmission is started in the initialization mode, correct operation is not guaranteed after the CAN controller has entered the normal operation mode with ABT.

3. Do not start automatic block transmission while FCNnCMCLCTL.FCNnCMCLSSTS is set to 1 (transmission in progress).
Confirm directly that FCNnCMCLSSTS = 0 before starting automatic block transmission.

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>1</td><td>FCNnGMABSEAC</td><td>Automatic block transmission engine clear request bit
0: The automatic block transmission engine is in the idle state or under operation.
1: Request clearing of the automatic block transmission engine. After the automatic block transmission engine has been cleared, automatic block transmission is started from message buffer 0 by setting the FCNnGMABCTL.FCNnGMABABTT = 1.</td></tr>
<tr><td>8,
0</td><td>FCNnGMABSEAT,
FCNnGMABCLAT</td><td>Automatic block transmission start bit
<table>
<tr><th>FCNnGMABSEAT</th><th>FCNnGMABCLAT</th><th>Automatic Block Transmission Start Bit</th></tr>
<tr><td>0</td><td>1</td><td>Request automatic block transmission to be stopped.</td></tr>
<tr><td>1</td><td>0</td><td>Request automatic block transmission to be started.</td></tr>
<tr><td colspan="2">Other than above</td><td>The FCNnGMABCTL.FCNnGMABABTT bit is not changed.</td></tr>
</table></td></tr>
</table>

(4) FCNn Global Automatic Block Transmission Delay Register (FCNnGMADCTL)

This register is used to set the interval for transmitting data in the message buffer assigned to ABT in the normal operation mode with ABT.

- Access This register can be read or written in 8-bit units.
- Address <FCNn_base> + 0020H
- Initial Value 00H. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	FCNnGMADSSAD[3:0]			

Bit Position	Bit Name	Description
3 to 0	FCNnGMADSSAD[3:0]	Data frame interval during automatic block transmission (in units of DBT) Note

FCNnGMADSSAD[3:0]	Data Frame Interval during Automatic Block Transmission (in units of DBT) Note
0000B	0 DBT (default value)
0001B	2^5 DBT
0010B	2^6 DBT
0011B	2^7 DBT
0100B	2^8 DBT
0101B	2^9 DBT
0110B	2^{10} DBT
0111B	2^{11} DBT
1000B	2^{12} DBT
Other than above	Setting prohibited

Note: Unit: Data bit time (DBT)

Cautions 1. Do not change the setting of the FCNnGMADCTL register while FCNnGMABCTL.FCNnGMABCLRF = 1 (clearing of ABT is in progress).

2. The timing of the actual transmission of ABT messages to the CAN bus differs depending on the state of transmission from the other station or how a request for the transmission of messages other than ABT messages has been issued.

(5) FCNn Global Data Update Bit Monitor Register (FCNnDNBMRXk) (k = 0, 1)

These registers are used to read the data update bits of several message buffers at a time, globally.

- Access These registers can be read in 32-bit units.
- Address FCNnDNBMRX0: <FCNn_base> + 1 00C0H
  FCNnDNBMRX1: <FCNn_base> + 1 00D0H
- Initial Value 0000 0000H. This register is initialized by any reset.

31	30	29	28	27	26	25	24
FCNnDNBMSSDN[31:24]							

23	22	21	20	19	18	17	16
FCNnDNBMSSDN[23:16]							

15	14	13	12	11	10	9	8
FCNnDNBMSSDN[15:8]							

7	6	5	4	3	2	1	0
FCNnDNBMSSDN[7:0]							

Bit Position	Bit Name	Description
31 to 0	FCNnDNBMSSDN[31:0]	Message buffer data update bit 0: No remote or data frame has been stored into the message buffer. 1: A remote or data frame has been stored into the message buffer.

25.5.2 FCN Module Registers

(1) FCNn Module Mask Control Register (FCNnCMMKCTLaH, FCNnCMMKCTLaW)

These registers are used to increase the number of receivable messages which can be stored in the same message buffer by masking part of the message identifier (ID) to be compared and invalidating the ID of the masked part.

Two 16-bit registers FCNnCMMKCTLaH (a = 01 to 16) can also be accessed via a single 32-bit access to the registers FCNnCMMKCTLaW (a = 01, 03, 05, 07, 09, 11, 13, 15).

- Access The FCNnCMMKCTLaH registers can be read or written in 16-bit units.
  The FCNnCMMKCTLaW registers can be read or written in 32-bit units.
- Address FCNnCMMKCTL01H: <FCNn_base> + 0 8300H
  FCNnCMMKCTL02H: <FCNn_base> + 0 8308H
  FCNnCMMKCTL03H: <FCNn_base> + 0 8310H
  FCNnCMMKCTL04H: <FCNn_base> + 0 8318H
  FCNnCMMKCTL05H: <FCNn_base> + 0 8320H
  FCNnCMMKCTL06H: <FCNn_base> + 0 8328H
  FCNnCMMKCTL07H: <FCNn_base> + 0 8330H
  FCNnCMMKCTL08H: <FCNn_base> + 0 8338H
  FCNnCMMKCTL09H: <FCNn_base> + 0 8340H
  FCNnCMMKCTL10H: <FCNn_base> + 0 8348H
  FCNnCMMKCTL11H: <FCNn_base> + 0 8350H
  FCNnCMMKCTL12H: <FCNn_base> + 0 8358H
  FCNnCMMKCTL13H: <FCNn_base> + 0 8360H
  FCNnCMMKCTL14H: <FCNn_base> + 0 8368H
  FCNnCMMKCTL15H: <FCNn_base> + 0 8370H
  FCNnCMMKCTL16H: <FCNn_base> + 0 8378H
  FCNnCMMKCTL01W: <FCNn_base> + 1 0300H
  FCNnCMMKCTL03W: <FCNn_base> + 1 0310H
  FCNnCMMKCTL05W: <FCNn_base> + 1 0320H
  FCNnCMMKCTL07W: <FCNn_base> + 1 0330H
  FCNnCMMKCTL09W: <FCNn_base> + 1 0340H
  FCNnCMMKCTL11W: <FCNn_base> + 1 0350H
  FCNnCMMKCTL13W: <FCNn_base> + 1 0360H
  FCNnCMMKCTL15W: <FCNn_base> + 1 0370H
- Initial Value 0000H for FCNnCMMKCTLaH
  This register is initialized by any reset.
  0000 0000H for FCNnCMMKCTLaW
  This register is initialized by any reset.

(a) FCNnCMMKCTLaH (a = 01, 03, 05, 07, 09, 11, 13, 15)

15	14	13	12	11	10	9	8
FCNnCMMKSSID[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCNnCMMKSSID[7:0]							

(b) FCNnCMMKCTLaH (a = 02, 04, 06, 08, 10, 12, 14, 16)

15	14	13	12	11	10	9	8
0	0	0	FCNnCMMKSSID[28:24]				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCNnCMMKSSID[23:16]							

(c) FCNnCMMKCTLaW (a = 01, 03, 05, 07, 09, 11, 13, 15)

31	30	29	28	27	26	25	24
0	0	0	FCNnCMMKSSID[28:24]				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FCNnCMMKSSID[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FCNnCMMKSSID[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCNnCMMKSSID[7:0]							

Bit Position	Bit Name	Description
28 to 0	FCNnCMMKSSID[i[Note]]	Mask pattern setting of ID bit 0: The ID bit i of the message buffer m set by FCNnMmSSID[i] are compared with the ID bits of the received message frame. 1: The ID bit i of the message buffer m set by FCNnMmSSID[i] are not compared with the ID bits of the received message frame (they are masked).

Note: i = [28:0]

Remark: Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, FCNnCMMKSSID[17:0] are ignored. Therefore, only FCNnCMMKSSID[28:18] of the received ID are masked. The same mask can be used for both the standard and extended IDs.

(2) FCNn Module Control Register (FCNnCMCLCTL)

This register is used to control the operation mode of the FCN module.

- Access This register can be read or written in 16-bit units.
- Address <FCNn_base> + 0 8240H
- Initial Value 0000H. The register is initialized by any reset.

(a) When FCNnCMCLCTL is read

(1/4)

15	14	13	12	11	10	9	8
0	0	0	0	0	0	FCNnCM CLSSRS	FCNnCM CLSSTS

7	6	5	4	3	2	1	0
FCNnCM CLERCF	FCNnCM CLALBF	FCNnCM CLVALF	FCNnCM CLMDPF[1:0]		FCNnCM CLMDOF[2:0]		

Bit Position	Bit Name	Description
9	FCNnCMCLSSRS	Reception status bit 0: Reception is stopped. 1: Reception is in progress. **Remarks 1. FCNnCMCLSSRS is set to 1 under the following conditions (timing)** **• The SOF bit of a received frame is detected** **• On occurrence of arbitration loss during a transmission frame** **2. FCNnCMCLSSRS is cleared to 0 under the following conditions (timing)** **• When a recessive level is detected at the second bit of the interframe space** **• On transition to the initialization mode at the first bit of the interframe space**
8	FCNnCMCLSSTS	Transmission status bit 0: Transmission is stopped. 1: Transmission is in progress. **Remarks 1. FCNnCMCLSSTS is set to 1 under the following conditions (timing)** **• The SOF bit of a transmission frame is detected** **2. FCNnCMCLSSTS is cleared to 0 under the following conditions (timing)** **• During transition to bus-off state** **• On occurrence of arbitration loss in a transmission frame** **• On detection of recessive level at the second bit of the interframe space** **• On transition to the initialization mode at the first bit of the interframe space**

(2/4)

Bit Position	Bit Name	Description
7	FCNnCMCLERCF	Error counter clear bit 0: The FCNnCMERCNT and FCNnCMINSTR registers are not cleared in the initialization mode. 1: The FCNnCMERCNT and FCNnCMINSTR registers are cleared in the initialization mode. **Caution: FCNnCMCLERCF is used to clear the error counter FCNnCMERCNT and information register FCNnCMINSTR for re-initialization or forced recovery from the bus-off state. The error counter and the information register can be cleared under the following conditions (by setting FCNnCMCLERCF):** **- In the initialization mode during the bus-off period** **- In the initialization mode after the FCN module starts up (by changing FCNnGMCLPWOM from 0 to 1)** **- In the initialization mode entered after all the transmission requests have been cleared in accordance with the transmission abort processing shown in Figure 25.24, Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed) in an operation mode. (In normal operation mode with ABT, clear all the transmission requests in accordance with the transmission abort processing shown in Figure 25.25, Transmission Abort Processing (in Normal Operation Mode with ABT) – Repeat Option for Aborted Message.)** **Remarks 1. When the FCNnCMERCNT and FCNnCMINSTR registers have been cleared, FCNnCMCLERCF is also cleared to 0 automatically.** **2. FCNnCMCLERCF can be set to 1 at the same time as a request to change the initialization mode to an operation mode is issued.** **3. FCNnCMCLERCF is read-only in the FCN sleep mode or FCN stop mode.** **4. The error counter can also be cleared by a normal shutdown or forced shutdown of the CAN controller.**
6	FCNnCMCLALBF	Bit to set operation in case of arbitration loss 0: Re-transmission is not executed in case of an arbitration loss in the single-shot mode. 1: Re-transmission is executed in case of an arbitration loss in the single-shot mode. **Remark: FCNnCMCLALBF is valid only in the single-shot mode.**

(3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>5</td><td>FCNnCMCLVALF</td><td>Valid receive message frame detection bit
0: A valid message frame has not been received since FCNnCMCLVALF was last cleared to 0.
1: A valid message frame has been received since FCNnCMCLVALF was last cleared to 0.

Remarks 1. Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame/remote frame) or transmit message buffer (remote frame).
2. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal mode and the other in the receive-only mode, FCNnCMCLVALF is not set to 1 before the transmitting node enters the error passive state, because in receive-only mode no acknowledge is generated.
3. To clear FCNnCMCLVALF, set FCNnCMCLCLVL to 1 first and confirm that FCNnCMCLVALF is cleared. If it is not cleared, perform clearing processing again.</td></tr>
<tr><td>4, 3</td><td>FCNnCMCLMDPF[1:0]</td><td>Power save mode
<table><tr><th>FCNnCMCLMDPF[1:0]</th><th>Power Save Mode</th></tr><tr><td>00B</td><td>No power save mode is selected.</td></tr><tr><td>01B</td><td>FCN sleep mode</td></tr><tr><td>10B</td><td>Setting prohibited</td></tr><tr><td>11B</td><td>FCN stop mode</td></tr></table>
Cautions 1. Transition to and from the FCN stop mode must be made via FCN sleep mode. A request for direct transition to and from the FCN stop mode is ignored.
2. After release from power save mode, the FCNnGMCLSSMO flag of FCNnGMCLCTL must be checked prior to access to the message buffers again.
3. FCN sleep mode requests are kept pending, until they are cancelled by software or the transition to the appropriate bus state (bus idle). Software can check the actual state by reading FCNnCMCLMDPF[1:0].
4. Power save mode cannot be set in combination with the change of operation mode. Be sure to perform these operations in different steps.

Remark: When the system transitions from initialization mode to any communication mode, the FCN module participates in communications after confirming the CAN bus idle period. Although it is possible to transition to sleep mode before confirming the idle period, the wakeup condition is always a change from recessive level to dominant level.</td></tr>
</table>

(4/4)

Bit Position	Bit Name	Description
2 to 0	FCNnCMCLMDOF[2:0]	Operation mode

FCNnCMCLMDOF[2:0]	Operation Mode
000B	No operation mode is selected (FCN module is in the initialization mode).
001B	Normal operation mode
010B	Normal operation mode with automatic block transmission (normal operation mode with ABT)
011B	Receive-only mode
100B	Single-shot mode
101B	Self-test mode
Other than above	Setting prohibited

Cautions 1. Transition to initialization mode or power save mode may take time. Be sure to verify the success of mode change by reading the values before next processing.

2. If initialization mode is set while receiving data in operation mode, data in the message buffer that sets the FCNnMmDTNF flag might be received last.
However, the receive history list is cleared upon transition to operation mode. It is therefore necessary to confirm that initialization mode was set by reading the operation mode. Before restarting operation mode, make sure to clear all FCNnMmDTNF flags in all valid reception message buffers.

Remark: FCNnCM.FCNnCMCLMDOF[2:0] are read-only in the FCN sleep mode or FCN stop mode.

(b) When FCNnCMCLCTL is written

(1/2)

15	14	13	12	11	10	9	8
FCNnCM CLSERC	FCNnCM CLSEAL	0	FCNnCM CLSEPS[1:0]		FCNnCM CLSEOP[2:0]		

7	6	5	4	3	2	1	0
0	FCNnCM CLCLAL	FCNnCM CLCLVL	FCNnCM CLCLPS[1:0]		FCNnCM CLCLOP[2:0]		

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>15</td><td>FCNnCMCLSERC</td><td>Setting of FCNnCMCLERCF bit
0: FCNnCMCLERCF is not changed.
1: FCNnCMCLERCF is set to 1.</td></tr>
<tr><td>14,
6</td><td>FCNnCMCLSEAL,
FCNnCMCLCLAL</td><td><table>
<tr><th>FCNnCMCLSEAL</th><th>FCNnCMCLCLAL</th><th>Setting of FCNnCMCLALBF Bit</th></tr>
<tr><td>0</td><td>1</td><td>FCNnCMCLALBF is cleared to 0.</td></tr>
<tr><td>1</td><td>0</td><td>FCNnCMCLALBF is set to 1.</td></tr>
<tr><td colspan="2">Other than above</td><td>FCNnCMCLALBF is not changed.</td></tr>
</table></td></tr>
<tr><td>5</td><td>FCNnCMCLCLVL</td><td>Setting of FCNnCMCLVALF bit
0: FCNnCMCLVALF is not changed.
1: FCNnCMCLVALF is cleared to 0.</td></tr>
<tr><td>11,
3</td><td>FCNnCMCLSEPS0,
FCNnCMCLCLPS0</td><td><table>
<tr><th>FCNnCMCLSEPS0</th><th>FCNnCMCLCLPS0</th><th>Setting of FCNnCMCLMDPF0 Bit</th></tr>
<tr><td>0</td><td>1</td><td>FCNnCMCLMDPF0 is cleared to 0.</td></tr>
<tr><td>1</td><td>0</td><td>FCNnCMCLMDPF0 is set to 1.</td></tr>
<tr><td colspan="2">Other than above</td><td>FCNnCMCLMDPF0 is not changed.</td></tr>
</table></td></tr>
<tr><td>12,
4</td><td>FCNnCMCLSEPS1,
FCNnCMCLCLPS1</td><td><table>
<tr><th>FCNnCMCLSEPS1</th><th>FCNnCMCLCLPS1</th><th>Setting of FCNnCMCLMDPF1 Bit</th></tr>
<tr><td>0</td><td>1</td><td>FCNnCMCLMDPF1 is cleared to 0.</td></tr>
<tr><td>1</td><td>0</td><td>FCNnCMCLMDPF1 is set to 1</td></tr>
<tr><td colspan="2">Other than above</td><td>FCNnCMCLMDPF1 is not changed.</td></tr>
</table></td></tr>
<tr><td>8,
0</td><td>FCNnCMCLSEOP0,
FCNnCMCLCLOP0</td><td><table>
<tr><th>FCNnCMCLSEOP0</th><th>FCNnCMCLCLOP0</th><th>Setting of FCNnCMCLMDOF0 Bit</th></tr>
<tr><td>0</td><td>1</td><td>FCNnCMCLMDOF0 is cleared to 0.</td></tr>
<tr><td>1</td><td>0</td><td>FCNnCMCLMDOF0 is set to 1</td></tr>
<tr><td colspan="2">Other than above</td><td>FCNnCMCLMDOF0 is not changed.</td></tr>
</table></td></tr>
</table>

(2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>9,
1</td><td>FCNnCMCLSEOP1,
FCNnCMCLCLOP1</td><td><table>
<tr><th>FCNnCMCLSEOP1</th><th>FCNnCMCLCLOP1</th><th>Setting of FCNnCMCLMDOF1 Bit</th></tr>
<tr><td>0</td><td>1</td><td>FCNnCMCLMDOF1 is cleared to 0.</td></tr>
<tr><td>1</td><td>0</td><td>FCNnCMCLMDOF1 is set to 1</td></tr>
<tr><td colspan="2">Other than above</td><td>FCNnCMCLMDOF1 is not changed.</td></tr>
</table></td></tr>
<tr><td>10,
2</td><td>FCNnCMCLSEOP2,
FCNnCMCLCLOP2</td><td><table>
<tr><th>FCNnCMCLSEOP2</th><th>FCNnCMCLCLOP2</th><th>Setting of FCNnCMCLMDOF2 Bit</th></tr>
<tr><td>0</td><td>1</td><td>FCNnCMCLMDOF2 is cleared to 0.</td></tr>
<tr><td>1</td><td>0</td><td>FCNnCMCLMDOF2 is set to 1</td></tr>
<tr><td colspan="2">Other than above</td><td>FCNnCMCLMDOF2 is not changed.</td></tr>
</table></td></tr>
</table>

(3) FCNn Module Last Error Information Register (FCNnCMLCSTR)

This register provides the error information of the CAN protocol.

- Access This register can be read or written in 8-bit units.
- Address <FCNn_base> + 0 0248H
- Initial Value 00H. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCMLCSSLC[2:0]		

Remarks 1. The settings of the FCNnCMLCSTR register are not cleared even if the FCN module enters the initialization mode from the operation mode.

2. If an attempt is made to write a value other than 00H to the FCNnCMLCSTR register by software, the access is ignored.

Bit Position	Bit Name	Description
2 to 0	FCNnCMLCSSLC[2:0]	

FCNnCMLCSSLC[2:0]	Last CAN Protocol Error Information
000B	No error
001B	Stuff error
010B	Form error
011B	ACK error
100B	Bit error. (The FCN module tried to transmit a recessive level bit as part of a transmit message (except the arbitration field), but the value on the CAN bus is a dominant-level bit.)
101B	Bit error. (The FCN module tried to transmit a dominant level bit as part of a transmit message, ACK bit, error frame, or overload frame, but the value on the CAN bus is a recessive-level bit.)
110B	CRC error
111B	Undefined

(4) FCNn Module Information Register (FCNnCMINSTR)

This register indicates the state of the FCN module.

- Access This register is read-only in 8-bit units.
- Address <FCNn_base> + 0 024CH
- Initial Value 00H. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	FCNnCM INBOFF	FCNnCM INSSTE[1:0]		FCNnCM INSSRE[1:0]	

Bit Position	Bit Name	Description
4	FCNnCMINBOFF	(see table below)
3, 2	FCNnCMINSSTE[1:0]	(see table below)
1, 0	FCNnCMINSSRE[1:0]	(see table below)

FCNnCMINBOFF	Bus-Off State Bit
0	Not bus-off state (transmit error counter ≤ 255). (The value of the transmit counter is less than 256.)
1	Bus-off state (transmit error counter > 255). (The value of the transmit counter is 256 or above.)

FCNnCMINSSTE[1:0]	Transmission Error Counter Status Bit
00B	The value of the transmission error counter is less than that of the warning level (< 96).
01B	The value of the transmission error counter is in the range of the warning level (96 to 127).
10B	Undefined
11B	The value of the transmission error counter is in the range of the error passive or bus-off status (≥ 128).

FCNnCMINSSRE[1:0]	Reception Error Counter Status Bit
00B	The value of the reception error counter is less than that of the warning level (< 96).
01B	The value of the reception error counter is in the range of the warning level (96 to 127).
10B	Undefined
11B	The value of the reception error counter is in the error passive range (≥ 128).

(5) FCNn Module Error Counter Register (FCNnCMERCNT)

This register indicates the value of the transmission/reception error counter.

- Access This register is read-only in 16-bit units.
- Address <FCNn_base> + 0 8250H
- Initial Value 0000H. The register is initialized by any reset.

15	14	13	12	11	10	9	8
FCNnCM ERRPSF	FCNnCMERRECF[6:0]						

7	6	5	4	3	2	1	0
FCNnCMERTECF[7:0]							

Bit Position	Bit Name	Description
15	FCNnCMERRPSF	(see below)
14 to 8	FCNnCMERRECF[6:0]	(see below)
7 to 0	FCNnCMERTECF[7:0]	(see below)

FCNnCMERRPSF	Reception Error Passive Status Bit
0	The reception error counter is not in the error passive range (< 128)
1	The reception error counter is in the error passive range (≥ 128)

FCNnCMERRECF[6:0]	Reception Error Counter Bit
0–127	The number of errors counted. These bits reflect the state of the reception error counter. The number of errors counted is defined by the CAN protocol.

Remark: FCNnCMERRECF[6:0] are invalid in the reception error passive state (FCNnCMINSTR.FCNnCMINSSRE[1:0] = 11B).

FCNnCMERTECF[7:0]	Transmission Error Counter Bit
0–255	Number of transmission errors counted. These bits reflect the state of the transmission error counter. The number of errors counted is defined by the CAN protocol.

Remark: FCNnCMERTECF[7:0] are invalid in the bus-off state (FCNnCMINSTR.FCNnCMINBOFF = 1).

(6) FCNn Module Interrupt Enable Register (FCNnCMIECTL)

This register is used to enable or disable interrupts from the FCN module.

- Access This register can be read or written in 16-bit units.
- Address <FCNn_base> + 0 8258H
- Initial Value 0000H. The register is initialized by any reset.

(a) When FCNnCMIECTL is read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	FCNnCMIEINTF[6:0]						

Bit Position	Bit Name	Description
6 to 0	FCNnCMIEINTF[6:0]	See table below

FCNnCMIEINTF[6:0]	FCN Module Interrupt Enable Bit
0	Output of the interrupt corresponding to interrupt status register FCNnCMISCTL is disabled.
1	Output of the interrupt corresponding to interrupt status register FCNnCMISCTL is enabled.

(b) When FCNnCMIECTL is written

15	14	13	12	11	10	9	8
0	FCNnCMIESEIE[6:0]						

7	6	5	4	3	2	1	0
0	FCNnCMIECLIE[6:0]						

Bit Position	Bit Name	Description
14 to 8, 6 to 0	FCNnCMIESEIE[6:0], FCNnCMIECLIE[6:0]	See table below

FCNnCMIESEIE[6:0]	FCNnCMIECLIE[6:0]	Setting of FCNnCMIEINTF[6:0] Bit
0	1	FCNnCMIEINTF[6:0] bit is cleared to 0.
1	0	FCNnCMIEINTF[6:0] bit is set to 1.
Other than above		FCNnCMIEINTF[6:0] bit is not to change.

(7) FCNn Module Interrupt Status Register (FCNnCMISCTL)

This register indicates the state of the interrupt from the FCN module.

- Access This register can be read or written in 16-bit units.
- Address <FCNn_base> + 0 8260H
- Initial Value 0000H. The register is initialized by any reset.

(a) When FCNnCMISCTL is read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	FCNnCMISITSF[6:0]						

Bit Position	Bit Name	Description
6 to 0	FCNnCMISITSF[6:0]	(see tables below)

FCNnCMISITSF[6:0]	FCN Interrupt Status Bit
0	No related interrupt source event is pending
1	A related interrupt source event is pending

Interrupt Status Bit	Related Interrupt Source Event
FCNnCMISITSF6	FCN module transmission abort interrupt status bit
FCNnCMISITSF5	Wakeup interrupt from FCN sleep mode[Note]
FCNnCMISITSF4	Arbitration loss interrupt
FCNnCMISITSF3	CAN protocol error interrupt
FCNnCMISITSF2	CAN error status interrupt
FCNnCMISITSF1	Interrupt on completion of reception of valid message frame to message buffer m
FCNnCMISITSF0	Interrupt on normal completion of transmission of message frame from message buffer m

Note: FCNnCMISITSF5 is set only when the FCN module is woken up from the FCN sleep mode by operation on the CAN bus. It is not set when the FCN module is released from FCN sleep mode by software.

(b) When FCNnCMISCTL is written

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	FCNnCMISCLTS[6:0]						

Bit Position	Bit Name	Description
6 to 0	FCNnCMISCLTS[6:0]	(see table below)

FCNnCMISCLTS[6:0]	Clearing of FCNnCMISITSF[6:0]
0	FCNnCMISITSF[6:0] bits are not changed.
1	FCNnCMISITSF[6:0] bits are cleared to 0.

Caution: Clear the status bit of this register by software when interrupt processing requires confirmation of each status, because these bits are not cleared automatically.

(8) FCNn Module Bit Rate Prescaler Register (FCNnCMBRPRS)

This register is used to select the CAN protocol layer basic system clock (f_{TQ}). The communication baud rate is set in accord with the setting of the FCNnCMBTCTL register.

- Access This register can be read or written in 8-bit units.
- Address <FCNn_base> + 0 0268H
- Initial Value FFH. The register is initialized by any reset.

7	6	5	4	3	2	1	0
FCNnCMBRPRS[7:0]							

Bit Position	Bit Name	Description
7 to 0	FCNnCMBRPRS[7:0]	(see below)

FCNnCMBRPRS[7:0]	CAN Protocol Layer Basic System Clock (f_{TQ})
0x00	f_{CANMOD} / 1
0x01	f_{CANMOD} / 2
n	f_{CANMOD} / (n+1)
:	:
0xff	f_{CANMOD} / 256(default value)

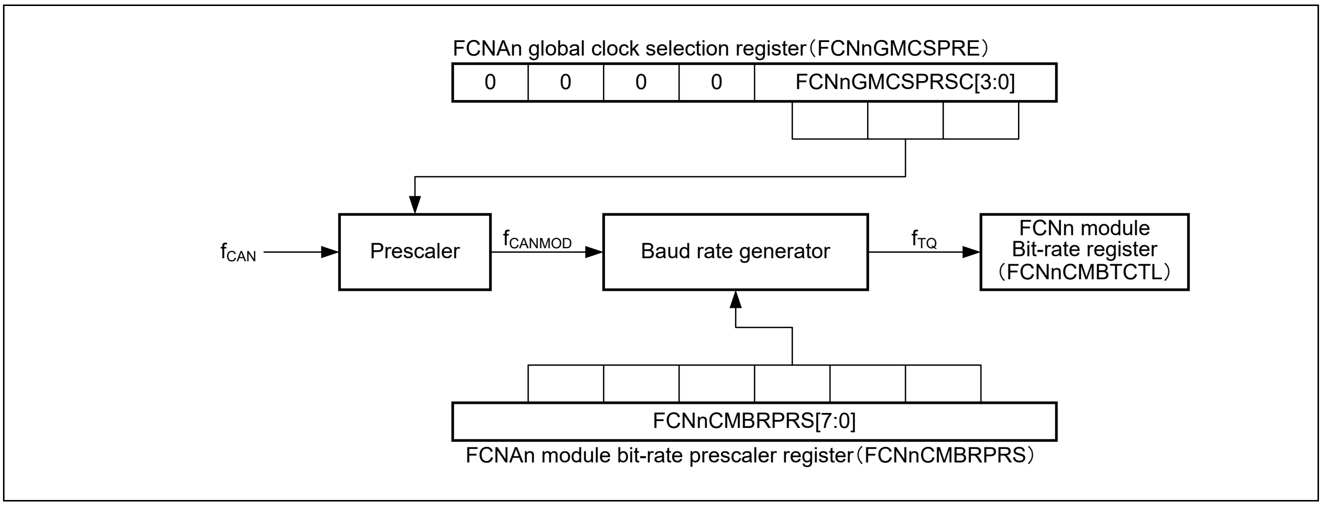


Figure 25.2 FCN Module Clock

Remark: f_{CAN} **Clock supplied to FCN**
f_{CANMOD} **FCN module system clock**
f_{TQ} **CAN protocol layer basic system clock**

Caution: FCNnCMBRPRS can be write-accessed only in the initialization mode.

(9) FCNn Module Bit Rate Register (FCNnCMBTCTL)

This register is used to control the data bit time of the communication baud rate.

- Access: This register can be read or written in 16-bit units.
- Address: <FCNn_base> + 0 8270H
- Initial Value: 370FH. The register is initialized by any reset.

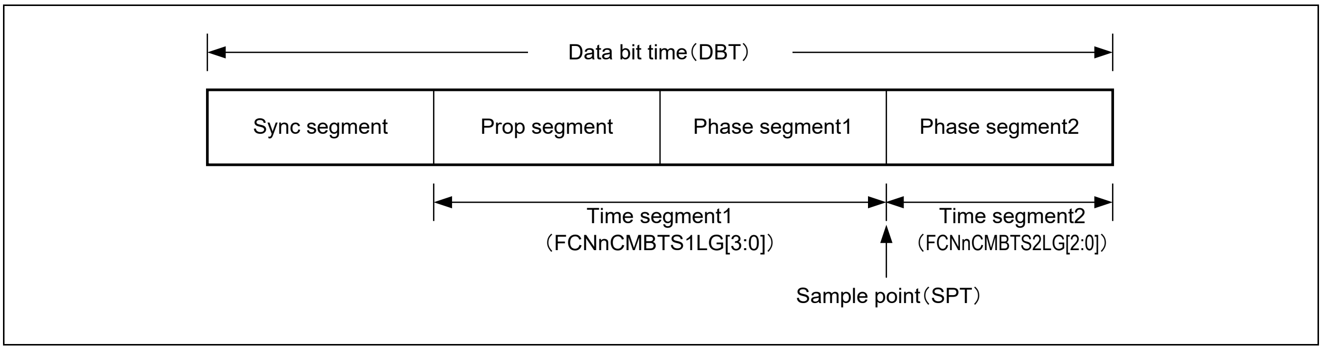


Figure 25.3 Data Bit Time

(1/2)

15	14	13	12	11	10	9	8
0	0	FCNnCM BTJWLG[1:0]		0	FCNnCM BTS2LG[2:0]		

7	6	5	4	3	2	1	0
0	0	0	0	FCNnCMBTS1LG[3:0]			

Bit Position	Bit Name	Description
13, 12	FCNnCMBTJWLG[1:0]	(see table below)

FCNnCMBTJWLG[1:0]	Length of Synchronization Jump Width
00B	$1T_Q$
01B	$2T_Q$
10B	$3T_Q$
11B	$4T_Q$ (Initial value)

Remark: $T_Q = 1 / f_{TQ}$ (f_{TQ}: CAN protocol layer basic system clock)

(2/2)

Bit Position	Bit Name	Description
10 to 8	FCNnCMBTS2LG[2:0]	FCNnCMBTS2LG[2:0] / Length of Time Segment 2: 000B: $1T_Q$ 001B: $2T_Q$ 010B: $3T_Q$ 011B: $4T_Q$ 100B: $5T_Q$ 101B: $6T_Q$ 110B: $7T_Q$ 111B: $8T_Q$ (Initial value)
3 to 0	FCNnCMBTS1LG[3:0]	FCNnCMBTS1LG[3:0] / Length of Time Segment 1: 0000B: Setting prohibited 0001B: Setting prohibited 0010B: Setting prohibited 0011B: $4T_Q$ 0100B: $5T_Q$ 0101B: $6T_Q$ 0110B: $7T_Q$ 0111B: $8T_Q$ 1000B: $9T_Q$ 1001B: $10T_Q$ 1010B: $11T_Q$ 1011B: $12T_Q$ 1100B: $13T_Q$ 1101B: $14T_Q$ 1110B: $15T_Q$ 1111B: $16T_Q$ (Initial value)

Remark: $T_Q = 1 / f_{TQ}$ (f_{TQ}: CAN protocol layer basic system clock)

(10) FCNn Module Last In-Pointer Register (FCNnCMLISTR)

This register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

- Access This register is read-only in 8-bit units.
- Address <FCNn_base> + 0 0278H
- Initial Value Undefined.

7	6	5	4	3	2	1	0
FCNnCMLISSLR[7:0]							

Bit Position	Bit Name	Description
7 to 0	FCNnCMLISSLR[7:0]	FCNnCMLISSLR[7:0]: Last In-Pointer Register of Receive History List 0–63: Reading the FCNnCMLISTR register obtains the number of the message buffer storing the last data frame or remote frame to be received. **Remark: The read value of FCNnCMLISTR is undefined if a data frame or a remote frame has never been received and stored in the message buffer. Therefore, if FCNnCMRGRX.FCNnCMRGSSPM is set to 1 after the FCN module entered any operation mode from the initialization mode, the read value of FCNnCMLISTR is undefined.**

(11) FCNn Module Receive History List Register (FCNnCMRGRX)

This register is used to read the receive history list (RHL).

- Access This register can be read or written in 16-bit units.
- Address <FCNn_base> + 0 8280H
- Initial Value xx02H. The register is initialized by any reset.

(a) When FCNnCMRGRX is read

(1/2)

15	14	13	12	11	10	9	8
FCNnCMRGSSPT[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnCM RGSSPM	FCNnCM RGRVFF

Bit Position	Bit Name	Description
15 to 8	FCNnCMRGSSPT[7:0]	FCNnCMRGSSPT[7:0]: Receive History List Read Pointer 0–63: When FCNnCMRGRX is read, the contents of the element indexed by the read pointer (FCNnCMRGRX.FCNnCMRGSSPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame has been stored.
1	FCNnCMRGSSPM[Note]	FCNnCMRGSSPM[Note]: Receive History List Pointer Match 0: The receive history list has at least one message buffer number that has not been read. 1: The receive history list has no message buffer numbers that have not been read. **Note: The read value of FCNnCMTGSSPT[7:0] is invalid while FCNnCMTGSSPM = 1.**

(2/2)

Bit Position	Bit Name	Description
0	FCNnCMRGRVFF[Note1]	(see table below)

FCNnCMRGRVFF[Note1]	Receive History List Overflow Bit[Note2]
0	All the message buffer numbers that have not been read are stored. All the numbers of the message buffers in which a new data frame or remote frame has been received and stored are recorded in the receive history list (the receive history list has a vacant element).
1	At least **(i)** entries have been stored since the host processor has serviced the RHL last time (i.e. read FCNnCMRGRX). The first **(i-1)** entries are sequentially stored while the last entry can have been overwritten whenever newly received message is stored, because all buffer numbers are stored at position **(i)**, when FCNnCMRGRVFF is set. Thus the sequence of receptions cannot be recovered completely now.

Notes 1. If FCNnCMRGRVFF is set, FCNnCMRGSSPM is not cleared even when messages are saved, but FCNnCMRGSSPM is still set, if all entries of FCNnCMRGRX are read by software.

2. i = 47

(b) When FCNnCMRGRX is written

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnCMRGCLRV

Bit Position	Bit Name	Description
0	FCNnCMRGCLRV	(see table below)

FCNnCMRGCLRV	Clearing of FCNnCMRGRVFF Bit
0	The FCNnCMRGRVFF bit is not changed
1	The FCNnCMRGRVFF bit is cleared to 0.

(12) FCNn Module Last Out-Pointer Register (FCNnCMLOSTR)

This register indicates the number of the message buffer, from which a data frame or a remote frame was most recently transmitted.

- Access This register is read-only in 8-bit units.
- Address <FCNn_base> + 0 0288H
- Initial Value Undefined

7	6	5	4	3	2	1	0
FCNnCMLOSSLT[7:0]							

Bit Position	Bit Name	Description
7 to 0	FCNnCMLOSSLT[7:0]	FCNnCMLOSSLT[7:0]: Last Out-Pointer of Transmit History List 0–63: When the FCNnCMLOSTR register is read, the number of the message buffer from which a data frame or a remote frame was most recently transmitted.

Caution: The value read from the FCNnCMLOSTR register is undefined if no data frame or remote frame has been transmitted from the message buffer.

(13) FCNn Module Transmit History List Register (FCNnCMTGTX)

This register is used to read the transmit history list (THL).

- Access This register can be read or written in 16-bit units.
- Address <FCNn_base> + 0 8290H
- Initial Value xx02H. The register is initialized by any reset.

(a) When FCNnCMTGTX is read

(1/2)

15	14	13	12	11	10	9	8
FCNnCMTGSSPT[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnCM TGSSPM	FCNnCM TGTVFF

Bit Position	Bit Name	Description
15 to 8	FCNnCMTGSSPT[7:0]	**FCNnCMTGSSPT[7:0]**: Transmit History List Read Pointer 0–63: When the FCNnCMTGTX register is read, the contents of the element indexed by the read pointer (FCNnCMTGSSPT[7:0]) of the transmit history list are read. These contents indicate the number of the message buffer from which a data frame or a remote frame was most recently transmitted.
1	FCNnCMTGSSPM[Note]	**FCNnCMTGSSPM[Note]**: Transmit History Pointer Match 0: The transmit history list has at least one message buffer number that has not been read. 1: The transmit history list has no message buffer numbers that have not been read. **Note. The read value of FCNnCMTGSSPT[7:0] is invalid when FCNnCMTGSSPM = 1.**

(2/2)

Bit Position	Bit Name	Description
0	FCNnCMTGTVFF[Note1]	see table below

FCNnCMTGTVFF[Note1]	Transmit History List Overflow Bit[Note2]
0	All the message buffer numbers that have not been read are stored. All the numbers of the message buffers from which a new data frame or remote frame has been transmitted are recorded in the transmit history list (the transmit history list has a vacant element).
1	At least **(i)** entries have been stored since the host processor has serviced the THL last time (i.e. read FCNnCMTGTX). The first **(i-1)** entries are sequentially stored while the last entry can have been overwritten whenever newly received message is stored, because all buffer numbers are stored at position **(i)**, when FCNnCMTGTVFF is set. Thus the sequence of receptions cannot be recovered completely now.

Notes 1. If FCNnCMTGTVFF is set, FCNnCMTGSSPM is not cleared in response to transmission of messages, but FCNnCMTGSSPM is still set, if all entries of FCNnCMTGTX are read by software.

2. i = 15

Remark: Transmission from the following message buffers is not recorded in the transmit history list in the normal operation mode with ABT.

- **0–16**

(b) When FCNnCMTGTX is written

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnCMTGCLTV

Bit Position	Bit Name	Description
0	FCNnCMTGCLTV	see table below

FCNnCMTGCLTV	Setting of FCNnCMTGTVFF Bit
0	FCNnCMTGTVFF bit is not changed.
1	FCNnCMTGTVFF bit is cleared to 0.

(14) FCNn Module Timestamp Register (FCNnCMTSCTL)

This register is used to control timestamping.

- Access This register can be read or written in 16-bit units.
- Address <FCNn_base> + 0 8298H
- Initial Value 0000H. The register is initialized by any reset.

(a) When FCNnCMTSCTL is read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCM TSLOKE	FCNnCM TSSELE	FCNnCM TSTSGE

Remark: Timestamp locking must not be used when the FCN module is in the normal operation mode with ABT.

Bit Position	Bit Name	Description
2	FCNnCMTSLOKE	(see table below)
1	FCNnCMTSSELE	(see table below)
0	FCNnCMTSTSGE	(see table below)

FCNnCMTSLOKE	Timestamp Locking Enable Bit
0	Timestamp locking is stopped. The TSOUT signal is toggled each time the selected timestamp capture event occurs.
1	Timestamp locking is enabled. The TSOUT signal is toggled each time the selected time stamp capture event occurs. However, the TSOUT output signal is locked when message buffer 0 has received a data frame correctly.Note

Note: FCNnCMTSTSGE is automatically cleared to 0.

FCNnCMTSSELE	Timestamp Capture Event Selection Bit
0	The timestamp capture event is SOF.
1	The timestamp capture event is the last bit of EOF.

FCNnCMTSTSGE	TSOUT Operation Setting Bit
0	TSOUT toggle operation is disabled.
1	TSOUT toggle operation is enabled.

(b) When FCNnCMTSCTL is written

15	14	13	12	11	10	9	8
0	0	0	0	0	FCNnCM TSSELK	FCNnCM TSSESL	FCNnCM TSSETS

7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCM TSCLLK	FCNnCM TSCLSL	FCNnCM TSCLTS

Bit Position	Bit Name	Description
10, 2	FCNnCMTSSELK, FCNnCMTSCLLK	(see table below)
9, 1	FCNnCMTSSESL, FCNnCMTSCLSL	(see table below)
8, 0	FCNnCMTSSETS, FCNnCMTSCLTS	(see table below)

FCNnCMTSSELK	FCNnCMTSCLLK	Setting of FCNnCMTSLOKE Bit
0	1	FCNnCMTSLOKE is cleared to 0.
1	0	FCNnCMTSLOKE is set to 1.
Other than above		FCNnCMTSLOKE is not changed.

FCNnCMTSSESL	FCNnCMTSCLSL	Setting of FCNnCMTSSELE Bit
0	1	FCNnCMTSSELE is cleared to 0.
1	0	FCNnCMTSSELE is set to 1
Other than above		FCNnCMTSSELE is not changed.

FCNnCMTSSETS	FCNnCMTSCLTS	Setting of FCNnCMTSTSGE Bit
0	1	FCNnCMTSTSGE is cleared to 0.
1	0	FCNnCMTSTSGE is set to 1
Other than above		FCNnCMTSTSGE is not changed.

25.5.3 FCN Message Buffer Registers

(1) FCNn Message Data Byte Registers (FCNnMmDATxB/H/W)

These registers are used to store the data of transmit/receive messages.

- Access The FCNnMmDATxW registers can be read or written in 32-bit units.
  The FCNnMmDATxH registers can be read or written in 16-bit units.
  The FCNnMmDATxB registers can be read or written in 8-bit units.
- Address FCNnMmDAT0B: <FCNn_base> + 0 1000H + m × 40H
  FCNnMmDAT1B: <FCNn_base> + 0 1004H + m × 40H
  FCNnMmDAT2B: <FCNn_base> + 0 1008H + m × 40H
  FCNnMmDAT3B: <FCNn_base> + 0 100CH + m × 40H
  FCNnMmDAT4B: <FCNn_base> + 0 1010H + m × 40H
  FCNnMmDAT5B: <FCNn_base> + 0 1014H + m × 40H
  FCNnMmDAT6B: <FCNn_base> + 0 1018H + m × 40H
  FCNnMmDAT7B: <FCNn_base> + 0 101CH + m × 40H
  FCNnMmDAT0H: <FCNn_base> + 0 9000H + m × 40H
  FCNnMmDAT2H: <FCNn_base> + 0 9008H + m × 40H
  FCNnMmDAT4H: <FCNn_base> + 0 9010H + m × 40H
  FCNnMmDAT6H: <FCNn_base> + 0 9018H + m × 40H
  FCNnMmDAT0W: <FCNn_base> + 1 1000H + m × 40H
  FCNnMmDAT4W: <FCNn_base> + 1 1010H + m × 40H
- Initial Value 00000000H for FCNnMmDATxW.
  This register is initialized by any reset.
  0000H for FCNnMmDATxH.
  This register is initialized by any reset.
  00H for FCNnMmDATxB.
  This register is initialized by any reset.

(a) FCNnCMmDATxB (x = 0 to 7)

7	6	5	4	3	2	1	0
FCNnMmSSD0, FCNnMmSSD1, FCNnMmSSD2, FCNnMmSSD3, FCNnMmSSD4, FCNnMmSSD5, FCNnMmSSD6, FCNnMmSSD7							

(b) FCNnCMmDATxH (x = 0, 2, 4, 6)

15	14	13	12	11	10	9	8
FCNnMmSSD0, FCNnMmSSD2, FCNnMmSSD4, FCNnMmSSD6							
7	6	5	4	3	2	1	0
FCNnMmSSD1, FCNnMmSSD3, FCNnMmSSD5, FCNnMmSSD7							

(c) FCNnCMmDATxW (x = 0, 4)

31	30	29	28	27	26	25	24
FCNnMmSSD0, FCNnMmSSD4							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FCNnMmSSD1, FCNnMmSSD5							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FCNnMmSSD2, FCNnMmSSD6							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCNnMmSSD3, FCNnMmSSD7							

(2) FCNn Message Data Length Register m (FCNnMmDTLGB)

This register is used to set the number of bytes of the data field of a message buffer (DLC).

- Access This register can be read or written in 8-bit units.
- Address <FCNn_base> + 0 1020H + m × 40H
- Initial Value 00H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	FCNnMmDTLG[3:0]			

Bit Position	Bit Name	Description
3 to 0	FCNnMmDTLG[3:0]	(see table below)

FCNnMmDTLG[3:0]	Data Length of Transmit/Receive Message
0000B	0 bytes
0001B	1 byte
0010B	2 bytes
0011B	3 bytes
0100B	4 bytes
0101B	5 bytes
0110B	6 bytes
0111B	7 bytes
1000B	8 bytes
1001B 1010B 1011B 1100B 1101B 1110B 1111B	Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the FCNnMmDTLG[3:0] value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.) Note

Note: The data and DLC value actually transmitted to CAN bus are as follows.

Type of Transmit Frame	Length of Transmit Data	DLC Transmitted
Data frame	Number of bits specified by FCNnMmDTLG[3:0] (However, 8 bytes if value ≥ 8)	Setting of FCNnMmDTLGB.FCNnMmDTLG [3:0] bits
Remote frame	0 bytes	

Cautions 1. Be sure to set bits 7 to 4 to 0000B.

2. Received data is stored in FCNnMmDATxB registers, the number of which is the same as the number of bytes (however, the upper limit is 8) corresponding to DLC of the received frame. The FCNnMmDATxB register in which no data is stored is undefined.

3. On reception, FCNnMmDTLGB is updated according to the received frame.

(3) FCNn Message Configuration Register m (FCNnMmSTRB)

This register is used to specify the type of message buffer and to set a mask.

- Access This register can be read or written in 8-bit units.
- Address <FCNn_base> + 0 1024H + m × 40H
- Initial Value 00H. This register is initialized by any reset.

(1/2)

7	6	5	4	3	2	1	0
FCNnMm SSOW	FCNnMm SSMT[3:0]				FCNnMm SSRT	0	FCNnMm SSAM

Bit Position	Bit Name	Description
7	FCNnMmSSOW	(see table below)

FCNnMmSSOW	Overwrite Control Bit
0	The message buffer that has already received a data frame[Note] is not overwritten by a newly received data frame. The newly received data frame is discarded.
1	The message buffer that has already received a data frame[Note] is overwritten by a newly received data frame.

Note: The "message buffer that has already received a data frame" is a receive message buffer for which the FCNnMmCTL.FCNnMmDTNF bit has been set to 1.

Remark: A remote frame is received and stored, regardless of the setting of FCNnMmCTL.FCNnMmSSOW and FCNnMmCTL.FCNnMmDTNF. A remote frame that satisfies the other conditions is always received and stored in the corresponding message buffer (interrupt generated, FCNnMmDTNF flag set, FCNnMmDTLGB.FCNnMmDTLG[3:0] updated, and recorded in the receive history list).

(2/2)

Bit Position	Bit Name	Description
6 to 3	FCNnMmSSMT[3:0]	<table><tr><td>FCNnMmSSMT[3:0]</td><td>Message Buffer Type Setting Bit</td></tr><tr><td>0000B</td><td>Transmit message buffer</td></tr><tr><td>0001B</td><td>Receive message buffer (no mask setting)</td></tr><tr><td>0010B</td><td>Receive message buffer (mask 1 set)</td></tr><tr><td>0011B</td><td>Receive message buffer (mask 2 set)</td></tr><tr><td>0100B</td><td>Receive message buffer (mask 3 set)</td></tr><tr><td>0101B</td><td>Receive message buffer (mask 4 set)</td></tr><tr><td>0110B</td><td>Receive message buffer (mask 5 set)</td></tr><tr><td>0111B</td><td>Receive message buffer (mask 6 set)</td></tr><tr><td>1000B</td><td>Receive message buffer (mask 7 set)</td></tr><tr><td>1001B</td><td>Receive message buffer (mask 8 set)</td></tr><tr><td>Other than above</td><td>Setting prohibited</td></tr></table> **Remark: The setting of FCNnMmSSMT also selects a mask in conjunction with reception of remote frames. To receive remote frames in receive message buffers, flag FCNnMmSSRT of the message buffer must be set.**
2	FCNnMmSSRT	Specifies the type of message frame for transmission to or reception from a message buffer. <table><tr><td>FCNnMmSSRT</td><td>Remote Frame Request Bit</td></tr><tr><td>0</td><td>Transmit or receive a data frame.</td></tr><tr><td>1</td><td>Transmit or receive a remote frame.</td></tr></table> **Remarks 1. If the message buffer is defined as a transmit message buffer, and this buffer is to receive a remote frame, the FCNnMmSSRT bit must be cleared.** **2. Even if a valid remote frame has been received in a transmit message buffer, the FCNnMmSSRT bit of the transmit message buffer that has received the frame remains cleared to 0.** **3. Even when a remote frame whose ID matches has been received from the CAN bus, if the FCNnMmSSRT bit of a transmit message buffer is set to 1 (to transmit a remote frame), that remote frame is not stored in this transmit message buffer.** **4. If the message buffer is defined as a receive message buffer, the FCNnMmSSRT bit must be set, in order to receive remote frames instead of data frames.**
0	FCNnMmSSAM	<table><tr><td>FCNnMmSSAM</td><td>Message Buffer Assignment Bit</td></tr><tr><td>0</td><td>Message buffer not used.</td></tr><tr><td>1</td><td>Message buffer used.</td></tr></table>

Caution: Be sure to write 0 to bit 1.

(4) FCNn Message ID Register m (FCNnMmMID0H, FCNnMmMID1H, FCNnMmMID0W)

These registers are used to set an identifier (ID).

- Access FCNnMmMID0H and FCNnMmMID1H can be read or written in 16-bit units.
  FCNnMmMID0W can be read or written in 32-bit units.
- Address FCNnMmMID0H: <FCNn_base> + 0 9028H + m × 40H
  FCNnMmMID1H: <FCNn_base> + 0 9030H + m × 40H
  FCNnMmMID0W: <FCNn_base> + 1 1028H + m × 40H
- Initial Value 0000H for FCNnMmMID0H and FCNnMmMID1H.
  These registers are initialized by any reset.
  0000 0000H for FCNnMmMID0W.
  This register is initialized by any reset.

(a) FCNnMmMID0H

15	14	13	12	11	10	9	8
FCNnMmSSID[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCNnMmSSID[7:0]							

(b) FCNnMmMID1H

15	14	13	12	11	10	9	8
FCNnMm SSIE	0	0	FCNnMmSSID[28:24]				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCNnMmSSID[23:16]							

(c) FCNnCMmMID0W

31	30	29	28	27	26	25	24
FCNnMm SSIE	0	0	FCNnMmSSID[28:24]				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FCNnMmSSID[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FCNnMmSSID[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCNnMmSSID[7:0]							

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>31</td><td>FCNnMmSSIE</td><td><table><tr><th>FCNnMmSSIE</th><th>Format Mode Specification Bit</th></tr><tr><td>0</td><td>Standard format mode (FCNnMmSSID[28:18]: 11 bits. FCNnMmSSID[17:0] are not used)</td></tr><tr><td>1</td><td>Extended format mode (FCNnMmSSID[28:0]: 29 bits)</td></tr></table></td></tr>
<tr><td>28 to 0</td><td>FCNnMmSSID[28:0]</td><td><table><tr><th>FCNnMmSSID[28:0]</th><th>Message ID</th></tr><tr><td>FCNnMmSSID[28:18]</td><td>11 bits of standard ID value (FCNnMmSSIE = 0)</td></tr><tr><td>FCNnMmSSID[28:0]</td><td>29 bits of extended ID value (FCNnMmSSIE = 1)</td></tr></table></td></tr>
</table>

Cautions 1. Be sure to write 0 to bits 14 and 13 of FC NnMmMID1H and bits 30 and 29 of the FCNnMmMID0W register, respectively.

2. Align ID values with the selected range of bit positions in these registers. Note that for a standard ID, the ID value must be shifted to fit into the FCNnMmSSID[28:18] bits.

(5) FCNn Message Control Register m (FCNnMmCTL)

This register is used to control operation of the message buffer.

- Access This register can be read or written in 16-bit units.
- Address <FCNn_base> + 0 9038H + m × 40H
- Initial Value 0000H. This register is initialized by any reset.

(a) When FCNnMmCTL is read

(1/3)

15	14	13	12	11	10	9	8
0	0	FCNnMm MUCF	0	0	0	FCNnMm TCPF	0

7	6	5	4	3	2	1	0
0	FCNnMm NHMF	0	FCNnMm MOWF	FCNnMm IENF	FCNnMm DTNF	FCNnMm TRQF	FCNnMm RDYF

Bit Position	Bit Name	Description
13	FCNnMmMUCF	<table><tr><td>FCNnMmMUCF</td><td>Bit indicating that Message Buffer Data is being Updated</td></tr><tr><td>0</td><td>The FCN module is not updating the message buffer (no data is being received and stored).</td></tr><tr><td>1</td><td>The FCN module is updating the message buffer (data is being received and stored).</td></tr></table>
9	FCNnMmTCPF[Note1]	<table><tr><td>FCNnMmTCPF[Note1]</td><td>Transmission Complete Flag</td></tr><tr><td>0</td><td>Transmission failed.[Note2]</td></tr><tr><td>1</td><td>Transmission is complete.</td></tr></table> **Notes 1. FCNnMmTCPF is cleared if FCNnMmRDYF is changed or FCNnMmTRQF is set.** **2. If transmission abort was requested by clearing the FCNnMmTRQF flag by the application, FCNnMmTCPF = 0 indicates that the transmission has been successfully aborted.**
6	FCNnMmNHMF	<table><tr><td>FCNnMmNHMF</td><td>History Mask Flag[Note3]</td></tr><tr><td>0</td><td>Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is not masked.</td></tr><tr><td>1</td><td>Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is masked.</td></tr></table> **3. If updating is masked, transmit and receive history lists are not updated even when reception or transmission on the corresponding message buffer finishes.**

(2/3)

Bit Position	Bit Name	Description
4	FCNnMmMOWF	<table><tr><th>FCNnMmMOWF</th><th>Message Buffer Overwrite Status Bit</th></tr><tr><td>0</td><td>The message buffer is not overwritten by a newly received data or remote frame.</td></tr><tr><td>1</td><td>The message buffer is overwritten by a newly received data or remote frame.</td></tr></table> **Remark: This bit will not be set (1) if a remote frame is received and stored in a transmit message buffer with FCNnMmDTNF = 1.**
3	FCNnMmIENF	<table><tr><th>FCNnMmIENF</th><th>Message Buffer Interrupt Request Enable Bit</th></tr><tr><td>0</td><td>Receive message buffer: Valid message reception completion interrupt is disabled. Transmit message buffer: Normal message transmission completion interrupt and transmit abort interrupt are disabled.</td></tr><tr><td>1</td><td>Receive message buffer: Valid message reception completion interrupt is enabled. Transmit message buffer: Normal message transmission completion interrupt and transmit abort interrupt are enabled.</td></tr></table> **Caution: Always set FCNnMmIENF and FCNnMmRDYF separately.**
2	FCNnMmDTNF	<table><tr><th>FCNnMmDTNF</th><th>Message Buffer Data Update Bit</th></tr><tr><td>0</td><td>No new data frame or remote frame has been stored in the message buffer.</td></tr><tr><td>1</td><td>A new data frame or remote frame has been stored in the message buffer.</td></tr></table> **Caution: Do not set FCNnMmDTNF to 1 by software. Be sure to write 0 to bit 10.**

(3/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>1</td><td>FCNnMmTRQF</td><td>
<table>
<tr><td>FCNnMmTRQF</td><td>Message Buffer Transmission Request Bit</td></tr>
<tr><td>0</td><td>No message frame transmitting request that is pending or being transmitted is in the message buffer.</td></tr>
<tr><td>1</td><td>The message buffer is holding a message frame pending for transmission or is transmitting a message frame.</td></tr>
</table>
Cautions 1. Do not set FCNnMmTRQF and FCNnMmRDYF to 1 at the same time.
Set FCNnMmRDYF = 1 before setting FCNnMmTRQF = 1.
2. Only set FCNnMmTRQF to 1 for buffers other than transmit message buffers (buffers with FCNnMmSSMT[3:0] ≠ 4'b0000 or FCNnMmSSAM = 0).
</td></tr>
<tr><td>0</td><td>FCNnMmRDYF</td><td>
<table>
<tr><td>FCNnMmRDYF</td><td>Message Buffer Ready Bit</td></tr>
<tr><td>0</td><td>The message buffer can be written by software. The FCN module cannot write to the message buffer.</td></tr>
<tr><td>1</td><td>Writing the message buffer by software is ignored (except a write access to the FCNnMmRDYF, FCNnMmTRQF, FCNnMmDTNF and CNnMmMOWF). The FCN module can write to the message buffer.</td></tr>
</table>
Cautions 1. Always set FCNnMmIENF and FCNnMmRDYF separately.
2. Do not set FCNnMmTRQF and FCNnMmRDYF to 1 at the same time. Set FCNnMmRDYF = 1 before setting FCNnMmTRQF = 1.
3. Do not clear FCNnMmRDYF to "0" during message transmission. Execute transmission abort processing to clear FCNnMmRDYF to redefine the message buffer.
4. Clearing of FCNnMmRDYF may take time, depending on the operating condition of the CAN controller. Repeat access for clearing until the clearing of FCNnMmRDYF is confirmed by reading this bit.
5. Do not write to another FCN message buffer register until the clearing of FCNnMmRDYF is confirmed by checking its state.
</td></tr>
</table>

(b) When FCNnMmCTL is written

15	14	13	12	11	10	9	8
0	FCNnMm SENH	0	0	FCNnMm SEIE	0	FCNnMm SETR	FCNnMm SERY

7	6	5	4	3	2	1	0
0	FCNnMm CLNH	0	FCNnMm CLMW	FCNnMm CLIE	FCNnMm CLDN	FCNnMm CLTR	FCNnMm CLRY

Bit Position	Bit Name	Description
14, 6	FCNnMmSENH, FCNnMmCLNH	<table><tr><td>FCNnMmSENH</td><td>FCNnMmCLNH</td><td>Setting of FCNnMmNHMF Bit</td></tr><tr><td>0</td><td>1</td><td>FCNnMmNHMF is cleared (0).</td></tr><tr><td>1</td><td>0</td><td>FCNnMmNHMF is set (1).</td></tr><tr><td colspan="2">Other than above</td><td>FCNnMmNHMF is not changed.</td></tr></table>
4	FCNnMmCLMW	<table><tr><td>FCNnMmCLMW</td><td>Setting of FCNnMmMOWF Bit</td></tr><tr><td>0</td><td>FCNnMmMOWF is not changed.</td></tr><tr><td>1</td><td>FCNnMmMOWF is cleared (0).</td></tr></table>
11, 3	FCNnMmSEIE, FCNnMmCLIE	<table><tr><td>FCNnMmSEIE</td><td>FCNnMmCLIE</td><td>Setting of FCNnMmIENF Bit</td></tr><tr><td>0</td><td>1</td><td>FCNnMmIENF is cleared (0).</td></tr><tr><td>1</td><td>0</td><td>FCNnMmIENF is set (1).</td></tr><tr><td colspan="2">Other than above</td><td>FCNnMmIENF is not changed.</td></tr></table>
2	FCNnMmCLDN	<table><tr><td>FCNnMmCLDN</td><td>Setting of FCNnMmDTNF Bit</td></tr><tr><td>0</td><td>FCNnMmDTNF is not changed.</td></tr><tr><td>1</td><td>FCNnMmDTNF is cleared (0).</td></tr></table> **Remark: If FCNnMmDTNF is cleared at the end of ID field reception, the frames being received will be saved into the corresponding message buffer.**
9, 1	FCNnMmSETR, FCNnMmCLTR	<table><tr><td>FCNnMmSETR</td><td>FCNnMmCLTR</td><td>Setting of FCNnMmTRQF Bit</td></tr><tr><td>0</td><td>1</td><td>FCNnMmTRQF is cleared (0).</td></tr><tr><td>1</td><td>0</td><td>FCNnMmTRQF is set (1).</td></tr><tr><td colspan="2">Other than above</td><td>FCNnMmTRQF is not changed.</td></tr></table>
8, 0	FCNnMmSERY, FCNnMmCLRY	<table><tr><td>FCNnMmSERY</td><td>FCNnMmCLRY</td><td>Setting of FCNnMmRDYF Bit</td></tr><tr><td>0</td><td>1</td><td>FCNnMmRDYF is cleared (0).</td></tr><tr><td>1</td><td>0</td><td>FCNnMmRDYF is set (1).</td></tr><tr><td colspan="2">Other than above</td><td>FCNnMmRDYF is not changed.</td></tr></table>

25.6 Initialization of CAN Controller

25.6.1 Initialization of FCN Module

To enable operation of the FCN module, the FCN module system clock needs to be determined by setting FCNnGMCSPRE.FCNnGMCSPRSC[3:0] by software. Do not change the setting for the FCN module system clock after FCN module operation is enabled.

The FCN module is enabled by setting FCNnGMCLCTL.FCNnGMCLPWOM.

For the procedure of initializing the FCN module, refer to section 25.14, Operation of the CAN Controller.

25.6.2 Initialization of Message Buffer

After the FCN module is enabled, the message buffers might contain an undefined value (except after software reset). A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the FCN module from the initialization mode to any operation mode.

- Clear FCNnMmRDYF, FCNnMmTRQF and FCNnMmDTNF of the FCNnMmCTL registers to 0.
- Clear all FCNnMmSTRB.FCNnMmSSAM to 0.

25.6.3 Redefinition of Message Buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) To Redefine Message Buffer in Initialization Mode

Place the FCN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the FCN module to the operation mode.

(2) To Redefine Message Buffer during Reception

Redefine the message buffer by following the procedure described in Figure 25.17, Message Buffer Redefinition during Reception.

(3) To Redefine Message Buffer during Transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (see 25.8.4(1), Aborting Transmission Other than Automatic Block Transmission (ABT), and 25.8.4(2), Aborting Automatic Block Transmission (ABT), for details). Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining the transmit message buffer, set a transmission request by following the procedure described below.

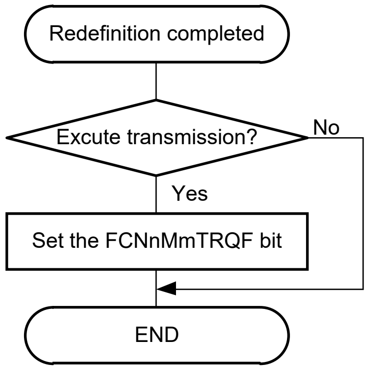


Figure 25.4 Setting Transmission Request (FCNnMmCTL.FCNnMmTRQF) to Transmit Message Buffer after Redefinition

Cautions 1. When a message is received, reception filtering is performed in accordance with the ID and mask set for each receive message buffer. If the procedure in Figure 25.17, Message Buffer Redefinition during Reception, is not followed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.

2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and FCNnMmSTRB.FCNnMmSSRT set for each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in Figure 25.4, Setting Transmission Request (FCNnMmCTL.FCNnMmTRQF) to Transmit Message Buffer after Redefinition, is not followed, a message with an ID having the highest priority may not be transmitted after redefinition.

25.6.4 Transition from Initialization Mode to Operation Mode

The FCN module can be switched to either of the following operation modes.

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

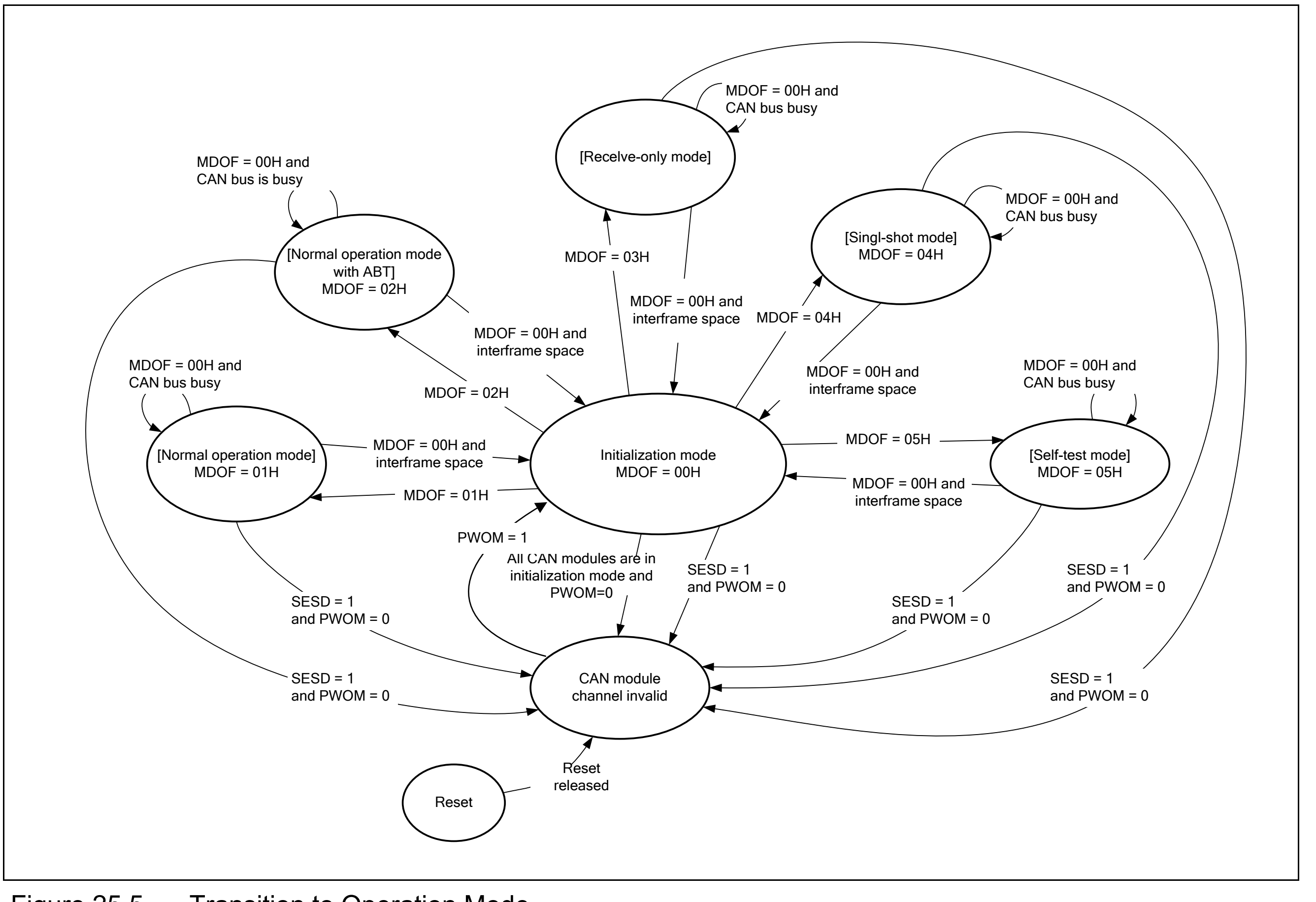


Figure 25.5 Transition to Operation Mode

Remark: In the figure above, following abbreviations are used:

- **MDOF = FCNnCMCLCTL.FCNnCMCLMDOF[2:0]**
- **PWOM = FCNnGMCLCTL.FCNnGMCLPWOM**
- **SESD = FCNnGMCLCTL.FCNnGMCLSESD**

The transition from the initialization mode to an operation mode is controlled by the FCNnCM.FCNnCMCLMDOF[2:0] bits.

Changing from one operation mode into another operation mode requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise operation will not be guaranteed.

Requests for transition from an operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the FCN module enters the initialization mode at the first bit in the interframe space (the values of the FCNnCMCLCTL.FCNnCMCLMDOF[2:0] are changed to 000B). After issuing a request to change the mode to the initialization mode, read FCNnCMCLCTL.FCNnCMCLMDOF[2:0] until their value becomes 000B to confirm that the module has entered the initialization mode (see Figure 25.14, Re-initialization without Using the Software Reset).

25.7 Message Reception

25.7.1 Message Reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer (FCNnMmSTRB.FCNnMmSSAM = 1)
- Set as a receive message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001B to 1001B)
- Ready for reception (FCNnMmCTL.FCNnMmRDYF = 1)

When two or more message buffers of the FCN module are found to be able to receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to store a message (i.e., when FCNnMmCTL.FCNnMmDTNF = 1 indicating that a message has already been received, but rewriting is disabled because FCNnMmSTRB.FCNnMmSSOW = 0). In this case, the message is not actually stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Table 25.15 Multi-Buffer Receive Block (MBRB) Priorities

Priority	Storing Condition if Same ID is Set	
1 (high)	Unmasked message buffer	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
2	Message buffer linked to mask 1	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
3	Message buffer linked to mask 2	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
...	...	...
9 (low)	Message buffer linked to mask 8	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1

25.7.2 Receive Data Read

To keep data consistency when reading FCN message buffers, perform the data reading according to Figure 25.31, Reception via Interrupt (Using FCNnCMLISTR Register), to Figure 25.34, Reception via Software Polling.

During message reception, the FCN module sets FCNnMmCTL.FCNnMmDTNF two times: at the beginning of the storage process of data to the message buffer, and again at the end of this storage process. During this storage process, FCNnMmCTL.FCNnMmMUCF of the message buffer is set (refer to Figure 25.6, Reception Timing).

The receive history list is also updated just before the storage process. In addition, during storage process (FCNnMmCTL.FCNnMmMUCF = 1), FCNnMmCTL.FCNnMmRDYF of the message buffer is locked to avoid writing of data by the CPU. Note that the storage process may be disturbed (delayed) when the CPU accesses the message buffer.

Caution: To reliably store a message in a message buffer, the DN bit for that buffer must be cleared before message search processing starts (after a frame ID is output on the bus). This might occur as early as the 15th CAN bit following the EOF of the previous frame. To reliably receive CAN frames successively sent over the bus, we recommend using two or more message buffers for frame reception.

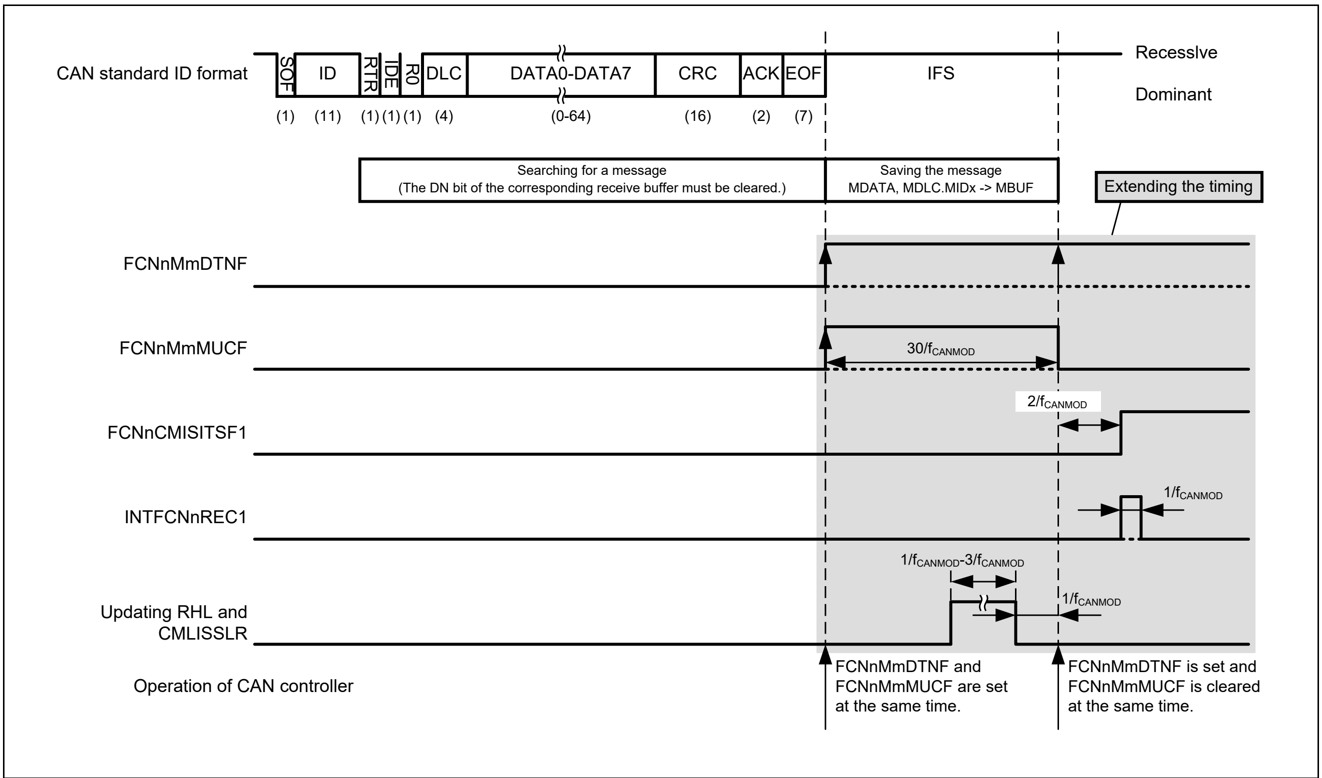


Figure 25.6 Reception Timing

25.7.3 Receive History List Function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 47 messages (on 64 message buffer FCN) or up to 95 messages (on 128 message buffer FCN), the last in-message pointer FCNnCMLISSLR[7:0] with the corresponding FCNnCMLISTR register and the receive history list get pointer FCNnCMRGSSPT with the corresponding FCNnCMRGRX register.

The RHL is undefined immediately after the transition of the FCN module from the initialization mode to one of the operation modes.

The FCNnCMLISTR register holds the contents of the RHL element indicated by the value of the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer minus 1. It is therefore possible to check the number of the message buffer that received and stored the last data frame or remote frame by reading the FCNnCMLISTR register. The FCNnCMLISSLR[7:0] pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the FCNnCMLISSLR[7:0] pointer. Each time recording to the RHL has been completed, the FCNnCMLISSLR[7:0] pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

For message buffers, where the flag FCNnMmCTL.FCNnMmNHMF is set, no entry in the history lists is recorded.

The FCNnCMRGRX.FCNnCMRGSSPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL.

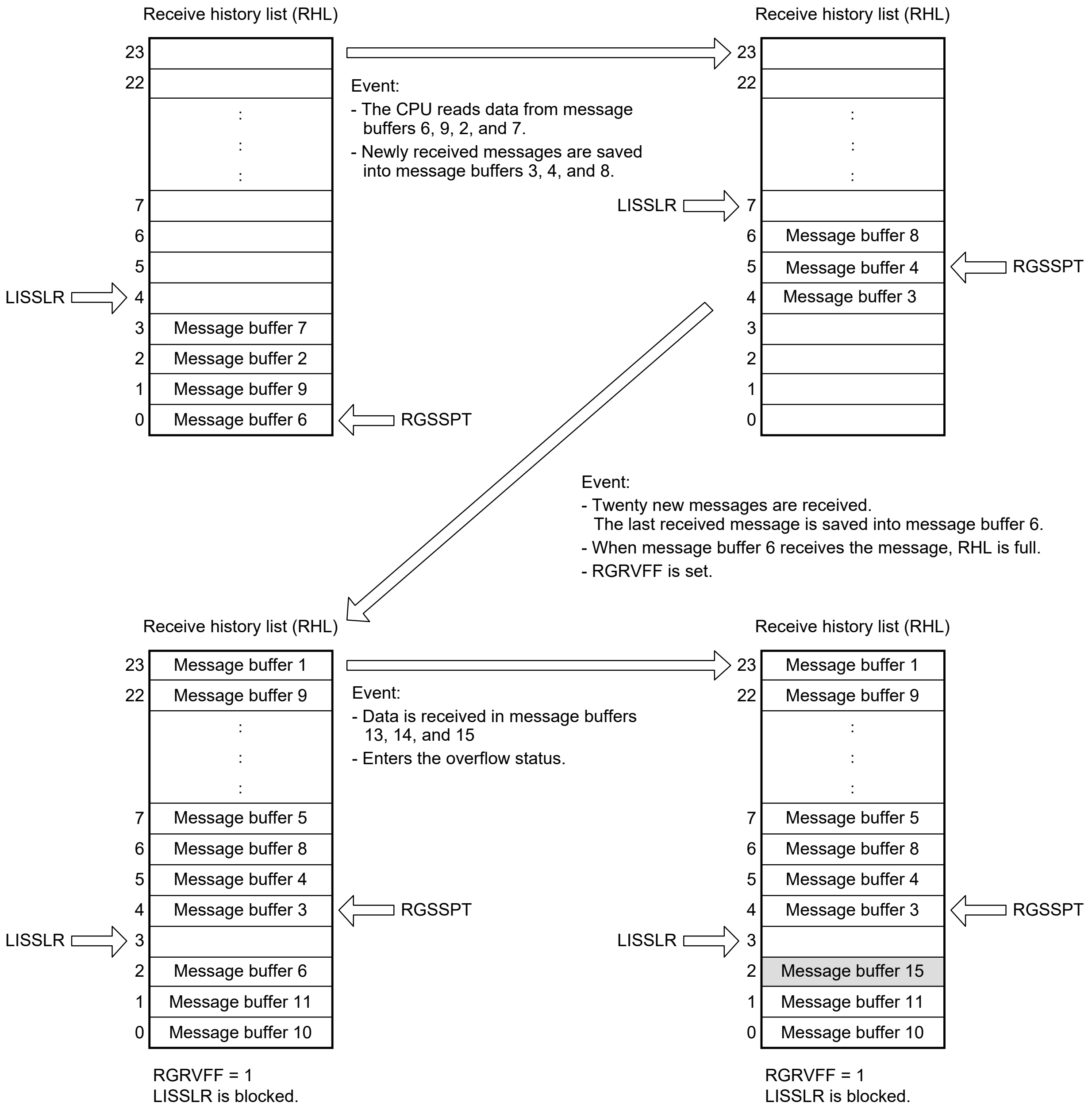
This pointer indicates the first RHL element that the CPU has not read yet. By reading the FCNnCMRGRX register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the FCNnCMRGRX register, the FCNnCMRGSSPT pointer is automatically incremented.

If the value of the FCNnCMRGRX.FCNnCMRGSSPT pointer matches the value of the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer, FCNnCMRGRX.FCNnCMRGSSPM (receive history list pointer match) is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the FCNnCMLISSLR[7:0] pointer is incremented and because its value no longer matches the value of the FCNnCMRGSSPT pointer, FCNnCMRGSSPM is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer is incremented and matches the value of the FCNnCMRGRX.FCNnCMRGSSPT pointer minus 1, FCNnCMRGRX.FCNnCMRGRVFF (receive history list overflow) is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the newly received message. In this case, after FCNnCMRGRVFF has been set (1), the recorded message buffer numbers in the RHL do not completely reflect the chronological order. However messages itself are not lost and can be located by CPU search in message buffer memory with the help of FCNnMmCTL.FCNnMmDTNF, or by reading the global registers FCNnDNBMRX.

Caution: If the receive history list overflows (FCNnCMRGRX.FCNnCMRGRVFF is set), reading the history list contents is still possible, until the receive history list is empty (indicated by the FCNnCMRGRX.FCNnCMRGSSPM flag being set). However, the history list remains overflowed until FCNnCMRGRVFF is cleared by software. If FCNnCMRGRVFF is not cleared, the FCNnCMRGSSPM flag will also not be updated (cleared) even if a newly received message in frames is stored. If this is the case, FCNnCMRGSSPM may indicate that the history list is empty (FCNnCMRGRVFF and FCNnCMRGSSPM are set), although reception has proceeded while the history list overflowed.

As long as the RHL has free entries, the order of reception is maintained. If further reception has proceeded before the host processor reads the RHL, the order of reception cannot be completely restored.



RGRVFF = 1 denoted that LISSLR equals RGSSPT - 1 while message buffer number stored to element indicated by LISSLR - 1.

Notes 1. The following abbreviations are used in the figure above.
- LISSLR = Last reception message pointer FCNnCMLISTR.FCNnCMLISSLR[7:0]
- RGSSPT = Reception history list acquisition pointer FCNnCMRGRX.FCNnCMRGSSPT[7:0]
- RGRVFF = Reception history list overflow flag FCNnCMRGRX.FCNnCMRGRVFF

2. If a message buffer with the FCNnMmCTL.FCNnMmNHMF flag set receives a message, the message buffer number indicated by LISSLR and the message buffer number written to RHL will differ.

Figure 25.7 Receive History List

25.7.4 Mask Function

Any message buffer, which is used for reception, can be assigned to one of eight global reception masks (or no mask).

By using the mask function, the bits for comparison of the message ID are reduced by masked bits, allowing the reception of several different IDs by one buffer.
While the mask function is in effect, an identifier bit that is defined to be 1 by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.
However, comparison is performed for any bit whose value is defined as 0 by the mask.

For example, assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are 0 and bits ID24 and ID22 are 1, are to be stored in message buffer 14. In this case, settings are as follows.

(1) Identifier to be Stored in Message Buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
X	0	0	0	1	X	1	X	X	X	X

(2) Identifier to be Configured in Message Buffer 14 (example)
(using FCN1M014MID0W register)

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
X	0	0	0	1	X	1	X	X	X	X
ID17	**ID16**	**ID15**	**ID14**	**ID13**	**ID12**	**ID11**	**ID10**	**ID9**	**ID8**	**ID7**
X	X	X	X	X	X	X	X	X	X	X
ID6	**ID5**	**ID4**	**ID3**	**ID2**	**ID1**	**ID0**				
X	X	X	X	X	X	X				

Remarks 1. IDs with the ID27 to ID25 bits cleared to 0 and the ID24 and the ID22 bits set to 1 are registered (initialized) in message buffer 14.

2. Message buffer 14 is set as a standard format identifier that is linked to mask 1 (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0010B).

Mask setting for FCN module 1 (mask 1) (example)
(using CAN1 address mask 1 register FCNnCMMKCTL01)

FNCnCMMKSSID[..]

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
1	0	0	0	1	1	1	1	1	1	1
ID17	**ID16**	**ID15**	**ID14**	**ID13**	**ID12**	**ID11**	**ID10**	**ID9**	**ID8**	**ID7**
1	1	1	1	1	1	1	1	1	1	1
ID6	**ID5**	**ID4**	**ID3**	**ID2**	**ID1**	**ID0**				
1	1	1	1	1	1	1				

1: Not compared (masked)
0: Compared

FCNnCMMKSSID[27:24] and FCNnCMMKSSID[21] are cleared to 0, and
FCNnCMMKSSID[28], FCNnCMMKSSID[23], and FCNnCMMKSSID[21:0] are set to 1.

25.7.5 Multi-Buffer Reception Blocking

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially without intervention by the CPU, by setting the same ID to two or more message buffers with the same message buffer type. These message buffers can be allocated anywhere in the message buffer memory, they do not even have to follow each other adjacently.

Suppose, for example, the same message buffer type is set to 10 message buffers, message buffers 10 to 19, and the same ID is set to each message buffer. If the first message whose ID matches an ID of the message buffers is received, it is stored in message buffer 10. At this point, FCNnMmCTL.FCNnMmDTNF of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.

When the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and so on. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.

Whether a data block has been received and stored can be checked by setting FCNnMmCTL.FCNnMmIENF of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. FCNnMmIENF in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and FCNnMmIENF in message buffer k-1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k-1, indicating that MBRB has become full. Alternatively, by clearing FCNnMmIENF of message buffers 0 to (k-3) and setting FCNnMmIENF of message buffer k-2, a warning that MBRB is about to overflow can be issued.

The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

Cautions 1. MBRB can be configured for each type of message buffer.
Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.

2. MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.

3. MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.

4. With MBRB, "matching ID" means "matching ID after mask". Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the destination to store messages.

5. The priority between MBRBs is mentioned in Table 25.16, List of FCN Module Interrupt Sources.

25.7.6 Remote Frame Reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers which satisfy the following conditions (conditions 1 and 2; condition 1 is given priority on reception). If condition 1 is not fulfilled, the remaining message buffers are searched to confirm whether condition 2 could be fulfilled.

(a) Condition 1:

Set as a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000B)

- Used as a message buffer (FCNnMmSTRB.FCNnMmSSAM = 1)
- Ready for reception (FCNnMmCTL.FCNnMmRDYF = 1)
- Set to data frame message type (FCNnMmSTRB.FCNnMmSSRT = 0)
- Transmission request is not set (FCNnMmCTL.FCNnMmTRQF = 0)

(b) Condition 2:

Set as a receive message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001B ... 1001B)

- Used as a message buffer (FCNnMmSTRB.FCNnMmSSAM = 1)
- Ready for reception (FCNnMmCTL.FCNnMmRDYF = 1)
- Set to remote frame message type (FCNnMmSTRB.FCNnMmSSRT = 1)
- Buffer is ready to store a message (FCNnMmCTL.FCNnMmDTNF = 0, or FCNnMmSTRB.FCNnMmSSOW = 1 with FCNnMmCTL.FCNnMmDTNF = 1)

Upon reception of a remote frame, the following actions are executed if the ID of the received remote frame matches the ID of a message buffer that satisfies the above conditions.

- The FCNnMmDTLG[3:0] bit string in the FCNnMmDTLGB register store the received DLC value.
- When received in a transmit message buffer, registers FCNnMmDAT0B to FCNnMmDAT7B in the data area will not be updated (the data from before reception is stored).
- FCNnMmCTL.FCNnMmDTNF is set to 1.
- FCNnCMISCTL.FCNnCMISITSF1 is set to 1 (if FCNnMmCTL.FCNnMmIENF of the message buffer that receives and stores the frame is set to 1).
- The receive completion interrupt (INTCnREC) is output (if FCNnMmCTL.FCNnMmIENF of the message buffer that receives and stores the frame is set to 1 and if FCNnCMIECTL.FCNnCMIEINTF1 is set to 1).
- The message buffer number is recorded in the receive history list, if the flag FCNnMmCTL.FCNnMmNHMF is not set.

Caution: When a transmit message buffer is found as a message buffer for receiving and storing a remote frame, overwrite control by FCNnMmSTRB.FCNnMmSSOW of the message buffer and FCNnMmCTL.FCNnMmDTNF are not checked. The setting of FCNnMmSSOW is ignored, and FCNnMmDTNF is set in any case.

Remarks 1. If more than one transmit message buffer has the same ID and the ID of the received remote frame matches that ID, the remote frame is stored in the transmit message buffer with the lowest message buffer number.

2. If transmit and receive message buffers are found, which could receive a remote frame matching with its ID, either masked or unmasked, the remote frame is stored in the transmit message buffer.

3. If several receive message buffers satisfy the conditions for reception of a remote frame, the reception priority is identical as for a data frame.

4. If a receive message buffer is found to match for a remote frame reception, and selected for storage, but this receive message buffer does not allow the storage because FCNnMmDTNF is set, and FCNnMmSSOW is not set, the remote frame is not stored at all.

25.8 Message Transmission

25.8.1 Transmission of Messages

Message buffers with its FCNnMmCTL.FCNnMmTRQF bit set to 1 are searched to find the message buffer for transmission of the highest-priority message if the following conditions are fulfilled. This processing is valid in any operation mode.

- Used as a message buffer (FCNnMmSTRB.FCNnMmSSAM = 1)
- Set as a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000B)
- Ready for transmission (FCNnMmCTL.FCNnMmRDYF = 1)

The CAN system is a multi-master communication system. In a system like this, the priority of message transmission is determined based on message identifiers (IDs).
To facilitate transmission processing by software when there are several messages awaiting transmission, the FCN module uses hardware to check the ID of the message with the highest priority and automatically identifies that message. This eliminates the need for software-based priority control.

The transmission priority is controlled by the identifier (ID).

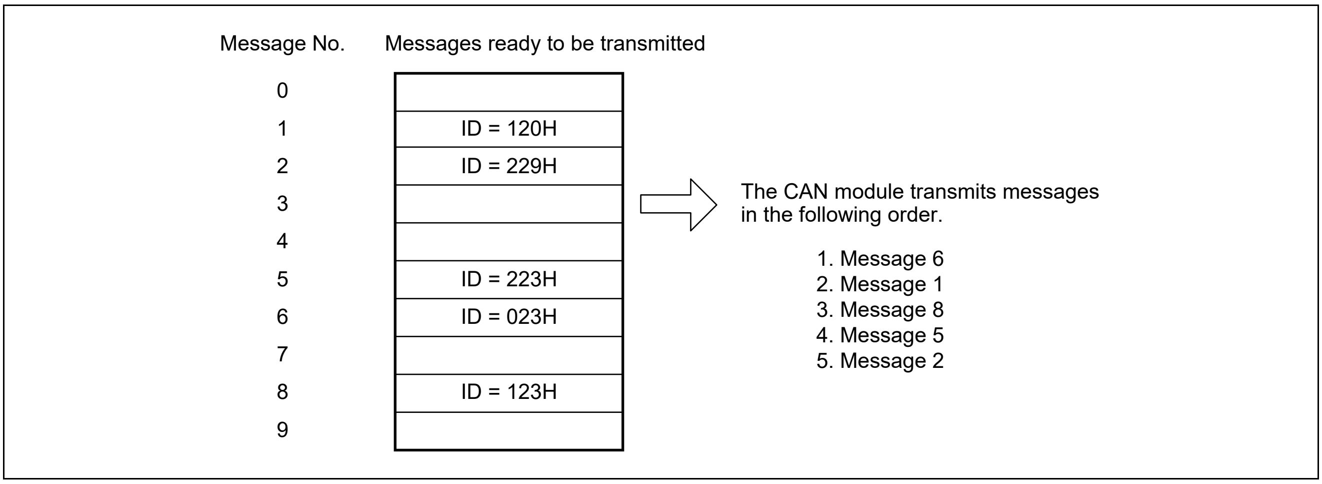


Figure 25.8 Message Processing Example

After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the FCNnMmCTL.FCNnMmTRQF bit set to 1 in advance) is transmitted.

If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. To solve this priority inversion effect, the software can issue a transmission abort request for the lower priority message. The order of priority is determined according to the following rules.

Priority	Conditions	Description
1 (high)	Value of higher-order 11 bits of ID (ID28 to ID18)	The message frame with the lowest value represented by the higher-order 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the higher-order 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than a message frame with a 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (FCNnMmSTRB.FCNnMmSSRT cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (bit FCNnMmSSIE in the message buffer identifier register FCNnCMmMID0W is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID (ID17 to ID0)	If one or more transmission-pending extended ID message frame has equal values in the higher-order 11 bits of the ID and the same frame type (equal FCNnMmSTRB.FCNnMmSSRT bit values), the message frame with the lowest value in the lower-order 18 bits of its extended ID is transmitted first.
5 (low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

Remarks 1. If the automatic block transmission request bit FCNnGMABCTL.FCNnGMABABTT is set to 1 in the normal operation mode with ABT, FCNnMmCTL.FCNnMmTRQF is set to 1 only for one message buffer in the ABT message buffer group.

If ABT mode is triggered by setting FCNnGMABCTL.FCNnGMABSEAT = 1, then one of the FCNnMmCTL.FCNnMmTRQF in the ABT area (64 message buffers FCN: 0 to 15, and 128 message buffers FCN: 0 to 31) will be set to 1. After this transmit request, the application can request transmission (set FCNnMmTRQF to 1) for other TX-message buffers that do not belong to the ABT area. In that case an interval arbitration process (TX-search) evaluates all TX-message buffers with FCNnMmTRQF set to 1 and chooses the message buffer that contains the highest-priority identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message stored in the lowest message buffer number is transmitted at first.

Upon successful transmission of a message frame, the following operations are performed.

- **The FCNnMmCTL.FCNnMmTRQF flag of the corresponding transmit message buffer is automatically cleared to 0.**
- **The transmission completion status bit FCNnCMISCTL.FCNnCMISITSF0 is set to 1 (if the interrupt enable bit FCNnMmIENF of the corresponding transmit message buffer is set to 1)**
- **An interrupt request signal INTCnTRX is output (if FCNnCMIECTL.FCNnCMIEINTF0 is set to1 and if the interrupt enable bit FCNnMmIENF of the corresponding transmit message buffer is set to 1).**

2. When changing the contents of a transmit buffer, the FCNnMmCTL.FCNnMmRDYF flag of this buffer must be cleared before updating the buffer contents. Since the FCNnMmRDYF flag may be locked temporarily during operation for internal transfer, etc., the status of the FCNnMmRDYF flag must be checked by software after changing it.

25.8.2 Transmit History List Function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been were sent. The THL consists of storage elements equivalent to up to 15 messages (on 64 message buffer FCN) or up to 31 messages (on 128 message buffer FCN), the last out-message pointer FCNnCMLOSTR[7:0] with the corresponding FCNnCMLOSTR register, and the transmit history list get pointer FCNnCMTGSSPT[7:0] with the corresponding FCNnCMTGTX register.

The THL is undefined immediately after the transition of the FCN module from the initialization mode to one of the operation modes.

The FCNnCMLOSTR register holds the contents of the THL element indicated by the value of the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer minus 1. By reading the FCNnCMLOSTR register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The FCNnCMLOSTR[7:0] pointer is utilized as a write pointer that indicates to what part of the THL a message buffer number is recorded. Any time a data frame or remote frame is transmitted, the corresponding message buffer number is recorded to the THL element indicated by the FCNnCMLOSTR[7:0] pointer. Each time recording to the THL has been completed, the FCNnCMLOSTR[7:0] pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

For message buffers, where the flag FCNnMmCTL.FCNnMmNHMF is set, no entry in the history lists is recorded.

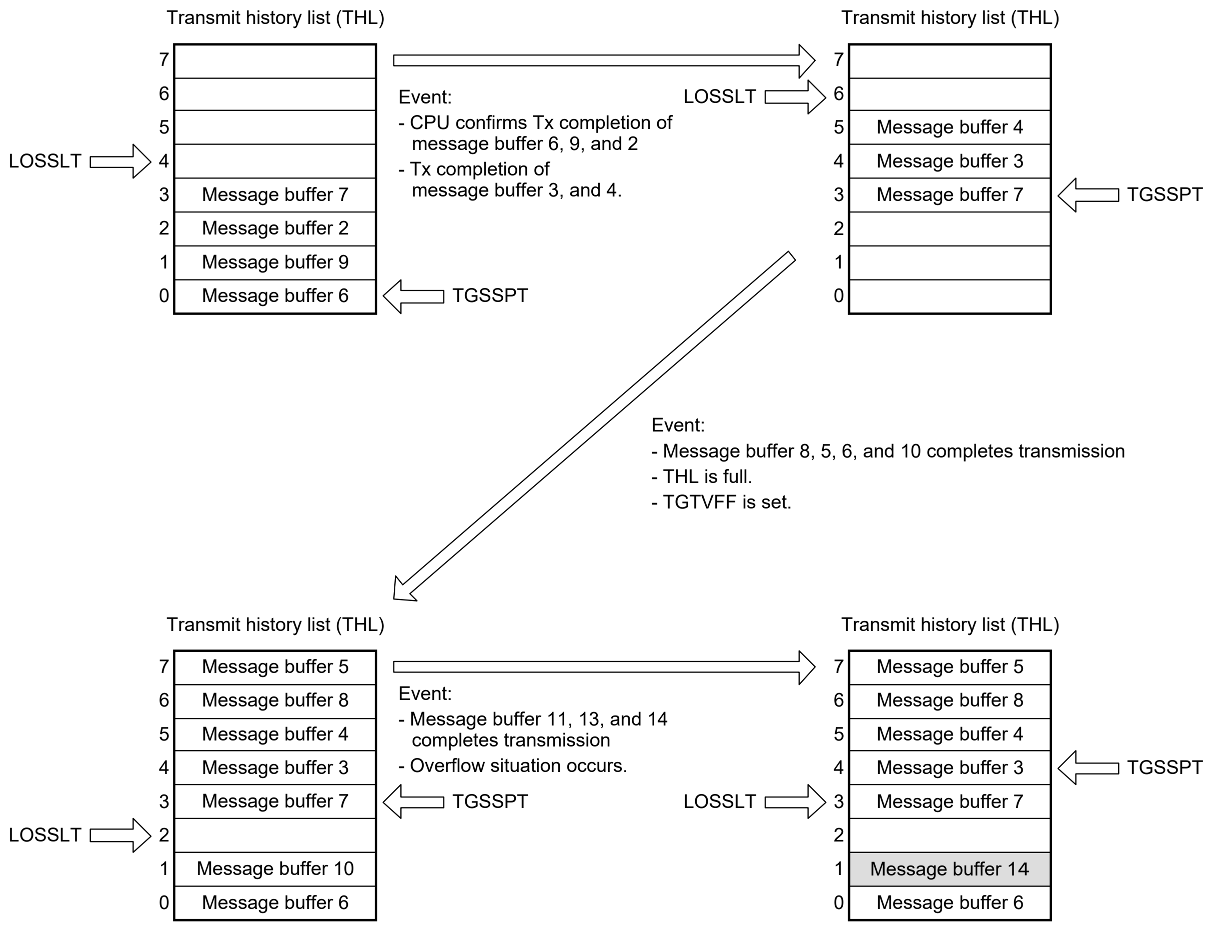
The FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer is utilized as a read pointer that reads a recorded message buffer number from the THL. This pointer indicates the first THL element that the CPU has not yet read. By reading the FCNnCMTGTX register by software, the number of a message buffer that has completed transmission can be read. Each time a message buffer number is read from the FCNnCMTGTX register, the FCNnCMTGSSPT[7:0] pointer is automatically incremented.

If the value of the FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer matches the value of the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer, FCNnCMTGTX.FCNnCMTGSSPM (transmit history list pointer match) is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the FCNnCMLOSTR[7:0] pointer is incremented and because its value no longer matches the value of the FCNnCMTGSSPT[7:0] pointer, FCNnCMTGSSPM is cleared. In other words, the numbers of the unread message buffers exist in the THL.

If the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer is incremented and matches the value of the FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer minus 1, FCNnCMTGTX.FCNnCMTGTVFF (transmit history list overflow) is set to 1. This indicates that the THL is full of message buffer numbers that have not been read. If a new message is received and stored, the message buffer number recorded last is overwritten by the message buffer number that transmitted its message afterwards. In this case, after FCNnCMTGTVFF has been set (1), therefore, the recorded message buffer numbers in the THL do not completely reflect the chronological order. Even in this case, however, the CPU can identify the number of the message buffer that completed reception by searching all reception buffers (the CPU does this before resetting transmission).

Regardless of the FCNnCMTGTX.FCNnCMTVFF setting, 14 (64 message buffers) or 30 (128 message buffer) transmit message buffer numbers are stored in THL.

Caution: If the transmit history list overflows (FCNnCMTGTX.FCNnCMTGTVFF is set), reading the history list contents is still possible, until the transmit history list is empty (indicated by the FCNnCMTGTX.FCNnCMTGSSPM flag being set). However, the history list remains overflowed until FCNnCMTGTVFF is cleared by software. If FCNnCMTGTVFF is not cleared, the FCNnCMTGTX.FCNnCMTGSSPM flag will also not be updated (cleared) upon successful transmission of a new message. If this is the case, FCNnCMTGSSPM may indicate that the history list is empty (FCNnCMTGTVFF and FCNnCMTGSSPM are set), although transmission has succeeded while the history list overflowed.



TGTVFF = 1 denoted that LOSSLT equals TGSSPT - 1 while message buffer number stored to element indicated by LOSSLT - 1.

Notes 1. The following abbreviations are used in the figure above.
- LOSSLT = Last transmission message pointer FCNnCMLOSTR.FCNnCMLOSSLT[7:0]
- TGSSPT = Transmission history list acquisition pointer FCNnCMTGTX.FCNnCMTGSSPT[7:0]
- TGTVFF = Transmission history list overflow flag FCNnCMTGTX.FCNnCMTGTVFF

2. If a message buffer with the FCNnMmCTL.FCNnMmNHMF flag set transmits a message, the message buffer number indicated by LOSSLT and the message buffer number written to THL will differ.

Figure 25.9 Transmit History List

25.8.3 Automatic Block Transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively without intervention by the CPU. The maximum number of transmit message buffers assigned to the ABT function is 16 (for 64 message buffer FCN) or 32 (for 128 message buffer FCN), always located in the lowest message buffers.

By setting FCNnCM.FCNnCMCLMDOF[2:0] to 010B, "normal operation mode with automatic block transmission function" ("ABT mode") can be selected.

To issue an ABT transmission request, define the message buffers by software first. Set FCNnMmSTRB.FCNnMmSSAM = 1in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting the FCNnMmSTRB.FCNnMmSSMT[3:0] bits to 0000B. Be sure to set the same ID for the message buffers for ABT even when that ID is being used for all the message buffers. To use two or more IDs, set the ID of each message buffer by using the FCNnMmMID0H and FCNnMmMID1H or FCNnMmMID0W registers. Set the FCN message data bytes registers before issuing a transmission request for the ABT function.

After initialization of message buffers for ABT is finished, FCNnMmCTL.FCNnMmRDYF needs to be set to 1. In the ABT mode, FCNnMmCTL.FCNnMmTRQF does not have to be manipulated by software.

After the data for the ABT message buffers has been prepared, set FCNnGMABCTL.FCNnGMABSEAT = 1. Automatic block transmission is then started. When ABT is started, FCNnMmCTL.FCNnMmTRQF in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 is finished, the FCNnMmTRQF of the next message buffer, message buffer 1, is set automatically. In this way, transmission is executed successively.

A delay time can be inserted by program in the interval in which the transmission request FCNnMmCTL.FCNnMmTRQF is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the FCNnGMADCTL register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the FCNnCMBRPRS and FCNnCMBTCTL registers.

Among transmit objects within the ABT-area, the priority of the transmission ID is not evaluated. Messages are sent by order of message number, starting with message buffer 0. When the transmission of the data frame from the last message buffer is complete, FCNnGMABCTL.FCNnGMABABTT is automatically cleared to 0, and ABT operation completes.

If there is an ABT message buffer for which FCNnMmCTL.FCNnMmRDYF is cleared during ABT, no data frame is transmitted from that buffer, ABT is stopped, and FCNnGMABCTL.FCNnGMABABTT is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting FCNnMmRDYF and FCNnGMABABTT to 1 by software. To not resume transmission from the message buffer where ABT stopped, the internal ABT engine can be reset by setting the FCNnGMABCTL.FCNnGMABCLRF bit to 1 while ABT mode is stopped and FCNnGMABABTT is cleared to 0. In this case, transmission is started from message buffer 0 if FCNnGMABCTL.FCNnGMABSEAC is cleared to 0 and then FFCNnGMABABTT is set to 1.

An interrupt can be used to check if data frames have been transmitted from all the message buffers for ABT. To do so, FCNnMmCTL.FCNnMmIENF of each message buffer except the last message buffer needs to be cleared (0).

If a transmit message buffer other than those used by the ABT function is assigned to a transmit message buffer, the message to be transmitted next is determined by the priority of the transmission ID of the ABT message buffer whose transmission is currently held pending and the transmission ID of the message buffers other than those used by the ABT function.

Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

Cautions 1. Set FCNnGMABCTL.FCNnGMABSEAC = 1 while FCNnGMABCTL.FCNnGMABABTT is cleared to 0 in order to resume ABT operation buffer No. 0. If FCNnGMABSEAC is set to 1 while FCNnGMABABTT is set to 1, the subsequent operation is not guaranteed.

2. If the automatic block transmission engine is cleared by setting FCNnGMABCTL.FCNnGMABSEAC = 1, FCNnGMABSEAC is automatically cleared immediately after the processing of the clearing request is completed.

3. Do not trigger automatic block transmission in the initialization mode. If FCNnGMABCTL.FCNnGMABSEAT is set in the initialization mode, proper operation is not guaranteed after the mode is changed from the initialization mode to the ABT mode.

4. Do not set FCNnMmCTL.FCNnMmTRQF of the ABT message buffers to 1 by software in the normal operation mode with ABT. Otherwise, correct operation is not guaranteed.

5. The FCNnGMADCTL register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of FCNnMmCTL.FCNnMmTRQF for the next ABT message when the transmission requests are set in the order of message numbers for each message for ABT that is successively transmitted in the ABT mode. The timing at which the messages are actually transmitted onto the CAN bus varies depending on the state of transmission from other stations and the setting of the transmission request for messages other than the ABT messages.

6. If a transmission request is issued for a message other than an ABT message and if no delay time is inserted in the interval in which transmission requests for ABT are automatically set (FCNnGMADCTL = 00H), messages other than ABT messages may be transmitted regardless of the difference in the priority of the ABT message.

7. Do not clear FCNnMmCTL.FCNnMmRDYF to 0 when FCNnGMABCTL.FCNnGMABABTT = 1.

25.8.4 Aborting Transmission

(1) Aborting Transmission Other than Automatic Block Transmission (ABT)

The user can clear FCNnMmCTL.FCNnMmTRQF to 0 to abort a transmission request. FCNnMmTRQF will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using FCNnCMCLCTL.FCNnCMCLSSTS and the FCNnCMTGTX register, or the FCNnMmCTL.FCNnMmTCPF flag, which indicate the transmission status on the CAN bus (for details, refer to the processing in Figure 25.24, Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed)).

(2) Aborting Automatic Block Transmission (ABT)

To abort the ABT that was already started, clear FCNnGMABCTL.FCNnGMABABTT to 0. In this case, FCNnGMABCTL.FCNnGMABABTT remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.

If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.

In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of FCNnMmCTL.FCNnMmTRQF in the last transmitted message buffer. If FCNnMmTRQF is cleared to 0 when clearing FCNnGMABCTL.FCNnGMABABTT is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area (for details, refer to the process in Figure 25.26, ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1)).

Caution: Be sure to abort the ABT by clearing FCNnGMABCTL.FCNnGMABCLAT to 0. Correct operation is not guaranteed if aborting transmission is requested by clearing FCNnMmCTL.FCNnMmRDYF.

When the normal operation mode with ABT is resumed after the ABT has been aborted and FCNnGMABCTL.FCNnGMABSEAT is set to 1, the next ABT message buffer to be transmitted can be determined from the following table.

Status of FCNnMmCTL.FCNnMmTRQF of ABT Message Buffer	The ABT is Aborted after Successful Transmission	The ABT is Aborted after Failure in the Transmission
Set (1)	Next message buffer in the ABT area[Note]	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area[Note]	Next message buffer in the ABT area[Note]

Note: The above resumption operation can be performed only if a message buffer ready for ABT exists in the ABT area. For example, an abort request that is issued while the ABT of the message buffer with the highest number is in progress is regarded as completion of ABT, rather than abort, if transmission of this message buffer has been successfully completed, even if FCNnGMABCTL.FCNnGMABABTT is cleared to 0. If FCNnMmCTL.FCNnMmRDYF in the next message buffer in the ABT area is cleared to 0, the internal ABT pointer is retained, but the resumption operation is not performed even if FCNnGMABABTT is set to 1, and ABT ends immediately.

25.8.5 Remote Frame Transmission

Remote frames can be transmitted only from transmit message buffers.

Set whether a data frame or remote frame is transmitted via FCNnMmSTRB.FCNnMmSSRT. Setting FCNnMmSSRT = 1 sets remote frame transmission.

25.9 Power Saving Modes

25.9.1 FCN Sleep Mode

The FCN sleep mode can be used to set the CAN controller to standby mode in order to reduce power consumption.
The FCN module can enter the FCN sleep mode from any operation mode. Release from the FCN sleep mode returns the FCN module to the same operation mode from which the FCN sleep mode was entered.

In the FCN sleep mode, the FCN module does not transmit messages, even when transmission requests are issued or pending.

(1) Transition to FCN Sleep Mode

The CPU issues a FCN sleep mode transition request by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01B.

This transition request is acknowledged only under the following conditions.

1. The FCN module is already in one of the following operation modes
   - Normal operation mode
   - Normal operation mode with ABT
   - Receive-only mode
   - Single-shot mode
   - Self-test mode
   - FCN stop mode in all the above operation modes
2. The CAN bus is in the idle state (the 4th bit in the interframe space is recessive).
   If the CAN bus is fixed to dominant, the request for transition to the FCN sleep mode is held pending. Also the transition from FCN stop mode to FCN sleep mode is independent of the CAN bus state.
3. No transmission request is pending.
4. Power save mode cannot be set in combination with the change of operation mode.
   Be sure to perform these operations in different steps.

Remark: If a sleep mode request is pending, and at the same time a message is received in a message box, the sleep mode request is not cancelled, but is executed right after message storage has been finished. This may result in the FCN being placed in sleep mode, while the CPU is executing the RX interrupt routine. Therefore, the interrupt routine must check the access to the message buffers as well as reception history list registers by using the FCNnGMCLSSMO flag, if sleep mode is used.

If any one of the conditions mentioned above is not met, the FCN module will operate as follows.

- **If the FCN sleep mode is requested from the initialization mode, the FCN sleep mode transition request is ignored and the FCN module remains in the initialization mode.**
- **If the CAN bus is not in the idle state (i.e. the CAN bus state is either transmitting or receiving) when the FCN sleep mode is requested in one of the operation modes, immediate transition to the FCN sleep mode is not possible. In this case, the FCN sleep mode transition request is held pending until the CAN bus becomes idle (the 4th bit in the interframe space is recessive). In the time from the FCN sleep mode request to successful transition, FCNnCMCLCTL.FCNnCMCLMDPF[1:0] remain 00B. When the module has entered the FCN sleep mode, the FCNnCMCLMDPF[1:0] bits are set to 01B.**
- **If a request for transition to the initialization mode and a request for transition to the FCN sleep mode are made at the same time while the FCN module is in one of the operation modes, the request for the initialization mode is enabled. The FCN module enters the initialization mode at a predetermined timing. At this time, the FCN sleep mode request is not held pending and is ignored.**
- **Even when initialization mode and sleep mode are not requested simultaneously (i.e. the first request has not been acknowledged while the second request is issued), the request for initialization is given priority over the sleep mode request. The sleep mode request is cancelled when the initialization mode is requested. When a pending request for initialization mode is present, a subsequent request for sleep mode request is cancelled at the point at which it was submitted.**

(2) Status in FCN Sleep Mode

The FCN module is in the following state after it enters the FCN sleep mode:

- The internal operating clock is stopped and the power consumption is minimized.
- The function to detect the falling edge of the FCN reception pin (CRXDn) remains in effect to wake up the FCN module from the CAN bus.
- To wake up the FCN module from the CPU, data can be set to FCNnCMCLCTL.FCNnCMCLMDPF[1:0], but nothing can be written to other FCN module registers or bits.
- The FCN module registers can be read, except for the FCNnCMLISTR, FCNnCMRGRX, FCNnCMLOSTR, and FCNnCMTGTX registers.
- The FCN message buffer registers cannot be written or read.
- FCNnGMCLCTL.FCNnGMCLSSMO is cleared.
- The registers FCNnDNBMRX cannot be read.
- A request for transition to the initialization mode is not acknowledged and is ignored.

(3) Release from FCN Sleep Mode

The FCN module is released from FCN sleep mode by the following events:

- When the CPU sets FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 00B
- A falling edge of the signal on the FCN reception pin CRXDn (i.e. the CAN bus level shifts from recessive to dominant)

Caution: Even if the falling edge belongs to the SOF of a receive message, this message will not be received and stored. If the CPU has turned off the clock supply to the FCN module while the FCN module was in sleep mode, even subsequently the FCN sleep mode will not be released and FCNnCMCLMDPF[1:0] will remain 01B unless the clock to the FCN module is supplied again. In addition to this, the receive message will not be received after that.

After release from the sleep mode, the FCN module returns to the operation mode from which the FCN sleep mode was requested and FCNnCMCLCTL.FCNnCMCLMDPF[1:0] must be reset by software to 00B. If the FCN sleep mode is released by a change in the CAN bus state, FCNnCMISCTL.FCNnCMISITSF5 is set to 1, regardless of FCNnCMIECTL.FCNnCMIEINTF[6:0]. After the FCN module is released from the FCN sleep mode, it participates in the CAN bus communications again by automatically detecting 11 consecutive recessive-level bits on the CAN bus. The user application has to wait until FCNnGMCLCTL.FCNnGMCLSSMO = 1, before accessing message buffers again.

When a request for transition to the initialization mode is made while the FCN module is in the FCN sleep mode, that request is ignored; the FCN module has to be released from sleep mode by software first before entering the initialization mode.

Cautions 1. Be aware that the release from FCN sleep mode by CAN bus event, i.e., the wakeup interrupt may occur at any time even right after the transition to sleep mode has been requested, if a CAN bus event occurs.

2. After wakeup from FCN sleep mode, always reset the FCNnCMCLCTL.FCNnCMCLMDPF[1:0] bits to 00B before accessing any other registers of the FCN module.

3. After wakeup from FCN sleep mode, always clear the interrupt flag FCNnCMISCTL.FCNnCMISITSF5.

25.9.2 FCN Stop Mode

The FCN stop mode can be used to place the CAN controller in standby mode to reduce power consumption. The FCN module can enter the FCN stop mode only from the FCN sleep mode.

Release from the FCN stop mode places the FCN module in the FCN sleep mode.

The FCN stop mode can only be released (entering FCN sleep mode) by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 01B and not by a change in the CAN bus state. While the FCN module is in the FCN stop mode, no message is transmitted even when transmission requests are issued or pending.

(1) Transition to FCN Stop Mode

A FCN stop mode transition request is issued by setting 11B to FCNnCMCLCTL.FCNnCMCLMDPF[1:0].

A FCN stop mode request is only acknowledged when the FCN module is in the FCN sleep mode. In any other mode, the request is ignored.

Caution: To set the FCN module to the FCN stop mode, the module must be in the FCN sleep mode. To confirm that the module is in the sleep mode, check that the FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01B, and then issue a request for transition to the FCN stop mode. If a bus change occurs at the FCN reception pin CRXDn while this processing is in progress, the FCN sleep mode is automatically released. In this case, the FCN stop mode transition request cannot be acknowledged.

(2) Status in FCN Stop Mode

The FCN module is in the following state after it enters the FCN stop mode.

- The internal operating clock is stopped and the power consumption is minimized.
- To wake up the FCN module from the CPU, data can be set in FCNnCMCLCTL.FCNnCMCLMDPF[1:0], but nothing can be written to other FCN module registers or bits.
- The FCN module registers can be read, except for the FCNnCMLISTR, FCNnCMRGRX, FCNnCMLOSTR, and FCNnCMTGTX registers.
- The FCN message buffer registers cannot be written or read.
- FCNnGMCLCTL.FCNnGMCLSSMO is cleared.
- The registers FCNnDNBMRX cannot be read.
- An initialization mode transition request is not acknowledged and is ignored.

(3) Release from FCN Stop Mode

The FCN module can only be released from FCN stop mode by writing 01B to FCNnCMCLCTL.FCNnCMCLMDPF[1:0]. After release from the FCN stop mode, the FCN module enters the FCN sleep mode.

When the initialization mode is requested while the FCN module is in the FCN stop mode, that request will be ignored; the CPU has to release the stop mode and subsequently FCN sleep mode before entering the initialization mode. Direct transition from the FCN stop mode to another operation mode without entering the FCN sleep mode is not possible. Such transition request will be ignored.

25.9.3 Example of Using Power Saving Mode

In some application systems, it may be necessary to place the CPU in power saving mode to reduce power consumption. By using the power saving mode specific to the FCN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving state by the CAN bus.

Here is an example for using the power saving mode.

- First, put the FCN module in the FCN sleep mode (FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01B).
  After successfully confirming this state by reading back the sleep mode status, put the CPU in the power saving mode. Disable interrupts for the CPU, while processing additional tasks after the FCN module is in sleep mode, to avoid that the FCN wakeup interrupt is acknowledged.
  If a rising edge from recessive to dominant is detected on the CRXDn FCN reception pin in this state, FCNnCMISCTL.FCNnCMISITSF5 in the FCN module will be set to 1. If FCNnCNIECTL.FCNnCMIEINT5 is set to 1, a wakeup interrupt (INTCnWUP) is generated.
  The FCN module is automatically released from FCN sleep mode (FCNnCMCLMDPF[1:0] = 00B) and returns to normal operation mode.
- The CPU, in response to INTCnWUP, can release its own power saving mode and return to normal operation mode. To further reduce the power consumption of the CPU, the internal clock - including that of the FCN module - may be stopped. In this case, the operating clock supplied to the FCN module is stopped after the FCN module has been put in FCN sleep mode. Then the CPU enters a power saving mode in which the clock supplied to the CPU is stopped.
- If an edge transition from recessive to dominant is detected at the FCN reception pin CRXDn in this status, the FCN module can set FCNnCMISCTL.FCNnCMISITSF5 to 1 and generate the wakeup interrupt INTCnWUP even if it is not supplied with the clock.
- The other functions, however, do not operate, because clock supply to the FCN module is stopped, and the module remains in FCN sleep mode.
- The CPU, in response to INTCnWUP,
  - releases its power saving mode,
  - resumes supply of the internal clocks - including the clock to the FCN module - after the oscillation stabilization time has elapsed, and
  - starts instruction execution.
- The FCN module is immediately released from the FCN sleep mode when clock supply is resumed, and returns to the normal operation mode (FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 00B).

25.10 Interrupts

The FCN module has 6 different interrupt sources.

The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 25.16 List of FCN Module Interrupt Sources

No.	Interrupt Status Bit FCNnCMISCTL.	Interrupt Enable Bit FCNnCMIECTL.Note	Interrupt Request Signal	Interrupt Source Description
1	FCNnCMISITSF0	FCNnCMIESEIE0	INTCnTRX	Message frame successfully transmitted from message buffer m
2	FCNnCMISITSF1	FCNnCMIESEIE1	INTCnREC	Valid message frame reception in message buffer m
3	FCNnCMISITSF2	FCNnCMIESEIE2	INTCnERR	FCN module error state interrupt • This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.
4	FCNnCMISITSF3	FCNnCMIESEIE3		FCN module protocol error interrupt • This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.
5	FCNnCMISITSF4	FCNnCMIESEIE4		FCN module arbitration loss interrupt
6	FCNnCMISITSF5	FCNnCMIESEIE5	INTCnWUP	FCN module wakeup interrupt from FCN sleep mode • This interrupt is generated when the FCN module wakes up from FCN sleep mode, due to detection of a rising edge on the FCN reception pin (change of CAN bus from recessive to dominant).
7	FCNnCMISITSF6	FCNnCMIESEIE6		FCN module transmit abort interrupt status • This interrupt is generated when the abortion of a transmission was successful (aborted message was not sent).

Note: The message buffer interrupt enable bit FCNnMmCTL.FCNnMmIENF of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

25.11 Diagnosis and Special Operation Modes

The FCN module has a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis or the special CAN communication methods.

25.11.1 Receive-Only Mode

The receive-only mode is used to monitor receive messages without any interference on the CAN bus and can be used for CAN bus analysis nodes.

For example, this mode can be used for automatic baud-rate detection. The baud rate in the FCN module is changed until "valid reception" is detected, so that the baud rates in the module match ("valid reception" means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus).

A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting FCNnCMCLCTL.FCNnCMCLVALF = 1.

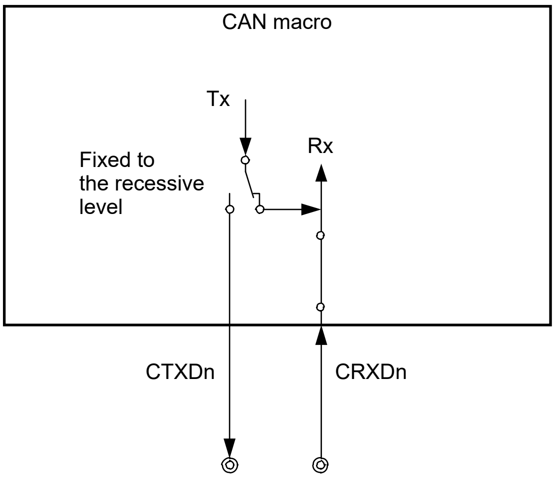


Figure 25.10 FCN Module Terminal Connection in Receive-Only Mode

In the receive-only mode, no message frames can be transmitted from the FCN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.

In the receive-only mode, the FCN transmission pin CTXDn in the FCN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the FCN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the FCN module, the transmission error counter the FCNnCMERCNT.FCNnCMERTECF[7:0] bits are never updated. Therefore, a FCN module in the receive-only mode does not enter the bus-off state.

Furthermore, in the receive-only mode ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

Caution: If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). After the message frame for the 17th time is transmitted, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the FCNnCMCLCTL.FCNnCMCLVALF bit is set to 1 for the first time.

25.11.2 Single-Shot Mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off. (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.) All other behavior of single shot mode is identical to normal operation mode.

Features of single shot mode cannot be used in combination with normal mode with ABT.

The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of FCNnCMCLCTL.FCNnCMCLALBF. When FCNnCMCLALBF is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If FCNnCMCLALBF is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, FCNnMmCTL.FCNnMmTRQF in a message buffer defined as a transmit message buffer is cleared to 0 by the following events:

- Successful transmission of the message frame
- Arbitration loss while sending the message frame
- Error occurrence while sending the message frame

The events arbitration loss and error occurrence can be distinguished by checking FCNnCMISCTL.FCNnCMISITSF4 and FCNnCMISCTL.FCNnCMISITSF3 respectively, and the type of the error can be identified by reading FCNnCMLCSTR.FCNnCMLCSSLC[2:0].

Upon successful transmission of the message frame, the transmit completion interrupt bit FCNnCMISCTL.FCNnCMISITSF0 is set to 1. If FCNnCMIECTL.FCNnCMIEINTF0 is set to 1 at this time, an interrupt request signal is output.

The single-shot mode can be used when emulating time-triggered communication methods (e.g., TTCAN level 1).

Caution: FCNnCMCLCTL.FCNnCMCLALBF is only valid in single-shot mode. It does not influence the operation of re-transmission upon arbitration loss in the other operation modes.

25.11.3 Self-Test Mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In the self-test mode, the FCN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The FCN transmission pin CTXDn is fixed to the recessive level.

If the falling edge on the FCN reception pin CRXDn is detected after the FCN module has entered the FCN sleep mode from the self-test mode, however, the module is released from the FCN sleep mode in the same manner as the other operation modes. Use the CRXDn FCN reception pin as a port pin in order to keep the module in FCN sleep mode.

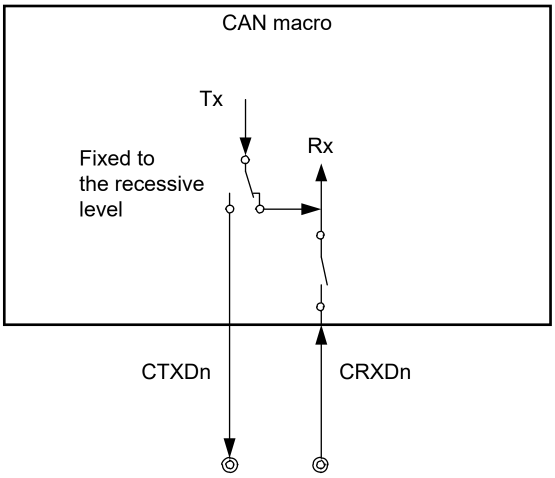


Figure 25.11 FCN Module Terminal Connection in Self-Test Mode

25.11.4 Receive/Transmit Operation in Each Operation Mode

The following table shows the outline of the receive/transmit operation in each operation mode.

Table 25.17 Outline of the Receive/Transmit in Each Operation Mode

Operation Mode	Transmission of Data/Remote Frame	Transmission of ACK	Transmission of Error/Overload Frame	Transmission Retry	Automatic Block Transmission (ABT)	Set of FCNnCMCLVALF Bit	Store Data to Message Buffer
Initialization Mode	No	No	No	No	No	No	No
Normal operation mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal operation mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive only mode	No	No	No	No	No	Yes	Yes
Single-shot mode	Yes	Yes	Yes	No[Note1]	No	Yes	Yes
Self-test mode	Yes[Note2]	Yes[Note2]	Yes[Note2]	Yes[Note2]	No	Yes[Note2]	Yes[Note2]

Notes 1. When the arbitration lost occurs, control of re-transmission is possible by FCNnCMCLCTL.FCNnCMCLALBF.

2. Generated signals are not externally output, but stay in the FCN module.

25.12 Timestamping

CAN is an asynchronous serial communication protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies).

In some applications, however, a common time base over the network (= global time base) is required. In order to build up a global time base, timestamping is used. The essential mechanism of timestamping is the capture of timer values triggered by signals on the CAN bus.

25.12.1 Timestamping

The CAN controller supports the capturing of timer values triggered by a specific frame. An on-chip 32-bit capture timer unit (TAUJ2) in a microcontroller system is used in addition to the CAN controller. The 32-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN Controller.

The CPU can retrieve the time when the capture event occurred, i.e., the timestamp of the message received from the CAN bus, by reading the captured value. The TSOUT signal can be selected from the following two event sources and is specified by FCNnCMTSCTL.FCNnCMTSSELE.

- SOF event (start of frame) (FCNnCMTSCTL.FCNnCMTSSELE = 0)
- EOF event (the least significant bit of the end of frame) (FCNnCMTSCTL.FCNnCMTSSELE = 1)

The TSOUT signal is enabled by setting FCNnCMTSCTL.FCNnCMTSTSGE = 1.

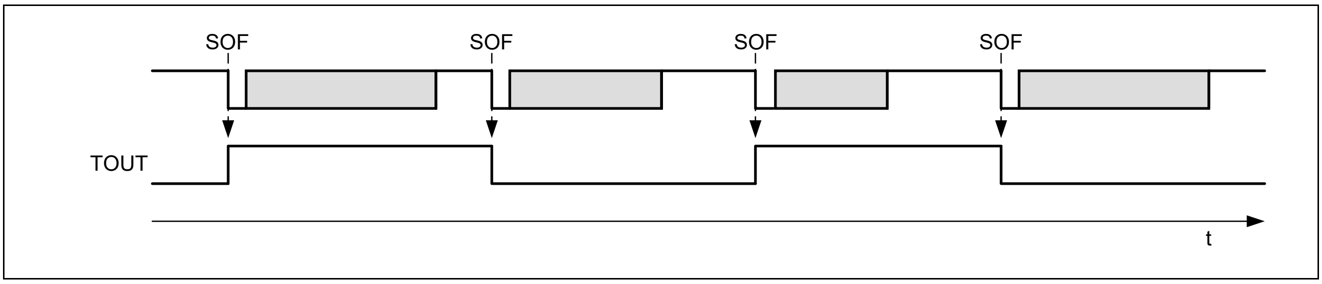


Figure 25.12 Timing Diagram of Capture Signal TSOUT

The TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in Figure 25.12, Timing Diagram of Capture Signal TSOUT, the SOF is used as the trigger event source). To capture a timer value by using the TSOUT signal, the capture timer unit must detect the capture signal at both the rising edge and falling edge.

This timestamping is controlled by the FCNnCMTSLOKE bit of the FCNnCMTSCTL register. When FCNnCMTSLOKE is cleared to 0, the TSOUT signal toggles upon occurrence of the selected event. If FCNnCMTSLOKE is set to 1, the TSOUT signal toggles upon occurrence of the selected event, but the toggle is stopped as FCNnCMTSCTL.FCNnCMTSTSGE is automatically cleared to 0 as soon as storing of messages in message buffer 0 starts.

This suppresses the subsequent toggle occurrence by the TSOUT signal, so that the timestamp value toggled last (= captured last) can be saved as the timestamp value of the time at which the data frame was received in message buffer 0.

Caution: Timestamping which uses the FCNnCMTSLOKE bit stops toggling of the TSOUT signal by receiving a data frame in message buffer 0. Toggling of the TSOUT signal does not stop when a data frame is received in a message buffer other than message buffer 0. A data frame cannot be received in message buffer 0 when the FCN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer.

In this operation mode, therefore, the function to stop toggling of the TSOUT signal by the FCNnCMTSLOKE bit cannot be used.

25.13 Baud Rate Settings

25.13.1 Baud Rate Setting Conditions

Make sure that the settings are within the range of the limit values below to ensure correct operation of the CAN controller.

- 5 TQ ≤ SPT (sampling point) ≤ 17 TQ
  SPT = FCNnCMBTS1LG[3:0] + 1
- 8 TQ ≤ DBT (data bit time) ≤ 25 TQ
  DBT = FCNnCMBTS1LG[3:0] + FCNnCMBTS2LG[2:0] + 1 TQ = FCNnCMBTS2LG[2:0] + SPT
- 1 TQ ≤ FCNnCMBTJWLG[1:0] (synchronization jump width) ≤ 4 TQ
  FCNnCMBTJWLG[1:0] ≤ DBT – SPT
- 4 TQ ≤ TSEG1 ≤ 16 TQ [3 ≤ FCNnCMBTS1LG[3:0] ≤ 15]
- 1 ≤ TSEG2[2:0] ≤ 8 [0 ≤ FCNnCMBTS2LG[2:0] ≤ 7]
- 75 [nsec][Note] < 5 [nsec] + 1 TQ – 20 [nsec][Note]

Note: 75 nsec: This value is the max value of internal delay time for CAN interface (t_{NODE}) which is listed in data sheet.
20 nsec: This value is from 2 PCLK. (PCLK is 100 MHz clock)

Remarks 1. TQ = 1/fTQ (fTQ: CAN protocol layer basic system clock)
2. The values of FCNnCMBTS1LG[3:0], FCNnCMBTS2LG[2:0] and FCNnCMBTJWLG[1:0] are specified by the FCNnCMBTCTL register.

Table 25.18, Combinations of Available Bit Rate Settings, shows the combinations of bit rates that satisfy the above conditions.

Table 25.18 Combinations of Available Bit Rate Settings

(1/3)

Valid Bit Rate Setting					FCNnCMBTCTL Register Setting Value		Sampling Point (unit: %)
DBT Length	SUNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMB TS1LG[3:0]	FCNnCMB TS2LG[2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7

Table 25.18 Combinations of Available Bit Rate Settings

(2/3)

Valid Bit Rate Setting					FCNnCMBTCTL Register Setting Value		Sampling Point (unit: %)
DBT Length	SUNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMB TS1LG[3:0]	FCNnCMB TS2LG[2:0]	
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0

Table 25.18 Combinations of Available Bit Rate Settings

(3/3)

Valid Bit Rate Setting					FCNnCMBTCTL Register Setting Value		Sampling Point (unit: %)
DBT Length	SUNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMB TS1LG[3:0]	FCNnCMB TS2LG[2:0]	
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7 [Note]	1	2	2	2	0011	001	71.4
7 [Note]	1	4	1	1	0100	000	85.7
6 [Note]	1	1	2	2	0010	001	66.7
6 [Note]	1	3	1	1	0011	000	83.3

Note: Setting of the DBT value of 7 or less is valid only when the value of the FCNnCMBRPRS register is other than 00H.

Caution: The values in Table 25.18, Combinations, do not guarantee proper operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

25.13.2 Representative Examples of Baud Rate Settings

Table 25.19 Representative Examples of Baud Rate Settings (f_{CANMOD} = 20 MHz)

(1/2)

Baud Rate Value (unit: kbps)	Division Ratio of FCNnCMB RPRS Register	FCNnCMB RPRS Register Value	Valid Bit Rate Setting (unit: TQ)					FCNnCMBTCTL Register Setting		Sampling Point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMB TS1LG[3:0]	FCNnCMB TS2LG[2:0]	
1000	1	00000000	20	1	3	8	8	1010	111	60.0
1000	1	00000000	20	1	5	7	7	1011	110	65.0
1000	1	00000000	20	1	7	6	6	1100	101	70.0
1000	1	00000000	20	1	7	4	4	1101	100	75.0
1000	1	00000000	20	1	9	5	5	1110	011	80.0
1000	1	00000000	20	1	11	4	4	1111	010	85.0
1000	2	00000001	10	1	1	4	4	0100	011	60.0
1000	2	00000001	10	1	3	3	3	0101	010	70.0
1000	2	00000001	10	1	5	2	2	0110	001	80.0
1000	2	00000001	10	1	7	1	1	0111	000	90.0
500	2	00000001	20	1	3	8	8	1010	111	60.0
500	2	00000001	20	1	5	7	7	1011	110	65.0
500	2	00000001	20	1	7	6	6	1100	101	70.0
500	2	00000001	20	1	7	4	4	1101	100	75.0
500	2	00000001	20	1	9	5	5	1110	011	80.0
500	2	00000001	20	1	11	4	4	1111	010	85.0
500	4	00000011	10	1	1	4	4	0100	011	60.0
500	4	00000011	10	1	3	3	3	0101	010	70.0
500	4	00000011	10	1	5	2	2	0110	001	80.0
500	4	00000011	10	1	7	1	1	0111	000	90.0
250	4	00000011	20	1	5	7	7	1011	110	65.0
250	4	00000011	20	1	7	6	6	1100	101	70.0
250	4	00000011	20	1	9	5	5	1101	100	75.0
250	4	00000011	20	1	11	4	4	1110	011	80.0
250	8	00000111	10	1	3	3	3	0101	010	70.0
250	8	00000111	10	1	5	2	2	0110	001	80.0
125	8	00000111	20	1	5	7	7	1011	110	65.0
125	8	00000111	20	1	7	6	6	1100	101	70.0
125	8	00000111	20	1	9	5	5	1101	100	75.0
125	8	00000111	20	1	11	4	4	1110	011	80.0
125	16	00001111	10	1	3	3	3	0101	010	70.0
125	16	00001111	10	1	5	2	2	0110	001	80.0
100	10	00001001	20	1	5	7	7	1011	110	65.0
100	10	00001001	20	1	7	6	6	1100	101	70.0
100	10	00001001	20	1	7	4	4	1101	100	75.0
100	10	00001001	20	1	9	5	5	1110	011	80.0
100	20	00010011	10	1	3	3	3	0101	010	70.0
100	20	00010011	10	1	5	2	2	0110	001	80.0

Table 25.19 Representative Examples of Baud Rate Settings (f_{CANMOD} = 20 MHz)

(2/2)

Baud Rate Value (unit: kbps)	Division Ratio of FCNnCMB RPRS Register	FCNnCMB RPRS Register Value	Valid Bit Rate Setting (unit: TQ)					FCNnCMBTCTL Register Setting		Sampling Point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMB TS1LG[3:0]	FCNnCMB TS2LG[2:0]	
83.3	10	00001001	24	1	7	8	8	1110	111	66.7
83.3	10	00001001	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	20	1	5	7	7	1011	110	65.0
83.3	12	00001011	20	1	7	6	6	1100	101	70.0
83.3	12	00001011	20	1	9	5	5	1101	100	75.0
83.3	12	00001011	20	1	11	4	4	1110	011	80.0
83.3	16	00001111	15	1	4	5	5	1000	100	66.7
83.3	16	00001111	15	1	6	4	4	1001	011	73.3
83.3	16	00001111	15	1	8	3	3	1010	010	80.0
83.3	16	00001111	15	1	10	2	2	1011	001	86.7
83.3	24	00010111	10	1	3	3	3	0101	010	70.0
83.3	24	00010111	10	1	5	2	2	0110	001	80.0
83.3	30	00011101	8	1	3	2	2	0100	001	75.0
83.3	30	00011101	8	1	5	1	1	0101	000	87.5
33.3	25	00011000	24	1	7	8	8	1110	111	66.7
33.3	25	00011000	24	1	9	7	7	1111	110	70.8
33.3	30	00011101	20	1	5	7	7	1011	110	65.0
33.3	30	00011101	20	1	7	6	6	1100	101	70.0
33.3	30	00011101	20	1	9	5	5	1101	100	75.0
33.3	30	00011101	20	1	11	4	4	1110	011	80.0
33.3	33	00100000	18	1	3	7	7	1001	110	61.1
33.3	33	00100000	18	1	5	6	6	1010	101	66.7
33.3	33	00100000	18	1	7	5	5	1011	100	72.2
33.3	33	00100000	18	1	9	4	4	1100	011	77.8
33.3	33	00100000	18	1	11	3	3	1101	010	83.3
33.3	33	00100000	18	1	13	2	2	1110	001	88.9
33.3	40	00100111	15	1	4	5	5	1000	100	66.7
33.3	40	00100111	15	1	6	4	4	1001	011	73.3
33.3	40	00100111	15	1	8	3	3	1010	010	80.0
33.3	40	00100111	15	1	10	2	2	1011	001	86.7
33.3	50	00110001	12	1	3	4	4	0110	011	66.7
33.3	50	00110001	12	1	5	3	3	0111	010	75.0
33.3	50	00110001	12	1	7	2	2	1000	001	83.3
33.3	60	00111011	10	1	3	3	3	0101	010	70.0
33.3	60	00111011	10	1	5	2	2	0110	001	80.0

Caution: The values in Table 25.19, Representative Examples of Baud Rate Settings (fCANMOD = 20 MHz), do not guarantee proper operation of the network system.
Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

25.14 Operation of the CAN Controller

The processing procedure described in this section is recommended for operating the FCN.

Refer to the recommended processing procedure when developing the program.

25.14.1 Initialization

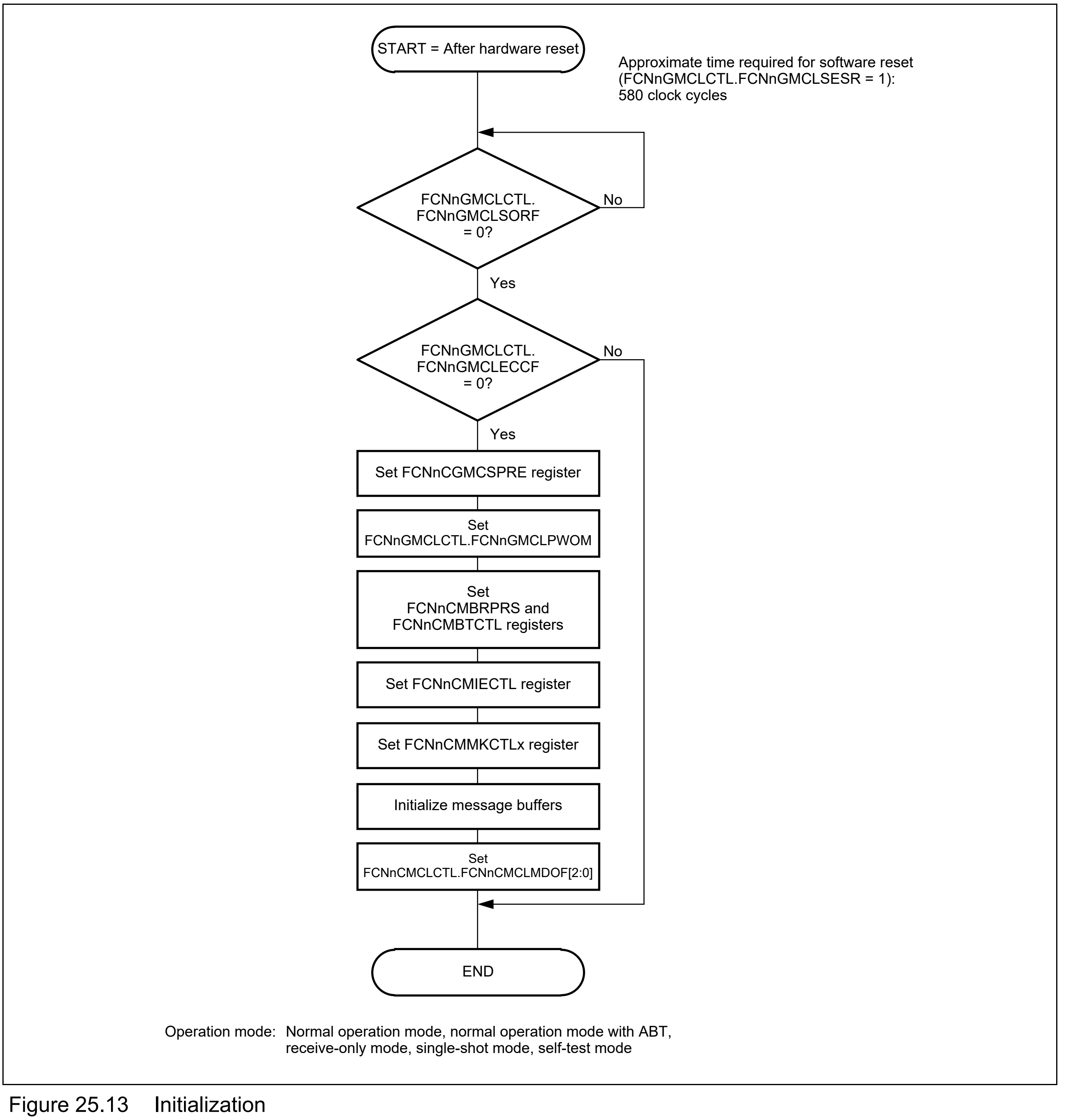


Figure 25.13 Initialization

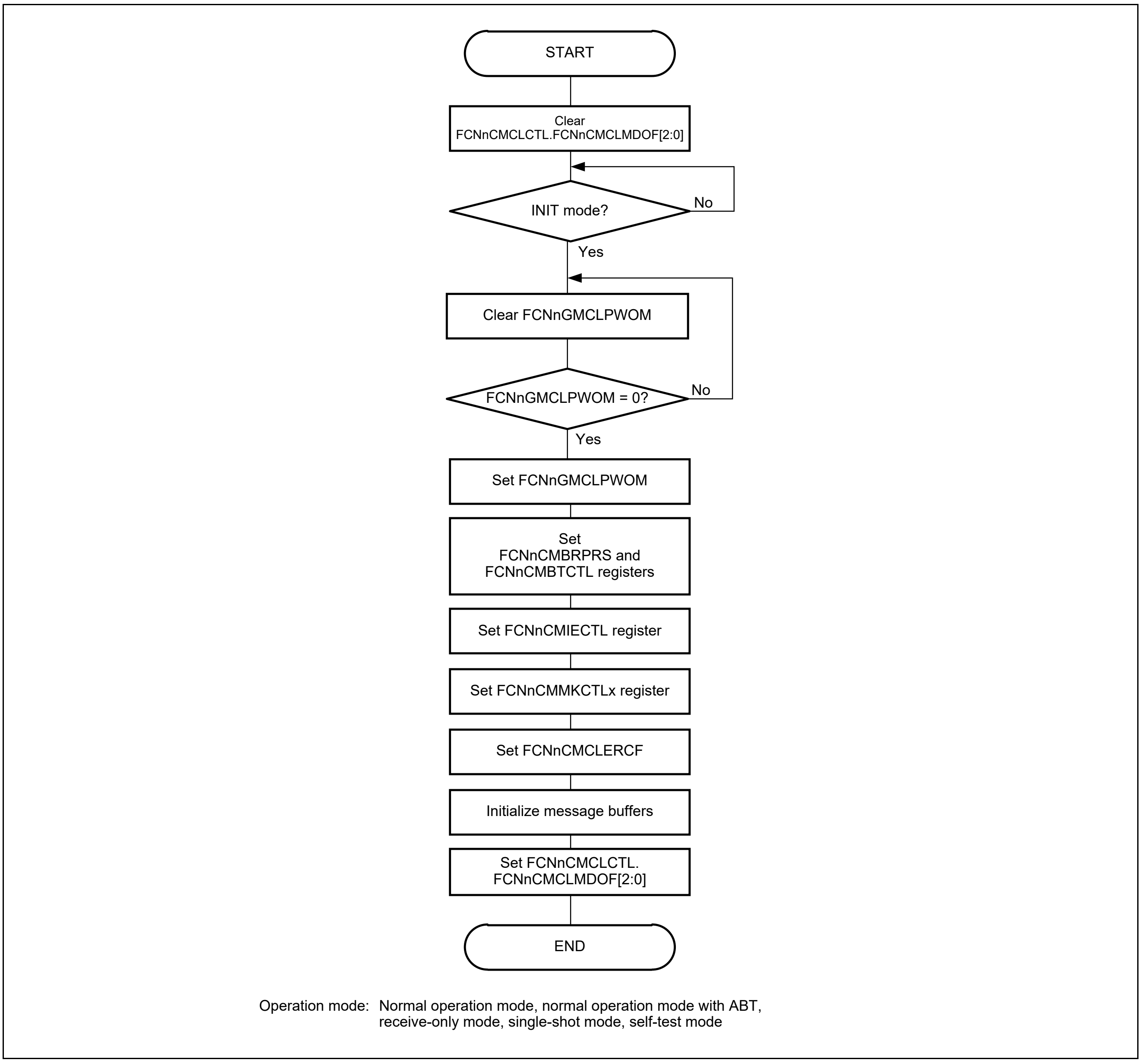
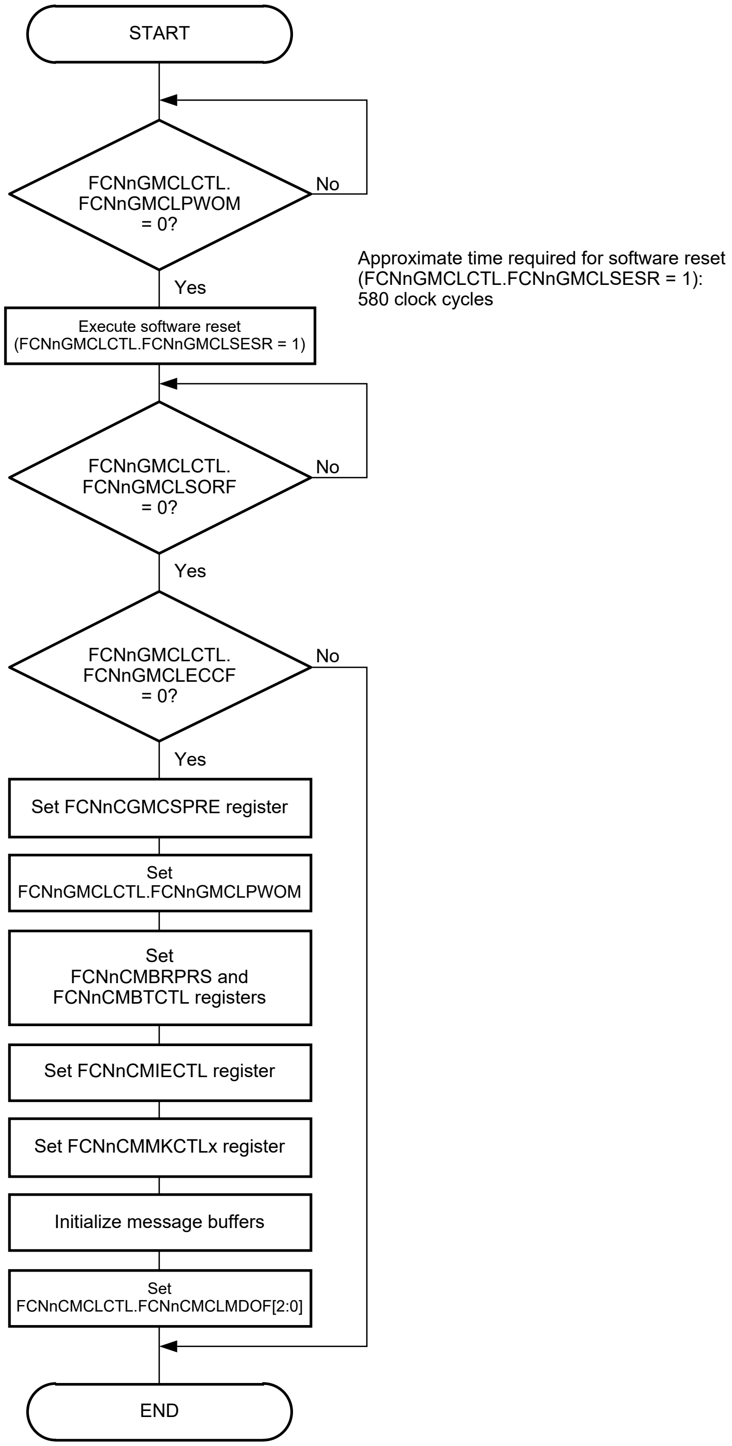


Figure 25.14 Re-initialization without Using the Software Reset

Caution: To clear the error counter (by setting FCNnCMCLERCF) during re-initialization, do so in either of the following states.

- **In the initialization mode following the start of the FCN module (by setting FCNnGMCLPWOM while FCNnGMCLPWOM = 0)**
- **In the initialization mode following clearing of all transmission requests according to the transmission abort processing described in Figure 25.24, Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed), during the operation mode (clear all the transmission requests according to the transmission abort processing described in Figure 25.25, Transmission Abort Processing (in Normal Operation Mode with ABT) – Repeat Option for Aborted Message, in the normal operation mode with ABT).**



Operation mode: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 25.15 Re-Initialization with Software Reset

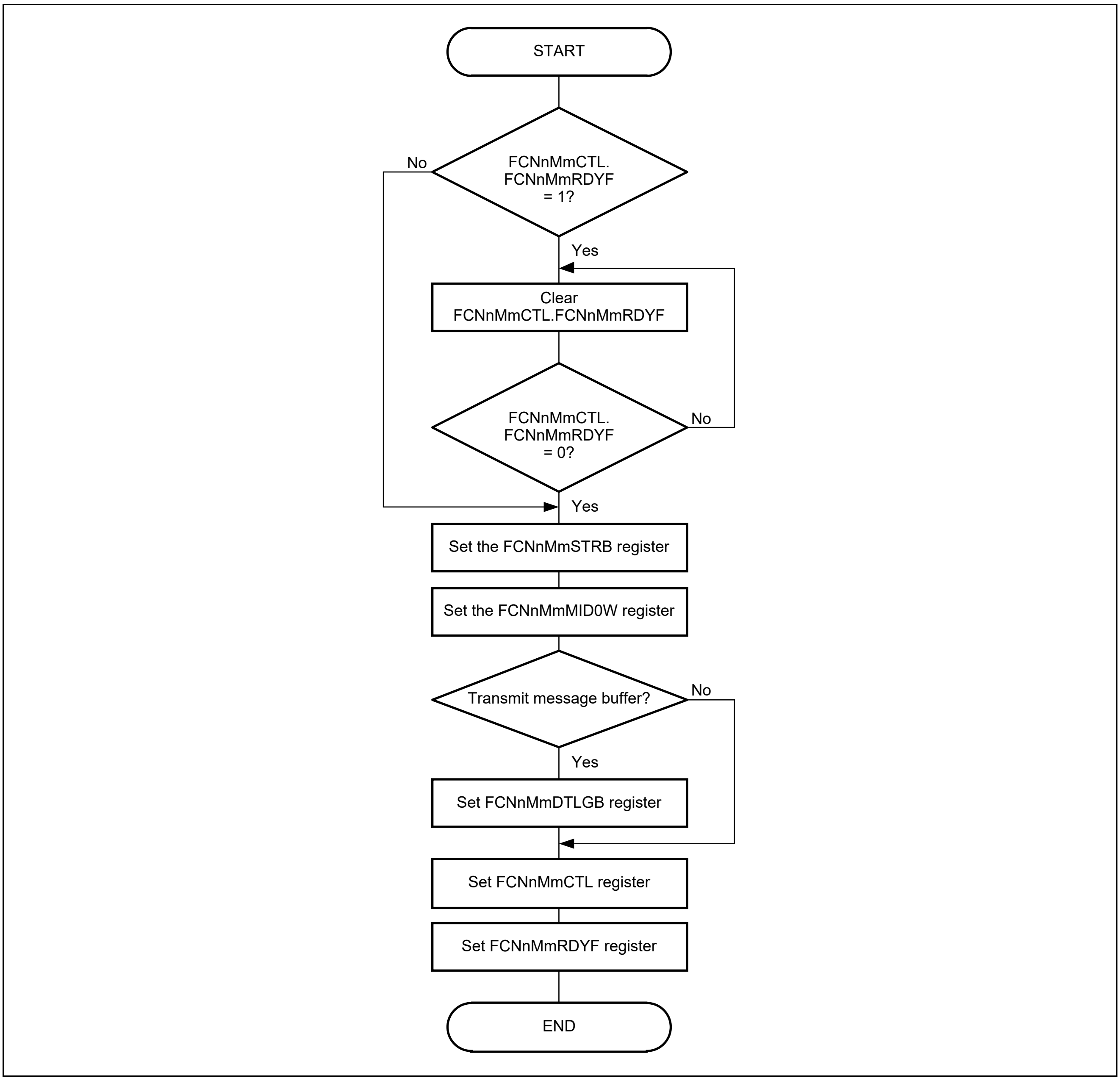


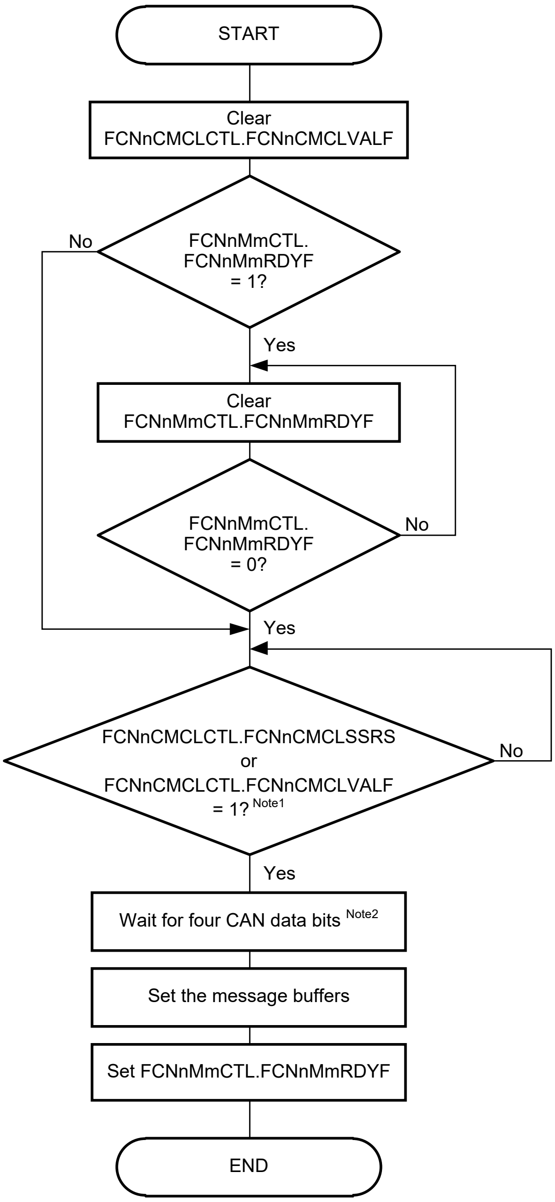
Figure 25.16 Message Buffer Initialization

Cautions 1. Before a message buffer is initialized, FCNnMmCTL.FCNnMmRDYF must be cleared.

2. Make the following settings for message buffers not used by the application.

- **Clear FCNnMmRDYF, FCNnMmTRQF, and FCNnMmDTNF bits of the FCNnMmCTL register to 0.**
- **Clear FCNnMmSTRB.FCNnMmSSAM to 0.**

Figure 25.17, Message Buffer Redefinition during Reception, shows the processing for a receive message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001B to 1000B).



Notes 1. Since FCNmMmCTL.FCNnMmRDFY must be set after a message is completely received, confirm that the message is being received, .

2. Wait for four CAN data bits so that message buffers are not redefined while saving the received messages.

Figure 25.17 Message Buffer Redefinition during Reception

Figure 25.18, Message Buffer Redefinition during Transmission, shows the processing for a transmit message buffer during transmission (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000B).

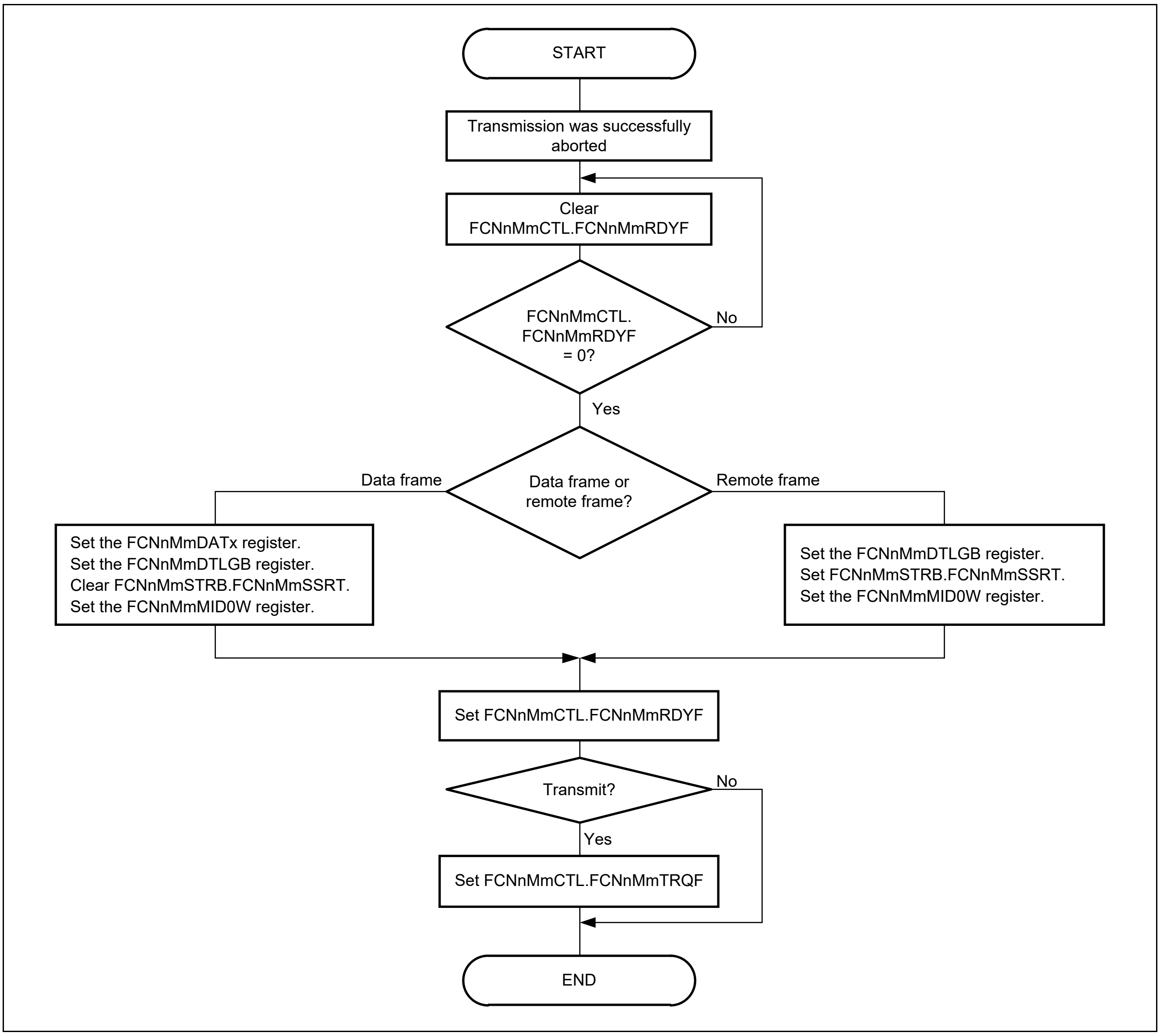


Figure 25.18 Message Buffer Redefinition during Transmission

25.14.2 Message Transmission

Figure 25.19, Message Transmit Processing, shows the processing for a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000B).

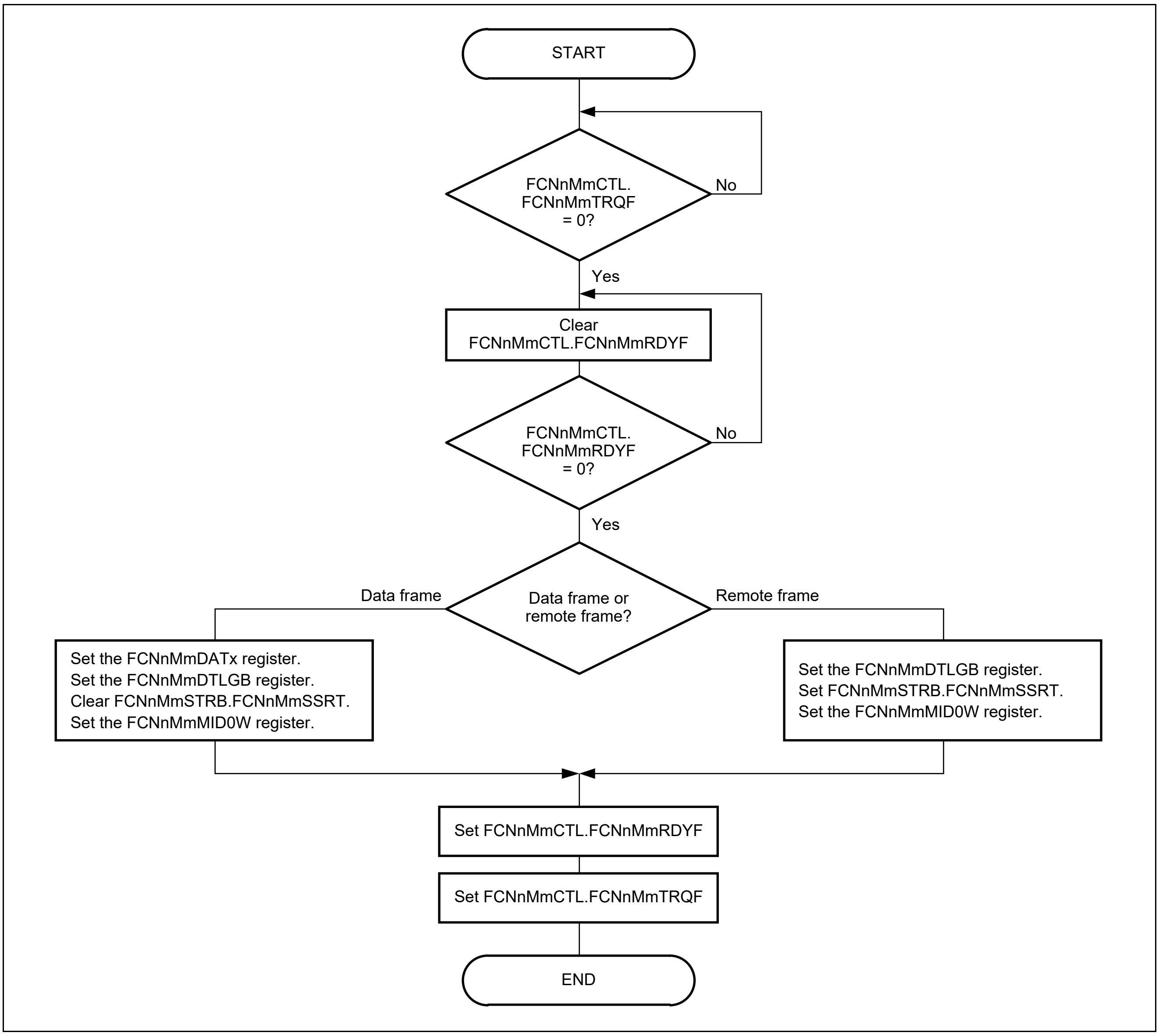


Figure 25.19 Message Transmit Processing

Cautions 1. FCNnMmCTL.FCNnMmTRQF should be set after FCNnMmCTL.FCNnMmRDYF is set.

2. FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF should not be set at the same time.

Figure 25.20, ABT Message Transmit Processing, shows the processing for a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000B)

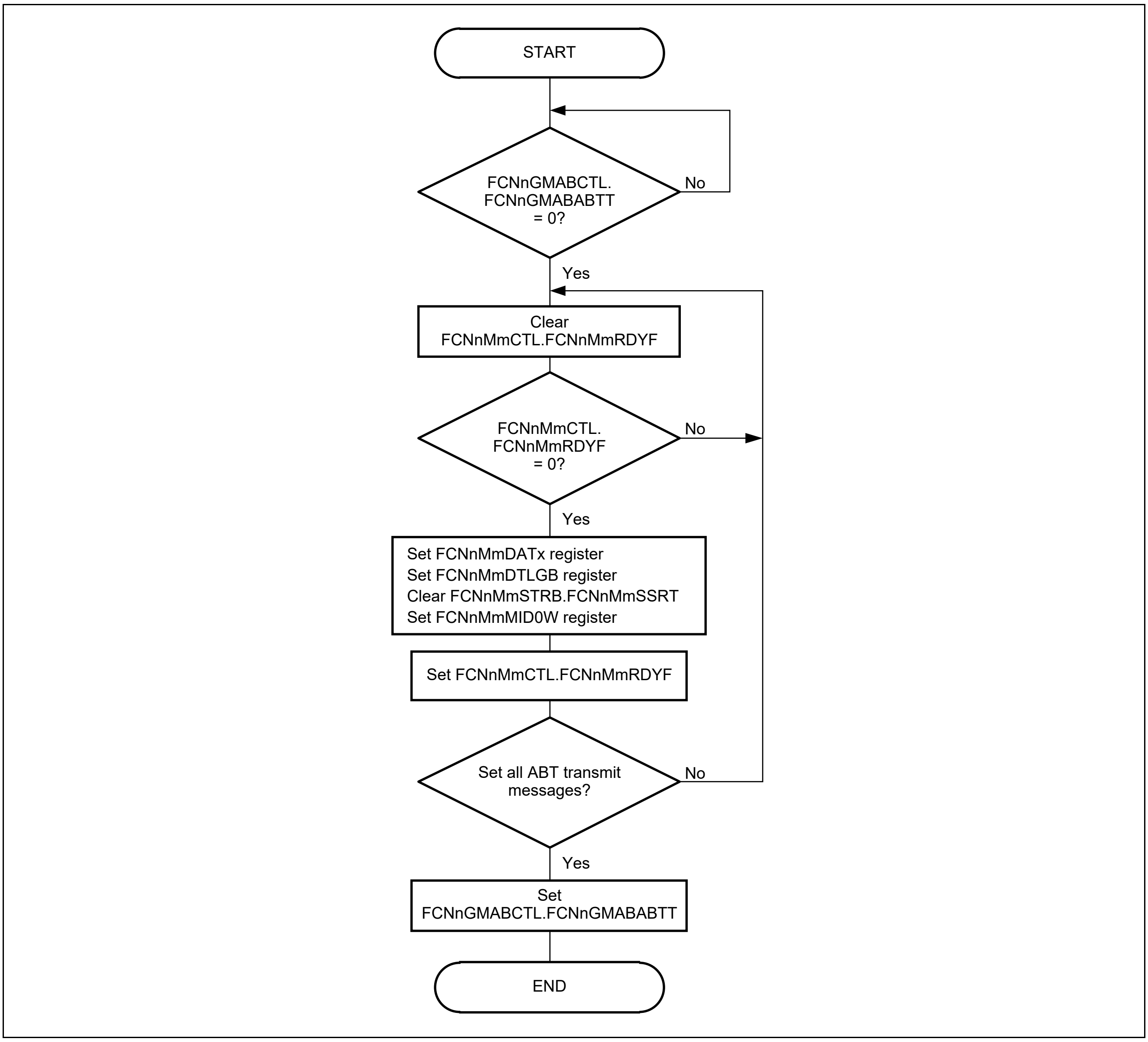


Figure 25.20 ABT Message Transmit Processing

Remark: This processing (normal operation mode with ABT) can only be applied to message buffers that are available in ABT mode. For the message buffers other than the ABT message buffers, see Figure 25.19, Message Transmit Processing.

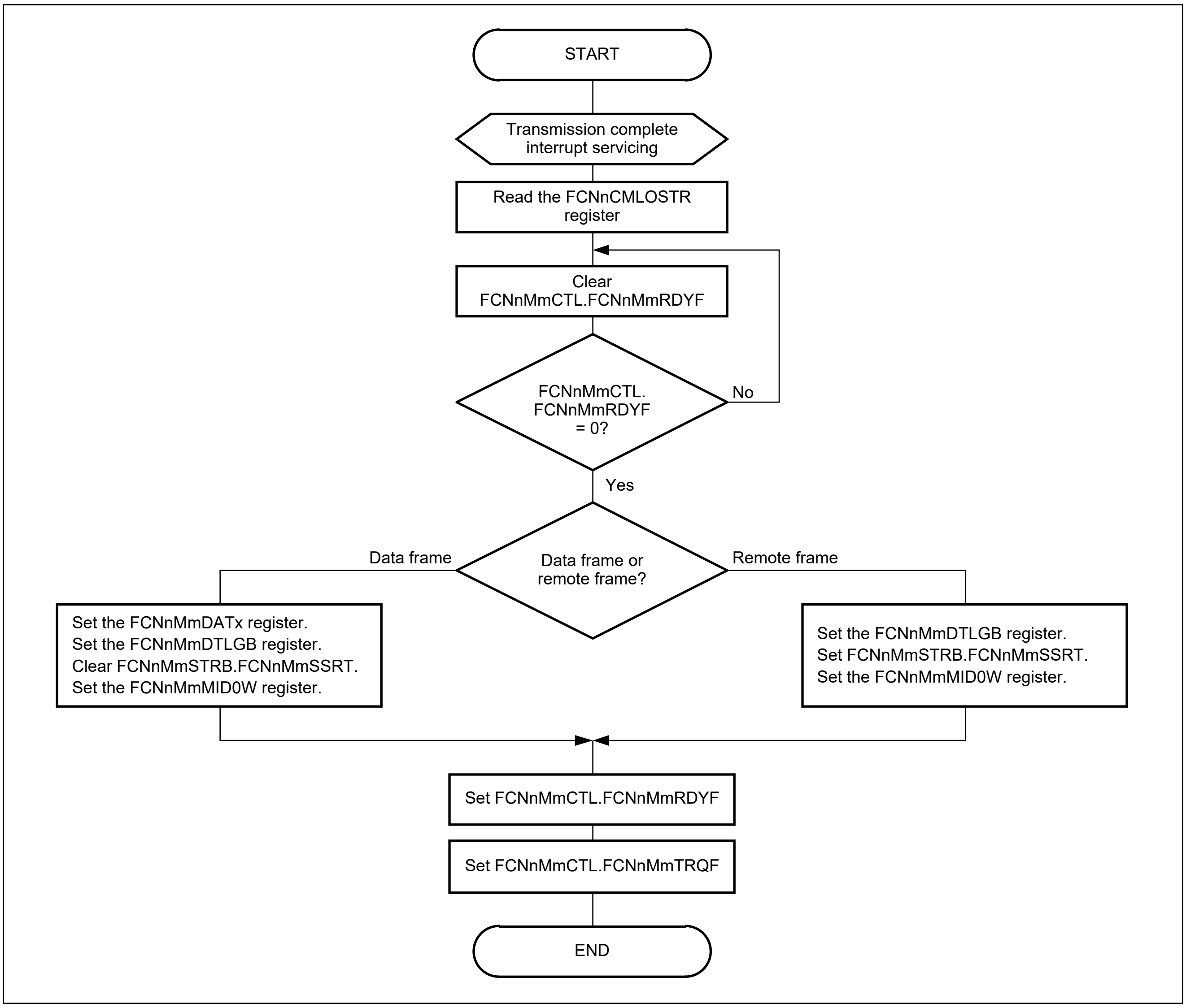


Figure 25.21 Transmission via Interrupt (Using FCNnCMLOSTR Register)

Cautions 1. FCNnMmCTL.FCNnMmTRQF should be set after FCNnMmCTL.FCNnMmRDYF is set.

2. FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF should not be set at the same time.

Remark: Since pending sleep mode may be executed, the FCNnGMCLSSMO flag must be checked at the beginning and at the end of the interrupt routine to check the access to the message buffers as well as TX history list registers. If FCNnGMCLSSMO is found to have been cleared at the time of checking, set FCNnGMCLSSMO again, and then discard the actions and results of the processing and execute the processing again. It is recommended to cancel any sleep mode requests before processing TX interrupts.

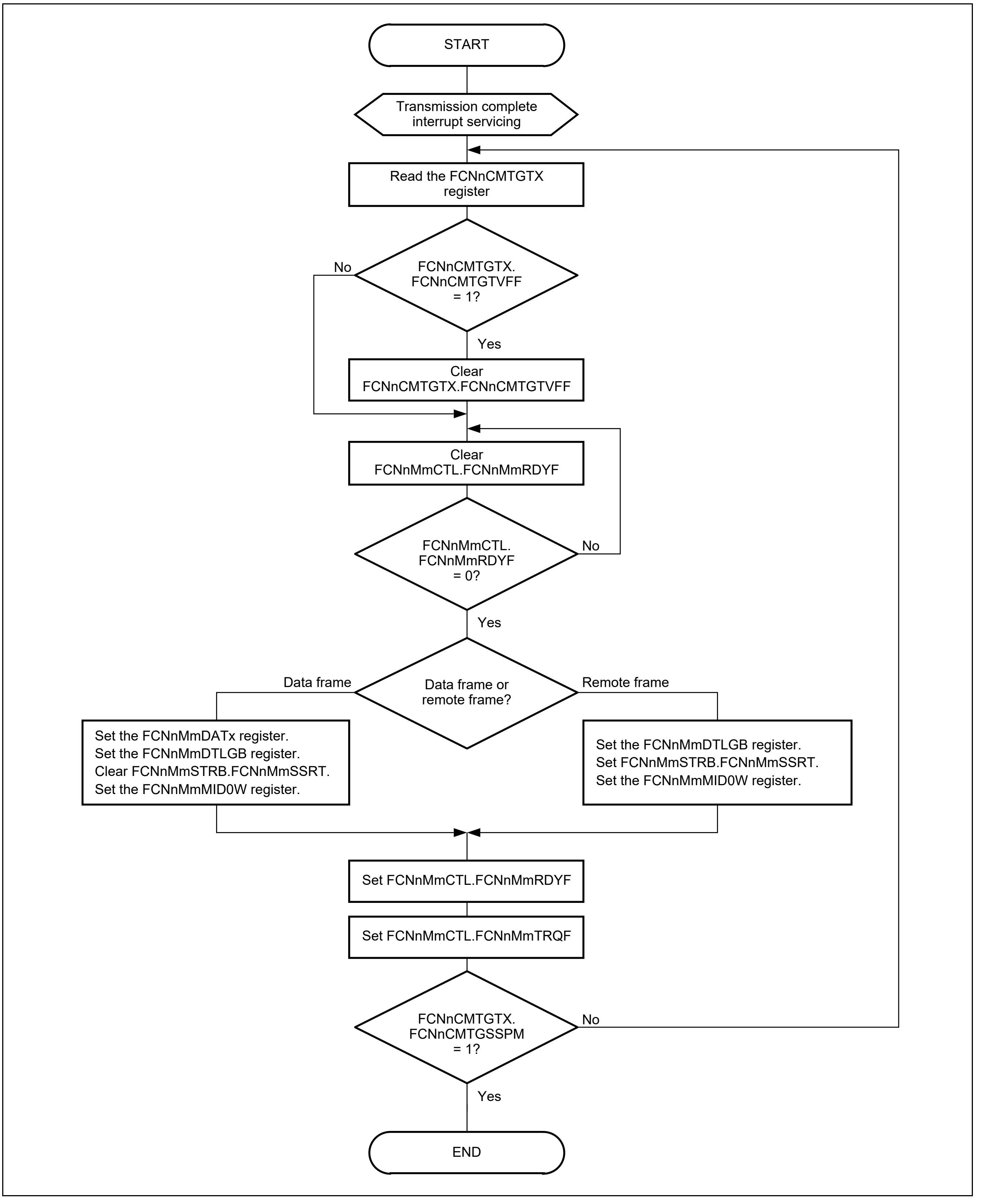


Figure 25.22 Transmission via Interrupt (Using FCNnCMTGTX Register)

Cautions 1. FCNnMmCTL.FCNnMmTRQF should be set after FCNnMmCTL.FCNnMmRDYF is set.

2. FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF should not be set at the same time.

Remarks 1. Since pending sleep mode may be executed, the FCNnGMCLSSMO flag must be checked at the beginning and at the end of the interrupt routine to check the access to the message buffers as well as TX history list registers. If FCNnGMCLSSMO is found to have been cleared at the time of checking, set FCNnGMCLSSMO again, and then discard the actions and results of the processing and execute the processing again. It is recommended to cancel any sleep mode requests before processing TX interrupts.

2. Once FCNnCMTGTX.FCNnCMTGTVFF is set, the transmit history list becomes inconsistent. Consider checking all configured transmit buffers to confirm completed transmissions.

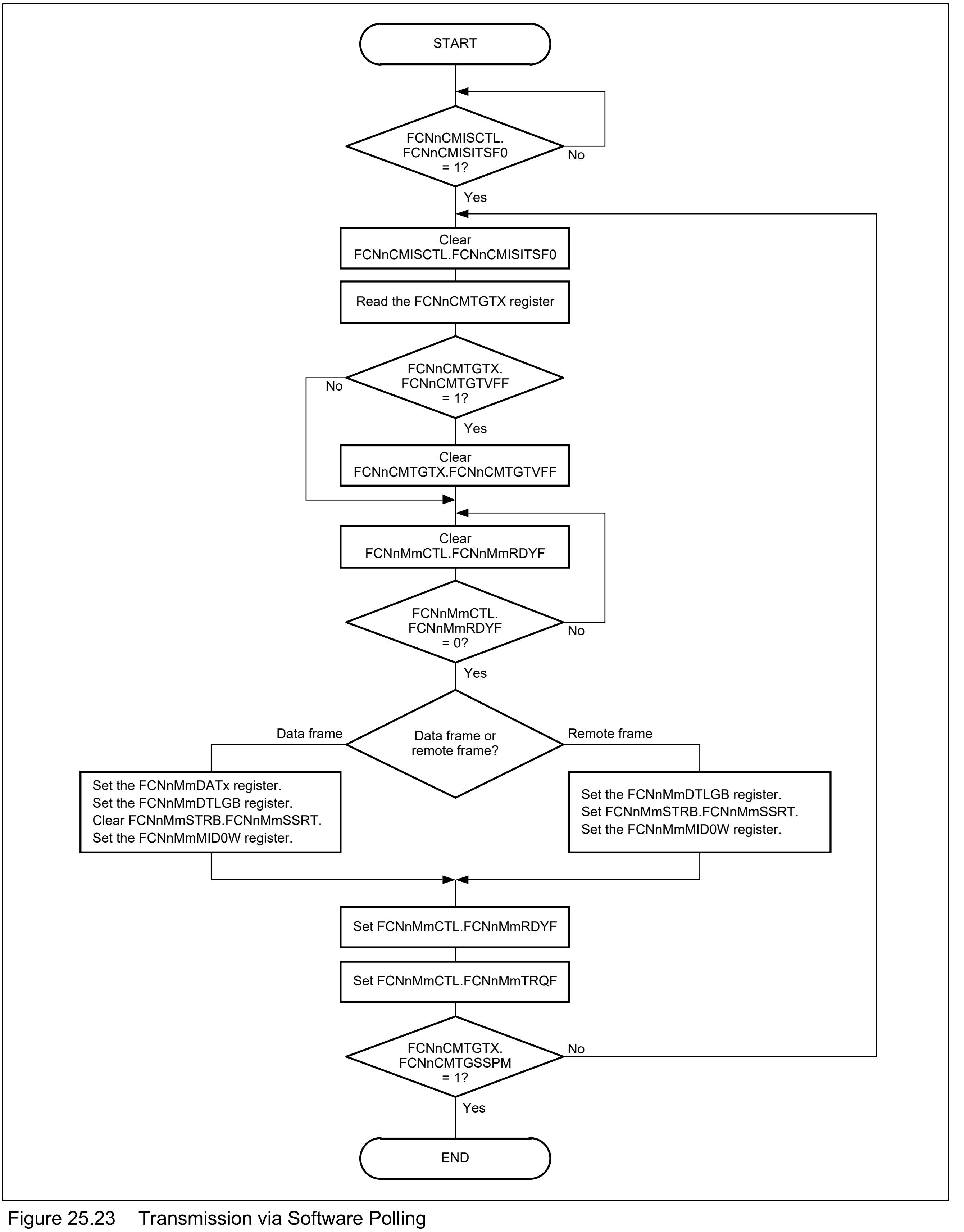


Figure 25.23 Transmission via Software Polling

Cautions 1. FCNnMmCTL.FCNnMmTRQF should be set after FCNnMmCTL.FCNnMmRDYF is set.

2. FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF should not be set at the same time.

Remarks 1. Since pending sleep mode may be executed, the FCNnGMCLSSMO flag must be checked at the beginning and at the end of the interrupt routine to check the access to the message buffers as well as TX history list registers. If FCNnGMCLSSMO is found to have been cleared at the time of checking, set FCNnGMCLSSMO again, and then discard the actions and results of the processing and execute the processing again.

2. Once FCNnCMTGTX.FCNnCMTGTVFF is set, the transmit history list becomes inconsistent. Consider checking all configured transmit buffers to confirm completed transmissions.

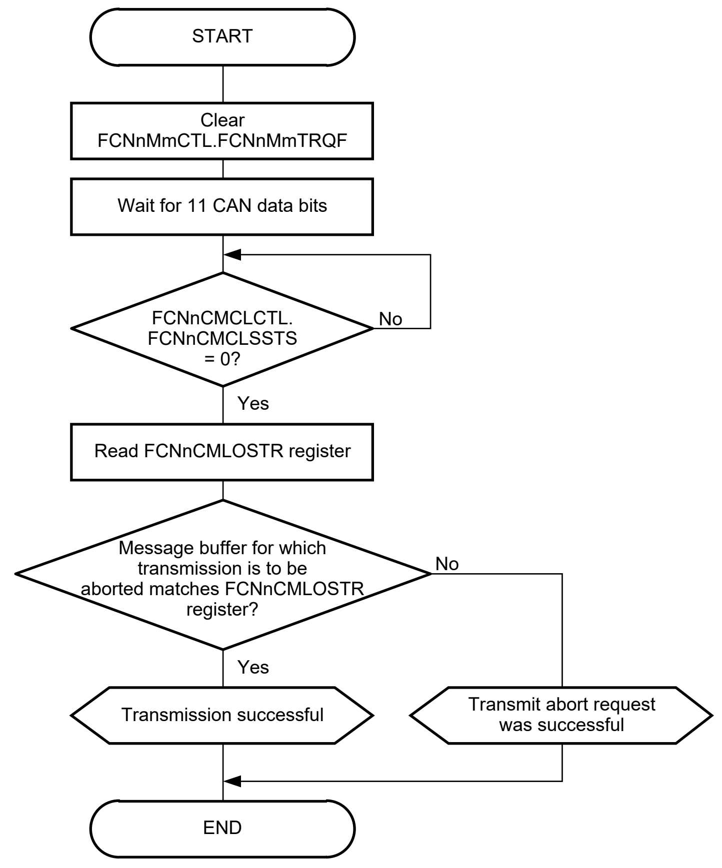


Figure 25.24 Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed)

Cautions 1. To issue a request for aborting the transmission, clear FCNnMmCTL.FCNnMmTRQF instead of FCNnMmCTL.FCNnMmRDYF.

2. Before issuing a request for transition to sleep mode, confirm that no transmission request which uses this processing remains.

3. FCNnCMCLCTL.FCNnCMCLSSTS can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute any new transmission request including transmission from the other message buffers while transmission abort processing is in progress.

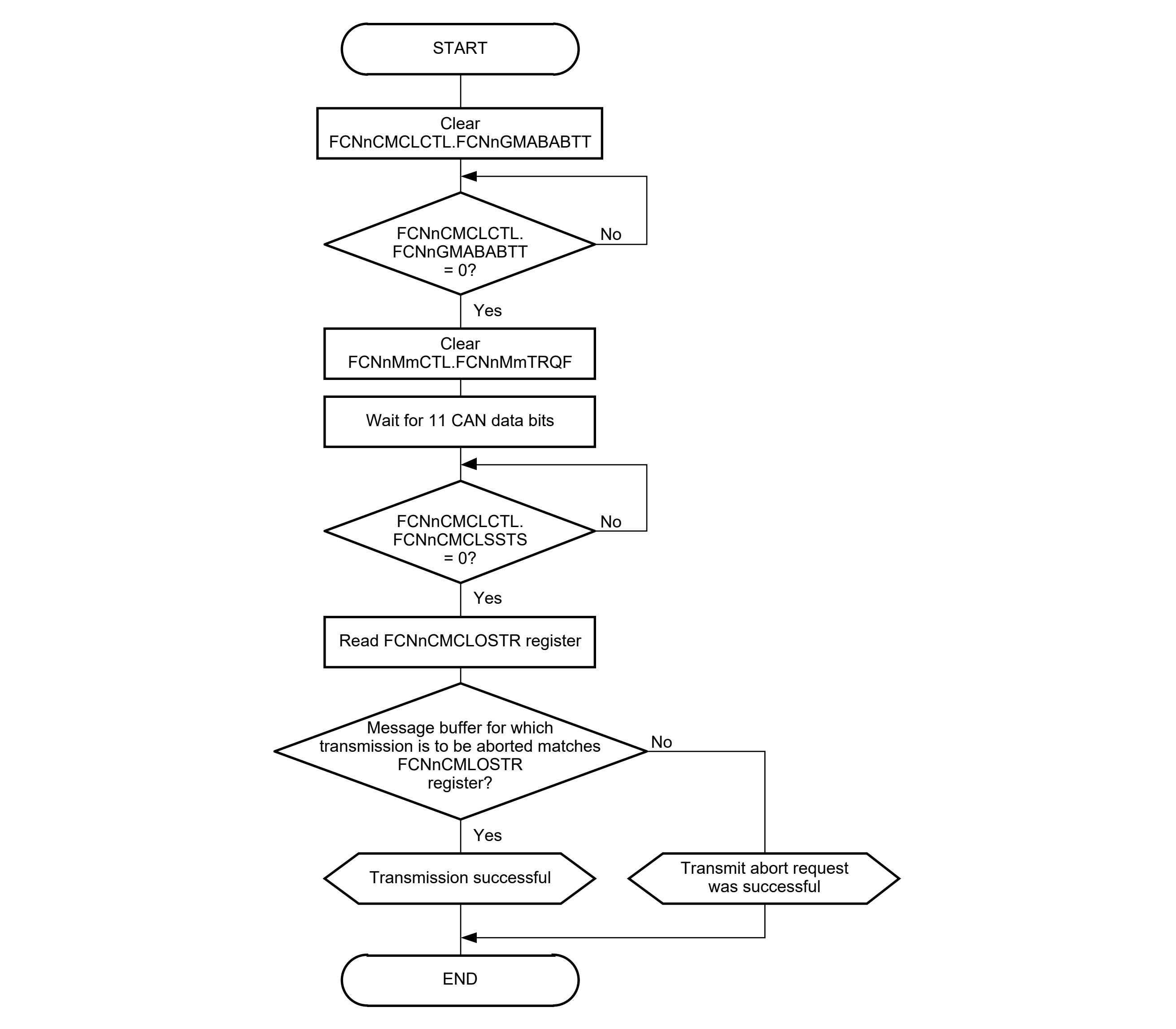


Figure 25.25 Transmission Abort Processing (in Normal Operation Mode with ABT) – Repeat Option for Aborted Message

Cautions 1. To issue a request for aborting the transmission, clear FCNnMmCTL.FCNnMmTRQF instead of FCNnMmCTL.FCNnMmRDYF.

2. Before issuing a request for transition to sleep mode, confirm that no transmission request which uses this processing remains.

3. FCNnCMCLCTL.FCNnCMCLSSTS can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute any new transmission request including transmission from the other message buffers while transmission abort processing is in progress.

Figure 25.26, ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1), shows the processing which does not skip resumption of message transmission that was stopped when transmission from an ABT message buffer was aborted.

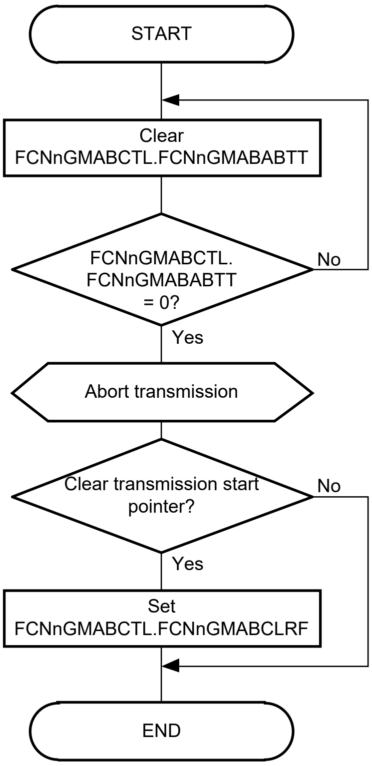


Figure 25.26 ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1)

Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Issue a request for transition to FCN sleep mode/FCN stop mode after FCNnGMABCTL.FCNnGMABABTT has been cleared (after ABT mode has been stopped) following the procedure shown in Figure 25.26, ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1), or Figure 25.27, ABT Transmission Request Abort Processing (In Normal Operation Mode with ABT) (2). When clearing a transmission request in the area other than the ABT area, follow the procedure shown in Figure 25.24, Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed).

Figure 25.27, ABT Transmission Request Abort Processing (In Normal Operation Mode with ABT) (2), shows the processing which does not skip resumption of message transmission that was stopped when transmission from an ABT message buffer was aborted.

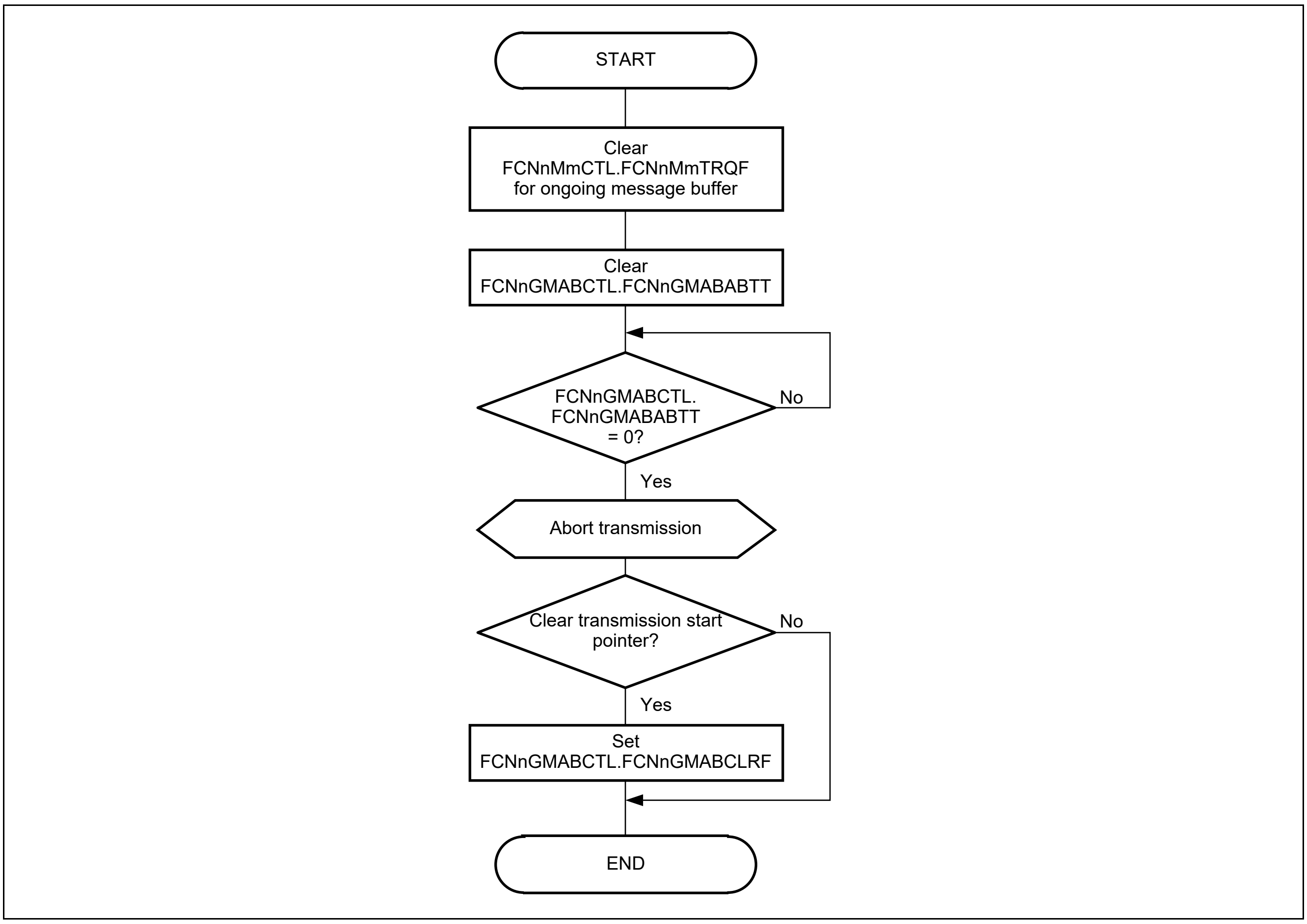


Figure 25.27 ABT Transmission Request Abort Processing (In Normal Operation Mode with ABT) (2)

Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Issue a request for transition to FCN sleep mode/FCN stop mode after FCNnGMABCTL.FCNnGMABABTT has been cleared (after ABT mode has been stopped) following the procedure shown in Figure 25.26, ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1), or Figure 25.27, ABT Transmission Request Abort Processing (In Normal Operation Mode with ABT) (2). When clearing a transmission request in the area other than the ABT area, follow the procedure shown in Figure 25.24, Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed).

Figure 25.28 shows the processing on ABT mode using the Transmit Abort functionality (transmission complete flag). The box "Transmission successfully aborted" indicates confirming whether transmission has been successfully aborted by checking the FCNnMmTCPF flag within the ABT message buffers.

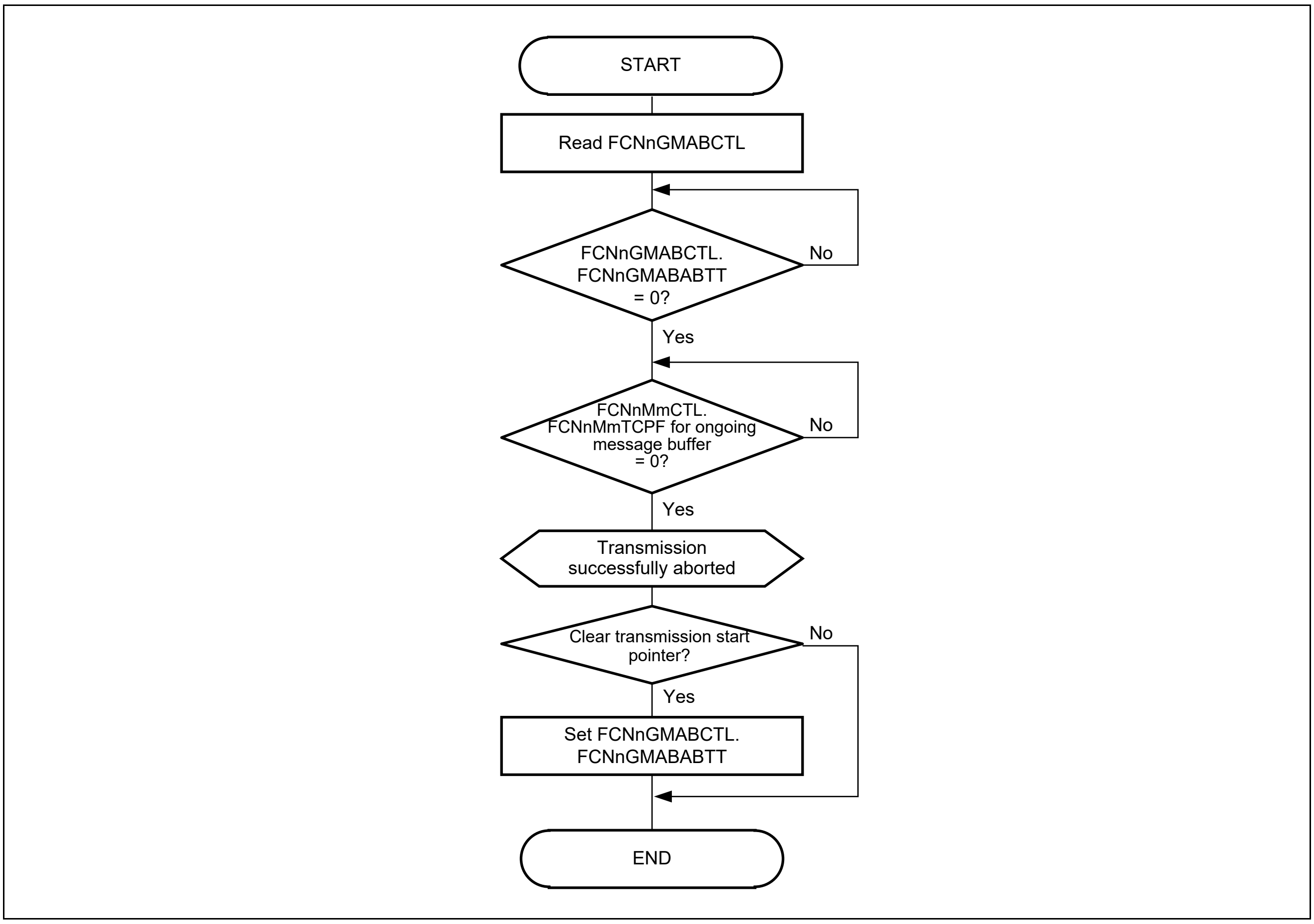


Figure 25.28 ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) with Transmission Complete Flag

Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Issue a request for transition to FCN sleep mode/FCN stop mode after FCNnGMABCTL.FCNnGMABABTT has been cleared (after ABT mode has been stopped) following the procedure shown in Figure 25.26, ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1), or Figure 25.27, ABT Transmission Request Abort Processing (In Normal Operation Mode with ABT) (2). When clearing a transmission request in the area other than the ABT area, follow the procedure shown in Figure 25.24, Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed).

Remark: All ABT may be transmitted completely even if ABT transmission abort processing is performed successfully. In such cases, you can check which message has been transmitted.

Figure 25.29, Transmission Abort Processing with Transmission Abort Interrupt and Transmission Complete Flag, shows the processing when using the Transmit Abort functionality (transmission abort interrupt).

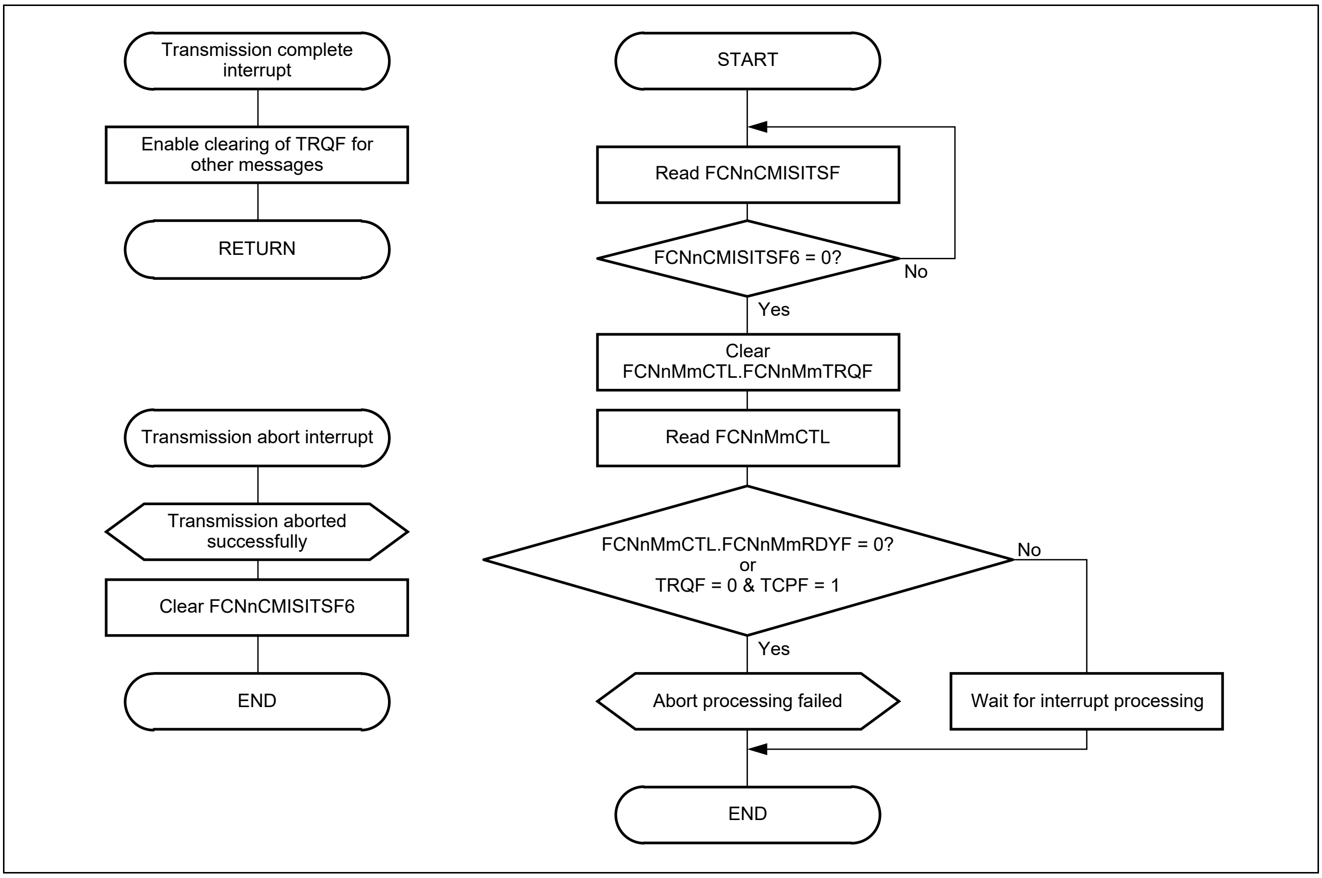


Figure 25.29 Transmission Abort Processing with Transmission Abort Interrupt and Transmission Complete Flag

Remark: FCNnMmRDYF=0 is judged considering the case where FCNnMmRDYF is cleared during transmission completion processing in response to interrupts.

Cautions 1. Transmission must be aborted by clearing FCNnMmTRQF rather than by clearing FCNnMmRDYF.

2. Before issuing a request for sleep, make sure that the transmission request has completely ended according to this flow.

3. Do not update the messages subject to transmission abort processing (FCNnMmRDYF or FCNnMmTRQF is set) while it is in progress by transmission complete interrupt processing, etc.

4. Do not clear FCNnMmTRQF for other message buffers while the transmission is being aborted.

5. If you set the ID with a lower priority than the original ID after transmission abort processing, wait for at least one frame after clearing FCNnMmTRQF before sending a transmission request.

6. Always read FCNnMmTRQF and FCNnMmTCPF at a time.

Figure 25.30, Transmission Abort Processing with Transmission Complete Flag, shows the processing when using the Transmit Abort functionality (transmission complete flag FCNnMmTCPF).

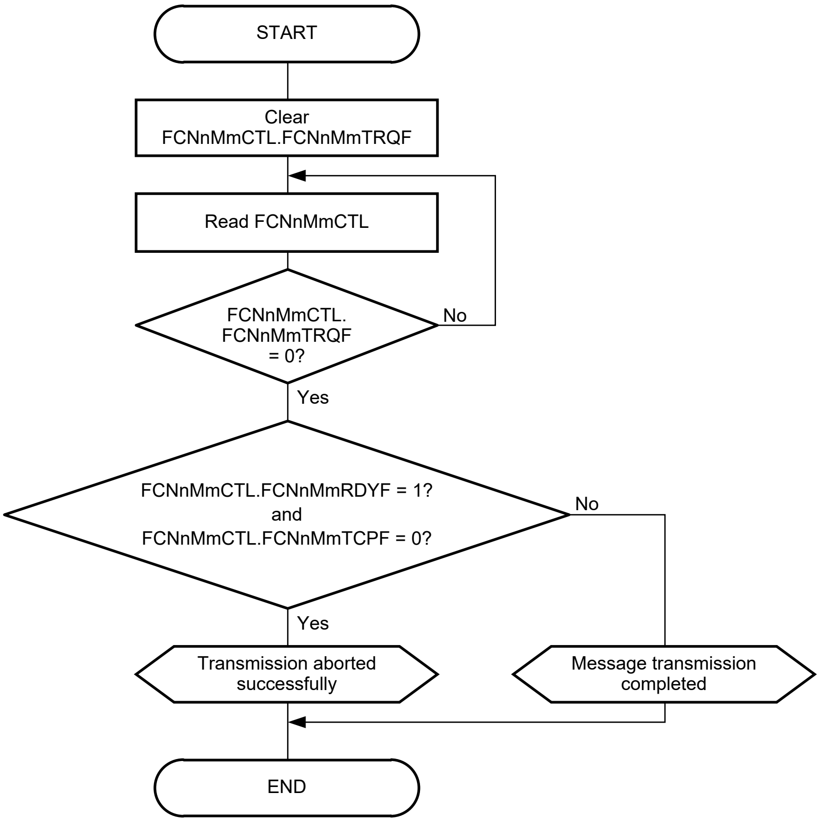


Figure 25.30 Transmission Abort Processing with Transmission Complete Flag

Remark: FCNnMmRDYF=0 is judged considering the case where FCNnMmRDYF is cleared during transmission completion processing in response to interrupts.

Cautions 1. Transmission must be aborted by clearing FCNnMmTRQF rather than by clearing FCNnMmRDYF.

2. Before issuing a request for sleep, make sure that the transmission request has completely ended according to this flow.

3. Do not update the messages subject to transmission abort processing (FCNnMmRDYF or FCNnMmTRQF is set) while it is in progress by transmission complete interrupt processing, etc.

4. If you set the ID with a lower priority than the original ID after transmission abort processing, wait for at least one frame after clearing FCNnMmTRQF before sending a transmission request.

5. Always read FCNnMmTRQF and FCNnMmTCPF at a time.

25.14.3 Message Reception

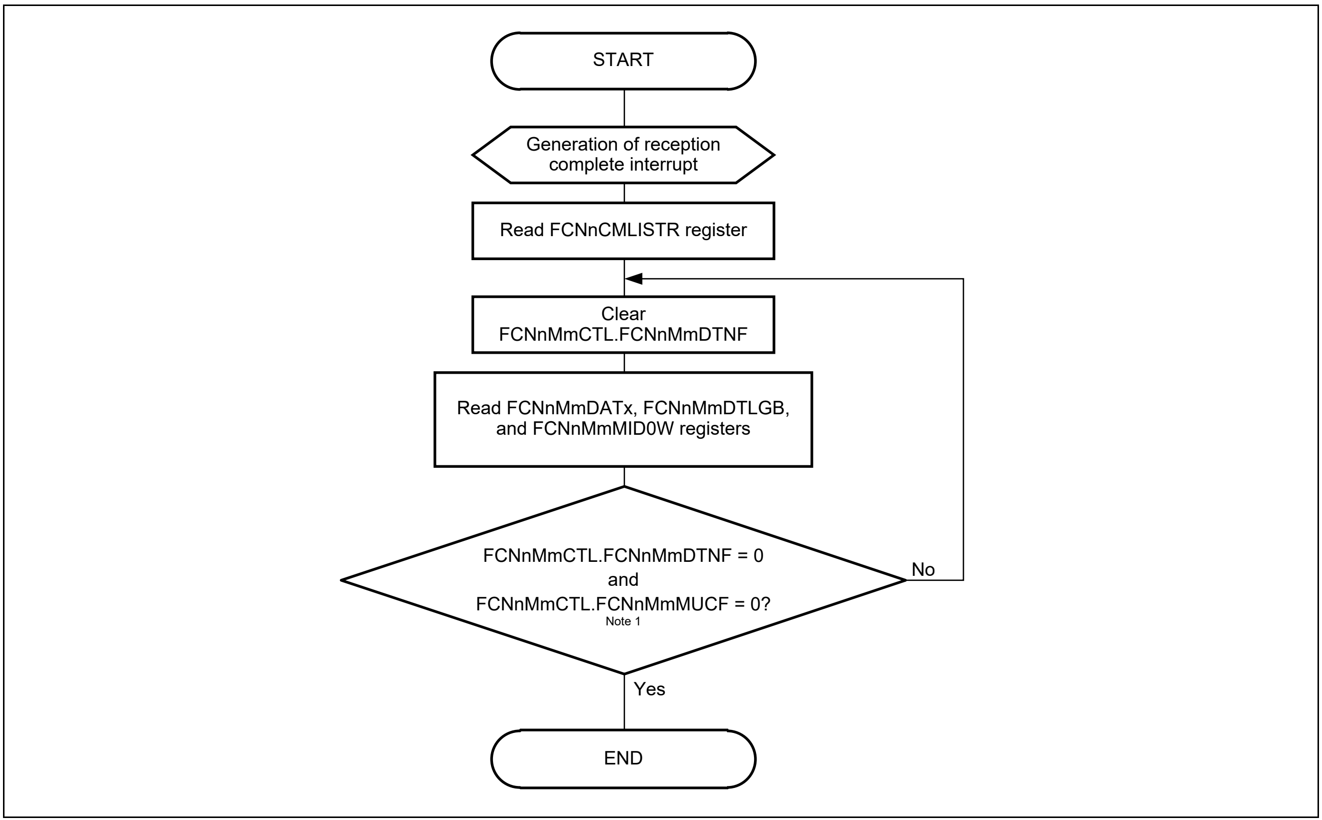


Figure 25.31 Reception via Interrupt (Using FCNnCMLISTR Register)

Remarks 1. Check the FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits via a single read access.

2. Since pending sleep mode may be executed, the FCNnGMCLSSMO flag must be checked at the beginning and at the end of the interrupt routine to check the access to the message buffers as well as reception history list registers. If FCNnGMCLSSMO is found to have been cleared at the time of checking, set FCNnGMCLSSMO again, and then discard the actions and results of the processing and execute the processing again. It is recommended to cancel any sleep mode requests before processing RX interrupts.

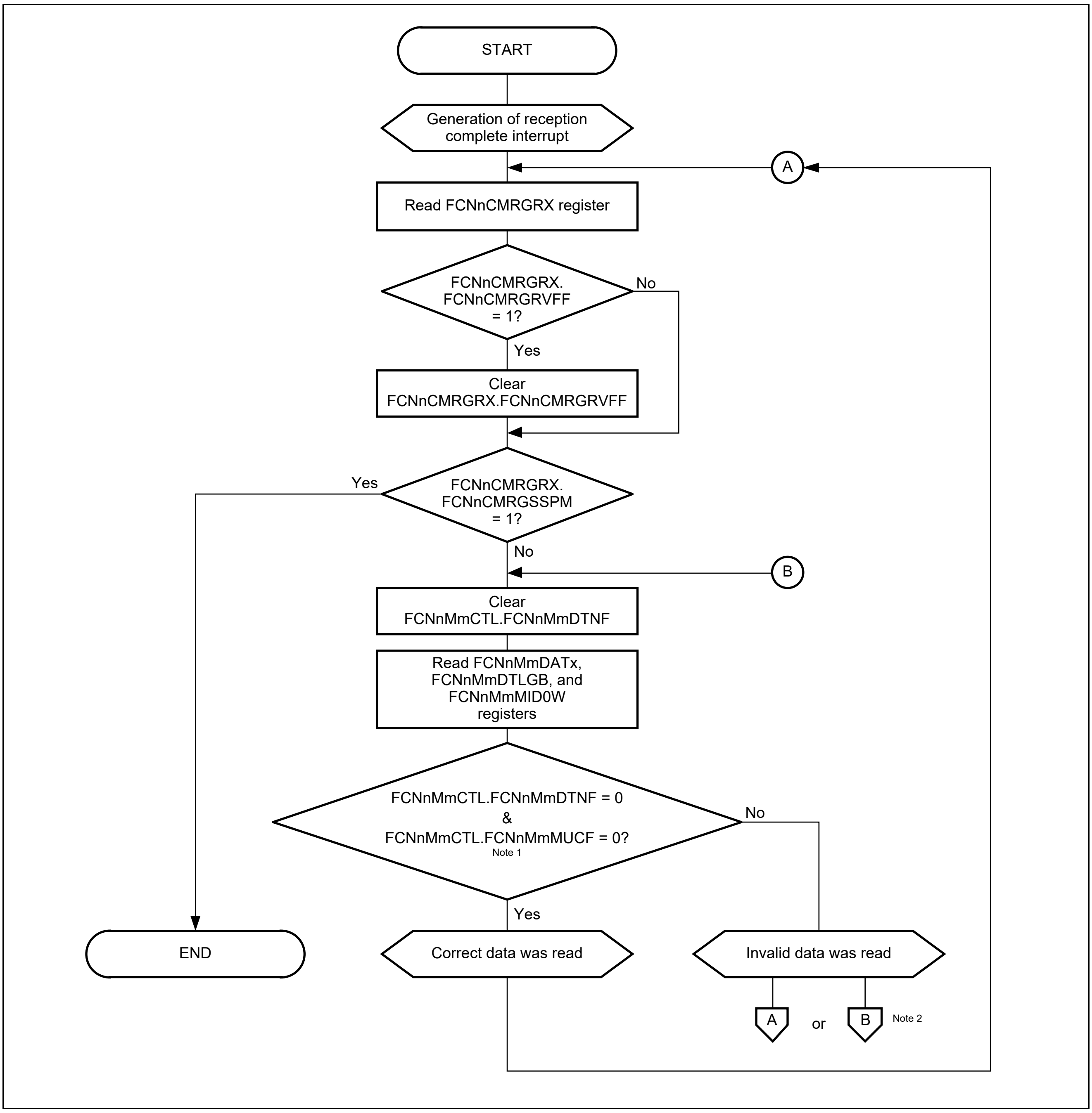


Figure 25.32 Reception via Interrupt (Using FCNnCMRGRX Register)

Remarks 1. Check the FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits via a single read access.

2. There are two ways of processing depending on the target for processing by the application:

- **Way A: A message is not processed on the current path, but on the next path, depending on the latest timing at which it is processed in response to the next reception interrupt. Other messages are processed earlier.**
- **Way B: A message is processed on the current path, and the loop enters the wait state in the current message. Other messages are processed later.**

3. Since pending sleep mode may be executed, the FCNnGMCLSSMO flag must be checked at the beginning and at the end of the interrupt routine to check the access to the message buffers as well as reception history list registers. If FCNnGMCLSSMO is found to have been cleared at the time of checking, set FCNnGMCLSSMO again, and then discard the actions and results of the processing and execute the processing again. It is recommended to cancel any sleep mode requests before processing RX interrupts.

4. Once FCNnCMRGRX.FCNnCMRGRVFF is set, the receive history list becomes inconsistent. Consider checking all configured receive buffers to confirm the reception.

5. Instead of the processing shown in Figure 25.32, Reception via Interrupt (Using FCNnCMRGRX Register), the processing shown in Figure 25.33, Another Way of Reception via Interrupt (Using FCNnCMRGRX Register), can be used.

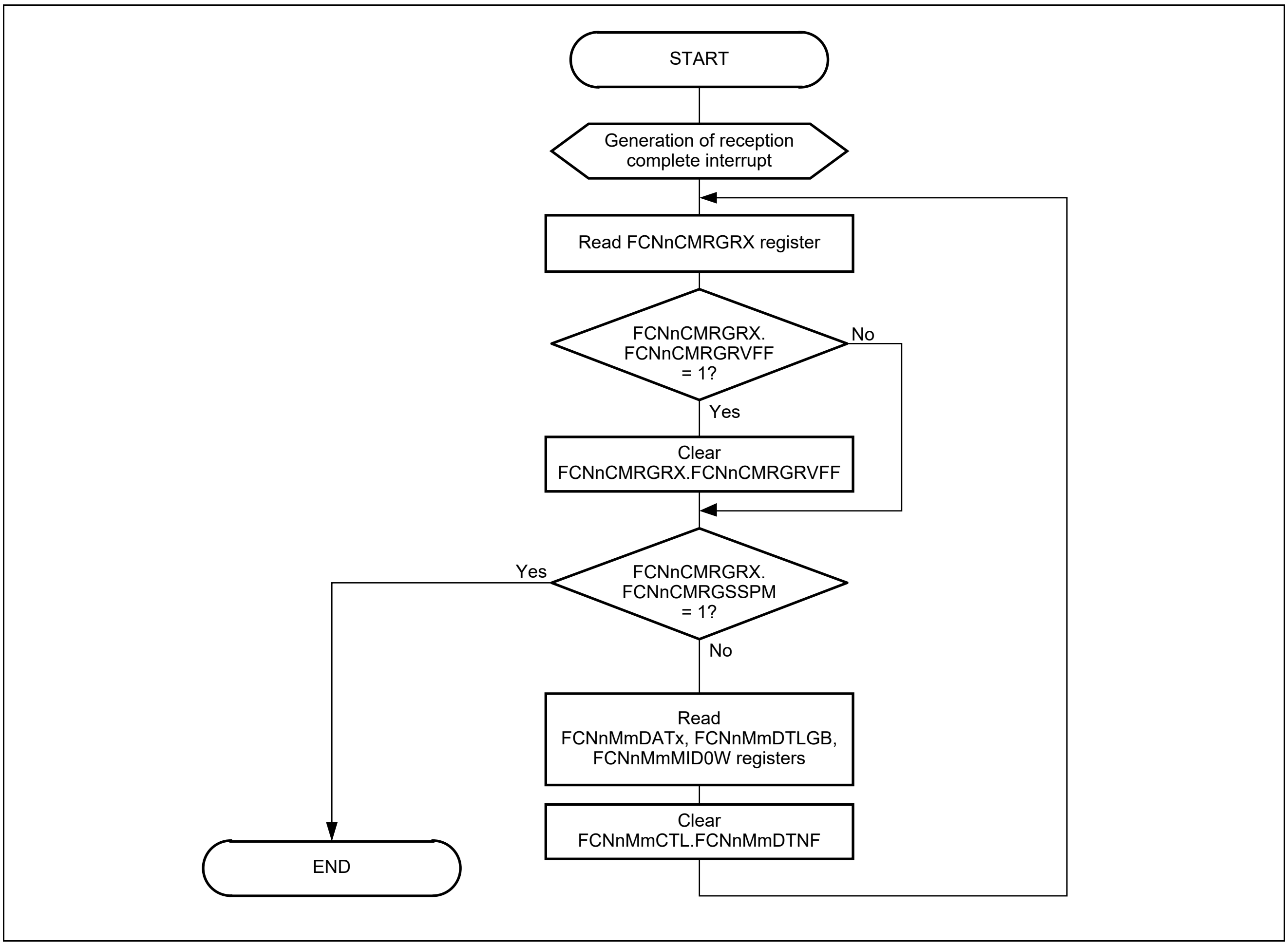


Figure 25.33 Another Way of Reception via Interrupt (Using FCNnCMRGRX Register)

Remarks 1. Since pending sleep mode may be executed, the FCNnGMCLSSMO flag must be checked at the beginning and at the end of the interrupt routine to check the access to the message buffers as well as reception history list registers. If FCNnGMCLSSMO is found to have been cleared at the time of checking, set FCNnGMCLSSMO again, and then discard the actions and results of the processing and execute the processing again.
It is recommended to cancel any sleep mode requests before processing RX interrupts.

2. Once FCNnCMRGRX.FCNnCMRGRVFF is set, the receive history list becomes inconsistent. Consider checking all configured receive buffers to confirm the reception.

3. If this flow is used, the application cannot obtain the latest received data. However, due to a low amount of processing, interrupt loads will be reduced.

4. Do not use overwriting (FCNnMmSTRB.FCNnMmSSOW = 1) with this flow, as this may lead to a loss of data consistency.

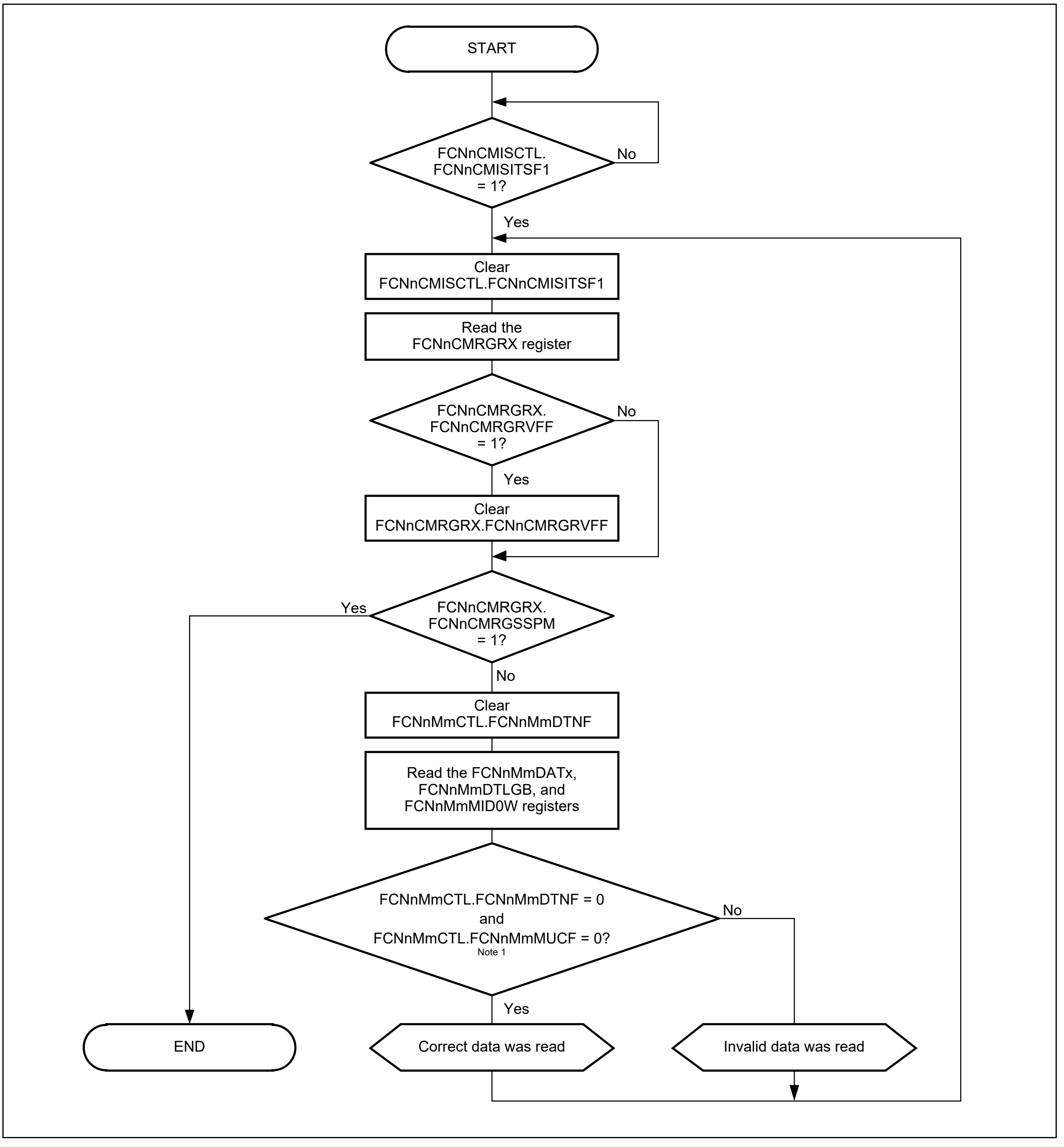


Figure 25.34 Reception via Software Polling

Remarks 1. Check the FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits via a single read access.

2. Since pending sleep mode may be executed, the FCNnGMCLSSMO flag must be checked at the beginning and at the end of the interrupt routine to check the access to the message buffers as well as reception history list registers. If FCNnGMCLSSMO is found to have been cleared at the time of checking, set FCNnGMCLSSMO again, and then discard the actions and results of the processing and execute the processing again.

3. Once FCNnCMRGRX.FCNnCMRGRVFF is set, the receive history list becomes inconsistent. Consider checking all configured receive buffers to confirm the reception.

25.14.4 Power Safe Mode

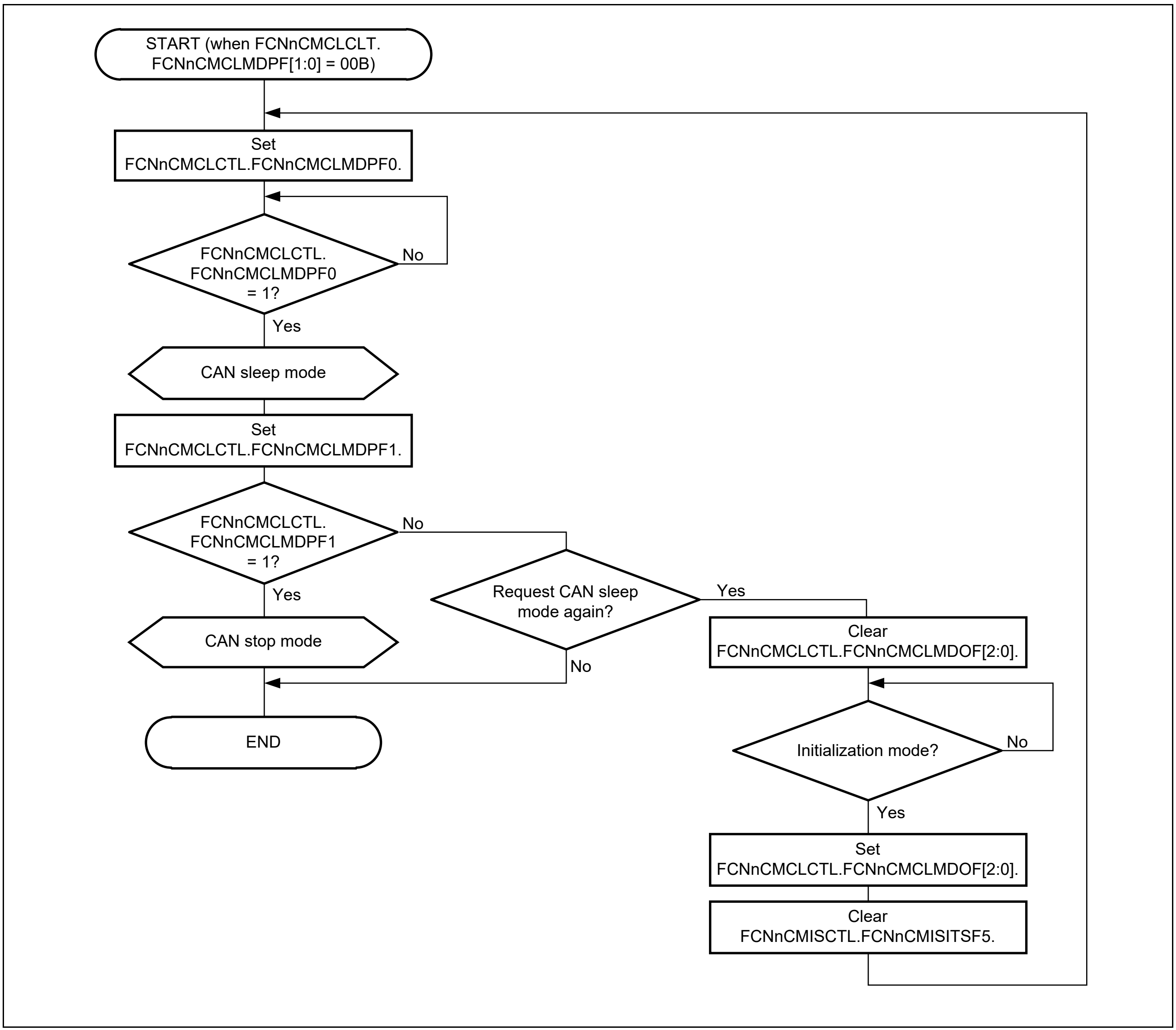


Figure 25.35 Setting FCN Sleep Mode/Stop Mode

Caution: To abort the transmission before issuing a request for transition to FCN sleep mode, perform transmission abort processing according to the previous flowcharts.

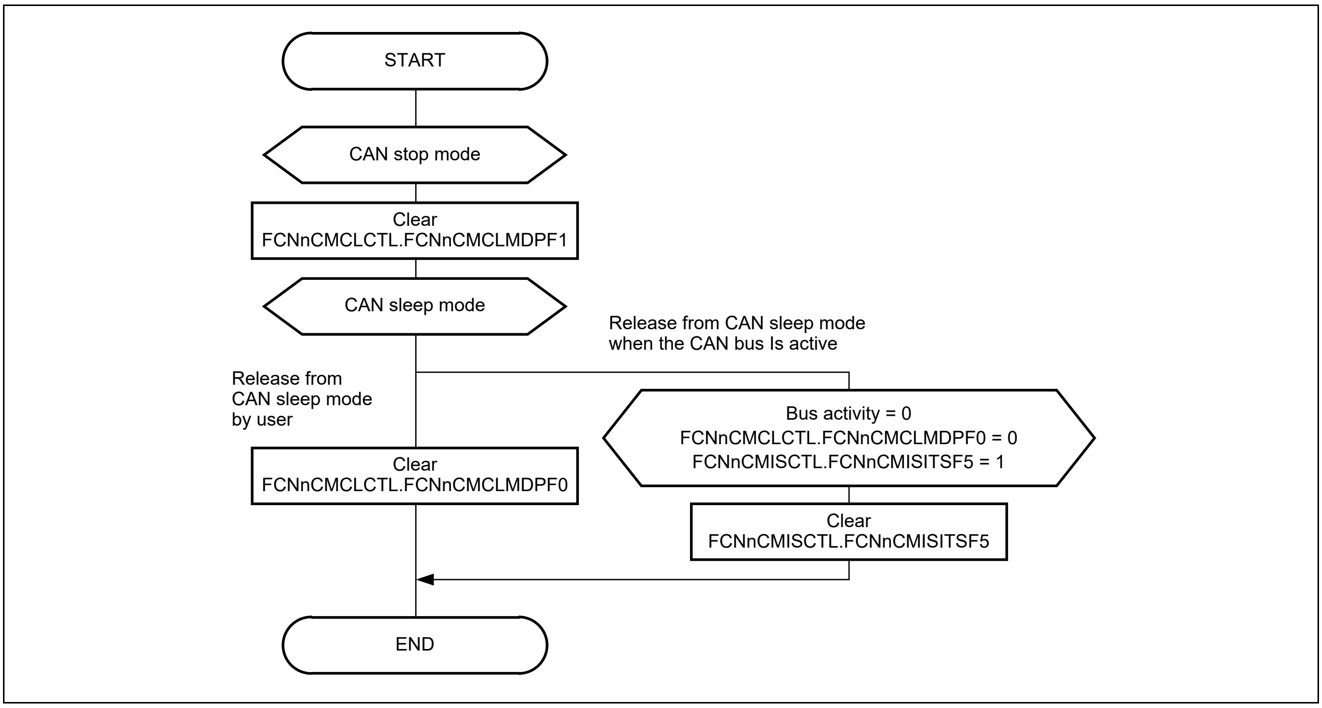


Figure 25.36 Release from FCN Sleep/Stop Mode

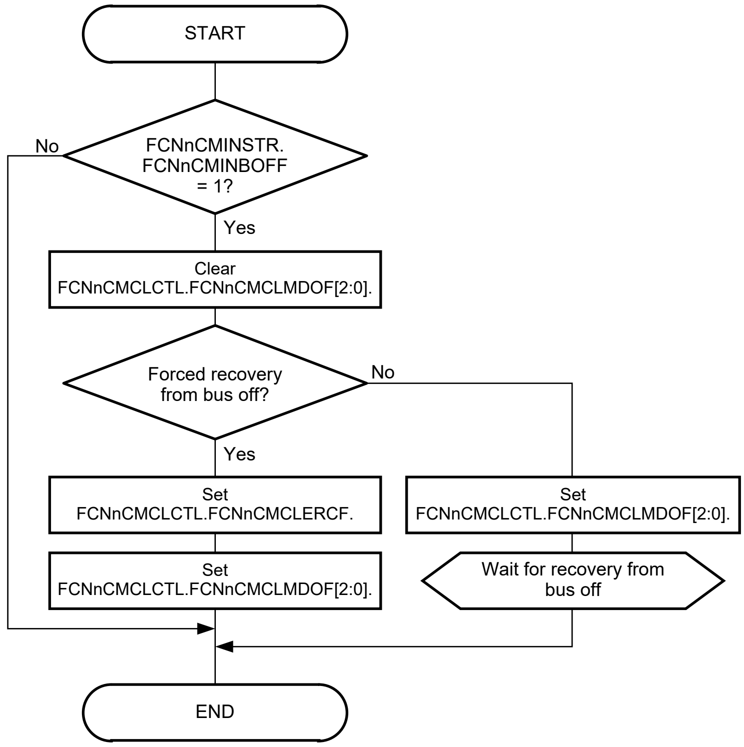


Figure 25.37 Recovery from Bus-Off

Caution: When a request for transition from the initialization mode to any operating mode is issued during the bus-off recovery sequence and the bus-off recovery sequence is executed again, the reception error counter is cleared.
Therefore, 11 consecutive recessive-level bits must be detected on the bus 128 times again.

Remark: Operation mode: Normal operation mode, normal operation mode with ABT, receive-only mode, single shot mode, self-test mode.

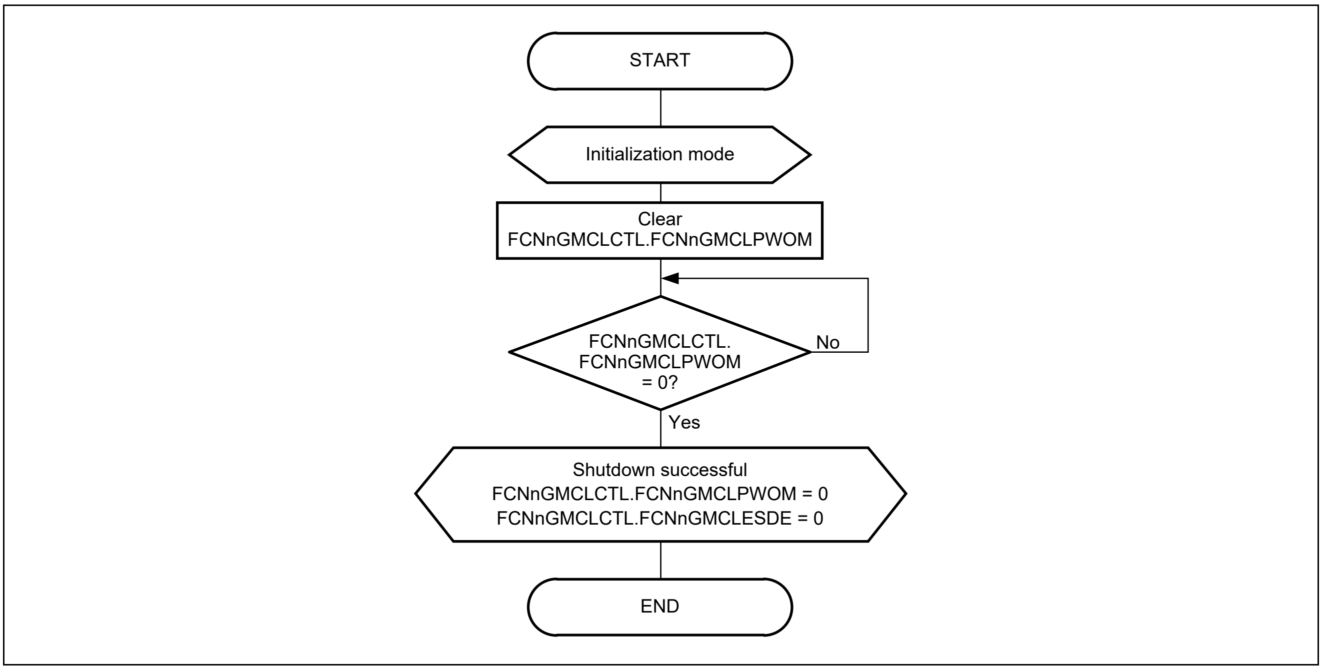


Figure 25.38 Normal Shutdown Processing

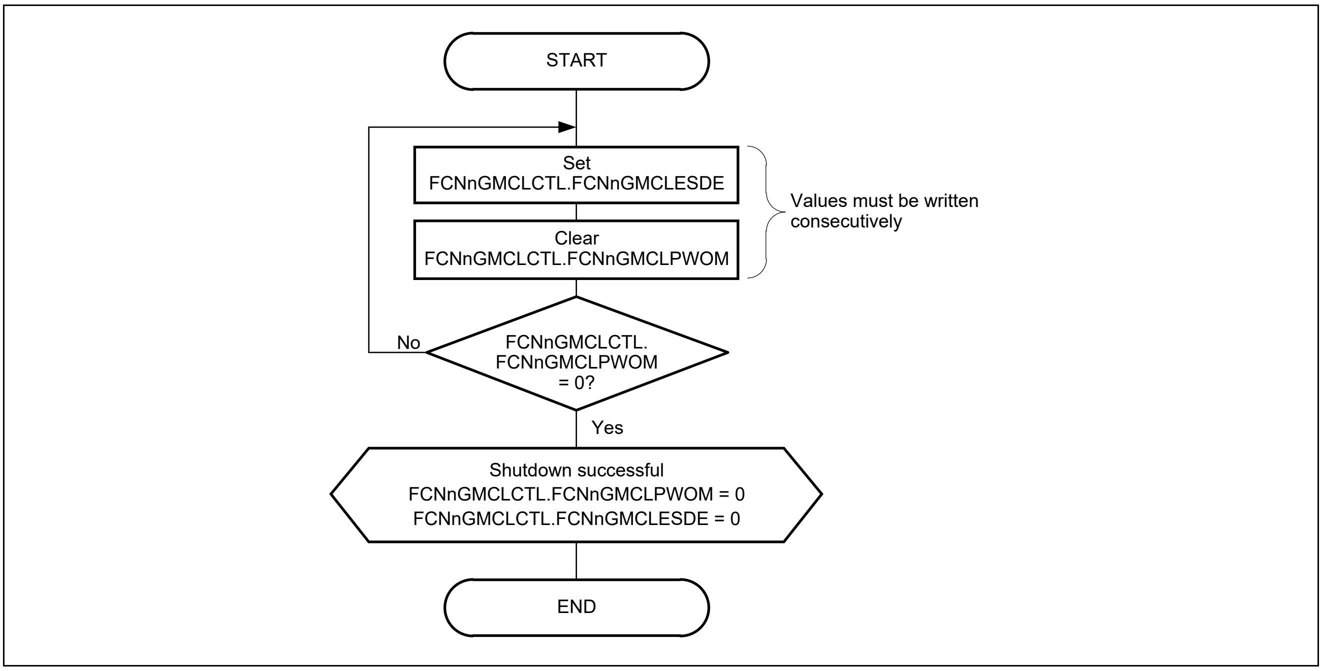


Figure 25.39 Forced Shutdown Processing

Caution: Do not read or write any registers by software between setting of the FCNnGMCLESDE bit and clearing of the FCNnGMCLPWOM bit.

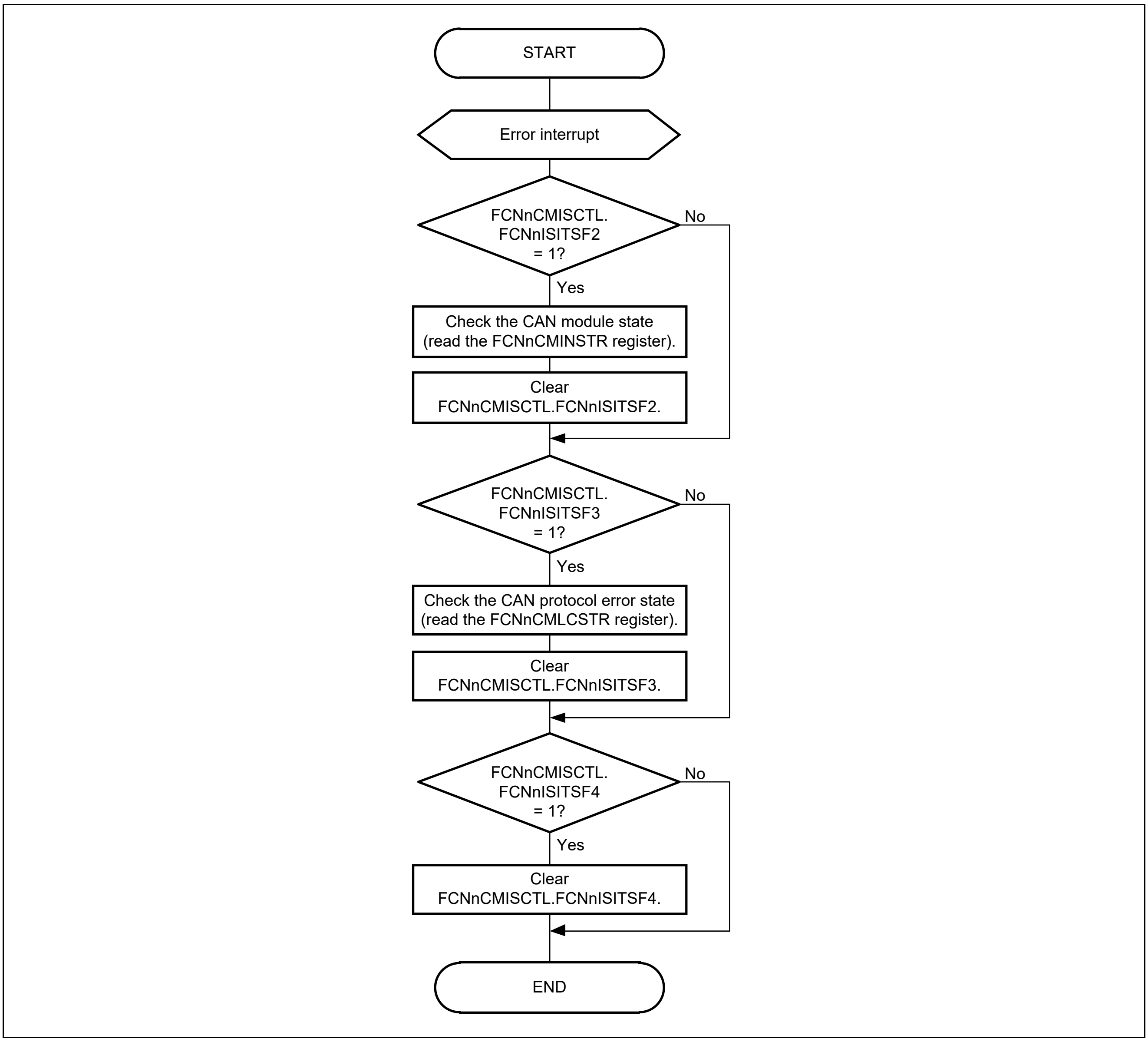


Figure 25.40 Error Handling

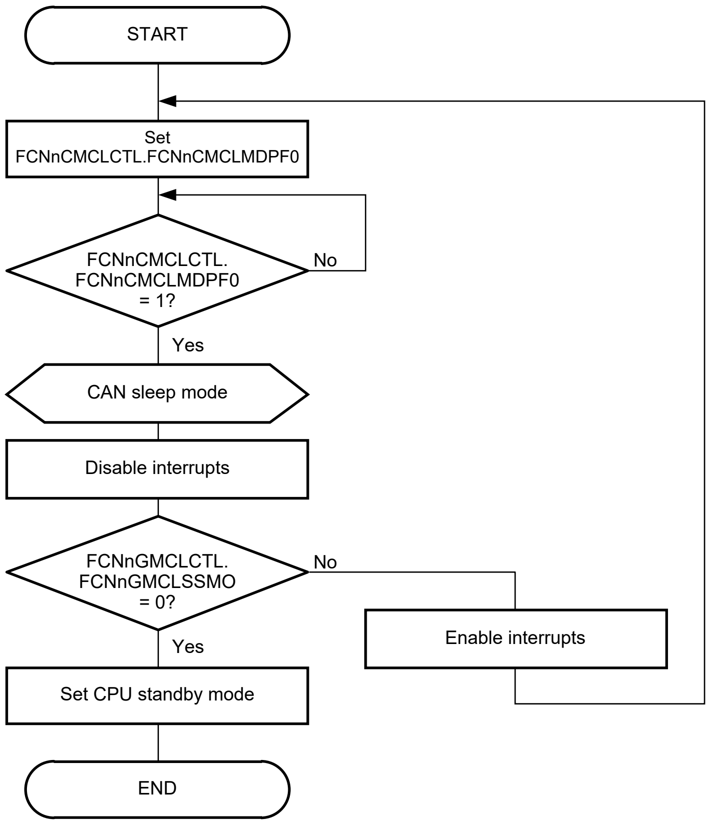


Figure 25.41   Setting CPU Standby (from FCN Sleep Mode)

Remarks 1. Before the CPU is set in the CPU standby mode, check if the FCN sleep mode has been entered.
However, checking the FCN sleep mode may lead to cancelation of this mode until the CPU is placed in the CPU standby mode by a wakeup on the CAN bus.

2. A wakeup may occur on the CAN bus between checking of FCNnGMCLSSMO = 0 and setting of the CPU standby mode. If this is the case, if the CAN module is release from sleep mode, the FCNnCMISITSF5 bit is set, and interrupts are enabled, a wakeup interrupt will be generated.

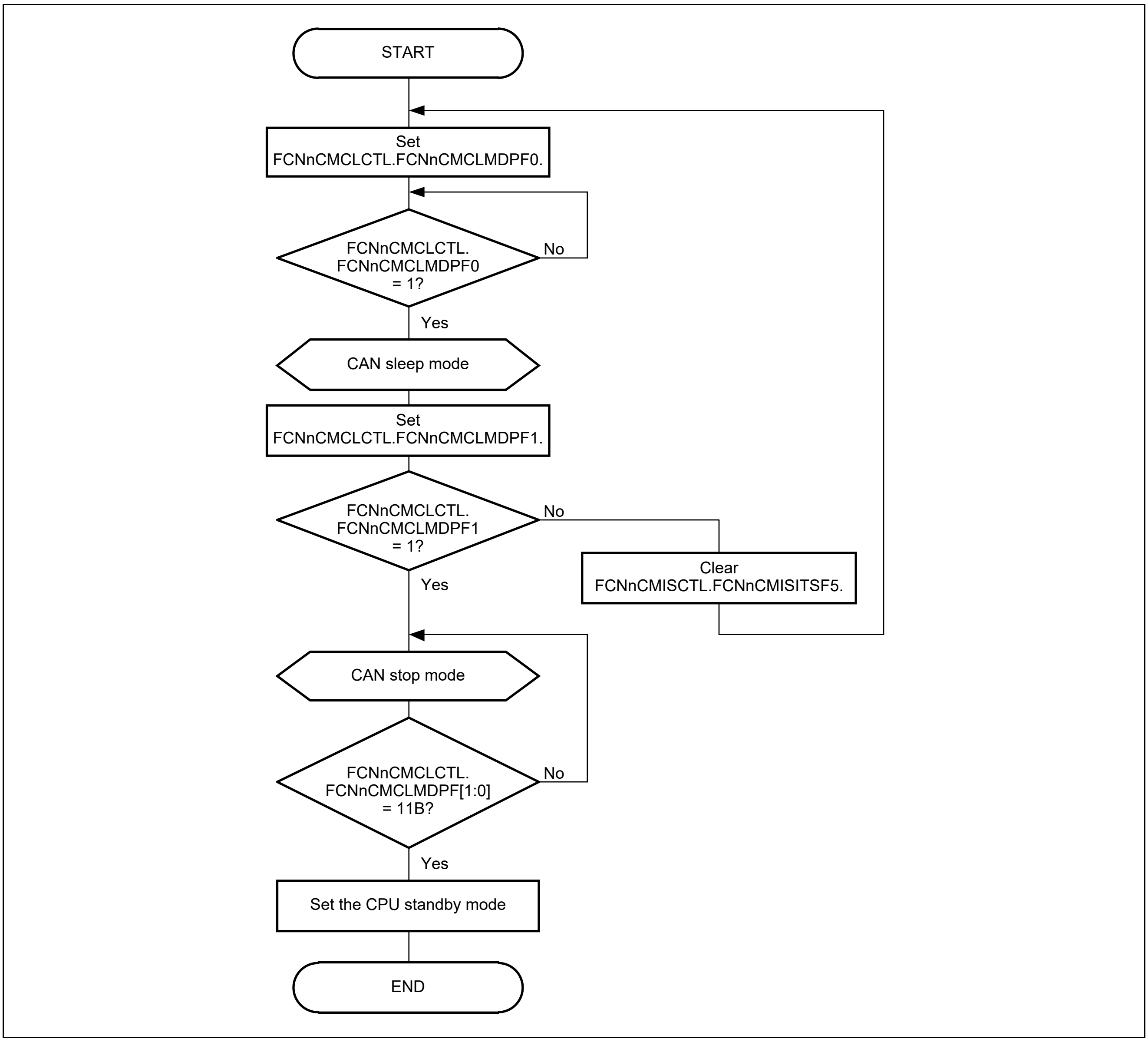


Figure 25.42 Setting CPU Standby (from FCN Stop Mode)

Caution: The FCN stop mode can only be released by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 01B. This mode is not released by a change in the state of the FCN bus.

26. CC-Link IE Function

R-IN32M4-CL3 supports CC-Link IE Field and CC-Link IE TSN, which are CC-Link IE functions.
The setting information for using each function is described below.

26.1 CC-Link IE Function Selection

In CC-Link IE, switch to CC-Link IE Field or CC-Link IE TSN by using the respective firmware, so use them.

26.2 CC-Link IE Clock Selection Function

In R-IN32M4-CL3, a 2 MHz clock used in CC-Link IE can be selected with the CLK2MSEL pin.
When using CC-Link IE TSN, the input clock to CCI_CLK2_097M can be omitted by setting the CLK2MSEL pin to High.

Table 26.1 CC-Link IE Clock Related Pin Settings

	CLK2MSEL	CCI_CLK2_097M
When using CC-Link IE Field	0	Input 2.097152 MHz clock
When using CC-Link IE TSN	1	Connect to GND (Use internally generated 2 MHz clock.)
	0	Input 2 MHz / 2.097152 MHz clock

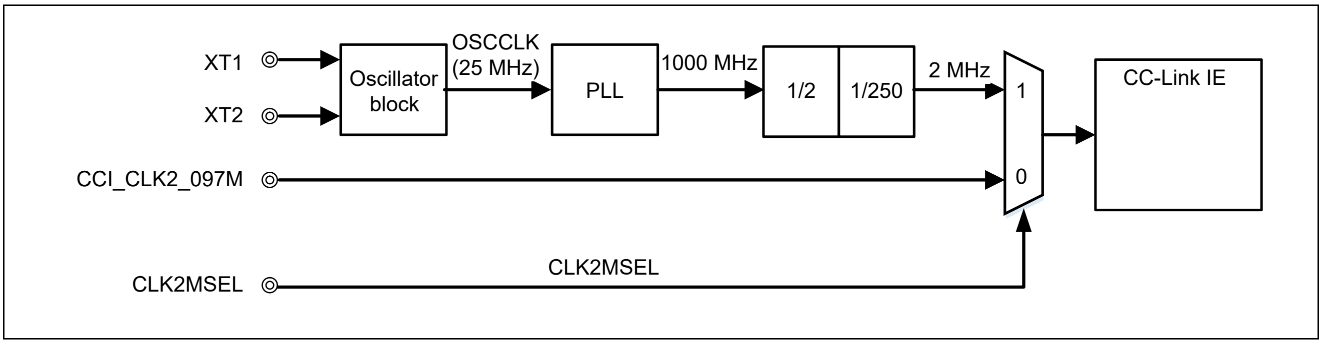


Figure 26.1 CC-Link IE Clock Selection Circuit

26.3 CC-Link IE Field Function

The outline specifications of CC-Link IE Field are as follows. For detailed specifications related to CC-Link IE Field, refer to the “R-IN32M4-CL3 User's Manual: CC-Link IE Field edition”.

Table 26.2 Outline Specifications of CC-Link IE Field

Item	Specification
Ethernet standard	IEEE802.3ab (1000BASE-T) compliant
Communication speed	1 Gbps
Network topology	Line, star, and ring
Number of connections	Up to 254 stations (master and slave stations)
Maximum station-to-station distance	100 m

26.3.1 CC-Link IE Field Bus Bridge (CCLIE MEMC)

The CPU and CC-Link IE Field are connected via a bus bridge (CCLIE MEMC). To access the CC-Link IE Field, it is necessary to set a control register that adjusts the access timing of the bus bridge (CCLIE MEMC).

Table 26.3 Registers Overview of Bus Control Function

Register Name	Abbreviation	Address
CC-Link IE Field Bus Size Control Register	CIEBSC	400A 4004H
CC-Link IE Field Bus Bridge Control Register	CIESMC	400A 4008H

26.3.1.1 CC-Link IE Field Bus Size Control Register (CIEBSC)

The CIEBSC register is used to set the data bus width for access to the CC-Link IE Field area. When using the CC-Link IE Field functions, set the bits of this register to 0000 FFFFh.

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

Address 400A 4004H, Initial value 0000 FFFFH

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIEBSC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIEBSC	CIEBSC15	CIEBSC14	CIEBSC13	CIEBSC12	CIEBSC11	CIEBSC10	CIEBSC9	CIEBSC8	CIEBSC7	CIEBSC6	CIEBSC5	CIEBSC4	CIEBSC3	CIEBSC2	CIEBSC1	CIEBSC0
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
15–0	CIEBSC15–0	Set these bits to 0000 FFFFh.

26.3.1.2 CC-Link IE Field Bus Bridge Control Register (CIESMC)

The CIESMC register is used for access control of the CC-Link IE Field area. When using the CC-Link IE Field functions, set the bits of this register to 0000 0050h.

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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address 400A 4008H, Initial value 0000 FFFFH
CIESMC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CIESMC15	CIESMC14	CIESMC13	CIESMC12	CIESMC11	CIESMC10	CIESMC9	CIESMC8	CIESMC7	CIESMC6	CIESMC5	CIESMC4	CIESMC3	CIESMC2	CIESMC1	CIESMC0	
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
15–0	CIESMC15–0	Set these bits to 0000 0050h.

26.4 CC-Link IE TSN Function

For detailed specifications related to CC-Link IE TSN, refer to the “R-IN32M4-CL3 User's Manual: CC-Link IE TSN edition”.

27. Port Functions

27.1 Features

- I/O port pins: 106 (23 mm Square Package)
  101 (17 mm Square Package)
  The number of available I/O pins depends on the chip package.
- Multiplexed with I/O pin functions of peripheral modules
- Input or output can be specified in bit units.

Cautions 1. Switching from a signal for a peripheral module that is multiplexed with a port pin to the port mode by changing the multiplexed function might lead to a spike, depending on the state of the pin at the time.
The following general countermeasure for spikes should, therefore, be implemented in software.
• Switch the pin function while the peripheral module is stopped.
• If the multiplexed pin in use is an interrupt signal, clear the interrupt request flag, and then unmask the interrupt.
• Only switch the mode after the output value is fixed.
2. Do not externally apply an intermediate voltage to input buffers because these buffers do not implement through-current countermeasures.

27.2 Port Configuration

R-IN32M4-CL3 incorporates twelve 8-bit ports and one 15-bit port (EXTP). Nine are 3-state I/O ports (including EXTP) and four are for real-time control. Input or output can be specified for ports in 1-bit units. The basic structure of ports is the 8-bit unit, but ports P0x–P3x, P4x–P7x, RP0x–RP3x (x = 0–7), and EXTP0–EXTP9 can also be grouped to enable reading and writing in 32-bit units. The real-time port pins (RP00–RP37) can be used for input and output in synchronization with interrupt signals.
Each port can be accessed in 8/16/32-bit units by register settings.

Each port has the following registers for setting I/O modes and selecting multiplexed functions to be used.
Figure 27.1 "Basic Circuit Configuration of Ports" shows the basic circuit configuration of ports.

Register Name	Application and Operation	
	Read	Write
Port registers (Pn, RPm, EXTPp)	Used to read the value of the output latch.	Used to set a value to the output latch.
Port mode registers (PMn, RPMm, EXTPMp)	Used to read whether the port is in input or output mode.	Used to set the port to input or output mode.
Port mode control registers (PMCn, RPMCm, EXTPMCp)	Used to read whether the port pins are selected as port pins or as multiplexed function pins.	Used to select whether the port pins are used as port pins or as multiplexed function pins.
Port function control registers (PFCn, RPFCm, EXTPFCp)	Used to read the selection status of multiplexed functions.	Used to select multiplexed functions.
Port function control expansion registers (PFCEn, RPFCEm, EXTPFCEp)		
Port pin input registers (PINn, RPINm, EXTPINp)	Used to read the input level of the port pin.	Cannot be written.

Cautions 1. Some I/O ports are not available depending on the package.

2. Operation is not guaranteed if the setting has been made not to allocate a multiplexed pin. For example, if multiplexed function 4 is not allocated in the same way as the P05 pin, operation does not proceed correctly even if the multiplexed function 4 is selected. For the allocation of multiplexed pins, refer to Section 27.5 "List of Selectable Multiplexed Functions".

Remark: n = 0–7, m = 0–3, p = 0–14

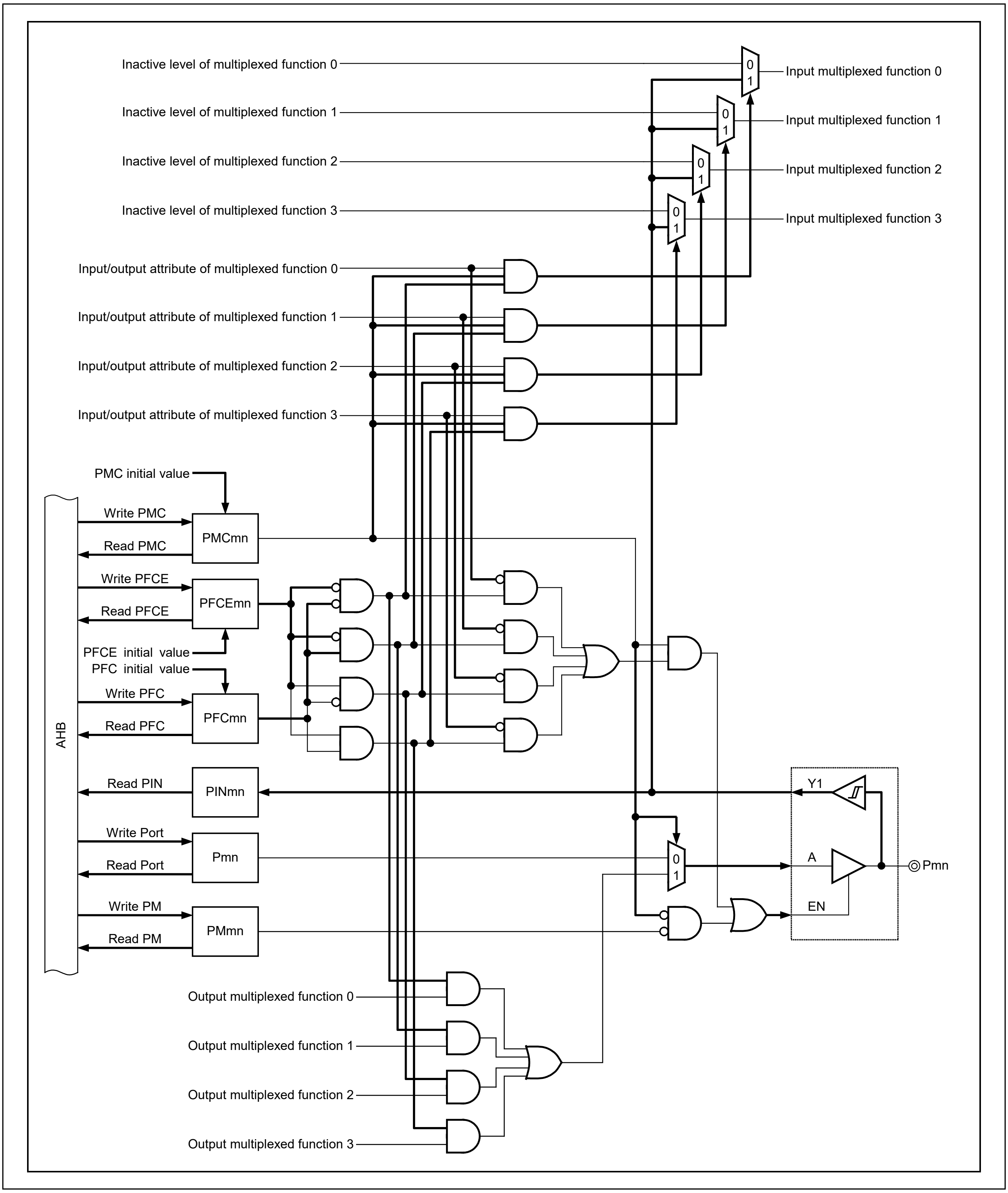


Figure 27.1 Basic Circuit Configuration of Ports

27.3 Register List

(1/7)

Register Name	Symbol	Address
Port register 0 (8-bit)	P0B	400A 3000H
Port register 1 (8-bit)	P1B	400A 3001H
Port register 2 (8-bit)	P2B	400A 3002H
Port register 3 (8-bit)	P3B	400A 3003H
Port register 4 (8-bit)	P4B	400A 3004H
Port register 5 (8-bit)	P5B	400A 3005H
Port register 6 (8-bit)	P6B	400A 3006H
Port register 7 (8-bit)	P7B	400A 3007H
Port register 0 (16-bit)	P0H	400A 3000H
Port register 2 (16-bit)	P2H	400A 3002H
Port register 4 (16-bit)	P4H	400A 3004H
Port register 6 (16-bit)	P6H	400A 3006H
Port register 0 (32-bit)	P0W	400A 3000H
Port register 4 (32-bit)	P4W	400A 3004H
Port mode register 0 (8-bit)	PM0B	400A 3010H
Port mode register 1 (8-bit)	PM1B	400A 3011H
Port mode register 2 (8-bit)	PM2B	400A 3012H
Port mode register 3 (8-bit)	PM3B	400A 3013H
Port mode register 4 (8-bit)	PM4B	400A 3014H
Port mode register 5 (8-bit)	PM5B	400A 3015H
Port mode register 6 (8-bit)	PM6B	400A 3016H
Port mode register 7 (8-bit)	PM7B	400A 3017H
Port mode register 0 (16-bit)	PM0H	400A 3010H
Port mode register 2 (16-bit)	PM2H	400A 3012H
Port mode register 4 (16-bit)	PM4H	400A 3014H
Port mode register 6 (16-bit)	PM6H	400A 3016H
Port mode register 0 (32-bit)	PM0W	400A 3010H
Port mode register 4 (32-bit)	PM4W	400A 3014H

(2/7)

Register Name	Symbol	Address
Port mode control register 0 (8-bit)	PMC0B	400A 3020H
Port mode control register 1 (8-bit)	PMC1B	400A 3021H
Port mode control register 2 (8-bit)	PMC2B	400A 3022H
Port mode control register 3 (8-bit)	PMC3B	400A 3023H
Port mode control register 4 (8-bit)	PMC4B	400A 3024H
Port mode control register 5 (8-bit)	PMC5B	400A 3025H
Port mode control register 6 (8-bit)	PMC6B	400A 3026H
Port mode control register 7 (8-bit)	PMC7B	400A 3027H
Port mode control register 0 (16-bit)	PMC0H	400A 3020H
Port mode control register 2 (16-bit)	PMC2H	400A 3022H
Port mode control register 4 (16-bit)	PMC4H	400A 3024H
Port mode control register 6 (16-bit)	PMC6H	400A 3026H
Port mode control register 0 (32-bit)	PMC0W	400A 3020H
Port mode control register 4 (32-bit)	PMC4W	400A 3024H
Port function control register 0 (8-bit)	PFC0B	400A 3030H
Port function control register 1 (8-bit)	PFC1B	400A 3031H
Port function control register 2 (8-bit)	PFC2B	400A 3032H
Port function control register 3 (8-bit)	PFC3B	400A 3033H
Port function control register 4 (8-bit)	PFC4B	400A 3034H
Port function control register 5 (8-bit)	PFC5B	400A 3035H
Port function control register 6 (8-bit)	PFC6B	400A 3036H
Port function control register 7 (8-bit)	PFC7B	400A 3037H
Port function control register 0 (16-bit)	PFC0H	400A 3030H
Port function control register 2 (16-bit)	PFC2H	400A 3032H
Port function control register 4 (16-bit)	PFC4H	400A 3034H
Port function control register 6 (16-bit)	PFC6H	400A 3036H
Port function control register 0 (32-bit)	PFC0W	400A 3030H
Port function control register 4 (32-bit)	PFC4W	400A 3034H

(3/7)

Register Name	Symbol	Address
Port function control expansion register 0 (8-bit)	PFCE0B	400A 3040H
Port function control expansion register 1 (8-bit)	PFCE1B	400A 3041H
Port function control expansion register 2 (8-bit)	PFCE2B	400A 3042H
Port function control expansion register 3 (8-bit)	PFCE3B	400A 3043H
Port function control expansion register 4 (8-bit)	PFCE4B	400A 3044H
Port function control expansion register 5 (8-bit)	PFCE5B	400A 3045H
Port function control expansion register 6 (8-bit)	PFCE6B	400A 3046H
Port function control expansion register 7 (8-bit)	PFCE7B	400A 3047H
Port function control expansion register 0 (16-bit)	PFCE0H	400A 3040H
Port function control expansion register 2 (16-bit)	PFCE2H	400A 3042H
Port function control expansion register 4 (16-bit)	PFCE4H	400A 3044H
Port function control expansion register 6 (16-bit)	PFCE6H	400A 3046H
Port function control expansion register 0 (32-bit)	PFCE0W	400A 3040H
Port function control expansion register 4 (32-bit)	PFCE4W	400A 3044H
Port pin input register 0 (8-bit)	PIN0B	400A 3050H
Port pin input register 1 (8-bit)	PIN1B	400A 3051H
Port pin input register 2 (8-bit)	PIN2B	400A 3052H
Port pin input register 3 (8-bit)	PIN3B	400A 3053H
Port pin input register 4 (8-bit)	PIN4B	400A 3054H
Port pin input register 5 (8-bit)	PIN5B	400A 3055H
Port pin input register 6 (8-bit)	PIN6B	400A 3056H
Port pin input register 7 (8-bit)	PIN7B	400A 3057H
Port pin input register 0 (16-bit)	PIN0H	400A 3050H
Port pin input register 2 (16-bit)	PIN2H	400A 3052H
Port pin input register 4 (16-bit)	PIN4H	400A 3054H
Port pin input register 6 (16-bit)	PIN6H	400A 3056H
Port pin input register 0 (32-bit)	PIN0W	400A 3050H
Port pin input register 4 (32-bit)	PIN4W	400A 3054H

(4/7)

Register Name	Symbol	Address
Real-Time port register 0 (8-bit)	RP0B	400A 3400H
Real-Time port register 1 (8-bit)	RP1B	400A 3401H
Real-Time port register 2 (8-bit)	RP2B	400A 3402H
Real-Time port register 3 (8-bit)	RP3B	400A 3403H
Real-Time port register 0 (16-bit)	RP0H	400A 3400H
Real-Time port register 2 (16-bit)	RP2H	400A 3402H
Real-Time port register 0 (32-bit)	RP0W	400A 3400H
Real-Time port mode register 0 (8-bit)	RPM0B	400A 3410H
Real-Time port mode register 1 (8-bit)	RPM1B	400A 3411H
Real-Time port mode register 2 (8-bit)	RPM2B	400A 3412H
Real-Time port mode register 3 (8-bit)	RPM3B	400A 3413H
Real-Time port mode register 0 (16-bit)	RPM0H	400A 3410H
Real-Time port mode register 2 (16-bit)	RPM2H	400A 3412H
Real-Time port mode register 0 (32-bit)	RPM0W	400A 3410H
Real-Time port mode control register 0 (8-bit)	RPMC0B	400A 3420H
Real-Time port mode control register 1 (8-bit)	RPMC1B	400A 3421H
Real-Time port mode control register 2 (8-bit)	RPMC2B	400A 3422H
Real-Time port mode control register 3 (8-bit)	RPMC3B	400A 3423H
Real-Time port mode control register 0 (16-bit)	RPMC0H	400A 3420H
Real-Time port mode control register 2 (16-bit)	RPMC2H	400A 3422H
Real-Time port mode control register 0 (32-bit)	RPMC0W	400A 3420H
Real-Time port function control register 0 (8-bit)	RPFC0B	400A 3430H
Real-Time port function control register 1 (8-bit)	RPFC1B	400A 3431H
Real-Time port function control register 2 (8-bit)	RPFC2B	400A 3432H
Real-Time port function control register 3 (8-bit)	RPFC3B	400A 3433H
Real-Time port function control register 0 (16-bit)	RPFC0H	400A 3430H
Real-Time port function control register 2 (16-bit)	RPFC2H	400A 3432H
Real-Time port function control register 0 (32-bit)	RPFC0W	400A 3430H

(5/7)

Register Name	Symbol	Address
Real-Time port function control expansion register 0 (8-bit)	RPFCE0B	400A 3440H
Real-Time port function control expansion register 1 (8-bit)	RPFCE1B	400A 3441H
Real-Time port function control expansion register 2 (8-bit)	RPFCE2B	400A 3442H
Real-Time port function control expansion register 3 (8-bit)	RPFCE3B	400A 3443H
Real-Time port function control expansion register 0 (16-bit)	RPFCE0H	400A 3440H
Real-Time port function control expansion register 2 (16-bit)	RPFCE2H	400A 3442H
Real-Time port function control expansion register 0 (32-bit)	RPFCE0W	400A 3440H
Real-Time port pin input register 0 (8-bit)	RPIN0B	400A 3450H
Real-Time port pin input register 1 (8-bit)	RPIN1B	400A 3451H
Real-Time port pin input register 2 (8-bit)	RPIN2B	400A 3452H
Real-Time port pin input register 3 (8-bit)	RPIN3B	400A 3453H
Real-Time port pin input register 0 (16-bit)	RPIN0H	400A 3450H
Real-Time port pin input register 2 (16-bit)	RPIN2H	400A 3452H
Real-Time port pin input register 0 (32-bit)	RPIN0W	400A 3450H

(6/7)

Register Name	Symbol	Address
EXT port register 0 (8-bit)	EXTP0B	400A 3800H
EXT port register 1 (8-bit)	EXTP1B	400A 3801H
EXT port register 0 (16-bit)	EXTP0H	400A 3800H
EXT port register 0 (32-bit)	EXTP0W	400A 3800H
EXT port mode register 0 (8-bit)	EXTPM0B	400A 3810H
EXT port mode register 1 (8-bit)	EXTPM1B	400A 3811H
EXT port mode register 0 (16-bit)	EXTPM0H	400A 3810H
EXT port mode register 0 (32-bit)	EXTPM0W	400A 3810H
EXT port mode control register 0 (8-bit)	EXTPMC0B	400A 3820H
EXT port mode control register 1 (8-bit)	EXTPMC1B	400A 3821H
EXT port mode control register 0 (16-bit)	EXTPMC0H	400A 3820H
EXT port mode control register 0 (32-bit)	EXTPMC0W	400A 3820H
EXT port function control register 0 (8-bit)	EXTPFC0B	400A 3830H
EXT port function control register 1 (8-bit)	EXTPFC1B	400A 3831H
EXT port function control register 0 (16-bit)	EXTPFC0H	400A 3830H
EXT port function control register 0 (32-bit)	EXTPFC0W	400A 3830H
EXT port function control expansion register 0 (8-bit)	EXTPFCE0B	400A 3840H
EXT port function control expansion register 1 (8-bit)	EXTPFCE1B	400A 3841H
EXT port function control expansion register 0 (16-bit)	EXTPFCE0H	400A 3840H
EXT port function control expansion register 0 (32-bit)	EXTPFCE0W	400A 3840H
EXT port pin input register 0 (8-bit)	EXTPIN0B	400A 3850H
EXT port pin input register 1 (8-bit)	EXTPIN1B	400A 3851H
EXT port pin input register 0 (16-bit)	EXTPIN0H	400A 3850H
EXT port pin input register 0 (32-bit)	EXTPIN0W	400A 3850H

(7/7)

Register Name	Symbol	Address
Buffer switching register P0L	DRCTLP0L	4001 0220H
Buffer switching register P0H	DRCTLP0H	4001 0224H
Buffer switching register P1L	DRCTLP1L	4001 0228H
Buffer switching register P1H	DRCTLP1H	4001 022CH
Buffer switching register P2L	DRCTLP2L	4001 0230H
Buffer switching register P2H	DRCTLP2H	4001 0234H
Buffer switching register P3L	DRCTLP3L	4001 0238H
Buffer switching register P3H	DRCTLP3H	4001 023CH
Buffer switching register P4L	DRCTLP4L	4001 0240H
Buffer switching register P4H	DRCTLP4H	4001 0244H
Buffer switching register P5L	DRCTLP5L	4001 0248H
Buffer switching register P5H	DRCTLP5H	4001 024CH
Buffer switching register P6L	DRCTLP6L	4001 0250H
Buffer switching register P6H	DRCTLP6H	4001 0254H
Buffer switching register P7L	DRCTLP7L	4001 0258H
Buffer switching register P7H	DRCTLP7H	4001 025CH
Buffer switching register RP0L	DRCTLRP0L	4001 0260H
Buffer switching register RP0H	DRCTLRP0H	4001 0264H
Buffer switching register RP1L	DRCTLRP1L	4001 0268H
Buffer switching register RP1H	DRCTLRP1H	4001 026CH
Buffer switching register RP2L	DRCTLRP2L	4001 0270H
Buffer switching register RP2H	DRCTLRP2H	4001 0274H
Buffer switching register RP3L	DRCTLRP3L	4001 0278H
Buffer switching register RP3H	DRCTLRP3H	4001 027CH
Buffer switching register EXTP0L	DRCTLEXTP0L	4001 0280H
Buffer switching register EXTP0H	DRCTLEXTP0H	4001 0284H
Buffer switching register EXTP1L	DRCTLEXTP1L	4001 0288H
Buffer switching register EXTP1H	DRCTLEXTP1H	4001 028CH

27.4 Register Details

27.4.1 Port Registers (P, RP, EXTP)

R-IN32M4-CL3 incorporates twelve 8-bit ports and one 15-bit port (EXTP).
Nine are 3-state I/O ports (including EXTP) and four are for real-time control. Input or output can be specified for ports in 1-bit units. The port registers are used for writing the output levels for output port pins. When read, the value of the given port register is read. The PIN, RPIN, and EXTPIN registers are used to read the levels on input pins.

	7	6	5	4	3	2	1	0	Address	Initial value
P0B	P07	P06	P05	P04	P03	P02	P01	P00	400A 3000H	00H
P1B	P17	P16	P15	P14	P13	P12	P11	P10	400A 3001H	00H
P2B	P27	P26	P25	P24	P23	P22	P21	P20	400A 3002H	00H
P3B	P37	P36	P35	P34	P33	P32	P31	P30	400A 3003H	00H
P4B	P47	P46	P45	P44	P43	P42	P41	P40	400A 3004H	00H
P5B	P57	P56	P55	P54	P53	P52	P51	P50	400A 3005H	00H
P6B	P67	P66	P65	P64	P63	P62	P61	P60	400A 3006H	00H
P7B	P77	P76	P75	P74	P73	P72	P71	P70	400A 3007H	00H
EXTP0B	EXTP7	EXTP6	EXTP5	EXTP4	EXTP3	EXTP2	EXTP1	EXTP0	400A 3800H	00H
EXTP1B	0	EXTP14	EXTP13	EXTP12	EXTP11	EXTP10	EXTP9	EXTP8	400A 3801H	00H
RP0B	RP07	RP06	RP05	RP04	RP03	RP02	RP01	RP00	400A 3400H	00H
RP1B	RP17	RP16	RP15	RP14	RP13	RP12	RP11	RP10	400A 3401H	00H
RP2B	RP27	RP26	RP25	RP24	RP23	RP22	RP21	RP20	400A 3402H	00H
RP3B	RP37	RP36	RP35	RP34	RP33	RP32	RP31	RP30	400A 3403H	00H

Bit Position	Bit Name	Description
7–0	Pmn/RPln EXTPp	These bits set the value of the output latch when the port is used in output mode. If read, the value of the output latch is read.

Figure 27.2 Port Registers (8-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

P0H	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
	P17	P16	P15	P14	P13	P12	P11	P10	P07	P06	P05	P04	P03	P02	P01	P00	400A 3000H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

P2H	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
	P37	P36	P35	P34	P33	P32	P31	P30	P27	P26	P25	P24	P23	P22	P21	P20	400A 3002H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

P4H	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
	P57	P56	P55	P54	P53	P52	P51	P50	P47	P46	P45	P44	P43	P42	P41	P40	400A 3004H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

P6H	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
	P77	P76	P75	P74	P73	P72	P71	P70	P67	P66	P65	P64	P63	P62	P61	P60	400A 3006H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

EXTP0H	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
	0	EXTP 14	EXTP 13	EXTP 12	EXTP 11	EXTP 10	EXTP 9	EXTP 8	EXTP 7	EXTP 6	EXTP 5	EXTP 4	EXTP 3	EXTP 2	EXTP 1	EXTP 0	400A 3800H
	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

RP0H	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
	RP17	RP16	RP15	RP14	RP13	RP12	RP11	RP10	RP07	RP06	RP05	RP04	RP03	RP02	RP01	RP00	400A 3400H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

RP2H	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
	RP37	RP36	RP35	RP34	RP33	RP32	RP31	RP30	RP27	RP26	RP25	RP24	RP23	RP22	RP21	RP20	400A 3402H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Bit Position	Bit Name	Description
15–0	Pmn/RPln EXTPp	These bits set the value of the output latch when the port is used in output mode. If read, the value of the output latch is read.

Figure 27.3 Port Registers (16-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

P0W — Address 400A 3000H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0W	P37	P36	P35	P34	P33	P32	P31	P30	P27	P26	P25	P24	P23	P22	P21	P20	P17	P16	P15	P14	P13	P12	P11	P10	P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4W — Address 400A 3004H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P4W	P77	P76	P75	P74	P73	P72	P71	P70	P67	P66	P65	P64	P63	P62	P61	P60	P57	P56	P55	P54	P53	P52	P51	P50	P47	P46	P45	P44	P43	P42	P41	P40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EXTP0W — Address 400A 3800H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTP0W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EXTP14	EXTP13	EXTP12	EXTP11	EXTP10	EXTP9	EXTP8	EXTP7	EXTP6	EXTP5	EXTP4	EXTP3	EXTP2	EXTP1	EXTP0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RP0W — Address 400A 3400H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP0W	RP37	RP36	RP35	RP34	RP33	RP32	RP31	RP30	RP27	RP26	RP25	RP24	RP23	RP22	RP21	RP20	RP17	RP16	RP15	RP14	RP13	RP12	RP11	RP10	RP07	RP06	RP05	RP04	RP03	RP02	RP01	RP00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–0	Pmn/RPln/ EXTPp	These bits set the value of the output latch when the port is used in output mode. If read, the value of the output latch is read.

Figure 27.4 Port Registers (32-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

27.4.2 Port Mode Registers (PM, RPM, EXTPM)

These registers are used to set a port to input or output mode.

	7	6	5	4	3	2	1	0	Address	Initial value
PM0B	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	400A 3010H	FFH
PM1B	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	400A 3011H	FFH
PM2B	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	400A 3012H	FFH
PM3B	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	400A 3013H	FFH
PM4B	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	400A 3014H	FFH
PM5B	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	400A 3015H	FFH
PM6B	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	400A 3016H	FFH
PM7B	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	400A 3017H	FFH
EXTPM0B	EXTPM7	EXTPM6	EXTPM5	EXTPM4	EXTPM3	EXTPM2	EXTPM1	EXTPM0	400A 3810H	FFH
EXTPM1B	0	EXTPM14	EXTPM13	EXTPM12	EXTPM11	EXTPM10	EXTPM9	EXTPM8	400A 3811H	7FH
RPM0B	RPM07	RPM06	RPM05	RPM04	RPM03	RPM02	RPM01	RPM00	400A 3410H	FFH
RPM1B	RPM17	RPM16	RPM15	RPM14	RPM13	RPM12	RPM11	RPM10	400A 3411H	FFH
RPM2B	RPM27	RPM26	RPM25	RPM24	RPM23	RPM22	RPM21	RPM20	400A 3412H	FFH
RPM3B	RPM37	RPM36	RPM35	RPM34	RPM33	RPM32	RPM31	RPM30	400A 3413H	FFH

Bit Position	Bit Name	Description
7–0	PMmn/ RPMln/ EXTPMp	These bits set the port to input or output mode. 0: Output mode (output buffer is on) 1: Input mode (output buffer is off) (initial value)

Figure 27.5 Port Mode Registers (8-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

PM0H — Address: 400A 3010H, Initial value: FFFFH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PM2H — Address: 400A 3012H, Initial value: FFFFH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PM4H — Address: 400A 3014H, Initial value: FFFFH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PM6H — Address: 400A 3016H, Initial value: FFFFH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EXTPM0H — Address: 400A 3810H, Initial value: 7FFFH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	EXTPM14	EXTPM13	EXTPM12	EXTPM11	EXTPM10	EXTPM9	EXTPM8	EXTPM7	EXTPM6	EXTPM5	EXTPM4	EXTPM3	EXTPM2	EXTPM1	EXTPM0
Access	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RPM0H — Address: 400A 3410H, Initial value: FFFFH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RPM17	RPM16	RPM15	RPM14	RPM13	RPM12	RPM11	RPM10	RPM07	RPM06	RPM05	RPM04	RPM03	RPM02	RPM01	RPM00
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RPM2H — Address: 400A 3412H, Initial value: FFFFH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RPM37	RPM36	RPM35	RPM34	RPM33	RPM32	RPM31	RPM30	RPM27	RPM26	RPM25	RPM24	RPM23	RPM22	RPM21	RPM20
Access	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
15–0	PMmn/ RPMln/ EXTPMp	These bits set the port to input or output mode. 0: Output mode (output buffer is on) 1: Input mode (output buffer is off) (initial value)

Figure 27.6 Port Mode Registers (16-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

PM0W — Address 400A 3010H, Initial value FFFF FFFFH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PM0W	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PM4W — Address 400A 3014H, Initial value FFFF FFFFH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PM4W	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EXTPM0W — Address 400A 3810H, Initial value 0000 7FFFH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTPM0W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EXTPM14	EXTPM13	EXTPM12	EXTPM11	EXTPM10	EXTPM9	EXTPM8	EXTPM7	EXTPM6	EXTPM5	EXTPM4	EXTPM3	EXTPM2	EXTPM1	EXTPM0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RPM0W — Address 400A 3410H, Initial value FFFF FFFFH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPM0W	RPM37	RPM36	RPM35	RPM34	RPM33	RPM32	RPM31	RPM30	RPM27	RPM26	RPM25	RPM24	RPM23	RPM22	RPM21	RPM20	RPM17	RPM16	RPM15	RPM14	RPM13	RPM12	RPM11	RPM10	RPM07	RPM06	RPM05	RPM04	RPM03	RPM02	RPM01	RPM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–0	PMmn/RPMln/ EXTPMp	These bits set the port to input or output mode. 0: Output mode (output buffer is on) 1: Input mode (output buffer is off) (initial value)

Figure 27.7 Port Mode Registers (32-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

27.4.3 Port Mode Control Registers (PMC, RPMC, EXTPMC)

These registers are for selecting whether the port pins are used as port pins or as multiplexed function pins.

	7	6	5	4	3	2	1	0	Address	Initial value
PMC0B	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00	400A 3020H	00H
PMC1B	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10	400A 3021H	00H*1
PMC2B	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	400A 3022H	00H
PMC3B	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	400A 3023H	00H
PMC4B	PMC47	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40	400A 3024H	00H*1
PMC5B	PMC57	PMC56	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50	400A 3025H	00H
PMC6B	PMC67	PMC66	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60	400A 3026H	00H
PMC7B	PMC77	PMC76	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70	400A 3027H	00H
EXTPMC0B	EXTPMC7	EXTPMC6	EXTPMC5	EXTPMC4	EXTPMC3	EXTPMC2	EXTPMC1	EXTPMC0	400A 3820H	00H*1
EXTPMC1B	0	EXTPMC 14	EXTPMC 13	EXTPMC 12	EXTPMC 11	EXTPMC 10	EXTPMC9	EXTPMC8	400A 3821H	00H*1
RPMC0B	RPMC07	RPMC06	RPMC05	RPMC04	RPMC03	RPMC02	RPMC01	RPMC00	400A 3420H	00H*1
RPMC1B	RPMC17	RPMC16	RPMC15	RPMC14	RPMC13	RPMC12	RPMC11	RPMC10	400A 3421H	00H*1
RPMC2B	RPMC27	RPMC26	RPMC25	RPMC24	RPMC23	RPMC22	RPMC21	RPMC20	400A 3422H	00H*1
RPMC3B	RPMC37	RPMC36	RPMC35	RPMC34	RPMC33	RPMC32	RPMC31	RPMC30	400A 3423H	00H*1

Bit Position	Bit Name	Description
7–0	PMCmn / RPMCln / EXTPMCp	These bits select whether the port pins are used as port pins or as multiplexed function pins.*2 0: Port mode (the inactive level is input for multiplexed input pin functions.) 1: Multiplexed function (control mode)

Figure 27.8 Port Mode Control Registers (8-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 "Pin States".

2. The multiplexed functions are selected using the port mode control registers, port function control registers, and port function control expansion registers.
For details, refer to Section 27.5 "List of Selectable Multiplexed Functions".

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

PMC0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 3020H
Initial value 0000H*1

PMC2H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 3022H
Initial value 0000H

PMC4H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMC57	PMC56	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50	PMC47	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 3024H
Initial value 0000H*1

PMC6H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMC77	PMC76	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70	PMC67	PMC66	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 3026H
Initial value 0000H

EXTPMC0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	EXTPMC14	EXTPMC13	EXTPMC12	EXTPMC11	EXTPMC10	EXTPMC9	EXTPMC8	EXTPMC7	EXTPMC6	EXTPMC5	EXTPMC4	EXTPMC3	EXTPMC2	EXTPMC1	EXTPMC0
0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 3820H
Initial value 0000H*1

RPMC0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPMC17	RPMC16	RPMC15	RPMC14	RPMC13	RPMC12	RPMC11	RPMC10	RPMC07	RPMC06	RPMC05	RPMC04	RPMC03	RPMC02	RPMC01	RPMC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 3420H
Initial value 0000H*1

RPMC2H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPMC37	RPMC36	RPMC35	RPMC34	RPMC33	RPMC32	RPMC31	RPMC30	RPMC27	RPMC26	RPMC25	RPMC24	RPMC23	RPMC22	RPMC21	RPMC20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address 400A 3422H
Initial value 0000H*1

Bit Position	Bit Name	Description
15–0	PMCmn / RPMCln / EXTPMCp	These bits select whether the port pins are used as port pins or as multiplexed function pins.*2 0: Port mode (the inactive level is input for multiplexed input pin functions.) 1: Multiplexed function (control mode)

Figure 27.9 Port Mode Control Registers (16-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 “Pin States”.

2. The multiplexed functions are selected using the port mode control registers, port function control registers, and port function control expansion registers.
For details, refer to Section 27.5 “List of Selectable Multiplexed Functions”.

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

PMC0W — Address 400A 3020H, Initial value 0000 0000H*1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20
R/W	R/W	R/W	R/W	R/W	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
Name	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMC4W — Address 400A 3024H, Initial value 0000 0000H*1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PMC77	PMC76	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70	PMC67	PMC66	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60
R/W	R/W	R/W	R/W	R/W	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
Name	PMC57	PMC56	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50	PMC47	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EXTPMC0W — Address 400A 3820H, Initial value 0000 0000H*1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	EXTPMC14	EXTPMC13	EXTPMC12	EXTPMC11	EXTPMC10	EXTPMC9	EXTPMC8	EXTPMC7	EXTPMC6	EXTPMC5	EXTPMC4	EXTPMC3	EXTPMC2	EXTPMC1	EXTPMC0
R/W	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RPMC0W — Address 400A 3420H, Initial value 0000 0000H*1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RPMC37	RPMC36	RPMC35	RPMC34	RPMC33	RPMC32	RPMC31	RPMC30	RPMC27	RPMC26	RPMC25	RPMC24	RPMC23	RPMC22	RPMC21	RPMC20
R/W	R/W	R/W	R/W	R/W	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
Name	RPMC17	RPMC16	RPMC15	RPMC14	RPMC13	RPMC12	RPMC11	RPMC10	RPMC07	RPMC06	RPMC05	RPMC04	RPMC03	RPMC02	RPMC01	RPMC00
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–0	PMCmn / RPMCln / EXTPMCp	These bits select whether the port pins are used as port pins or as multiplexed function pins.*2 0: Port mode (the inactive level is input for multiplexed input pin functions.) 1: Multiplexed function (control mode)

Figure 27.10 Port Mode Control Registers (32-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 "Pin States".

2. The multiplexed functions are selected using the port mode control registers, port function control registers, and port function control expansion registers.
For details, refer to Section 27.5 "List of Selectable Multiplexed Functions".

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

27.4.4 Port Function Control Registers (PFC, RPFC, EXTPFC)

These registers are used to specify which multiplexed function is to be used. These registers can be set in 1-bit units.

	7	6	5	4	3	2	1	0	Address	Initial value
PFC0B	PFC07	PFC06	PFC05	PFC04	PFC03	PFC02	PFC01	PFC00	400A 3030H	00H
PFC1B	PFC17	PFC16	PFC15	PFC14	PFC13	PFC12	PFC11	PFC10	400A 3031H	00H
PFC2B	PFC27	PFC26	PFC25	PFC24	PFC23	PFC22	PFC21	PFC20	400A 3032H	00H
PFC3B	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30	400A 3033H	00H
PFC4B	PFC47	PFC46	PFC45	PFC44	PFC43	PFC42	PFC41	PFC40	400A 3034H	00H*1
PFC5B	PFC57	PFC56	PFC55	PFC54	PFC53	PFC52	PFC51	PFC50	400A 3035H	00H
PFC6B	PFC67	PFC66	PFC65	PFC64	PFC63	PFC62	PFC61	PFC60	400A 3036H	00H
PFC7B	PFC77	PFC76	PFC75	PFC74	PFC73	PFC72	PFC71	PFC70	400A 3037H	00H
EXTPFC0B	EXTPFC7	EXTPFC6	EXTPFC5	EXTPFC4	EXTPFC3	EXTPFC2	EXTPFC1	EXTPFC0	400A 3830H	00H
EXTPFC1B	0	EXTPFC14	EXTPFC13	EXTPFC12	EXTPFC11	EXTPFC10	EXTPFC9	EXTPFC8	400A 3831H	00H
RPFC0B	RPFC07	RPFC06	RPFC05	RPFC04	RPFC03	RPFC02	RPFC01	RPFC00	400A 3430H	00H*1
RPFC1B	RPFC17	RPFC16	RPFC15	RPFC14	RPFC13	RPFC12	RPFC11	RPFC10	400A 3431H	00H
RPFC2B	RPFC27	RPFC26	RPFC25	RPFC24	RPFC23	RPFC22	RPFC21	RPFC20	400A 3432H	00H*1
RPFC3B	RPFC37	RPFC36	RPFC35	RPFC34	RPFC33	RPFC32	RPFC31	RPFC30	400A 3433H	00H

Bit Position	Bit Name	Description
7–0	PFCmn / RPFCmn / EXTPFCp	These bits specify whether to use multiplexed functions 1 and 3 or multiplexed functions 2 and 4.*2 0: Multiplexed function 1 / Multiplexed function 3 1: Multiplexed function 2 / Multiplexed function 4

Figure 27.11 Port Function Control Registers (8-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 "Pin States".

2. The multiplexed functions are selected using the port mode control registers, port function control registers, and port function control expansion registers.
For details, refer to Section 27.5 "List of Selectable Multiplexed Functions".

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address / Initial value
PFC0H	PFC 17	PFC 16	PFC 15	PFC 14	PFC 13	PFC 12	PFC 11	PFC 10	PFC 07	PFC 06	PFC 05	PFC 04	PFC 03	PFC 02	PFC 01	PFC 00	Address 400A 3030H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address / Initial value
PFC2H	PFC 37	PFC 36	PFC 35	PFC 34	PFC 33	PFC 32	PFC 31	PFC 30	PFC 27	PFC 26	PFC 25	PFC 24	PFC 23	PFC 22	PFC 21	PFC 20	Address 400A 3032H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address / Initial value
PFC4H	PFC 57	PFC 56	PFC 55	PFC 54	PFC 53	PFC 52	PFC 51	PFC 50	PFC 47	PFC 46	PFC 45	PFC 44	PFC 43	PFC 42	PFC 41	PFC 40	Address 400A 3034H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H*1

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address / Initial value
PFC6H	PFC 77	PFC 76	PFC 75	PFC 74	PFC 73	PFC 72	PFC 71	PFC 70	PFC 67	PFC 66	PFC 65	PFC 64	PFC 63	PFC 62	PFC 61	PFC 60	Address 400A 3036H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address / Initial value
EXTPFC0H	0	EXTPFC14	EXTPFC13	EXTPFC12	EXTPFC11	EXTPFC10	EXTPFC9	EXTPFC8	EXTPFC7	EXTPFC6	EXTPFC5	EXTPFC4	EXTPFC3	EXTPFC2	EXTPFC1	EXTPFC0	Address 400A 3830H
	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address / Initial value
RPFC0H	RPFC 17	RPFC 16	RPFC 15	RPFC 14	RPFC 13	RPFC 12	RPFC 11	RPFC 10	RPFC 07	RPFC 06	RPFC 05	RPFC 04	RPFC 03	RPFC 02	RPFC 01	RPFC 00	Address 400A 3430H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H*1

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address / Initial value
RPFC2H	RPFC 37	RPFC 36	RPFC 35	RPFC 34	RPFC 33	RPFC 32	RPFC 31	RPFC 30	RPFC 27	RPFC 26	RPFC 25	RPFC 24	RPFC 23	RPFC 22	RPFC 21	RPFC 20	Address 400A 3432H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H*1

Bit Position	Bit Name	Description
15–0	PFCmn / RPFCln / EXTPFCp	These bits specify whether to use multiplexed functions 1 and 3 or multiplexed functions 2 and 4.*2 0: Multiplexed function 1 / Multiplexed function 3 1: Multiplexed function 2 / Multiplexed function 4

Figure 27.12 Port Function Control Registers (16-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 "Pin States".

2. The multiplexed functions are selected using the port mode control registers, port function control registers, and port function control expansion registers.
For details, refer to Section 27.5 "List of Selectable Multiplexed Functions".

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

PFC0W — Address 400A 3030H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30	PFC27	PFC26	PFC25	PFC24	PFC23	PFC22	PFC21	PFC20
R/W	R/W	R/W	R/W	R/W	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
Name	PFC17	PFC16	PFC15	PFC14	PFC13	PFC12	PFC11	PFC10	PFC07	PFC06	PFC05	PFC04	PFC03	PFC02	PFC01	PFC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFC4W — Address 400A 3034H, Initial value 0000 0000H*1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PFC77	PFC76	PFC75	PFC74	PFC73	PFC72	PFC71	PFC70	PFC67	PFC66	PFC65	PFC64	PFC63	PFC62	PFC61	PFC60
R/W	R/W	R/W	R/W	R/W	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
Name	PFC57	PFC56	PFC55	PFC54	PFC53	PFC52	PFC51	PFC50	PFC47	PFC46	PFC45	PFC44	PFC43	PFC42	PFC41	PFC40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EXTPFC0W — Address 400A 3830H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	EXTPFC14	EXTPFC13	EXTPFC12	EXTPFC11	EXTPFC10	EXTPFC9	EXTPFC8	EXTPFC7	EXTPFC6	EXTPFC5	EXTPFC4	EXTPFC3	EXTPFC2	EXTPFC1	EXTPFC0
R/W	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RPFC0W — Address 400A 3430H, Initial value 0000 0000H*1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RPFC37	RPFC36	RPFC35	RPFC34	RPFC33	RPFC32	RPFC31	RPFC30	RPFC27	RPFC26	RPFC25	RPFC24	RPFC23	RPFC22	RPFC21	RPFC20
R/W	R/W	R/W	R/W	R/W	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
Name	RPFC17	RPFC16	RPFC15	RPFC14	RPFC13	RPFC12	RPFC11	RPFC10	RPFC07	RPFC06	RPFC05	RPFC04	RPFC03	RPFC02	RPFC01	RPFC00
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–0	PFCmn / RPFCln / EXTPFCp	These bits specify whether to use multiplexed functions 1 and 3 or multiplexed functions 2 and 4.*2 0: Multiplexed function 1 / Multiplexed function 3 1: Multiplexed function 2 / Multiplexed function 4

Figure 27.13 Port Function Control Registers (32-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 “Pin States”.

2. The multiplexed functions are selected using the port mode control registers, port function control registers, and port function control expansion registers.
For details, refer to Section 27.5 “List of Selectable Multiplexed Functions”.

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

27.4.5 Port Function Control Expansion Registers (PFCE, RPFCE, EXTPFCE)

These registers are used to specify which multiplexed extended function is to be used. These registers can be set in 1-bit units.

	7	6	5	4	3	2	1	0	Address	Initial value
PFCE0B	PFCE07	PFCE06	PFCE05	PFCE04	PFCE03	PFCE02	PFCE01	PFCE00	400A 3040H	00H
PFCE1B	PFCE17	PFCE16	PFCE15	PFCE14	PFCE13	PFCE12	PFCE11	PFCE10	400A 3041H	00H
PFCE2B	PFCE27	PFCE26	PFCE25	PFCE24	PFCE23	PFCE22	PFCE21	PFCE20	400A 3042H	00H
PFCE3B	PFCE37	PFCE36	PFCE35	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30	400A 3043H	00H
PFCE4B	PFCE47	PFCE46	PFCE45	PFCE44	PFCE43	PFCE42	PFCE41	PFCE40	400A 3044H	00H
PFCE5B	PFCE57	PFCE56	PFCE55	PFCE54	PFCE53	PFCE52	PFCE51	PFCE50	400A 3045H	00H
PFCE6B	PFCE67	PFCE66	PFCE65	PFCE64	PFCE63	PFCE62	PFCE61	PFCE60	400A 3046H	00H
PFCE7B	PFCE77	PFCE76	PFCE75	PFCE74	PFCE73	PFCE72	PFCE71	PFCE70	400A 3047H	00H
EXTPFCE0B	EXTPFCE7	EXTPFCE6	EXTPFCE5	EXTPFCE4	EXTPFCE3	EXTPFCE2	EXTPFCE1	EXTPFCE0	400A 3840H	00H
EXTPFCE1B	0	EXTPFCE 14	EXTPFCE 13	EXTPFCE 12	EXTPFCE 11	EXTPFCE 10	EXTPFCE9	EXTPFCE8	400A 3841H	00H
RPFCE0B	RPFCE07	RPFCE06	RPFCE05	RPFCE04	RPFCE03	RPFCE02	RPFCE01	RPFCE00	400A 3440H	00H*1
RPFCE1B	RPFCE17	RPFCE16	RPFCE15	RPFCE14	RPFCE13	RPFCE12	RPFCE11	RPFCE10	400A 3441H	00H
RPFCE2B	RPFCE27	RPFCE26	RPFCE25	RPFCE24	RPFCE23	RPFCE22	RPFCE21	RPFCE20	400A 3442H	00H
RPFCE3B	RPFCE37	RPFCE36	RPFCE35	RPFCE34	RPFCE33	RPFCE32	RPFCE31	RPFCE30	400A 3443H	00H

Bit Position	Bit Name	Description
7–0	PFCEmn / RPFCEln / EXTPFCEp	These bits specify whether to use multiplexed functions 1 and 3 or multiplexed functions 2 and 4.*2 0: Multiplexed function 1 / Multiplexed function 2 1: Multiplexed function 3 / Multiplexed function 4

Figure 27.14 Port Function Control Expansion Registers (8-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 “Pin States”.

2. The multiplexed functions are selected using the port mode control registers, port function control registers, and port function control expansion registers.
For details, refer to Section 27.5 “List of Selectable Multiplexed Functions”.

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PFCE0H	PFCE17	PFCE16	PFCE15	PFCE14	PFCE13	PFCE12	PFCE11	PFCE10	PFCE07	PFCE06	PFCE05	PFCE04	PFCE03	PFCE02	PFCE01	PFCE00	Address 400A 3040H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PFCE2H	PFCE37	PFCE36	PFCE35	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30	PFCE27	PFCE26	PFCE25	PFCE24	PFCE23	PFCE22	PFCE21	PFCE20	Address 400A 3042H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PFCE4H	PFCE57	PFCE56	PFCE55	PFCE54	PFCE53	PFCE52	PFCE51	PFCE50	PFCE47	PFCE46	PFCE45	PFCE44	PFCE43	PFCE42	PFCE41	PFCE40	Address 400A 3044H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PFCE6H	PFCE77	PFCE76	PFCE75	PFCE74	PFCE73	PFCE72	PFCE71	PFCE70	PFCE67	PFCE66	PFCE65	PFCE64	PFCE63	PFCE62	PFCE61	PFCE60	Address 400A 3046H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
EXTPFCE0H	0	EXTPFCE14	EXTPFCE13	EXTPFCE12	EXTPFCE11	EXTPFCE10	EXTPFCE9	EXTPFCE8	EXTPFCE7	EXTPFCE6	EXTPFCE5	EXTPFCE4	EXTPFCE3	EXTPFCE2	EXTPFCE1	EXTPFCE0	Address 400A 3840H
	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RPFCE0H	RPFCE17	RPFCE16	RPFCE15	RPFCE14	RPFCE13	RPFCE12	RPFCE11	RPFCE10	RPFCE07	RPFCE06	RPFCE05	RPFCE04	RPFCE03	RPFCE02	RPFCE01	RPFCE00	Address 400A 3440H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H*1

Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RPFCE2H	RPFCE37	RPFCE36	RPFCE35	RPFCE34	RPFCE33	RPFCE32	RPFCE31	RPFCE30	RPFCE27	RPFCE26	RPFCE25	RPFCE24	RPFCE23	RPFCE22	RPFCE21	RPFCE20	Address 400A 3442H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 0000H

Bit Position	Bit Name	Description
15–0	PFCEmn / RPFCEln / EXTPFCEp	These bits specify whether to use multiplexed functions 1 and 3 or multiplexed functions 2 and 4.*2 0: Multiplexed function 1 / Multiplexed function 2 1: Multiplexed function 3 / Multiplexed function 4

Figure 27.15 Port Function Control Expansion Registers (16-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 "Pin States".

2. The multiplexed functions are selected using the port function control registers and port function control expansion registers.
For details, refer to Section 27.5 "List of Selectable Multiplexed Functions".

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

PFCE0W — Address 400A 3040H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PFCE37	PFCE36	PFCE35	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30	PFCE27	PFCE26	PFCE25	PFCE24	PFCE23	PFCE22	PFCE21	PFCE20
R/W	R/W	R/W	R/W	R/W	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
Name	PFCE17	PFCE16	PFCE15	PFCE14	PFCE13	PFCE12	PFCE11	PFCE10	PFCE07	PFCE06	PFCE05	PFCE04	PFCE03	PFCE02	PFCE01	PFCE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCE4W — Address 400A 3044H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PFCE77	PFCE76	PFCE75	PFCE74	PFCE73	PFCE72	PFCE71	PFCE70	PFCE67	PFCE66	PFCE65	PFCE64	PFCE63	PFCE62	PFCE61	PFCE60
R/W	R/W	R/W	R/W	R/W	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
Name	PFCE57	PFCE56	PFCE55	PFCE54	PFCE53	PFCE52	PFCE51	PFCE50	PFCE47	PFCE46	PFCE45	PFCE44	PFCE43	PFCE42	PFCE41	PFCE40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EXTPFCE0W — Address 400A 3840H, Initial value 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	EXTPFCE14	EXTPFCE13	EXTPFCE12	EXTPFCE11	EXTPFCE10	EXTPFCE9	EXTPFCE8	EXTPFCE7	EXTPFCE6	EXTPFCE5	EXTPFCE4	EXTPFCE3	EXTPFCE2	EXTPFCE1	EXTPFCE0
R/W	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RPFCE0W — Address 400A 3440H, Initial value 0000 0000H*1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RPFCE37	RPFCE36	RPFCE35	RPFCE34	RPFCE33	RPFCE32	RPFCE31	RPFCE30	RPFCE27	RPFCE26	RPFCE25	RPFCE24	RPFCE23	RPFCE22	RPFCE21	RPFCE20
R/W	R/W	R/W	R/W	R/W	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
Name	RPFCE17	RPFCE16	RPFCE15	RPFCE14	RPFCE13	RPFCE12	RPFCE11	RPFCE10	RPFCE07	RPFCE06	RPFCE05	RPFCE04	RPFCE03	RPFCE02	RPFCE01	RPFCE00
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–0	PFCEmn / RPFCEln / EXTPFCEp	These bits specify whether to use multiplexed functions 1 and 3 or multiplexed functions 2 and 4.*2 0: Multiplexed function 1 / Multiplexed function 2 1: Multiplexed function 3 / Multiplexed function 4

Figure 27.16 Port Function Control Expansion Registers (32-bit Notation)

Notes 1. The initial value depends on the state of the operation mode setting pins. For details, refer to Section 2.2 “Pin States”.

2. The multiplexed functions are selected using the port function control registers and port function control expansion registers.
For details, refer to Section 27.5 “List of Selectable Multiplexed Functions”.

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

27.4.6 Port Pin Input Registers (PIN, RPIN, EXTPIN)

These are read-only registers for reading the input level of port pins.

	7	6	5	4	3	2	1	0	Address	Initial value
PIN0B	PIN07	PIN06	PIN05	PIN04	PIN03	PIN02	PIN01	PIN00	400A 3050H	Pin level
PIN1B	PIN17	PIN16	PIN15	PIN14	PIN13	PIN12	PIN11	PIN10	400A 3051H	Pin level
PIN2B	PIN27	PIN26	PIN25	PIN24	PIN23	PIN22	PIN21	PIN20	400A 3052H	Pin level
PIN3B	PIN37	PIN36	PIN35	PIN34	PIN33	PIN32	PIN31	PIN30	400A 3053H	Pin level
PIN4B	PIN47	PIN46	PIN45	PIN44	PIN43	PIN42	PIN41	PIN40	400A 3054H	Pin level
PIN5B	PIN57	PIN56	PIN55	PIN54	PIN53	PIN52	PIN51	PIN50	400A 3055H	Pin level
PIN6B	PIN67	PIN66	PIN65	PIN64	PIN63	PIN62	PIN61	PIN60	400A 3056H	Pin level
PIN7B	PIN77	PIN76	PIN75	PIN74	PIN73	PIN72	PIN71	PIN70	400A 3057H	Pin level
EXTPIN0B	EXTPIN7	EXTPIN6	EXTPIN5	EXTPIN4	EXTPIN3	EXTPIN2	EXTPIN1	EXTPIN0	400A 3850H	Pin level
EXTPIN1B	0	EXTPIN14	EXTPIN13	EXTPIN12	EXTPIN11	EXTPIN10	EXTPIN9	EXTPIN8	400A 3851H	Pin level
RPIN0B	RPIN07	RPIN06	RPIN05	RPIN04	RPIN03	RPIN02	RPIN01	RPIN00	400A 3450H	Pin level
RPIN1B	RPIN17	RPIN16	RPIN15	RPIN14	RPIN13	RPIN12	RPIN11	RPIN10	400A 3451H	Pin level
RPIN2B	RPIN27	RPIN26	RPIN25	RPIN24	RPIN23	RPIN22	RPIN21	RPIN20	400A 3452H	Pin level
RPIN3B	RPIN37	RPIN36	RPIN35	RPIN34	RPIN33	RPIN32	RPIN31	RPIN30	400A 3453H	Pin level

Bit Position	Bit Name	Description
7–0	PINmn / RPINln / EXTPINp	These bits are for reading the input level of the port pins.

Figure 27.17 Port Pin Input Registers (8-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

PIN0H — Address 400A 3050H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PIN17	PIN16	PIN15	PIN14	PIN13	PIN12	PIN11	PIN10	PIN07	PIN06	PIN05	PIN04	PIN03	PIN02	PIN01	PIN00
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Initial value: Pin level

PIN2H — Address 400A 3052H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PIN37	PIN36	PIN35	PIN34	PIN33	PIN32	PIN31	PIN30	PIN27	PIN26	PIN25	PIN24	PIN23	PIN22	PIN21	PIN20
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Initial value: Pin level

PIN4H — Address 400A 3054H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PIN57	PIN56	PIN55	PIN54	PIN53	PIN52	PIN51	PIN50	PIN47	PIN46	PIN45	PIN44	PIN43	PIN42	PIN41	PIN40
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Initial value: Pin level

PIN6H — Address 400A 3056H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PIN77	PIN76	PIN75	PIN74	PIN73	PIN72	PIN71	PIN70	PIN67	PIN66	PIN65	PIN64	PIN63	PIN62	PIN61	PIN60
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Initial value: Pin level

EXTPIN0H — Address 400A 3850H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	EXTPIN14	EXTPIN13	EXTPIN12	EXTPIN11	EXTPIN10	EXTPIN9	EXTPIN8	EXTPIN7	EXTPIN6	EXTPIN5	EXTPIN4	EXTPIN3	EXTPIN2	EXTPIN1	EXTPIN0
	0	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Initial value: Pin level

RPIN0H — Address 400A 3450H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RPIN17	RPIN16	RPIN15	RPIN14	RPIN13	RPIN12	RPIN11	RPIN10	RPIN07	RPIN06	RPIN05	RPIN04	RPIN03	RPIN02	RPIN01	RPIN00
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Initial value: Pin level

RPIN2H — Address 400A 3452H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RPIN37	RPIN36	RPIN35	RPIN34	RPIN33	RPIN32	RPIN31	RPIN30	RPIN27	RPIN26	RPIN25	RPIN24	RPIN23	RPIN22	RPIN21	RPIN20
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Initial value: Pin level

Bit Position	Bit Name	Description
15–0	PINmn / RPINln / EXTPINp	These bits are for reading the input level of the port pins.

Figure 27.18 Port Pin Input Registers (16-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

PIN0W — Address: 400A 3050H, Initial value: Pin level

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PIN37	PIN36	PIN35	PIN34	PIN33	PIN32	PIN31	PIN30	PIN27	PIN26	PIN25	PIN24	PIN23	PIN22	PIN21	PIN20
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PIN17	PIN16	PIN15	PIN14	PIN13	PIN12	PIN11	PIN10	PIN07	PIN06	PIN05	PIN04	PIN03	PIN02	PIN01	PIN00
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

PIN4W — Address: 400A 3054H, Initial value: Pin level

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PIN77	PIN76	PIN75	PIN74	PIN73	PIN72	PIN71	PIN70	PIN67	PIN66	PIN65	PIN64	PIN63	PIN62	PIN61	PIN60
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PIN57	PIN56	PIN55	PIN54	PIN53	PIN52	PIN51	PIN50	PIN47	PIN46	PIN45	PIN44	PIN43	PIN42	PIN41	PIN40
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

EXTPIN0W — Address: 400A 3850H, Initial value: Pin level

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	EXTPIN14	EXTPIN13	EXTPIN12	EXTPIN11	EXTPIN10	EXTPIN9	EXTPIN8	EXTPIN7	EXTPIN6	EXTPIN5	EXTPIN4	EXTPIN3	EXTPIN2	EXTPIN1	EXTPIN0
R/W	0	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

RPIN0W — Address: 400A 3450H, Initial value: Pin level

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RPIN37	RPIN36	RPIN35	RPIN34	RPIN33	RPIN32	RPIN31	RPIN30	RPIN27	RPIN26	RPIN25	RPIN24	RPIN23	RPIN22	RPIN21	RPIN20
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RPIN17	RPIN16	RPIN15	RPIN14	RPIN13	RPIN12	RPIN11	RPIN10	RPIN07	RPIN06	RPIN05	RPIN04	RPIN03	RPIN02	RPIN01	RPIN00
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Description
31–0	PINmn / RPINln / EXTPINp	These bits are for reading the input level of the port pins.

Figure 27.19 Port Pin Input Registers (32-bit Notation)

Remark: l = 0–3, m = 0–7, n = 0–7, p = 0–14

27.5 List of Selectable Multiplexed Functions

The following tables provide the combinations of multiplexed functions that can be specified by using the port-related registers.

Cautions . Operation is not guaranteed if the setting has been made not to allocate a multiplexed pin.

(1) Ports (P00–P77)

(1/3)

Pin Name	PKG		PMCmn = 0 (Port Mode)		PMCmn = 1 (Control Mode)			
					PFCEmn = 0		PFCEmn = 1	
	23□	17□	PMmn = 0 (Output Port)	PMmn = 1 (Input Port))	PFCmn = 0 (Multiplexed Function 1)	PFCmn = 1 (Multiplexed Function 2)	PFCmn = 0 (Multiplexed Function 3)	PFCmn = 1 (Multiplexed Function 4)
P00	○	○	P00 (output mode)	P00 (input mode)	INTPZ0	—	CCI_RUNLEDZ	—
P01	○	○	P01 (output mode)	P01 (input mode)	INTPZ1	—	—	—
P02	○	○	P02 (output mode)	P02 (input mode)	INTPZ2	—	CCI_DLINKLEDZ	—
P03	○	○	P03 (output mode)	P03 (input mode)	INTPZ3	—	CCI_ERRLEDZ	—
P04	○	○	P04 (output mode)	P04 (input mode)	INTPZ4	—	CCI_LERR1LEDZ	—
P05	○	○	P05 (output mode)	P05 (input mode)	INTPZ5	—	CCI_LERR2LEDZ	—
P06	○	×	P06 (output mode)	P06 (input mode)	—	—	CCI_SDLEDZ	—
P07	○	×	P07 (output mode)	P07 (input mode)	—	—	CCI_RDLEDZ	—
P10	○	×	P10 (output mode)	P10 (input mode)	SMIO2	—	—	—
P11	○	×	P11 (output mode)	P11 (input mode)	SMIO3	—	—	—
P12	○	×	P12 (output mode)	P12 (input mode)	CSZ3	—	CCI_WDTIZ	—
P13	○	×	P13 (output mode)	P13 (input mode)	CSZ2	—	—	—
P14	○	○	P14 (output mode)	P14 (input mode)	SMSCK	—	—	—
P15	○	○	P15 (output mode)	P15 (input mode)	SMIO0	—	—	—
P16	○	○	P16 (output mode)	P16 (input mode)	SMIO1	—	—	—
P17	○	○	P17 (output mode)	P17 (input mode)	SMCSZ	—	—	—
P20	○	○	P20 (output mode)	P20 (input mode)	RXD0	—	—	—
P21	○	○	P21 (output mode)	P21 (input mode)	TXD0	—	—	—
P22	○	○	P22 (output mode)	P22 (input mode)	INTPZ8	—	—	—
P23	○	○	P23 (output mode)	P23 (input mode)	INTPZ9	—	—	—
P24	○	○	P24 (output mode)	P24 (input mode)	INTPZ10	ETHSWSYNCOUT	—	—
P25	○	○	P25 (output mode)	P25 (input mode)	WDTOUTZ	—	—	—
P26	○	○	P26 (output mode)	P26 (input mode)	TINJ1/TIND5	TOUTJ1/ TOUTD5	—	—
P27	○	○	P27 (output mode)	P27 (input mode)	TINJ0/TIND4	TOUTJ0/ TOUTD4	—	—

Remark: m = 0–7, n = 0–7

(2/3)

Pin Name	PKG 23□	PKG 17□	PMCmn = 0 (Port Mode) PMmn = 0 (Output Port)	PMmn = 1 (Input Port)	PMCmn = 1 (Control Mode) PFCEmn = 0 PFCmn = 0 (Multiplexed Function 1)	PFCmn = 1 (Multiplexed Function 2)	PFCEmn = 1 PFCmn = 0 (Multiplexed Function 3)	PFCmn = 1 (Multiplexed Function 4)
P30	○	○	P30 (output mode)	P30 (input mode)	RXD1	—	—	—
P31	○	○	P31 (output mode)	P31 (input mode)	TXD1	—	—	—
P32	○	○	P32 (output mode)	P32 (input mode)	DMAREQZ1	—	—	—
P33	○	○	P33 (output mode)	P33 (input mode)	DMAACKZ1	—	—	—
P34	○	○	P34 (output mode)	P34 (input mode)	DMATCZ1	—	—	—
P35	○	○	P35 (output mode)	P35 (input mode)	CSISCK1	INTPZ22	—	—
P36	○	○	P36 (output mode)	P36 (input mode)	CSISI1	INTPZ23	—	—
P37	○	○	P37 (output mode)	P37 (input mode)	CSISO1	INTPZ24	—	—
P40	○	○	P40 (output mode)	P40 (input mode)	A1	HA1	—	—
P41	○	○	P41 (output mode)	P41 (input mode)	WAITZ	HWAITZ	INTPZ29	—
P42	○	○	P42 (output mode)	P42 (input mode)	CSICS00	HERROUTZ	—	—
P43	○	○	P43 (output mode)	P43 (input mode)	CSICS01	HBUSCLK	—	—
P44	○	○	P44 (output mode)	P44 (input mode)	CSZ1	HPGCSZ	—	—
P45	○	○	P45 (output mode)	P45 (input mode)	CSISCK0	WAITZ1	—	—
P46	○	○	P46 (output mode)	P46 (input mode)	CSISI0	WAITZ2	—	—
P47	○	○	P47 (output mode)	P47 (input mode)	CSISO0	WAITZ3	—	—
P50	○	○	P50 (output mode)	P50 (input mode)	INTPZ6	—	—	—
P51	○	○	P51 (output mode)	P51 (input mode)	INTPZ7	—	—	—
P52	○	○	P52 (output mode)	P52 (input mode)	TINJ3/TIND7	TOUTJ3/ TOUTD7	CCI_NMIZ	—
P53	○	×	P53 (output mode)	P53 (input mode)	CRXD0	CCI_INTZ	—	—
P54	○	×	P54 (output mode)	P54 (input mode)	CTXD0	—	—	—
P55	○	×	P55 (output mode)	P55 (input mode)	CRXD1	—	—	—
P56	○	×	P56 (output mode)	P56 (input mode)	CTXD1	—	—	—
P57	○	○	P57 (output mode)	P57 (input mode)	TINJ2/TIND6	TOUTJ2/ TOUTD6	—	—

Remark: m = 0–7, n = 0–7

(3/3)

Pin Name	PKG 23□	PKG 17□	PMCmn = 0 (Port Mode) PMmn = 0 (Output Port)	PMCmn = 0 (Port Mode) PMmn = 1 (Input Port)	PMCmn = 1 (Control Mode) PFCEmn = 0 PFCmn = 0 (Multiplexed Function 1)	PFCEmn = 0 PFCmn = 1 (Multiplexed Function 2)	PFCEmn = 1 PFCmn = 0 (Multiplexed Function 3)	PFCEmn = 1 PFCmn = 1 (Multiplexed Function 4)
P60	○	○	P60 (output mode)	P60 (input mode)	SCL0	—	—	—
P61	○	○	P61 (output mode)	P61 (input mode)	SDA0	—	—	—
P62	○	○	P62 (output mode)	P62 (input mode)	RTDMAREQZ	—	—	—
P63	○	○	P63 (output mode)	P63 (input mode)	RTDMAACKZ	—	—	—
P64	○	○	P64 (output mode)	P64 (input mode)	RTDMATCZ	—	—	—
P65	○	○	P65 (output mode)	P65 (input mode)	DMAREQZ0	—	—	—
P66	○	○	P66 (output mode)	P66 (input mode)	DMAACKZ0	—	—	—
P67	○	○	P67 (output mode)	P67 (input mode)	DMATCZ0	—	—	—
P70	○	○	P70 (output mode)	P70 (input mode)	CSICS10	—	—	—
P71	○	○	P71 (output mode)	P71 (input mode)	CSICS11	—	—	—
P72	○	○	P72 (output mode)	P72 (input mode)	SLEEPING	—	—	—
P73	○	○	P73 (output mode)	P73 (input mode)	INTPZ11	—	—	—
P74	○	○	P74 (output mode)	P74 (input mode)	INTPZ12	—	—	—
P75	○	○	P75 (output mode)	P75 (input mode)	INTPZ13	—	—	—
P76	○	○	P76 (output mode)	P76 (input mode)	INTPZ14	—	—	—
P77	○	○	P77 (output mode)	P77 (input mode)	INTPZ15	—	—	—

Remark: m = 0–7, n = 0–7

(2) Real-Time Ports (RP00–RP37)

(1/2)

Pin Name	PKG		RPMCmn = 0 (Port Mode)		RPMCmn = 1 (Control Mode)			
					RPFCEmn = 0		RPFCEmn = 1	
	23□	17□	RPMmn = 0 (Output Port)	RPMmn = 1 (Input Port)	RPFCmn = 0 (Multiplexed Function 1)	RPFCmn = 1 (Multiplexed Function 2)	RPFCmn = 0 (Multiplexed Function 3)	RPFCmn = 1 (Multiplexed Function 4)
RP00	○	○	RP00 (output mode)	RP00 (input mode)	INTPZ16	SCL1	—	—
RP01	○	○	RP01 (output mode)	RP01 (input mode)	INTPZ17	SDA1	—	—
RP02	○	○	RP02 (output mode)	RP02 (input mode)	INTPZ18	—	—	—
RP03	○	○	RP03 (output mode)	RP03 (input mode)	INTPZ19	—	—	—
RP04	○	○	RP04 (output mode)	RP04 (input mode)	INTPZ20	—	—	—
RP05	○	○	RP05 (output mode)	RP05 (input mode)	INTPZ21	—	—	—
RP06	○	○	RP06 (output mode)	RP06 (input mode)	WRZ2/ BENZ2	HWRZ2/ HBENZ2	—	—
RP07	○	○	RP07 (output mode)	RP07 (input mode)	WRZ3/ BENZ3	HWRZ3/ HBENZ3	—	—
RP10	○	○	RP10 (output mode)	RP10 (input mode)	D24/MD24/HD24	LED0_PHY0	—	—
RP11	○	○	RP11 (output mode)	RP11 (input mode)	D25/MD25/HD25	LED1_PHY0	—	—
RP12	○	○	RP12 (output mode)	RP12 (input mode)	D26/MD26/HD26	LED2_PHY0	—	—
RP13	○	○	RP13 (output mode)	RP13 (input mode)	D27/MD27/HD27	LED3_PHY0	—	—
RP14	○	○	RP14 (output mode)	RP14 (input mode)	D28/MD28/HD28	LED0_PHY1	—	—
RP15	○	○	RP15 (output mode)	RP15 (input mode)	D29/MD29/HD29	LED1_PHY1	—	—
RP16	○	○	RP16 (output mode)	RP16 (input mode)	D30/MD30/HD30	LED2_PHY1	—	—
RP17	○	○	RP17 (output mode)	RP17 (input mode)	D31/MD31/HD31	LED3_PHY1	—	—

Remark: m = 0–3, n = 0–7

(2/2)

Pin Name	PKG 23☐	PKG 17☐	RPMCmn = 0 (Port Mode) RPMmn = 0 (Output Port)	RPMCmn = 0 (Port Mode) RPMmn = 1 (Input Port)	RPMCmn = 1 (Control Mode) RPFCEmn = 0 RPFCmn = 0 (Multiplexed Function 1)	RPMCmn = 1 (Control Mode) RPFCEmn = 0 RPFCmn = 1 (Multiplexed Function 2)	RPMCmn = 1 (Control Mode) RPFCEmn = 1 RPFCmn = 0 (Multiplexed Function 3)	RPMCmn = 1 (Control Mode) RPFCEmn = 1 RPFCmn = 1 (Multiplexed Function 4)
RP20	○	○	RP20 (output mode)	RP20 (input mode)	BCYSTZ / ADVZ	HBCYSTZ	—	—
RP21	○	○	RP21 (output mode)	RP21 (input mode)	A21/MA20	—	—	—
RP22	○	○	RP22 (output mode)	RP22 (input mode)	A22/MA21	—	—	—
RP23	○	○	RP23 (output mode)	RP23 (input mode)	A23/MA22	—	—	—
RP24	○	○	RP24 (output mode)	RP24 (input mode)	A24/MA23	INTPZ25	—	—
RP25	○	○	RP25 (output mode)	RP25 (input mode)	A25/MA24	INTPZ26	—	—
RP26	○	○	RP26 (output mode)	RP26 (input mode)	A26/MA25	INTPZ27	—	—
RP27	○	○	RP27 (output mode)	RP27 (input mode)	A27/MA26	INTPZ28	—	—
RP30	○	○	RP30 (output mode)	RP30 (input mode)	D16/HD16	TOUTD8	TIND8	—
RP31	○	○	RP31 (output mode)	RP31 (input mode)	D17/HD17	TOUTD9	TIND9	—
RP32	○	○	RP32 (output mode)	RP32 (input mode)	D18/HD18	TOUTD10	TIND10	—
RP33	○	○	RP33 (output mode)	RP33 (input mode)	D19/HD19	TOUTD11	TIND11	—
RP34	○	○	RP34 (output mode)	RP34 (input mode)	D20/HD20	TOUTD12	TIND12	—
RP35	○	○	RP35 (output mode)	RP35 (input mode)	D21/HD21	TOUTD13	TIND13	—
RP36	○	○	RP36 (output mode)	RP36 (input mode)	D22/HD22	TOUTD14	TIND14	—
RP37	○	○	RP37 (output mode)	RP37 (input mode)	D23/HD23	TOUTD15	TIND15	—

Remark: m = 0–3, n = 0–7

(3) EXT Ports (EXTP0–EXTP14)

Pin Name	PKG		EXTPMCp = 0 (Port Mode)		EXTPMCp = 1 (Control Mode)			
					EXTPFCEp = 0		EXTPFCEp = 1	
	23□	17□	EXTPMp = 0 (Output Port)	EXTPMp = 1 (Input Port)	EXTPFCp = 0 (Multiplexed Function 1)	EXTPFCp = 1 (Multiplexed Function 2)	EXTPFCp = 0 (Multiplexed Function 3)	EXTPFCp = 1 (Multiplexed Function 4)
EXTP0	○	○	EXTP0 (output mode)	EXTP0 (input mode)	—	TOUTD0	—	TIND0
EXTP1	○	○	EXTP1 (output mode)	EXTP1 (input mode)	—	TOUTD1	—	TIND1
EXTP2	○	○	EXTP2 (output mode)	EXTP2 (input mode)	—	TOUTD2	—	TIND2
EXTP3	○	○	EXTP3 (output mode)	EXTP3 (input mode)	WDTOUTZ	TOUTD3	—	TIND3
EXTP4	○	○	EXTP4 (output mode)	EXTP4 (input mode)	—	—	—	—
EXTP5	○	○	EXTP5 (output mode)	EXTP5 (input mode)	—	—	—	—
EXTP6	○	○	EXTP6 (output mode)	EXTP6 (input mode)	—	—	—	—
EXTP7	○	○	EXTP7 (output mode)	EXTP7 (input mode)	—	—	—	—
EXTP8	○	○	EXTP8 (output mode)	EXTP8 (input mode)	—	—	—	—
EXTP9	○	○	EXTP9 (output mode)	EXTP9 (input mode)	—	—	—	—
EXTP10	×	○	EXTP10 (output mode)	EXTP10 (input mode)	SMIO2	CCI_INTZ	—	—
EXTP11	×	○	EXTP11 (output mode)	EXTP11 (input mode)	SMIO3	CCI_WDTIZ	—	—
EXTP12	×	○	EXTP12 (output mode)	EXTP12 (input mode)	CSZ3	CCI_SDLEDZ	—	—
EXTP13	×	○	EXTP13 (output mode)	EXTP13 (input mode)	CSZ2	CCI_RDLEDZ	—	—
EXTP14	×	○	EXTP14 (output mode)	EXTP14 (input mode)	IETYPE_LED	—	—	—

Remark: p = 0–14

27.6 Buffer Switching Register (DRCTL)

For some port pins, the driving ability and the connection or disconnection of a pull-up or pull-down resistor is programmable.
Set up the DRCTL registers during initialization by a program after release from the reset state. After that, change the setting of a given DRCTL register only while the corresponding pins are not in use. For example, change the setting while only internal access is proceeding.
The settings of the DRCTL registers are effective for output pins regardless of their operating mode (port mode, or control mode, in which a multiplexed function is used).

- This register can be read and written in 32- or 16-bit units.

Cautions 1. This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)".
No special sequence is required for reading the register.

2. Take special care with pins in the high-impedance state, since changing the settings for the pull-up and pull-down resistors will change the levels on the pins.

27.6.1 Port 0 Buffer Switching Registers (DRCTLP0L, DRCTLP0H)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLP0L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOP03	PDIOP03	IOLP031	IOLP030	PUIOP02	PDIOP02	IOLP021	IOLP020	PUIOP01	PDIOP01	IOLP011	IOLP010	PUIOP00	PDIOP00	IOLP001	IOLP000	BASE + 0220H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLP0H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOP07	PDIOP07	IOLP071	IOLP070	PUIOP06	PDIOP06	IOLP061	IOLP060	PUIOP05	PDIOP05	IOLP051	IOLP050	PUIOP04	PDIOP04	IOLP041	IOLP040	BASE + 0224H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIOP0n, PDIOP0n	These bits specify whether to connect a pull-up or pull-down resistor to the P07–P00 pins.
13,12,9,8, 5,4,1,0	IOLP0n1, IOLP0n0	These bits specify the driving ability of the P07–P00 pins.

PUIOP0n	PDIOP0n	Connection of a Pull-Up or Pull-Down Resistor to the P07–P00 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLP0n1	IOLP0n0	Driving Ability of the P07–P00 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: n = 7–0

27.6.2 Port 1 Buffer Switching Registers (DRCTLP1L, DRCTLP1H)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLP1L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOP13	PDIOP13	0	1	PUIOP12	PDIOP12	0	1	PUIOP11	PDIOP11	0	1	PUIOP10	PDIOP10	0	1	BASE + 0228H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLP1H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOP17	PDIOP17	0	1	PUIOP16	PDIOP16	0	1	PUIOP15	PDIOP15	0	1	PUIOP14	PDIOP14	0	1	BASE + 022CH Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	

Bit Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIOP1n, PDIOP1n	These bits specify whether to connect a pull-up or pull-down resistor to the P17–P10 pins.

PUIOP1n	PDIOP1n	Connection of a Pull-Up or Pull-Down Resistor to the P17–P10 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

Remark: n = 7–0

27.6.3 Port 2 Buffer Switching Registers (DRCTLP2L, DRCTLP2H)

DRCTLP2L — Address: BASE + 0230H, Initial value: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLP2L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLP2L	PUIOP23	PDIOP23	IOLP231	IOLP230	PUIOP22	PDIOP22	IOLP221	IOLP220	PUIOP21	PDIOP21	IOLP211	IOLP210	PUIOP20	PDIOP20	IOLP201	IOLP200
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DRCTLP2H — Address: BASE + 0234H, Initial value: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLP2H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLP2H	PUIOP27	PDIOP27	IOLP271	IOLP270	PUIOP26	PDIOP26	IOLP261	IOLP260	PUIOP25	PDIOP25	IOLP251	IOLP250	PUIOP24	PDIOP24	0	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	0	1

Bit Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIOP2n, PDIOP2n	These bits specify whether to connect a pull-up or pull-down resistor to the P27–P20 pins.
13,12,9,8, 5,4,1,0	IOLP2m1, IOLP2m0	These bits specify the driving ability of the P27–P25 and P23–P20 pins.

PUIOP2n	PDIOP2n	Connection of a Pull-Up or Pull-Down Resistor to the P27–P20 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLP2m1	IOLP2m0	Driving Ability of the P27–P25 and P23– P20 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: m = 7–5, 3–0, n = 7–0

27.6.4 Port 3 Buffer Switching Registers (DRCTLP3L, DRCTLP3H)

DRCTLP3L (Address: BASE + 0238H, Initial value: 0000 9999H)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLP3L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLP3L	PUIOP33	PDIOP33	IOLP331	IOLP330	PUIOP32	PDIOP32	IOLP321	IOLP320	PUIOP31	PDIOP31	IOLP311	IOLP310	PUIOP30	PDIOP30	IOLP301	IOLP300
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DRCTLP3H (Address: BASE + 023CH, Initial value: 0000 5959H)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLP3H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLP3H	PUIOP37	PDIOP37	IOLP371	IOLP370	PUIOP36	PDIOP36	0	1	PUIOP35	PDIOP35	IOLP351	IOLP350	PUIOP34	PDIOP34	IOLP341	IOLP340
R/W	R/W	R/W	R/W	R/W	R/W	R/W	0	1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIOP3n, PDIOP3n	These bits specify whether to connect a pull-up or pull-down resistor to the P37–P30 pins.
13,12,9,8, 5,4,1,0	IOLP3m1, IOLP3m0	These bits specify the driving ability of the P37and P35–P30 pins.

PUIOP3n	1.PDIOP3n	Connection of a Pull-Up or Pull-Down Resistor to the P37–P30 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLP3m1	IOLP3m0	Driving Ability of the P37 and P35–P30 pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: m = 7, 5–0, n = 7–0

27.6.5 Port 4 Buffer Switching Registers (DRCTLP4L, DRCTLP4H)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLP4L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOP43	PDIOP43	0	1	PUIOP42	PDIOP42	0	1	PUIOP41	PDIOP41	0	1	PUIOP40	PDIOP40	0	1	BASE + 0240H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLP4H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOP47	PDIOP47	0	1	PUIOP46	PDIOP46	0	1	PUIOP45	PDIOP45	0	1	PUIOP44	PDIOP44	IOLP441	IOLP440	BASE + 0244H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	R/W	R/W	

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>31–16</td><td>—</td><td>Reserved (When writing to these bits, write 0. When read, 0 is returned.)</td></tr>
<tr><td>15,14,11,
10,7,6,3,2</td><td>PUIOP4n,
PDIOP4n</td><td>These bits specify whether to connect a pull-up or pull-down resistor to the P47–P40 pins.
<table>
<tr><th>PUIOP4n</th><th>PDIOP4n</th><th>Connection of a Pull-Up or Pull-Down Resistor to the P47–P40 Pins</th></tr>
<tr><td>0</td><td>0</td><td>Do not connect a pull-up or pull-down resistor.</td></tr>
<tr><td>0</td><td>1</td><td>Connect a pull-down resistor.</td></tr>
<tr><td>1</td><td>0</td><td>Connect a pull-up resistor.</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>1,0</td><td>IOLP441,
IOLP440</td><td>These bits specify the driving ability of the P44 pin.
<table>
<tr><th>IOLP441</th><th>IOLP440</th><th>Driving Ability of the P44 Pin</th></tr>
<tr><td>0</td><td>1</td><td>8 mA (recommended)</td></tr>
<tr><td>1</td><td>1</td><td>12 mA</td></tr>
<tr><td colspan="2">Other than the above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Remark: n = 7–0

27.6.6 Port 5 Buffer Switching Registers (DRCTLP5L, DRCTLP5H)

DRCTLP5L — Address: BASE + 0248H, Initial value: 0000 0959H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLP5L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLP5L	0	0	0	0	PUIOP52	PDIOP52	0	1	PUIOP51	PDIOP51	0	1	PUIOP50	PDIOP50	IOLP501	IOLP500
R/W	0	0	0	0	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	R/W	R/W

DRCTLP5H — Address: BASE + 024CH, Initial value: 0000 9000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLP5H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLP5H	PUIOP57	PDIOP57	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIOP5n, PDIOP5n	These bits specify whether to connect a pull-up or pull-down resistor to the P57 and P52–P50 pins.
1,0	IOLP501, IOLP500	These bits specify the driving ability of the P50 pin.

PUIOP5n	PDIOP5n	Connection of a Pull-Up or Pull-Down Resistor to the P57 and P52–P50 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLP501	IOLP500	Driving Ability of the P50 Pin
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: n = 7, 2–0

27.6.7 Port 6 Buffer Switching Registers (DRCTLP6L, DRCTLP6H)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLP6L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOP63	PDIOP63	IOLP631	IOLP630	PUIOP62	PDIOP62	IOLP621	IOLP620	PUIOP61	PDIOP61	IOLP611	IOLP610	PUIOP60	PDIOP60	IOLP601	IOLP600	BASE + 0250H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLP6H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOP67	PDIOP67	IOLP671	IOLP670	PUIOP66	PDIOP66	IOLP661	IOLP660	PUIOP65	PDIOP65	IOLP651	IOLP650	PUIOP64	PDIOP64	IOLP641	IOLP640	BASE + 0254H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIOP6n, PDIOP6n	These bits specify whether to connect a pull-up or pull-down resistor to the P67–P60 pins.
13,12,9,8, 5,4,1,0	IOLP6n1, IOLP6n0	These bits specify the driving ability of the P37and P67–P60 pins.

PUIOP6n	PDIOP6n	Connection of a Pull-Up or Pull-Down Resistor to the P67–P60 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLP6n1	IOLP6n0	Driving Ability of the P67–P60 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: n = 7–0

27.6.8 Port 7 Buffer Switching Registers (DRCTLP7L, DRCTLP7H)

DRCTLP7L — Address: BASE + 0258H, Initial value: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLP7L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLP7L	PUIOP73	PDIOP73	IOLP731	IOLP730	PUIOP72	PDIOP72	IOLP721	IOLP720	PUIOP71	PDIOP71	IOLP711	IOLP710	PUIOP70	PDIOP70	IOLP701	IOLP700
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DRCTLP7H — Address: BASE + 025CH, Initial value: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLP7H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLP7H	PUIOP77	PDIOP77	IOLP771	IOLP770	PUIOP76	PDIOP76	IOLP761	IOLP760	PUIOP75	PDIOP75	IOLP751	IOLP750	PUIOP74	PDIOP74	IOLP741	IOLP740
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIOP7n, PDIOP7n	These bits specify whether to connect a pull-up or pull-down resistor to the P77–P70 pins.
13,12,9,8, 5,4,1,0	IOLP7n1, IOLP7n0	These bits specify the driving ability of the P37and P77–P70 pins.

PUIOP7n	PDIOP7n	Connection of a Pull-Up or Pull-Down Resistor to the P77–P70 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLP7n1	IOLP7n0	Driving Ability of the P77–P70 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: n = 7–0

27.6.9 EXT Port 0 Buffer Switching Registers (DRCTLEXTP0L, DRCTLEXTP0H)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTL EXTP0L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOE03	PDIOE03	0	1	PUIOE02	PDIOE02	0	1	PUIOE01	PDIOE01	0	1	PUIOE00	PDIOE00	IOLE001	IOLE000	BASE + 0280H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	R/W	R/W	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTL EXTP0H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOE07	PDIOE07	IOLE071	IOLE070	PUIOE06	PDIOE06	IOLE061	IOLE060	PUIOE05	PDIOE05	0	1	PUIOE04	PDIOE04	0	1	BASE + 0284H Initial value 0000 9599H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0	1	R/W	R/W	0	1	

Bit Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIOE0n, PDIOE0n	These bits specify whether to connect a pull-up or pull-down resistor to the EXTP7–EXTP0 pins.
13,12,9,8, 1,0	IOLE0m1, IOLE0m0	These bits specify the driving ability of the EXTP7–EXTP6 and EXTP0 pins.

PUIOE0n	PDIOE0n	Connection of a Pull-Up or Pull-Down Resistor to the EXTP7–EXTP0 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLE0m1	IOLE0m0	Driving Ability of the EXTP7–EXTP6 and EXTP0 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: m = 7, 6, 0, n = 7–0

27.6.10 EXT Port 1 Buffer Switching Registers (DRCTLEXTP1L, DRCTLEXTP1H)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLEXTP1L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOE11	PDIOE11	IOLE111	IOLE110	PUIOE10	PDIOE10	IOLE101	IOLE100	PUIOE09	PDIOE09	0	1	PUIOE08	PDIOE08	0	1
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0	1	R/W	R/W	0	1

Address: BASE + 0288H
Initial value: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLEXTP1H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIOE14	PDIOE14	IOLE141	IOLE140	PUIOE13	PDIOE13	IOLE131	IOLE130	PUIOE12	IOLE100	IOLE121	IOLE120
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: BASE + 028CH
Initial value: 0000 0999H

Bit Position	Bit Name	Description
31–8	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
7,6,3,2	PUIOEn, PDIOEn	These bits specify whether to connect a pull-up or pull-down resistor to the EXTP14–EXTP8 pins.
13,12,9,8, 5,4,1,0	IOLE1m1, IOLE1m0	These bits specify the driving ability of the EXTP10–EXTP14 pins.

PUIOEn	PDIOEn	Connection of a Pull-Up or Pull-Down Resistor to the EXTP14–EXTP8 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLE1m1	IOLE1m0	Driving Ability of the EXTP10–EXTP14 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: n = 14–08

27.6.11 Real-Time Port 0 Buffer Switching Registers (DRCTLRP0L, DRCTLRP0H)

DRCTLRP0L — Address: BASE + 0260H, Initial value: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLRP0L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLRP0L	PUIORP03	PDIORP03	IOLRP031	IOLRP030	PUIORP02	PDIORP02	IOLRP021	IOLRP020	PUIORP01	PDIORP01	IOLRP011	IOLRP010	PUIORP00	PDIORP00	IOLRP001	IOLRP000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DRCTLRP0H — Address: BASE + 0264H, Initial value: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLRP0H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLRP0H	PUIORP07	PDIORP07	0	1	PUIORP06	PDIORP06	0	1	PUIORP05	PDIORP05	0	1	PUIORP04	PDIORP04	IOLRP041	IOLRP040
R/W	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIORP0n, PDIORP0n	These bits specify whether to connect a pull-up or pull-down resistor to the RP07–RP00 pins.
13,12,9,8, 5,4,1,0	IOLRP0m1, IOLRP0m0	These bits specify the driving ability of the RP04–RP00 pins.

PUIORP0n	PDIORP0n	Connection of a Pull-Up or Pull-Down Resistor to the RP07–RP00 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLRP0m1	IOLRP0m0	Driving Ability of the RP04–RP00 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: m = 4–0, n = 7–0

27.6.12 Real-Time Port 1 Buffer Switching Registers (DRCTLRP1L, DRCTLRP1H)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLRP1L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIORP13	PDIORP13	IOLRP131	IOLRP130	PUIORP12	PDIORP12	IOLRP121	IOLRP120	PUIORP11	PDIORP11	IOLRP111	IOLRP110	PUIORP10	PDIORP10	IOLRP101	IOLRP100	BASE + 0268H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLRP1H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIORP17	PDIORP17	IOLRP171	IOLRP170	PUIORP16	PDIORP16	IOLRP161	IOLRP160	PUIORP15	PDIORP15	IOLRP151	IOLRP150	PUIORP14	PDIORP14	IOLRP141	IOLRP140	BASE + 026CH Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIORP1n, PDIORP1n	These bits specify whether to connect a pull-up or pull-down resistor to the RP17–RP10 pins.
13,12,9,8, 5,4,1,0	IOLRP1n1, IOLRP1n0	These bits specify the driving ability of the RP17–RP10 pins.

PUIORP1n	PDIORP1n	Connection of a Pull-Up or Pull-Down Resistor to the RP17–RP10 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLRP1n1	IOLRP1n0	Driving Ability of the RP17–RP10 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: n = 7–0

27.6.13 Real-Time Port 2 Buffer Switching Registers (DRCTLRP2L, DRCTLRP2H)

DRCTLRP2L — Address: BASE + 0270H; Initial value 23□: 0000 5559H, 17□: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLRP2L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLRP2L	PUIORP23	PDIORP23	IOLRP231	IOLRP230	PUIORP22	PDIORP22	IOLRP221	IOLRP220	PUIORP21	PDIORP21	IOLRP211	IOLRP210	PUIORP20	PDIORP20	IOLRP201	IOLRP200
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DRCTLRP2H — Address: BASE + 0274H; Initial value 23□: 0000 5555H, 17□: 0000 9999H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRCTLRP2H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRCTLRP2H	PUIORP27	PDIORP27	IOLRP271	IOLRP270	PUIORP26	PDIORP26	IOLRP261	IOLRP260	PUIORP25	PDIORP25	IOLRP251	IOLRP250	PUIORP24	PDIORP24	IOLRP241	IOLRP240
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15,14,11, 10,7,6,3,2	PUIORP2n, PDIORP2n	These bits specify whether to connect a pull-up or pull-down resistor to the RP27–RP20 pins.
13,12,9,8, 5,4,1,0	IOLRP2n1, IOLRP2n0	These bits specify the driving ability of the RP27–RP20 pins.

PUIORP2n	PDIORP2n	Connection of a Pull-Up or Pull-Down Resistor to the RP27–RP20 Pins
0	0	Do not connect a pull-up or pull-down resistor.
0	1	Connect a pull-down resistor.
1	0	Connect a pull-up resistor.
1	1	Setting prohibited

IOLRP2n1	IOLRP2n0	Driving Ability of the RP27–RP20 Pins
0	1	8 mA (recommended)
1	1	12 mA
Other than the above		Setting prohibited

Remark: n = 7–0

27.6.14 Real-Time Port 3 Buffer Switching Registers (DRCTLRP3L, DRCTLRP3H)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLRP3L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIORP33	PDIORP33	IOLRP331	IOLRP330	PUIORP32	PDIORP32	0	1	PUIORP31	PDIORP31	0	1	PUIORP30	PDIORP30	0	1	BASE + 0278H Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DRCTLRP3H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PUIORP37	PDIORP37	0	1	PUIORP36	PDIORP36	0	1	PUIORP35	PDIORP35	0	1	PUIORP34	PDIORP34	IOLRP341	IOLRP340	BASE + 027CH Initial value 0000 9999H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	0	1	R/W	R/W	R/W	R/W	

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>31–16</td><td>—</td><td>Reserved (When writing to these bits, write 0. When read, 0 is returned.)</td></tr>
<tr><td>15,14,11,
10,7,6,3,2</td><td>PUIORP3n,
PDIORP3n</td><td>These bits specify whether to connect a pull-up or pull-down resistor to the RP37–RP30 pins.
<table>
<tr><th>PUIORP3n</th><th>PDIORP3n</th><th>Connection of a Pull-Up or Pull-Down Resistor to the RP37–RP30 Pins</th></tr>
<tr><td>0</td><td>0</td><td>Do not connect a pull-up or pull-down resistor.</td></tr>
<tr><td>0</td><td>1</td><td>Connect a pull-down resistor.</td></tr>
<tr><td>1</td><td>0</td><td>Connect a pull-up resistor.</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>13,12,1,0</td><td>IOLRP3m1,
IOLRP3m0</td><td>These bits specify the driving ability of the RP34 and RP33 pins.
<table>
<tr><th>IOLRP3n1</th><th>IOLRP3n0</th><th>Driving Ability of the RP37–RP30 Pins</th></tr>
<tr><td>0</td><td>1</td><td>8 mA (recommended)</td></tr>
<tr><td>1</td><td>1</td><td>12 mA</td></tr>
<tr><td colspan="2">Other than the above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Remark: m = 4–3, n = 7–0

27.7 Operation of Port Functions

Operation of the ports differs depending on the I/O mode setting as described below.

27.7.1 Reading and Writing via I/O Ports

(1) In output mode

If a value is written to a port register (Pn, RPm, or EXTPp), the value is written to that port's output latch (Pn, RPm, or EXTPp). The value of the output latch is output from the pin.
The value written to the output latch is held until another value is written.
The value of the output latch (Pn, RPm, or EXTPp) can be read by reading the port register (Pn, RPm, or EXTPp).
To directly read the pin level, read a port pin input register (PINn, RPINm, or EXTPINp).

Remark: n = 0–7, m = 0–3, p = 0–1

(2) In input mode

If a value is written to a port register (Pn, RPm, or EXTPp), the value is written to that port's output latch (Pn, RPm, or EXTPp).
However, the pin state does not change because the output buffer is off.
The value written to the output latch is held until another value is written.
To read the input level, read a port pin input register (PINn, RPINm, or EXTPINp).

Remark: n = 0–7, m = 0–3, p = 0–1

27.7.2 Multiplexed Function Pin Output State in Control Mode

The port pin level can be read directly by reading port pin input register n, m, or p (PINn, RPINm, or EXTPINp), regardless of the settings of the PMCn, RPMCm, and EXTPMCp registers, PMn, RPMm, and EXTPMp registers, PFCn, RPFCm, and EXTPFCp registers, and PFCEn, RPCEm, and EXTPEp registers.

Remark: n = 0–7, m = 0–3, p = 0–1

27.8 Trigger-Synchronous Ports (RP00–RP37)

The state of the 32-bit port pins RP00–RP37 can be updated in synchronization with an interrupt from an internal peripheral module.

Use the RPTRGMD register to set trigger-synchronous port control mode in 1-bit units. To select the target trigger, use the RPTFR0–RPTFR3 registers.

For details, refer to Section 28.13.2 “Trigger Synchronous Port Source Registers (RP0TFR–RP3TFR)”.

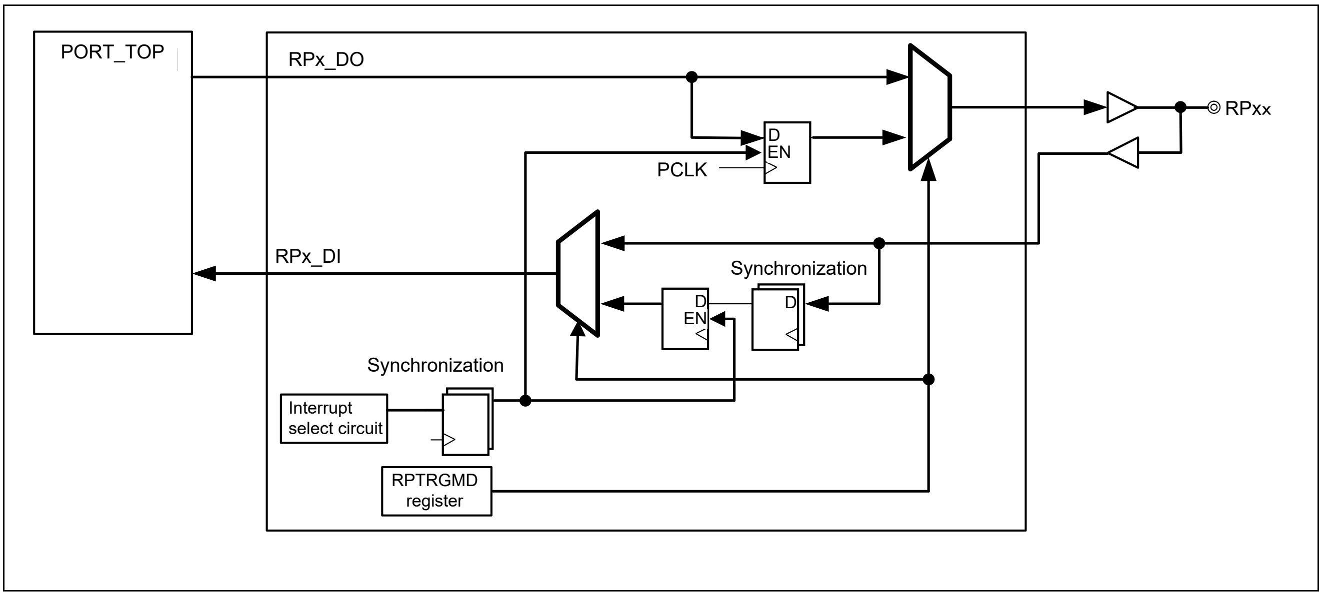


Figure 27.20 Configuration of Trigger-Synchronous Ports

28. System Registers (APB Peripheral Registers Area)

This section describes system registers (APB peripheral registers area) provided in R-IN32M4-CL3. These registers can be accessed from the external MCU interface. When the external MCU interface bus is used with a width of 16 bits, some registers are inaccessible. For details, refer to Section 28.1 “List of Registers”.

Caution: The addresses of registers given below are relative to the base addresses.
In access to the registers via the external MCU interface, the base address is D_0000H. In access by the internal CPU or DMA controller, the base address is 4001_0000H.
- In access by the CPU or DMA controller
BASE = 4001_0000H
- In access via the external MCU interface
BASE = D_0000H

28.1 List of Registers

(1/2)

Register Name	Symbol	Address	Protection Target*1	Valid Size		External MCU Operation
				16	32	
Operating Mode Monitor Register	MDMNT	BASE + 0000H	—	○	○	○
IDCODE Register	IDCODE	BASE + 0004H	—	×	○	○
Version Register	RINVER	BASE + 0008H	—	○	○	○
Watchdog Timer Input Clock Select Register	WDTCLKCFG	BASE + 0180H	○	×	○	×
CPURESET Register	CPURESET	BASE + 0210H	○	○	○	○
System Protect Command Register	SYSPCMD	BASE + 0300H	—	○	○	○
HW-RTOS Reset Register	RTOS_SOFTRST	BASE + 0400H	○	×	○	○
Timer Input Function Select Register	SELCNT	BASE + 0500H	○	×	○	×
Timer Input Function Select Register (TAUD)	SELCNTD	BASE + 0504H	○	×	○	×
Timer Trigger Source Register 0	TMTFR0	BASE + 0530H	○	×	○	×
Timer Trigger Source Register 1	TMTFR1	BASE + 0534H	○	×	○	×
Timer Trigger Source Register 2	TMTFR2	BASE + 0538H	○	×	○	×
Timer Trigger Source Register 3	TMTFR3	BASE + 053CH	○	×	○	×
Timer Trigger Source Register (TAUD) 0	TMDTFR0	BASE + 0D00H	○	×	○	×
Timer Trigger Source Register (TAUD) 1	TMDTFR1	BASE + 0D04H	○	×	○	×
Timer Trigger Source Register (TAUD) 2	TMDTFR2	BASE + 0D08H	○	×	○	×
Timer Trigger Source Register (TAUD) 3	TMDTFR3	BASE + 0D0CH	○	×	○	×

Note 1. ○: These registers are writable only when a protection release sequence is performed using Section 28.7 “System Protect Command Register (SYSPCMD)”.
—: This register is out of the scope of the write-protection.

(2/2)

Register Name	Symbol	Address	Protection Target*1	Valid Size 16	Valid Size 32	External MCU Operation
Timer Trigger Source Register (TAUD) 4	TMDTFR4	BASE + 0D10H	○	×	○	×
Timer Trigger Source Register (TAUD) 5	TMDTFR5	BASE + 0D14H	○	×	○	×
Timer Trigger Source Register (TAUD) 6	TMDTFR6	BASE + 0D18H	○	×	○	×
Timer Trigger Source Register (TAUD) 7	TMDTFR7	BASE + 0D1CH	○	×	○	×
Noise Filter Configuration Register 0	NFC0	BASE + 0700H	○	×	○	×
Noise Filter Configuration Register 1	NFC1	BASE + 0704H	○	×	○	×
Noise Filter Configuration Register 2	NFC2	BASE + 0708H	○	×	○	×
Noise Filter Configuration Register 3	NFC3	BASE + 070CH	○	×	○	×
Noise Filter Configuration Register 4	NFC4	BASE + 1250H	○	×	○	×
Interrupt Mode Register 0	INTM0	BASE + 0710H	○	×	○	×
Interrupt Mode Register 1	INTM1	BASE + 0714H	○	×	○	×
Interrupt Mode Register 2	INTM2	BASE + 0718H	○	×	○	×
Trigger Synchronous Port Control Mode Register	RPTRGMD	BASE + 0A00H	—	×	○	×
Trigger Synchronous Port Source Register 0	RP0TFR	BASE + 0A30H	○	×	○	×
Trigger Synchronous Port Source Register 1	RP1TFR	BASE + 0A34H	○	×	○	×
Trigger Synchronous Port Source Register 2	RP2TFR	BASE + 0A38H	○	×	○	×
Trigger Synchronous Port Source Register 3	RP3TFR	BASE + 0A3CH	○	×	○	×
Scratch Register 0	SCRATCH0	BASE + 0900H	—	○	○	○
Scratch Register 1	SCRATCH1	BASE + 0904H	—	○	○	○
Scratch Register 2	SCRATCH2	BASE + 0908H	—	○	○	○
Scratch Register 3	SCRATCH3	BASE + 090CH	—	○	○	○
Scratch Register 4	SCRATCH4	BASE + 0910H	—	○	○	○
Scratch Register 5	SCRATCH5	BASE + 0914H	—	○	○	○
Scratch Register 6	SCRATCH6	BASE + 0918H	—	○	○	○
Scratch Register 7	SCRATCH7	BASE + 091CH	—	○	○	○
Scratch Register 8	SCRATCH8	BASE + 0920H	—	○	○	○
Scratch Register 9	SCRATCH9	BASE + 0924H	—	○	○	○
Scratch Register A	SCRATCHA	BASE + 0928H	—	○	○	○
Scratch Register B	SCRATCHB	BASE + 092CH	—	○	○	○
Scratch Register C	SCRATCHC	BASE + 0930H	—	○	○	○
PHYLINK_ENABLE Register	PHYLINK_EN	BASE + 093CH	—	○	○	○
WDT Input Filter Select Register	WDTISEL	BASE + 1230H	—	○	○	○
Timer Interface Select Register	TMISEL	BASE + 1240H	—	○	○	×
INTPZ/timer Interrupt Select Register	INTSEL	BASE + 1244H	—	○	○	×

Note 1. **○: These registers are writable only when a protection release sequence is performed using Section 28.7 "System Protect Command Register (SYSPCMD)".**
—: This register is out of the scope of the write-protection.

28.2 Operating Mode Monitor Register (MDMNT)

The operating mode monitor register can monitor setting levels from external pins.

- Access This register can be read in 32- or 16-bit units.

Address: BASE + 0000H

Initial value: Pin setting

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDMNT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CLK2MSEL	PKGSEL	0	0	0/1	MEMCSEL	0	BOOT1	BOOT0	OSCTH	JTAGSEL	HWRZSEL	HIFSYNC	ADMUXMODE	MEMIFSEL	BUS32EN
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	R	0	0	0/1	R	R	R	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Description	
31–16	—	Reserved (When read, 0 is returned.)	
15	CLK2MSEL	This bit is used to read the level of the CLK2MSEL pin. 0: Select clock input from the CCI_CLK2_097M pin. 1: Select the PLL divided clock (2 MHz).	
14	PKGSEL	This bit is used to read the package type. 0: 23 mm Square Package 1: 17 mm Square Package	
13, 12	—	Reserved (When read, 0 is returned.)	
11	—	23 mm Square Package	Reserved (When read, 0 is returned.)
		17 mm Square Package	Reserved (When read, 1 is returned.)
9	—	Reserved (When read, 0 is returned.)	
10, 8–0	MEMCSEL, BOOT1, BOOT0, OSCTH, JTAGSEL, HWRZSEL, HIFSYNC, ADMUXMODE, MEMIFSEL, BUS32EN	These bits are used to read operating mode setting pin levels.	

28.3 IDCODE Register (IDCODE)

This register is used to identify an R-IN32M4-CL3. Reading this register makes it possible to read RIN2 in the ASCII code.

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

IDCODE — Address: BASE + 0004H, Initial value: 5249 4E32H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDCODE	IDCODE31	IDCODE30	IDCODE29	IDCODE28	IDCODE27	IDCODE26	IDCODE25	IDCODE24	IDCODE23	IDCODE22	IDCODE21	IDCODE20	IDCODE19	IDCODE18	IDCODE17	IDCODE16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDCODE	IDCODE15	IDCODE14	IDCODE13	IDCODE12	IDCODE11	IDCODE10	IDCODE9	IDCODE8	IDCODE7	IDCODE6	IDCODE5	IDCODE4	IDCODE3	IDCODE2	IDCODE1	IDCODE0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Description
31–0	IDCODE31–0	These bits are used to identify an R-IN32M4-CL3. Reading these bits allows reading RIN2 in the ASCII code.

Note: When this register is accessed by the external MCU, access can be in 16-bit units.

28.4 Version Register (RINVER)

This register is used to identify the version number of R-IN32M4-CL3.

- Access This register can be read in 32- or 16-bit units.

RINVER — Address: BASE + 0008H, Initial value: 0000 0021H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RINVER	RINVER31	RINVER30	RINVER29	RINVER28	RINVER27	RINVER26	RINVER25	RINVER24	RINVER23	RINVER22	RINVER21	RINVER20	RINVER19	RINVER18	RINVER17	RINVER16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RINVER	RINVER15	RINVER14	RINVER13	RINVER12	RINVER11	RINVER10	RINVER9	RINVER8	RINVER7	RINVER6	RINVER5	RINVER4	RINVER3	RINVER2	RINVER1	RINVER0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

28.5 Watchdog Timer Input Clock Select Register (WDTCLKCFG)

This register selects the frequency division ratio of the watchdog timer count clock.

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

Caution: This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 “System Protect Command Register (SYSPCMD)”.
No special sequence is required for reading the register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
WDTCLKCFG	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	WDTCNF3	WDTCNF2	WDTCNF1	WDTCNF0	BASE + 0180H Initial value 0000 0000H
R/W	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	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31–4	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
3–0	WDTCNF3–WDTCNF0	These bits select the clock to be supplied to the watchdog timer. 0000: HCLK/32 0001: HCLK/64 0010: HCLK/128 0011: HCLK/256 0100: HCLK/512 0101: HCLK/1024 0110: HCLK/2048 0111: HCLK/4096 1000: HCLK/8192 Other than above: Setting prohibited

28.6 CPURESET Register (CPURESET)

This register is used to release the CPU of an R-IN32M4-CL3 (Cortex-M4) from the reset state by the host CPU when booting the host.
The initial value after release from the reset state differs with the setting of the BOOT1 and BOOT0 pins. This register can be used to release the CPU from the reset state, but cannot reset the CPU again.

- Access This register can be read or written in 32- or 16-bit units.

Caution: This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)".
No special sequence is required for reading the register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
CPURESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RESET	BASE + 0210H
																																	Initial value
																																	Note
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

Bit Position	Bit Name	Description
31–1	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
0	RESET	This bit releases the CPU of an R-IN32M4-CL3 (Cortex-M4) from the reset state. 0: CPU is being reset. 1: CPU is released from the reset state.

Note: The initial value changes with the state of the BOOT1 and BOOT0 pins.

BOOT1	BOOT0	Boot Mode	Initial value
0	0	External memory boot	1
0	1	External serial flash ROM boot	1
1	0	External MCU boot	0
1	1	Instruction RAM boot (for debugging only)	1

28.7 System Protect Command Register (SYSPCMD)

SYSPCMD is a 32-bit register for use in protecting against inadvertent access to write-protected registers, i.e. registers to which writing raises the possibility of serious effects on application systems, such as programs crashing and the like.
This register can be read or written in 32- or 16-bit units. Unless the PROT bit is set to 1 for the write-protected registers, writing to these register is not possible.
When the SYSPCMD register is set to 1, writing to the write-protected registers is only possible with the sequence described below. No special sequence is required to clear this register to 0 or read it.

- Access This register can be read or written in 32- or 16-bit units.

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

Address: BASE + 0300H
Initial value: 0000 0000H

Bit Position	Bit Name	Description
31–1	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
0	PROT	This bit enables write access to the write-protected registers. 1: Enables write access. 0: Disables write access (write-protected).

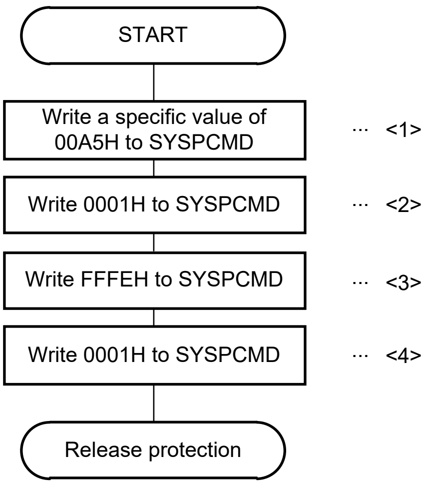


Figure 28.1 Protection Release Sequence

Cautions 1. The registers cannot be written in steps <1>, <2>, and <3>.
2. Be sure to clear this bit to 0 (setting for protection) after the completion of writing to an applicable register.

28.8 HW-RTOS Reset Register (RTOS_SOFTRST)

This register is used to reset the HW-RTOS and Gigabit Ethernet MAC areas by software.

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

Caution: This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)".
No special sequence is required for reading the register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address: BASE + 0400H
RTOS_SOFTRST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OSRST	Initial value 0000 0001H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	

Bit Position	Bit Name	Description
31–1	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
0	OSRST	Writing 0 to this bit resets the HW-RTOS and Gigabit Ethernet MAC areas. The following three memory map areas are to be reset. 4008 0000H to 4008 FFFFH: HW-RTOS (64 Kbytes) 4009 0000H to 4009 0FFFH: Gigabit Ethernet (4 Kbytes) 4009 1000H to 4009 1FFFH: QINT BUFID (4 Kbytes) To release the chip from the reset state, write 1 to this bit. 0: Reset state 1: Release from the reset state

28.9 Timer Input Function Select Register (SELCNT, SELCNTD)

The SELCNT and SELCNTD registers are used to select capture trigger input mode of TAUJ2 (4ch) and TAUD (16ch) signals. Timestamping is supported by the input of the trigger output signal (TSOUT) of the CAN macro to TAUJ2. Furthermore, the internal peripheral interrupt can be selected as a capture trigger. To select the internal peripheral interrupt, use the timer source select register.
The TIND15 to TIND8 pins of TAUD are not applicable.

- Access These registers can be read or written in 32-bit units.

Caution: This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)".
No special sequence is required for reading the register.

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
SELCNT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ISEL31	ISEL30	ISEL21	ISEL20	ISEL11	ISEL10	ISEL01	ISEL00	BASE + 0500H Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31–8	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
7, 6	ISEL31, ISEL30	These bits specify selection of the TIN3 (TAUJ2 ch3) input signal.
5, 4	ISEL21, ISEL20	These bits specify selection of the TIN2 (TAUJ2 ch2) input signal.

ISEL31	ISEL30	TIN3 Input Signal Selection
0	0	TIN3 (P52 pin)
0	1	Interrupt signal selected by the TMTFR3 register
1	0	TSOUT signal of CAN1
1	1	Setting prohibited (same as 00)

ISEL21	ISEL20	TIN2 Input Signal Selection
0	0	TIN2 (P57 pin)
0	1	Interrupt signal selected by the TMTFR2 register
1	0	Setting prohibited (same as 00)
1	1	Setting prohibited (same as 00)

(2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>3, 2</td><td>ISEL11,
ISEL10</td><td>These bits specify selection of the TIN1 (TAUJ2 ch1) input signal.
<table><tr><td>ISEL11</td><td>ISEL10</td><td>TIN1 Input Signal Selection</td></tr><tr><td>0</td><td>0</td><td>TIN1 (P26 pin)</td></tr><tr><td>0</td><td>1</td><td>Interrupt signal selected by the TMTFR1 register</td></tr><tr><td>1</td><td>0</td><td>TSOUT signal of CAN0</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited (same as 00)</td></tr></table></td></tr>
<tr><td>1, 0</td><td>ISEL01,
ISEL00</td><td>These bits specify selection of the TIN0 (TAUJ2 ch0) input signal.
<table><tr><td>ISEL01</td><td>ISEL00</td><td>TIN0 Input Signal Selection</td></tr><tr><td>0</td><td>0</td><td>TIN0 (P27 pin)</td></tr><tr><td>0</td><td>1</td><td>Interrupt signal selected by the TMTFR0 register</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited (same as 00)</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited (same as 00)</td></tr></table></td></tr>
</table>

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
SELCNTD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ISEL71	ISEL70	ISEL61	ISEL60	ISEL51	ISEL50	ISEL41	ISEL40	ISEL31	ISEL30	ISEL21	ISEL20	ISEL11	ISEL10	ISEL01	ISEL00	BASE + 0504H Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–16	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
15, 14	ISEL71, ISEL70	These bits specify selection of the TIND7 (TAUD ch7) input signal.
13, 12	ISEL61, ISEL60	These bits specify selection of the TIND6 (TAUD ch6) input signal.
11, 10	ISEL51, ISEL50	These bits specify selection of the TIND5 (TAUD ch5) input signal.
9, 8	ISEL41, ISEL40	These bits specify selection of the TIND4 (TAUD ch4) input signal.

ISEL71	ISEL70	TIND7 Input Signal Selection
0	0	TIND7 (P52)
0	1	Interrupt signal selected by the TMDTFR7 register
1	0	Setting prohibited (same as 00)
1	1	Setting prohibited (same as 00)

ISEL61	ISEL60	TIND6 Input Signal Selection
0	0	TIND6 (P57)
0	1	Interrupt signal selected by the TMDTFR6 register
1	0	Setting prohibited (same as 00)
1	1	Setting prohibited (same as 00)

ISEL51	ISEL50	TIND5 Input Signal Selection
0	0	TIND5 (P26)
0	1	Interrupt signal selected by the TMDTFR5 register
1	0	Setting prohibited (same as 00)
1	1	Setting prohibited (same as 00)

ISEL41	ISEL40	TIND4 Input Signal Selection
0	0	TIND4 (P27)
0	1	Interrupt signal selected by the TMDTFR4 register
1	0	Setting prohibited (same as 00)
1	1	Setting prohibited (same as 00)

(2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>7, 6</td><td>ISEL31,
ISEL30</td><td>These bits specify selection of the TIND3 (TAUD ch3) input signal.
<table><tr><td>ISEL31</td><td>ISEL30</td><td>TIND3 Input Signal Selection</td></tr><tr><td>0</td><td>0</td><td>TIND3 (EXTP3)</td></tr><tr><td>0</td><td>1</td><td>Interrupt signal selected by the TMDTFR3 register</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited (same as 00)</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited (same as 00)</td></tr></table></td></tr>
<tr><td>5, 4</td><td>ISEL21,
ISEL20</td><td>These bits specify selection of the TIND2 (TAUD ch2) input signal.
<table><tr><td>ISEL21</td><td>ISEL20</td><td>TIND2 Input Signal Selection</td></tr><tr><td>0</td><td>0</td><td>TIND2 (EXTP2)</td></tr><tr><td>0</td><td>1</td><td>Interrupt signal selected by the TMDTFR2 register</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited (same as 00)</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited (same as 00)</td></tr></table></td></tr>
<tr><td>3, 2</td><td>ISEL11,
ISEL10</td><td>These bits specify selection of the TIND1 (TAUD ch1) input signal.
<table><tr><td>ISEL11</td><td>ISEL10</td><td>TIND1 Input Signal Selection</td></tr><tr><td>0</td><td>0</td><td>TIND1 (EXTP1)</td></tr><tr><td>0</td><td>1</td><td>Interrupt signal selected by the TMDTFR1 register</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited (same as 00)</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited (same as 00)</td></tr></table></td></tr>
<tr><td>1, 0</td><td>ISEL01,
ISEL00</td><td>These bits specify selection of the TIND0 (TAUD ch0) input signal.
<table><tr><td>ISEL01</td><td>ISEL00</td><td>TIND0 Input Signal Selection</td></tr><tr><td>0</td><td>0</td><td>TIND0 (EXTP0)</td></tr><tr><td>0</td><td>1</td><td>Interrupt signal selected by the TMDTFR0 register</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited (same as 00)</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited (same as 00)</td></tr></table></td></tr>
</table>

28.10 Timer Trigger Source Register (TMTFR0–3, TMDTFR0–7)

A desired interrupt signal can be selected from interrupt request signals provided for TMTFRn and TMDTFRm registers, and the selected interrupt signal can be assigned to the TIN input of the timer. These registers are used to select the interrupt request signal to be assigned to the TIN input of the timer.

- TMTFR0–3 TINJ0–3 (TAUJ2) are assigned.
- TMDTFR0–7 TIND0–7 (TAUD) are assigned. (TIND8–15 are out of scope.)

- Access These registers can be read or written in 32-bit units.

Cautions 1. This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

2. To use these registers, set the prescaler to the maximum value 0000B (PCLK/2^0) in the TAUJ2TPS register of TAUJ2.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMTFRn TMDTFRm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IFC7	IFC6	IFC5	IFC4	IFC3	IFC2	IFC1	IFC0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address
BASE
+ 0530H + 4n
+ 0D00H + 4m

Initial value
0000 0000H

Bit Position	Bit Name	Description
31–8	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
7–0	IFC7–IFC0	These bits select trigger sources of timer channels n and m. (TAUJ2: n, TAUD: m)

IFC7–IFC0	Timer Count Trigger Source Selection
00H	Masks timer trigger sources.
01H–03H	Reserved (setting prohibited)
04H	TAUJ2 channel 0 interrupt
05H	TAUJ2 channel 1 interrupt
06H	TAUJ2 channel 2 interrupt
07H	TAUJ2 channel 3 interrupt
08H	UARTJ0 transmission interrupt
09H	UARTJ0 reception interrupt
0AH	UARTJ1 transmission interrupt
0BH	UARTJ1 reception interrupt
0CH	CSIH0 communication status interrupt
0DH	CSIH0 reception status interrupt
0EH	CSIH0 job end interrupt

Remark: n = 0–3; m = 0–7

Bit Position	Bit Name	Description
7–0	IFC7–IFC0	These bits select trigger sources of timer channels n and m. (TAUJ2: n, TAUD: m)

IFC7–IFC0	Timer Count Trigger Source Selection
0FH	CSIH1 communication status interrupt
10H	CSIH1 reception status interrupt
11H	CSIH1 job end interrupt
12H	IICB0 data transmission/reception interrupt
13H	IICB1 data transmission/reception interrupt
14H	FCN0 reception end interrupt
15H	FCN0 transmission end interrupt
16H	FCN0 sleep wakeup/transmission suspension interrupt
17H	FCN1 reception end interrupt
18H	FCN1 transmission end interrupt
19H	FCN1 sleep wakeup/transmission suspension interrupt
1AH	General DMAC channel 0 transfer end interrupt
1BH	General DMAC channel 1 transfer end interrupt
1CH	General DMAC channel 2 transfer end interrupt
1DH	General DMAC channel 3 transfer end interrupt
1EH	Real-time port DMAC transfer end interrupt
1FH	TAUD channel 0 interrupt
20H	TAUD channel 1 interrupt
21H	TAUD channel 2 interrupt
22H	TAUD channel 3 interrupt
23H	TAUD channel 4 interrupt
24H	Inter-buffer DMA transfer end interrupt
25H	Gigabit Ethernet PHY port 0 interrupt
26H	Gigabit Ethernet PHY port 1 interrupt
27H	Ethernet MII management access end interrupt
28H	Ethernet pause packet transmission end interrupt
29H	Ethernet transmission end interrupt
2AH	Ethernet SWITCH interrupt
2BH	Ethernet SWITCH DLR interrupt
2CH	Ethernet SWITCH SYNC interrupt
2DH, 2EH	Reserved (setting prohibited)
2FH	Ethernet MACDMA reception end interrupt
30H	Ethernet MACDMA transmission end interrupt

Remark: n = 0–3; m = 0–7

Bit Position	Bit Name	Description
7–0	IFC7–IFC0	These bits select trigger sources of timer channels n and m. (TAUJ2: n, TAUD: m)

IFC7–IFC0	Timer Count Trigger Source Selection
31H	reception frame normal interrupt
32H	Reserved (setting prohibited)
33H	INTPZ0 input*1
34H	INTPZ1 input*1
35H	INTPZ2 input*1
36H	INTPZ3 input*1
37H	INTPZ4 input*1
38H	INTPZ5 input*1
39H	INTPZ6 input*1
3AH	INTPZ7 input*1
3BH	INTPZ8 input*1
3CH	INTPZ9 input*1
3DH	INTPZ10 input*1
3EH	INTPZ11 input*1/TAUD channel 5 interrupt*2
3FH	INTPZ12 input*1/TAUD channel 6 interrupt*2
40H	INTPZ13 input*1/TAUD channel 7 interrupt*2
41H	INTPZ14 input*1/TAUD channel 8 interrupt*2
42H	INTPZ15 input*1/TAUD channel 9 interrupt*2
43H	INTPZ16 input*1/TAUD channel 10 interrupt*2
44H	INTPZ17 input*1/TAUD channel 11 interrupt*2
45H	INTPZ18 input*1/TAUD channel 12 interrupt*2
46H	INTPZ19 input*1/TAUD channel 13 interrupt*2
47H	INTPZ20 input*1/TAUD channel 14 interrupt*2
48H	INTPZ21 input*1/TAUD channel 15 interrupt*2
49H	INTPZ22 input*1/peak interrupt (TAPA)*2
4AH	INTPZ23 input*1/trough interrupt (TAPA)*2
4BH	INTPZ24 input*1
4CH	INTPZ25 input*1
4DH	INTPZ26 input*1
4EH	INTPZ27 input*1
4FH	INTPZ28 input*1

Notes 1. When using an external interrupt as a timer trigger source, be sure to specify edge detection. (Do not specify level detection.)

2. These interrupts are selected by the INTSEL register.
For details, refer to Section 28.18 “INTPZ/timer Interrupt Select Register (INTSEL)”.

Remark: n = 0–3; m = 0–7

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>7–0</td><td>IFC7–IFC0</td><td>These bits select trigger sources of timer channels n and m. (TAUJ2: n, TAUD: m)
<table>
<tr><th>IFC7–IFC0</th><th>Timer Count Trigger Source Selection</th></tr>
<tr><td>50H–62H</td><td>Reserved (setting prohibited)</td></tr>
<tr><td>63H</td><td>Gigabit Ethernet PHY LED0_PHY0 input interrupt</td></tr>
<tr><td>64H</td><td>Gigabit Ethernet PHY LED0_PHY1 input interrupt</td></tr>
<tr><td>65H–6EH</td><td>Reserved (setting prohibited)</td></tr>
<tr><td>6FH</td><td>CC-Link IE Field Network NMIZ interrupt</td></tr>
<tr><td>70H</td><td>CC-Link IE Field Network WDTZ interrupt</td></tr>
<tr><td>71H</td><td>CC-Link IE Field Network INTZ interrupt</td></tr>
<tr><td>72H</td><td>CC-Link IE Field Network CLKLOSSZ interrupt</td></tr>
<tr><td>73H–7CH</td><td>Reserved (setting prohibited)</td></tr>
<tr><td>7DH</td><td>Gigabit Ethernet PHY LED1_PHY0 input interrupt</td></tr>
<tr><td>7EH</td><td>Gigabit Ethernet PHY LED1_PHY1 input interrupt</td></tr>
<tr><td>7FH</td><td>INTPZ29 input*</td></tr>
</table>
</td></tr>
</table>

Note: When using an external interrupt as a timer trigger source, be sure to specify edge detection. (Do not specify level detection.)

Remark: n = 0–3; m = 0–7

Bit Position	Bit Name	Description
7–0	IFC7–IFC0	These bits select trigger sources of timer channels n and m. (TAUJ2: n, TAUD: m)

IFC7–IFC0	Timer Count Trigger Source Selection
80H–83H	Reserved (setting prohibited)
84H	Bus Error Response Interrupt (HSNGNREG)
85H	Bus Error Response Interrupt (HSNGN)
86H	Bus Error Response Interrupt (XSNGN)
87H	Reserved (setting prohibited)
88H	WDT Error Interrupt for External MCU
89H–97H	Reserved (setting prohibited)
98H	Built-in Regulator Interrupt
99H–FFH	Reserved (setting prohibited)

Remark: n = 0–3; m = 0–7

28.11 Noise Elimination Circuit

Digital noise filtering is used to eliminate noise from the external interrupt input signals, timer array input signals, UART serial data input signals, CC-Link IE Field and CC-Link WDT input signals, and the external AD trigger input signal.
Use the noise filter configuration registers 0–4 (NFC0–NFC4)to make settings for noise elimination.

Caution: This function can be changed only with the built-in CPU of R-IN32M4-CL3.

Table 28.1 Noise Elimination Target Signals

Target Signal	Internally Connected Unit	Signal Function
NMIZ	Interrupt controller	Non-maskable external interrupt input
INTPZ0–INTPZ29	Interrupt controller	Maskable external interrupt input
TIN0–TIN3	32-bit timer array unit (TAUJ2)	Timer input
TIND0–TIND15	16-bit timer array unit (TAUD)	Timer input
RXD0, RXD1	Asynchronous serial interface (UARTJ)	UART serial data input
CCI_WDTIZ	Interrupt controller, CC-Link IE	CC-Link IE input
LED2_PHY1–LED0_PHY1	Interrupt controller	Gigabit Ethernet PHY LED interrupt (PHY1)
LED2_PHY0–LED0_PHY0	Interrupt controller	Gigabit Ethernet PHY LED interrupt (PHY0)

28.11.1 Noise Filter Configuration Register (NFC0–4)

These registers are used to set a noise elimination level of the input signals shown in Table 28.1.

- Access These registers can be read or written in 32-bit units.

Cautions 1. When the input pulse width = NFC0-NFC4 setting value to (NFC0-NFC4 setting value – 1), whether to detect the signal as a valid signal or eliminate it as noise is undefined.

2. Interrupt input signals (INTPZ0 to INTPZ28 and NMIZ) and LED output signals of the Gigabit Ethernet PHY are transferred through the edge specification circuit, but alternative functions other than interrupts are not transferred through the edge specification circuit. Effective edges of timer array unit input pins are specified by the timer array unit edge specification register. No edge specification function is provided for the RXD0 and RXD1 input signals.

3. When NFC0 to NFC4 registers are modified, an unintended interrupt may be generated in each register. Modify these registers in the Disable IRQ state, and then clear the corresponding interrupt pending bit.

4. The noise elimination circuit performs signal synchronization processing. Therefore, these signals are delayed even if the filter stage is set to 0.

5. This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 “System Protect Command Register (SYSPCMD)”.
No special sequence is required for reading the register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
NFC0	0	0	0	0	0	0	0	0	0	0	0	0	NFP1L21	NFP1L20	NFP1L11	NFP1L10	NFP1L01	NFP1L00	0	0	NFP0L21	NFP0L20	NFP0L11	NFP0L10	NFP0L01	NFP0L00	0	0	0	0	NFPNM1	NFPNM0	BASE + 0700H Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	0	0	R/W	R/W	R/W	R/W	R/W	R/W	0	0	0	0	R/W	R/W	

Bit Position	Bit Name	Description
31–20, 13–12, 5–2	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
19–14	NFP1Ln1, NFP1Ln0	These bits set the noise filter stage of the LED2_PHY1–LED0_PHY1 inputs with the internal system clock as a reference. (n = 0–2) NFP1Ln1 / NFP1Ln0 / Noise Filter Stage: 0 / 0 / 0 × HCLK (Initial value) 0 / 1 / 4 × HCLK 1 / 0 / 8 × HCLK 1 / 1 / 16 × HCLK
11–6	NFP0Ln1, NFP0Ln0	These bits set the noise filter stage of the LED2_PHY0–LED0_PHY0 inputs with the internal system clock as a reference. (n = 0–2) NFP0Ln1 / NFP0Ln0 / Noise Filter Stage: 0 / 0 / 0 × HCLK (Initial value) 0 / 1 / 4 × HCLK 1 / 0 / 8 × HCLK 1 / 1 / 16 × HCLK
1, 0	NFPNM1, NFPNM0	These bits set the noise filter stage of the NMI input with the internal system clock as a reference. NFPNM1 / NFPNM0 / Noise Filter Stage: 0 / 0 / 0 × HCLK (Initial value) 0 / 1 / 4 × HCLK 1 / 0 / 8 × HCLK 1 / 1 / 16 × HCLK

NFC1 — Address: BASE + 0704H, Initial value: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC1	NFP151	NFP150	NFP141	NFP140	NFP131	NFP130	NFP121	NFP120	NFP111	NFP110	NFP101	NFP100	NFP91	NFP90	NFP81	NFP80
R/W	R/W	R/W	R/W	R/W	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
NFC1	NFP71	NFP70	NFP61	NFP60	NFP51	NFP50	NFP41	NFP40	NFP31	NFP30	NFP21	NFP20	NFP11	NFP10	NFP01	NFP00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NFC2 — Address: BASE + 0708H, Initial value: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC2	0	0	0	0	NFP291	NFP290	NFP281	NFP280	NFP271	NFP270	NFP261	NFP260	NFP251	NFP250	NFP241	NFP240
R/W	0	0	0	0	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
NFC2	NFP231	NFP230	NFP221	NFP220	NFP211	NFP210	NFP201	NFP200	NFP191	NFP190	NFP181	NFP180	NFP171	NFP170	NFP161	NFP160
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–0	NFPn1, NFPn0	These bits set the noise filter stage of the INTPZn inputs with the internal system clock as a reference.

NFPn1	NFPn0	Noise Filter Stage
0	0	0 × HCLK (Initial value)
0	1	4 × HCLK
1	0	8 × HCLK
1	1	16 × HCLK

Remark: n = 0–29

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
NFC3	0	0	0	0	0	0	0	0	0	0	NFWDT1	NFWDT0	NFRX11	NFRX10	NFRX01	NFRX00	0	0	0	0	0	0	0	0	NFTIN31	NFTIN30	NFTIN21	NFTIN20	NFTIN11	NFTIN10	NFTIN01	NFTIN00	BASE + 070CH Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31–22, 15–8	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
21, 20	NFWDT1, NFWDT0	These bits set the noise filter stage of the CCI_WDTIZ input with the internal system clock as a reference.
19–16	NFRXn1, NFRXn0	These bits set the noise filter stage of the RXD1 and RXD0 inputs with the internal system clock as a reference.
7–0	NFTINm1, NFTINm0	These bits set the noise filter stage of the TIN3–TIN0 inputs with the internal system clock as a reference.

NFWDT1	NFWDT0	Noise Filter Stage
0	0	0 × HCLK (Initial value)
0	1	4 × HCLK
1	0	8 × HCLK
1	1	16 × HCLK

NFRXn1	NFRXn0	Noise Filter Stage
0	0	0 × HCLK (Initial value)
0	1	4 × HCLK
1	0	8 × HCLK
1	1	16 × HCLK

NFTINm1	NTINm0	Noise Filter Stage
0	0	0 × HCLK (Initial value)
0	1	4 × HCLK
1	0	8 × HCLK
1	1	16 × HCLK

Remark: n = 1, 0; m = 3–0

NFC4 — Address: BASE + 1250H, Initial value: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC4	NFTIND151	NFTIND150	NFTIND141	NFTIND140	NFTIND131	NFTIND130	NFTIND121	NFTIND120	NFTIND111	NFTIND110	NFTIND101	NFTIND100	NFTIND91	NFTIND90	NFTIND81	NFTIND80
R/W	R/W	R/W	R/W	R/W	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
NFC4	NFTIND71	NFTIND70	NFTIND61	NFTIND60	NFTIND51	NFTIND50	NFTIND41	NFTIND40	NFTIND31	NFTIND30	NFTIND21	NFTIND20	NFTIND11	NFTIND10	NFTIND01	NFTIND00
R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–0	NFTINDm1, NFTINDm0	These bits set the noise filter stage of the TIND15–TIND0 inputs with the internal system clock as a reference.

NFTINDm1	NFTINDm0	Noise Filter Stage
0	0	0 × HCLK (Initial value)
0	1	4 × HCLK
1	0	8 × HCLK
1	1	16 × HCLK

Remark: m = 15–0

28.11.2 Noise Filter Operation

Input signals listed in Table 28.1 are sampled with the clock of the same frequency as the internal bus clock HCLK or ACLK, and their noise is eliminated as specified by the noise filter configuration registers (NFC0–NFC4). This sampling clock does not stop in standby mode, and therefore external interrupts NMI and INTPZ0 to INTPZ29 can be de-asserted from standby mode. In addition, for external interrupts INTPZ0 to INTPZ29, rising edges, falling edges, both edges, or Low active level can be selected as valid triggers.

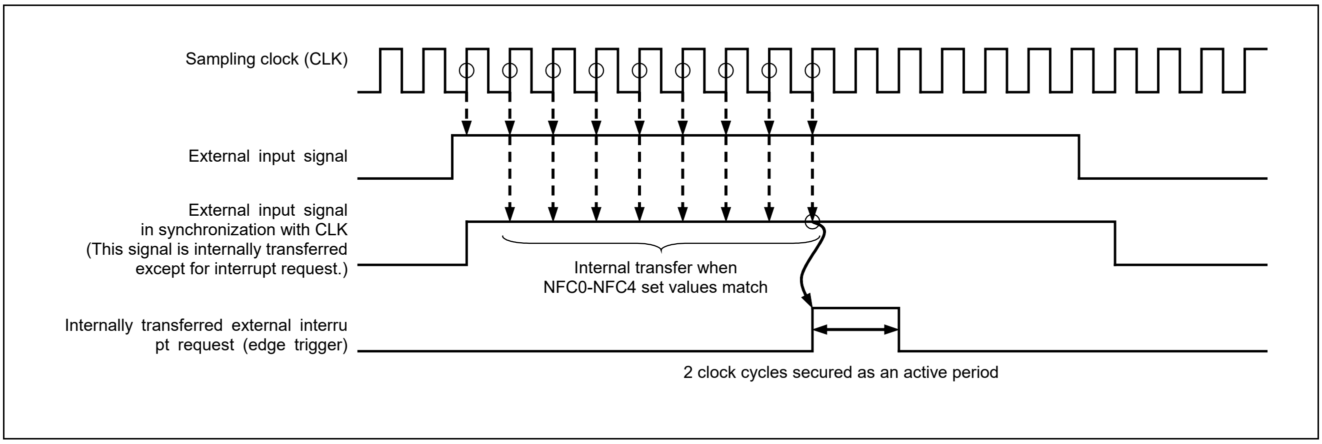


Figure 28.2 Operation of Digital Noise Filter for Interrupt Signals (Edge Trigger)

28.12 Interrupt Mode Register (INTM0–INTM2)

These registers specify the trigger mode of external interrupt requests (NMIZ, INTPZ0–INTPZ29) by external pins. The following shows the correspondence between the registers and external interrupt requests controlled by the corresponding register.

- Access These registers can be read or written in 32-bit units.

  - INTM0: NMIZ, LED2_PHY1–LED0_PHY1, LED2_PHY0–LED0_PHY0
  - INTM1: INTPZ0–INTPZ15
  - INTM2: INTPZ16–INTPZ29

Cautions 1. This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 “System Protect Command Register (SYSPCMD)”. No special sequence is required for reading the register.

2. The INTPZ0–INTPZ29 pins are also used as ports. When these pins are set as the interrupt function by the PMCm register, an unnecessary interrupt may be generated depending on the state immediately before. To prevent this problem, set the PMCm register with these interrupts masked, and then clear these interrupt request flags.

3. When the values of these registers are modified, an unnecessary interrupt may be generated depending on the state immediately before. Prior to the modification, take measures such as disabling interrupt processing of macros (CPU, DMA, timer, and others) that accept interrupt requests.

(1/2)

Address: BASE + 0710H
Initial value: 0040 0002H

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTM0	0	0	0	0	0	0	0	0	0	1	0	0	ESP1L21	ESP1L20	ESP1L11	ESP1L10
R/W	0	0	0	0	0	0	0	0	0	1	0	0	R/W	R/W	R/W	R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTM0	ESP1L01	ESP1L00	0	0	ESP0L21	ESP0L20	ESP0L11	ESP0L10	ESP0L01	ESP0L00	0	0	0	0	ESN01	ESN00
R/W	R/W	R/W	0	0	R/W	R/W	R/W	R/W	R/W	R/W	0	0	0	0	R/W	R/W

Bit Position	Bit Name	Description
31–23, 21, 20, 13, 12	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
22	—	Reserved (When writing to these bits, write 1. When read, 1 is returned.)

(2/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
INTM0	0	0	0	0	0	0	0	0	0	1	0	0	ESP1L21	ESP1L20	ESP1L11	ESP1L10	ESP1L01	ESP1L00	0	0	ESP0L21	ESP0L20	ESP0L11	ESP0L10	ESP0L01	ESP0L00	0	0	0	0	ESN01	ESN00	BASE + 0710H Initial value 0040 0002H
R/W	0	0	0	0	0	0	0	0	0	1	0	0	R/W	R/W	R/W	R/W	R/W	R/W	0	0	R/W	R/W	R/W	R/W	R/W	R/W	0	0	0	0	R/W	R/W	

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>19–14</td><td>ESP1Ln1,
ESP1Ln0</td><td>These bits specify the trigger mode of LED2_PHY1–LED0_PHY1. (n = 0–2)
<table><tr><th>ESP1Ln1</th><th>ESP1Ln0</th><th>Trigger Mode of LED2_PHY1–LED0_PHY1</th></tr><tr><td>0</td><td>0</td><td>Falling edge (Initial value)</td></tr><tr><td>0</td><td>1</td><td>Rising edge</td></tr><tr><td>1</td><td>0</td><td>Level detection (Low detection)</td></tr><tr><td>1</td><td>1</td><td>Rising and falling edges</td></tr></table></td></tr>
<tr><td>11–6</td><td>ESP0Ln1,
ESP0Ln0</td><td>These bits specify the trigger mode of LED2_PHY0–LED0_PHY0. (n = 0–2)
<table><tr><th>ESP0Ln1</th><th>ESP0Ln0</th><th>Trigger Mode of LED2_PHY0–LED0_PHY0</th></tr><tr><td>0</td><td>0</td><td>Falling edge (Initial value)</td></tr><tr><td>0</td><td>1</td><td>Rising edge</td></tr><tr><td>1</td><td>0</td><td>Level detection (Low detection)</td></tr><tr><td>1</td><td>1</td><td>Rising and falling edges</td></tr></table></td></tr>
<tr><td>5–2</td><td>—</td><td>Reserved (When writing to these bits, write 0. When read, 0 is returned.)</td></tr>
<tr><td>1, 0</td><td>ESN01,
ESN00</td><td>These bits specify the trigger mode of NMIZ.
<table><tr><th>ESN01</th><th>ESN00</th><th>Trigger Mode of NMIZ</th></tr><tr><td>0</td><td>0</td><td>Falling edge</td></tr><tr><td>0</td><td>1</td><td>Rising edge</td></tr><tr><td>1</td><td>0</td><td>Level detection (Low detection) (Initial value)*</td></tr><tr><td>1</td><td>1</td><td>Rising and falling edges</td></tr></table></td></tr>
</table>

Note: When the Low level of NMIZ is set to be detected, the signal is pulsed internally and an interrupt occurs once.

INTM1 — Address: BASE + 0714H, initial value: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTM1	ESP151	ESP150	ESP141	ESP140	ESP131	ESP130	ESP121	ESP120	ESP111	ESP110	ESP101	ESP100	ESP91	ESP90	ESP81	ESP80	ESP71	ESP70	ESP61	ESP60	ESP51	ESP50	ESP41	ESP40	ESP31	ESP30	ESP21	ESP20	ESP11	ESP10	ESP01	ESP00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INTM2 — Address: BASE + 0718H, initial value: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTM2	0	0	0	0	ESP291	ESP290	ESP281	ESP280	ESP271	ESP270	ESP261	ESP260	ESP251	ESP250	ESP241	ESP240	ESP231	ESP230	ESP221	ESP220	ESP211	ESP210	ESP201	ESP200	ESP191	ESP190	ESP181	ESP180	ESP171	ESP170	ESP161	ESP160
R/W	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 Position	Bit Name	Description
31–0	ESPn1, ESPn0	These bits specify the trigger mode of INTPZn.

ESPn1	ESPn0	Trigger Mode of INTPZ0–INTPZ29
0	0	Falling edge (initial value)
0	1	Rising edge
1	0	Level detection (Low detection)
1	1	Rising and falling edges

Remark: n = 0–29

28.13 Trigger Synchronous Port Function

The state of the 32-bit port pins RP00–RP37 can be updated in synchronization with an internal peripheral interrupt.
To set trigger synchronous port control mode, set the RPTRGMD register in 1-bit units. Use the RPTFR0 to RPTFR3 registers to select target triggers.

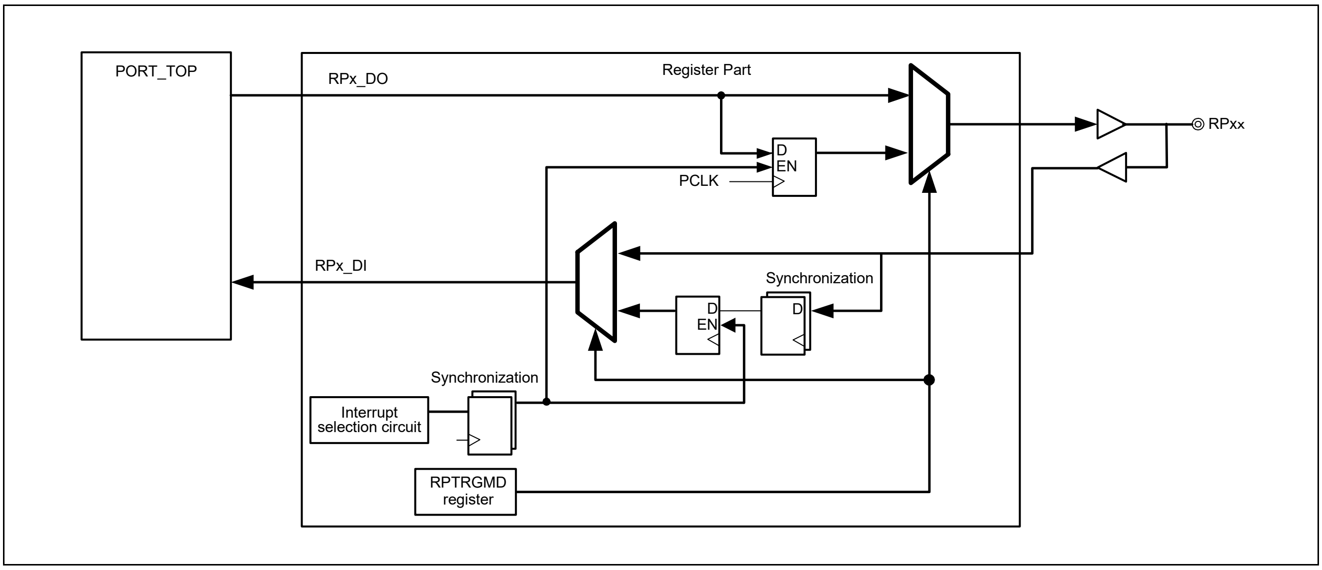


Figure 28.3 Configuration of the Trigger Synchronous Port

28.13.1 Trigger Synchronous Port Control Mode Register (RPTRGMD)

The RPTRGMD register selects whether to enable trigger synchronous port control mode for 32-bit port pins RP00–RP37 in 1-bit units.

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

	31–24	23–16	15–8	7–0	
RPTRGMD	RP37TRG–RP30TRG	RP27TRG–RP20TRG	RP17TRG–RP10TRG	RP07TRG–RP00TRG	Address BASE + 0A00H Initial value 0000 0000H
R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
7–0	RPmnTRG	These bits enable or disable trigger synchronous port control mode. 0: Used as a normal port. 1: Used as a trigger synchronous port.

Remark: m = 3–0; n = 7–0

28.13.2 Trigger Synchronous Port Source Registers (RP0TFR–RP3TFR)

A desired interrupt signal can be selected from interrupt request signals provided in RPTFR registers and the selected interrupt signal can be assigned to the trigger synchronizing signal. Interrupt signals are selectable by using the trigger synchronous port source registers (RP0TFR–RP3TFR). Trigger sources can be set in units of the 8-bit port of real-time ports (RP0x–RP3x).

- Access These registers can be read or written in 32-bit units.

Caution: This register is only writable after the protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For the protection release sequence, refer to Section 28.7 "System Protect Command Register (SYSPCMD)". No special sequence is required for reading the register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPnTFR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IFC7	IFC6	IFC5	IFC4	IFC3	IFC2	IFC1	IFC0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address BASE+0A30H +4n
Initial value 0000 0000H

Bit Position	Bit Name	Description
31–8	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
7–0	IFC7–IFC0	These bits select the trigger source of trigger synchronous port n.

IFC7–IFC0	Selection of Trigger Source of Trigger Synchronous Port n
00H	Timer output (TOUTJ0/TOUTD4) is used.
01H	Timer output (TOUTJ1/TOUTD5) is used.
02H	Timer output (TOUTJ2/TOUTD6) is used.
03H	Timer output (TOUTJ3/TOUTD7) is used.
04H	TAUJ2 channel 0 interrupt
05H	TAUJ2 channel 1 interrupt
06H	TAUJ2 channel 2 interrupt
07H	TAUJ2 channel 3 interrupt
08H	UARTJ0 transmission interrupt
09H	UARTJ0 reception interrupt
0AH	UARTJ1 transmission interrupt
0BH	UARTJ1 reception interrupt

Remark: n = 0–3

Bit Position	Bit Name	Description
7–0	IFC7–IFC0	These bits select the trigger source of trigger synchronous port n.

IFC7–IFC0	Selection of Trigger Source of Trigger Synchronous Port n
0CH	CSIH0 communication status interrupt
0DH	CSIH0 reception status interrupt
0EH	CSIH0 job end interrupt
0FH	CSIH1 communication status interrupt
10H	CSIH1 reception status interrupt
11H	CSIH1 job end interrupt
12H	IICB0 data transmission/reception interrupt
13H	IICB1 data transmission/reception interrupt
14H	FCN0 reception end interrupt
15H	FCN0 transmission end interrupt
16H	FCN0 sleep wakeup/transmission suspension interrupt
17H	FCN1 reception end interrupt
18H	FCN1 transmission end interrupt
19H	FCN1 sleep wakeup/transmission suspension interrupt
1AH	General-purpose DMAC channel 0 transfer end interrupt
1BH	General-purpose DMAC channel 1 transfer end interrupt
1CH	General-purpose DMAC channel 2 transfer end interrupt
1DH	General-purpose DMAC channel 3 transfer end interrupt
1EH	Real-time port DMAC transfer end interrupt
1FH	TAUD channel 0 interrupt
20H	TAUD channel 1 interrupt
21H	TAUD channel 2 interrupt
22H	TAUD channel 3 interrupt
23H	TAUD channel 4 interrupt
24H	Inter-Buffer DMA transfer end interrupt
25H	Gigabit Ethernet PHY Port 0 interrupt
26H	Gigabit Ethernet PHY Port 1 interrupt
27H	Ethernet MII management access end interrupt
28H	Ethernet pause packet transmission end interrupt
29H	Ethernet transmission end interrupt
2AH	Ethernet SWITCH interrupt
2BH	Ethernet SWITCH DLR interrupt
2CH	Ethernet SWITCH SYNC interrupt

Remark: n = 0–3

Bit Position	Bit Name	Description
7–0	IFC7–IFC0	These bits select the trigger source of trigger synchronous port n.

IFC7–IFC0	Selection of Trigger Source of Trigger Synchronous Port n
2DH, 2EH	Reserve (setting prohibited)
2FH	Ethernet MACDMA reception end interrupt
30H	Ethernet MACDMA transmission end interrupt
31H	Receive frame normal interrupt
32H	Reserve (setting prohibited)
33H	INTPZ0 input*1
34H	INTPZ1 input*1
35H	INTPZ2 input*1
36H	INTPZ3 input*1
37H	INTPZ4 input*1
38H	INTPZ5 input*1
39H	INTPZ6 input*1
3AH	INTPZ7 input*1
3BH	INTPZ8 input*1
3CH	INTPZ9 input*1
3DH	INTPZ10 input*1
3EH	INTPZ11 input*1/TAUD channel 5 interrupt*2
3FH	INTPZ12 input*1/TAUD channel 6 interrupt*2
40H	INTPZ13 input*1/TAUD channel 7 interrupt*2
41H	INTPZ14 input*1/TAUD channel 8 interrupt*2
42H	INTPZ15 input*1/TAUD channel 9 interrupt*2
43H	INTPZ16 input*1, 3/TAUD channel 10 interrupt*2
44H	INTPZ17 input*1, 3/TAUD channel 11 interrupt*2
45H	INTPZ18 input*1, 3/TAUD channel 12 interrupt*2
46H	INTPZ19 input*1, 3/TAUD channel 13 interrupt*2
47H	INTPZ20 input*1, 3/TAUD channel 14 interrupt*2
48H	INTPZ21 input*1, 3/TAUD channel 15 interrupt*2

Notes 1. When using an external interrupt as a source of trigger synchronous port n, be sure to specify edge detection. (Do not specify level detection.)

2. These interrupts are selected by the INTSEL register. For details, refer to Section 28.18 “ INTPZ/timer Interrupt Select Register (INTSEL)”.

3. The INTPZ16–INTPZ21 pin functions are multiplexed with the RP00–RP05 pin functions. The INTPZ25–INTPZ28 pin functions are multiplexed with RP24–RP27 pin functions.
For this reason, these pins cannot be selected as trigger sources for real-time ports to which external interrupt pins are assigned.

Remark: n = 0–3

Bit Position	Bit Name	Description
7–0	IFC7–IFC0	These bits select the trigger source of trigger synchronous port n.

IFC7–IFC0	Selection of Trigger Source of Trigger Synchronous Port n
49H	INTPZ22 input*1
4AH	INTPZ23 input*1
4BH	INTPZ24 input*1
4CH	INTPZ25 input*1, 3
4DH	INTPZ26 input*1, 3
4EH	INTPZ27 input*1, 3
4FH	INTPZ28 input*1, 3
50H–62H	Reserve (setting prohibited)
63H	Gigabit Ethernet PHY LED0_PHY0 input interrupt
64H	Gigabit Ethernet PHY LED0_PHY1 input interrupt
65H–6EH	Reserve (setting prohibited)
6FH	CC-Link IE Field NMIZ interrupt
70H	CC-Link IE Field WDTZ interrupt
71H	CC-Link IE Field INTZ interrupt
72H	CC-Link IE Field CLKLOSSZ interrupt
73H–78H	Reserve (setting prohibited)

Notes 1. When using an external interrupt as a source of trigger synchronous port n, be sure to specify edge detection. (Do not specify level detection.)

2. These interrupts are selected by the INTSEL register. For details, refer to Section 28.18 " INTPZ/timer Interrupt Select Register (INTSEL)".

3. The INTPZ16–INTPZ21 pin functions are multiplexed with the RP00–RP05 pin functions. The INTPZ25–INTPZ28 pin functions are multiplexed with RP24–RP27 pin functions.
For this reason, these pins cannot be selected as trigger sources for real-time ports to which external interrupt pins are assigned.

Remark: n = 0–3

Bit Position	Bit Name	Description
7–0	IFC7–IFC0	These bits select the trigger source of trigger synchronous port n.

IFC7–IFC0	Selection of Trigger Source of Trigger Synchronous Port n
79H–7CH	Reserve (setting prohibited)
7DH	Gigabit Ethernet PHY LED1_PHY0 input interrupt
7EH	Gigabit Ethernet PHY LED1_PHY1 input interrupt
7FH	INTPZ29 input*1
80H–83H	Reserve (setting prohibited)
84H	Bus error response interrupt (HSNGNREG)
85H	Bus error response interrupt (HSNGN)
86H	Bus error response interrupt (XSNGN)
87H	Reserve (setting prohibited)
88H	WDT error interrupt for external CPU
89H –97H	Reserve (setting prohibited)
98H	Built-in regulator interrupt
99H–FFH	Reserve (setting prohibited)

Note 1. When using an external interrupt as a source of trigger synchronous port n, be sure to specify edge detection. (Do not specify level detection.)

Remark: n = 0–3

28.14 Scratch Registers (SCRATCH0–SCRATCHC)

These registers are 16-bit general-purpose registers. These registers can also be used to transfer status information from and to the external MCU.

- Access These registers can be read or written in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
SCRATCHn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BASE + 0900H + 4nH
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
SCRATCHnF–SCRATCHn0																0000 0000H
R/W																

Remark: n = 0–C

28.15 PHYLINK_ENABLE Register (PHYLINK_EN)

This register enables or disables the LEDm-PHYn pin that the Gigabit Ethernet PHY outputs.
The LEDm-PHYn pin function is disabled by default. After the Gigabit Ethernet PHY has been initialized, control this register to enable the LEDm-PHYn pin function.

- Access This register can be read or written in 32- or 16-bit units.

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

Address BASE + 093CH
Initial value 0000 0000H

Bit Position	Bit Name	Description
31–1	—	Reserved (When writing to these bits, write 0. When read, 0 is returned.)
0	LINKEN	This bit enables or disables the LEDm-PHYn pin function that is output by Gigabit Ethernet PHY. 0: Disabled High level is always output from the LEDm-PHYn pin. 1: Enabled The LEDm-PHYn pin works as a function of Gigabit Ethernet PHY.

Caution: This register is not reset by the HOTRESETZ pin.

Remark: m = 0–3; n = 0–1

28.16 WDT Input Filter Select Register (WDTISEL)

This register selects whether to use the noise filter for the CC-Link IE Field WDT (CCI_WDTIZ) input.
When the noise filter is used, set this register to 1 and then set the noise filter stage in the noise filter configuration register (NFC3).

- Access This register can be read or written in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
																																	BASE + 1230H
WDTISEL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WDTISEL	Initial value 0000 0000H
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit Position	Bit Name	Description
0	WDTISEL	This bit selects whether to use the noise filter for the CC-Link IE Field WDT input (CCI_WDTIZ). 0: The noise filter is not used (through). 1: The noise filter is used.

28.17 Timer Interface Select Register (TMISEL)

Of the TAUJ2 and TAUD interfaces multiplexed on ports 26 (P26), 27 (P27), 52 (P52), and 57 (P57), this register selects which interface is to be used on which pins.

- Access This register can be read or written in 32- or 16-bit units.

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

Address
BASE + 1240H
Initial value
0000 0000H

Bit Position	Bit Name	Description
0	TMISEL	This bit selects the TAUJ2 or TAUD interface. 0: Enables TAUJ2. 1: Enables TAUD.

Alternative Port	TMISEL Setting	
	0H	1H
P26	TINJ1/TOUTJ1	TIND5/TOUTD5
P27	TINJ0/TOUTJ0	TIND4/TOUTD4
P52	TINJ3/TOUTJ3	TIND7/TOUTD7
P57	TINJ2/TOUTJ2	TIND6/TOUTD6

28.18 INTPZ/timer Interrupt Select Register (INTSEL)

In the use of interrupts for the CPU, RTOS, DMAC activation, timer triggers, or real-time port triggers, some INTPZ interrupts are shared with timer (TAUD) interrupts. The INTSEL register selects whether INTPZ or timer (TAUD) interrupts are to be used as the interrupt functions.

- Access This register can be read or written in 32- or 16-bit units.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSEL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	INTSEL10	INTSEL9	INTSEL8	INTSEL7	INTSEL6	INTSEL5	INTSEL4	INTSEL3	INTSEL2	INTSEL1	INTSEL0
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address
BASE + 1244H
Initial value
0000 0000H

Bit Position	Bit Name	Description
10–0	INTSEL10–0	These bits select INTPZ or TAUD interrupts. 0: Enables INTPZ. 1: Enables TAUD.

Exception No.	INTSELn	INTSELn Setting Value (n = 0–12)	
		0H	1H
74	INTSEL0	INTPZ11 input	TAUD channel 5 interrupt
75	INTSEL1	INTPZ12 input	TAUD channel 6 interrupt
76	INTSEL2	INTPZ13 input	TAUD channel 7 interrupt
77	INTSEL3	INTPZ14 input	TAUD channel 8 interrupt
78	INTSEL4	INTPZ15 input	TAUD channel 9 interrupt
79	INTSEL5	INTPZ16 input	TAUD channel 10 interrupt
80	INTSEL6	INTPZ17 input	TAUD channel 11 interrupt
81	INTSEL7	INTPZ18 input	TAUD channel 12 interrupt
82	INTSEL8	INTPZ19 input	TAUD channel 13 interrupt
83	INTSEL9	INTPZ20 input	TAUD channel 14 interrupt
84	INTSEL10	INTPZ21 input	TAUD channel 15 interrupt

29. Electrical Characteristics

29.1 Terminology

Table 29.1 Terms Used in Absolute Maximum Ratings

Parameter	Symbol	Meaning
Power supply voltage	V_{DD}	Indicates the voltage range within which damage or reduced reliability will not result when power is applied to a VDD pin.
Input voltage	V_{I}	Indicates the voltage range within which damage or reduced reliability will not result when power is applied to an input pin.
Output voltage	V_{O}	Indicates the voltage range within which damage or reduced reliability will not result when power is applied to an output pin.
Output current	I_{O}	Indicates the absolute tolerance value for DC current to prevent damage or reduced reliability when a current flows out of or into an output pin.
Operating ambient temperature	T_{A}	Indicates the ambient temperature range for normal logic operations.
Storage temperature	$T_{.Sgt.}$	Indicates the element temperature range within which damage or reduced reliability will not result while no voltage or current is being applied to the device.

Table 29.2 Terms Used in Recommended Operating Range

Parameter	Symbol	Meaning
Power supply voltage	V_{DD}	Indicates the voltage range for normal logic operations that occur when VSS = 0 V.
High-level input voltage	V_{IH}	A voltage, which is applied to the input pins of the R-IN32M4-CL3, indicating that the high level state for normal operation of the input buffer. – If a voltage that is equal to or greater than the minimum value is applied, the input voltage is guaranteed as a high level voltage.
Low-level input voltage	V_{IL}	A voltage, which is applied to the input pins of the R-IN32M4-CL3, indicating that the low level state for normal operation of the input buffer. – If a voltage that is equal to or less than the maximum value is applied, the input voltage is guaranteed as a low level voltage.
Positive trigger voltage	V_{P}	Indicates the input level at which the output level is inverted when the input to the R-IN32M4-CL3 is changed from the low-level side to the high-level side.
Negative trigger voltage	V_{N}	Indicates the input level at which the output level is inverted when the input to the R-IN32M4-CL3 is changed from the high-level side to the low-level side.
Hysteresis voltage	V_{H}	Indicates the differential between the positive trigger voltage and the negative trigger voltage.
Input rising time	t_{ried}, t_{ric}, t_{ris}	Indicates the limit value for the time period when an input voltage applied to R-IN32M4-CL3 rises from 10% to 90%. t_{ried}, t_{ric}, and t_{ris} each indicate the input rising time for the data, clock, and Schmitt buffer.
Input falling time	t_{fid}, t_{fic}, t_{fis}	Indicates the limit value for the time period when an input voltage applied to R-IN32M4-CL3 falls from 90% to 10%. t_{fid}, t_{fic}, and t_{fis} each indicate the input falling time for the data, clock, and Schmitt buffer.

Table 29.3 Terms Used for DC Characteristics

Parameter	Symbol	Meaning
Off-state output current	I_{OZ}	Indicates the current that flows via an output pin when the rated voltage is applied when a tri-state output has high impedance.
Output short circuit current	I_{OS}	Indicates the current that flows when the output pins are shorted to the ground when output is at high level.
Input leakage current	I_{LI}	Indicates the current that flows via an input pin when a voltage is applied to that pin.
Low-level output current	I_{OL}	Indicates the current that flows through the output pin when the rated low-level voltage is outputting.
High-level output current	I_{OH}	Indicates the current that flows from the output pin when the rated high-level voltage is outputting.
Low-level output voltage	V_{OL}	Indicates the output voltage at low level.
High-level output voltage	V_{OH}	Indicates the output voltage at high level.

29.2 Absolute Maximum Ratings

Table 29.4 Absolute Maximum Ratings

Item	Symbol	Conditions		Ratings	Unit
Power supply voltage	V_{DD}	1.15 V power supply		−0.3 to +1.265	V
		2.5 V power supply		−0.3 to +2.75	V
		3.3 V power supply		−0.3 to +4.20	V
I/O voltage	V_I /V_O	3.3 V buffer	V_I /V_O < V_{DD} + 0.3 V	−0.3 to +4.20	V
		Gigabit Ethernet PHY MDI (Px_DyP/Px_DyN) x = 0, 1, y = 0 to 3		−0.3 to +2.75	V
	V_I	5 V-tolerant buffer		−0.3 to +5.80	V
	V_O			−0.3 to +4.20	V
Output current (3.3 V buffer)	I_O	8mA type		16.0	mA
		10mA type		22.3	mA
		12mA type		27.6	mA
Output current (5 V-tolerant buffer)	I_O	4mA type (5 V-tolerant buffer)		10.2	mA
Operating ambient temperature	T_A	—		−40 to +85	°C
Storage temperature	$T_{.Sgt.}$	—		−55 to +125	°C
Junction temperature	T_j	—		−40 to +125	°C

Note. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark. Be sure to apply voltage to the I/O pins only after the supply voltage has been fixed.

29.3 Recommended Operating Conditions

Table 29.5 Recommended Operating Conditions

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage	V_{DD}	1.15 V power supply	1.09	1.15	1.21	V
		2.5 V power supply	2.375	2.5	2.625	V
		3.3 V power supply	3.135	3.3	3.465	V
Negative trigger voltage	V_N	3.3 V buffer	0.8	—	1.8	V
		5 V-tolerant buffer	0.8	—	1.8	V
Positive trigger voltage	V_P	3.3 V buffer	1.1	—	2.4	V
		5 V-tolerant buffer	1.1	—	2.1	V
Hysteresis voltage	V_H	3.3 V buffer	0.15	—	1.1	V
		5 V-tolerant buffer	0.15	—	1.1	V
Low-level input voltage	V_{IL}	3.3 V buffer	−0.3	—	0.8	V
		3.3 V OSC buffer	−0.3	—	0.8	V
		5 V-tolerant buffer	−0.3	—	0.8	V
High-level input voltage	V_{IH}	3.3 V buffer	2.2	—	V_{DD} + 0.3	V
		3.3 V OSC buffer	2.4	—	V_{DD} + 0.3	V
		5 V-tolerant buffer	2.2	—	5.8	V
Input rising/falling time	t_{ried}	—	0	—	200	ns
	t_{fid}	—	0	—	200	ns
Input rising/falling time (clock)	t_{ric}	—	0	—	4	ns
	t_{fic}	—	0	—	4	ns
Input rising/falling time (Schmitt input)	t_{ris}	—	0	—	1	ms
	t_{fis}	—	0	—	1	ms
Operating ambient temperature	T_A	—	−40	—	85	°C

29.4 DC Characteristics

Table 29.6 DC Characteristics (V_{DD} = 3.3 ± 0.165 V, T_A = −40 to +85°C) (1/2)

Item	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Operating current consumption	I_{DD}	$V_I = V_{DD}$ or GND	Without 2.5-V built-in Regulator	VDD11, VDD11A	—	325	515	mA
				VDD25A	—	280	320	
				VDD33*2	—	28	—	mA
			With 2.5-V built-in Regulator	VDD11, VDD11A	—	325	515	mA
				VDD25A	—	—	—	
				VDD33*2	—	28	—	mA
				VDDREG_33, AVDDREG_33	—	248	289	mA
Off-state current	I_{OZ}	$V_I = V_{DD}$ or GND	3.3 V output		—	—	±10	μA
		V_I = GND	5 V-tolerant buffer		—	—	−10	μA
		$V_I \leq$ 5.8 V			—	—	+10	μA
Output short circuit current*1	I_{OS}	V_O = GND	—		—	—	−250	mA
Input leakage current (3.3 V buffer)	I_I	$V_I = V_{DD}$ or GND	Normal input		—	—	±10	μA
		V_I = GND	With pull-up resistor (130 kΩ)		−6.7	—	−195	μA
		$V_I = V_{DD}$	With pull-down resistor (160 kΩ)		6.7	—	195	μA
Input leakage current (5 V-tolerant buffer)	I_I	V_I = GND	With pull-up resistor (130 kΩ)		−6.7	—	−195	μA

Note 1. The output short circuit time is no more than one second and is only for one pin.
2. The operating current of I/O differs depending on the conditions (for example, loads, waveform distortion, and toggle frequency). Measure actual current under the mounting environment.

Remark. The (+) and (–) signs in the table indicate the current direction. Current flowing to the device is indicated by (+) and current flowing out is indicated by (–).

Table 29.7 DC Characteristics (VDD = 3.3 ± 0.165 V, TA = −40 to +85°C) (2/2)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-level output current (3.3 V buffer)	I_{OL}	V_{OL} = 0.4 V	8 mA type	8.0	—	—	mA
			10 mA type	10.0	—	—	mA
			12 mA type	12.0	—	—	mA
Low-level output current (5 V-tolerant buffer)	I_{OL}	V_{OL} = 0.4 V	4 mA type	4.0	—	—	mA
High-level output current (3.3 V buffer)	I_{OH}	V_{OH} = V_{DD} − 0.4 V	8 mA type	−8.0	—	—	mA
			10 mA type	−10.0	—	—	mA
			12 mA type	−12.0	—	—	mA
High-level output current (5 V-tolerant buffer)	I_{OH}	V_{OH} = V_{DD} − 0.4 V	4 mA type	−4.0	—	—	mA
Low-level output voltage	V_{OL}	I_{OL} = 0 mA	3.3 V buffer	—	—	0.1	V
			5 V-tolerant buffer	—	—	0.1	V
High-level output voltage	V_{OH}	I_{OL} = 0 mA	3.3 V buffer	V_{DD} − 0.1 V	—	—	V
			5 V-tolerant buffer	V_{DD} − 0.1 V	—	—	V

Table 29.8 DC Characteristics (2.5-V built-in Regulator: V_{DD} = 3.3 ± 0.165 V, T_A = −40 to +85°C)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output voltage	V_{DD}	REG_OUT pin*	2.5 − 3%	—	2.5 + 3%	V
Output current	I_O		—	—	400	mA
Conversion efficiency	—	—	—	80	—	%

Note. The power supply of 2.5 V via the REG_OUT pin is dedicated to the VDD25A.
The pin is not available for 2.5 V power supply to other devices.

29.5 Pull-Up/Pull-Down Resistor Values

Table 29.9 Pull-Up/Pull-Down Resistor Values (V_{DD} = 3.3 ± 0.165 V, T_A = −40 to +85°C)

Item	Library Specification	MIN.	TYP.	MAX.	Unit
Pull-up resistor (3.3 V buffer)	130 kΩ	18	130	450	kΩ
Pull-down resistor (3.3 V buffer)	160 kΩ	18	160	450	kΩ
Pull-up resistor (5 V-tolerant buffer)	130 kΩ	18	130	450	kΩ

29.6 Pin Capacitance

Table 29.10 Pin Capacitance

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input buffer	C_B	XT2 pin	—	—	7.0	pF
		Other than XT2 pin	—	—	10.0	pF
Output buffer		—	—	—	10.0	pF
I/O buffer		—	—	—	10.0	pF

29.7 Power-On/Off Sequence

Table 29.11 lists external power supplies to R-IN32M4-CL3. Figure 29.1 and Figure 29.3 show the power-on/off sequence.
There is no particular rule for the power-on sequence. We recommend supplying external power voltage VDD11 first and then supplying external power voltage VDD33. On the other hand, when turning off the power, disconnect VDD33, then VDD11.

If VDD33 is supplied first, note that the I/O modes of the I/O buffers are not fixed and outputs become undefined over the period between VDD33 and VDD11 rising to their thresholds.
3.3 V must be applied to the input pins only after the power supply voltages have been applied.

Table 29.11 External Power Supplies

External power supply	Voltage [V]	External pin name
VDD33	3.3 ± 0.165*	VDD33 VDDREG_33 AVDDREG_33
VDD25	2.5 ± 0.125*	VDD25A
VDD11	1.15 ± 0.06*	VDD11 VDD11A PLL_VDD

Note. Ripple incorporated value. As a target value, set the DC component to within ±3% and the ripple component to within ±2%.

29.7.1 Power-On/Off Sequence without 2.5-V built-in Regulator

(1) Supplying Power Voltages

Supply power voltages so that the following two conditions are both satisfied.

1) The period from when VDD33, VDD25, or VDD11 reaches 10% VDD to when all of them reach 90% VDD or higher is within 100 ms.
2) The period from when VDD33, VDD25, or VDD11 reaches 95% VDD to when all of them reach 95% VDD or higher is within 50 ms.

(2) Turning Off Power Voltages

Turn off power voltages so that the following two conditions are both satisfied.

1) The period from when VDD33, VDD25, or VDD11 reaches 90% VDD to when all of them reach 10% VDD or lower is within 100 ms.
2) The period from when VDD33, VDD25, or VDD11 reaches 95% VDD to when all of them reach 95% VDD or lower is within 50 ms.

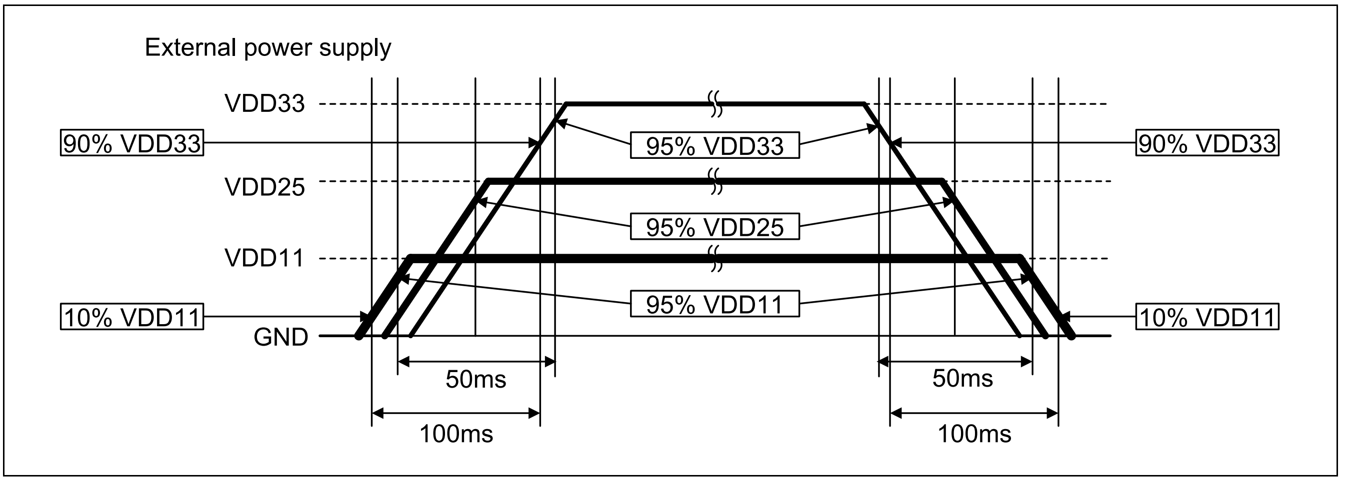


Figure 29.1 Power-On/Off Sequence (without 2.5-V built-in Regulator)

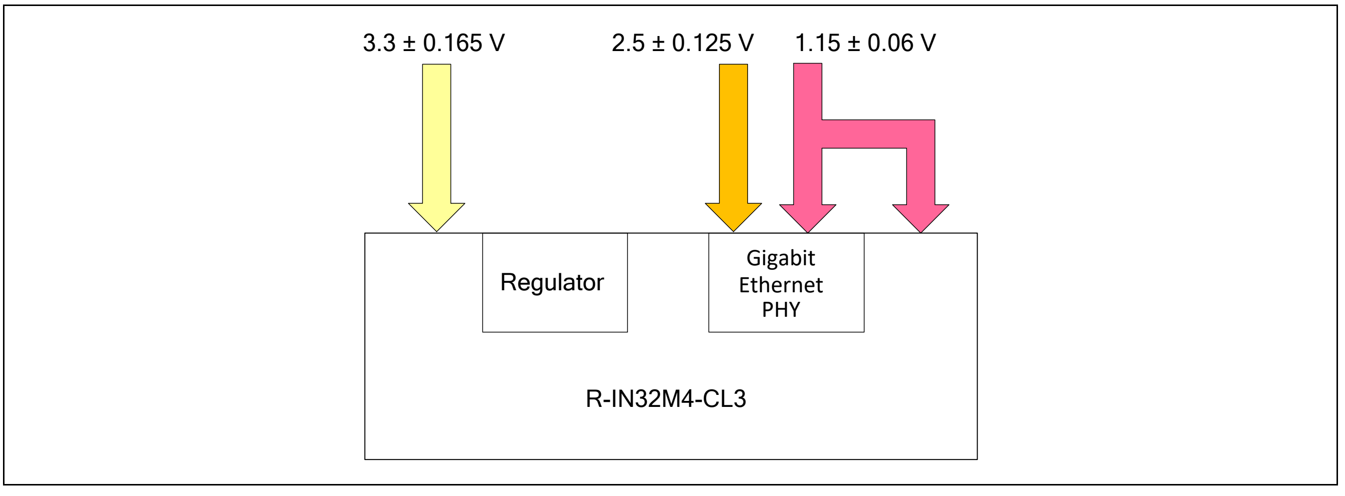


Figure 29.2 Power Supply Path to R-IN32M4-CL3 (without 2.5-V built-in Regulator)

29.7.2 Power-On/Off Sequence with 2.5-V built-in Regulator

(1) Supplying Power Voltages

Supply power voltages so that the following two conditions are both satisfied.

1) The period from when VDD33 or VDD11 reaches 10% VDD to when both of them reach 90% VDD or higher is within 100 ms.
2) The period from when VDD33 or VDD11 reaches 95% VDD to when both of them reach 95% VDD or higher is within 49 ms.

(2) Turning Off Power Voltages

Turn off power voltages so that the following two conditions are both satisfied.

1) The period from when VDD33 or VDD11 reaches 90% VDD to when both of them reach 10% VDD or lower is within 100 ms.
2) The period from when VDD33 or VDD11 reaches 95% VDD to when both of them reach 95% VDD or lower is within 49 ms.

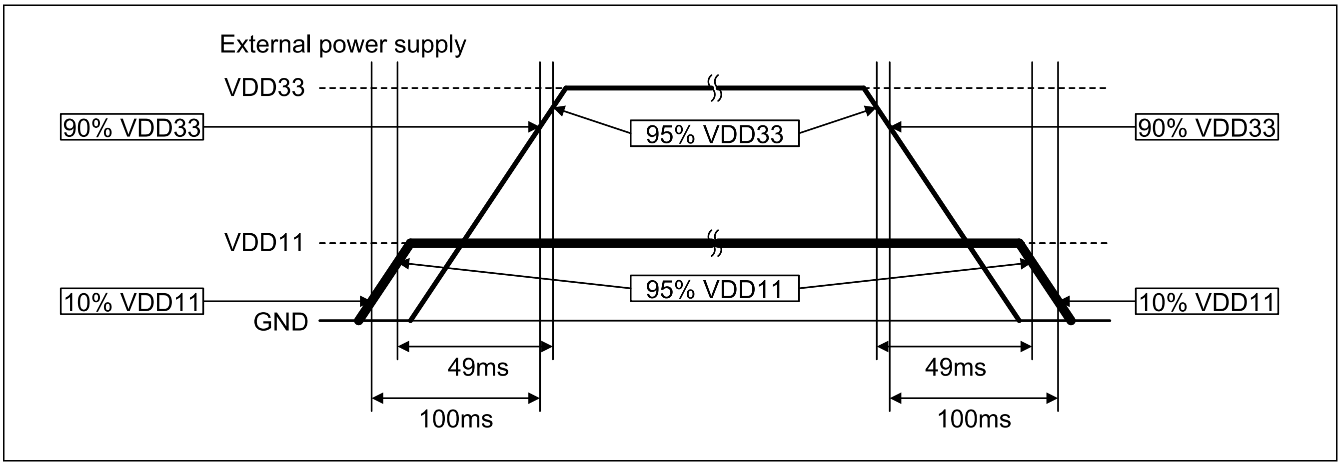


Figure 29.3 Power-On/Off Sequence (with 2.5-V built-in Regulator)

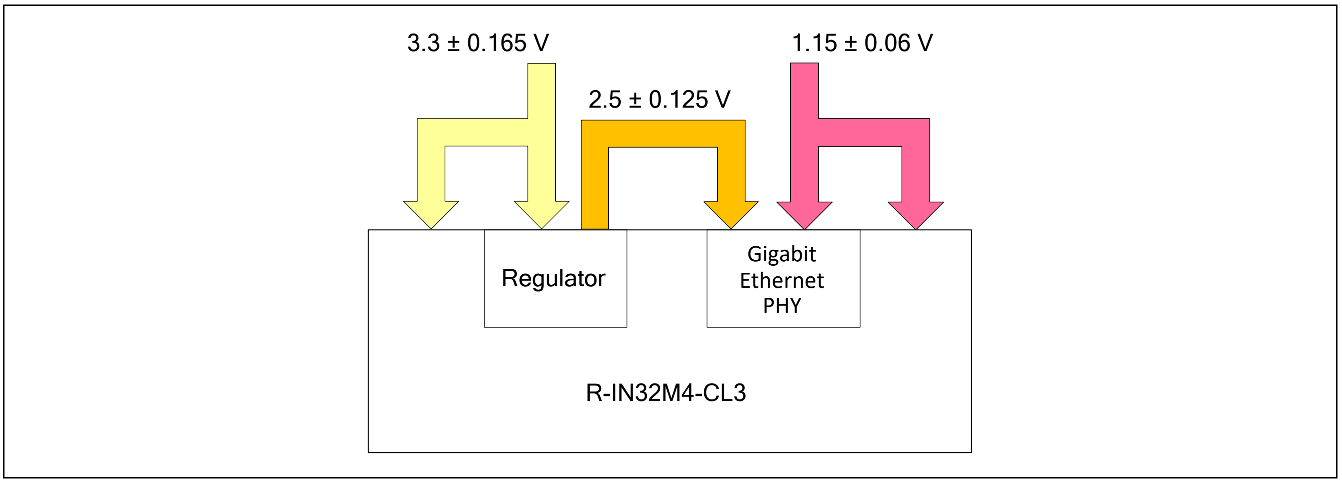


Figure 29.4 Power Supply Path to R-IN32M4-CL3 (with 2.5-V built-in Regulator)

29.8 AC Characteristics

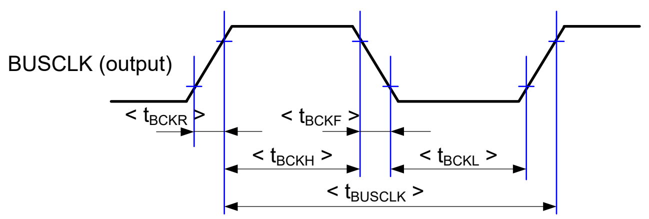
29.8.1 Clock Pins

(1) Input Clocks

Item	Symbol	Conditions	MIN	MAX	Unit
XT1 and XT2 clock frequency	t_{SYSCLK}	When the resonator is used (OSCTH pin = 0)	25 ± 50 ppm, 5 ps–rms		MHz
XT2 clock frequency		When the oscillator is used (OSCTH pin = 1)	25 ± 50 ppm, 5 ps–rms		MHz
XT2 clock duty			45	55	%
CCI_CLK2_097M	$t_{CCLIECLK}$	—	2.097152 ± 100 ppm		MHz
HBUSCLK	$t_{HBUSCLK}$	—	—	50	MHz
CSISCK0, CSISCK1	$t_{CSISSCK}$	Slave mode	—	16.6	MHz
TCK	t_{TCK}	—	—	50	MHz

(2) Output Clocks

Item	Symbol	Conditions	MIN	MAX	Unit
BUSCLK output cycle	t_{BUSCLK}	C_L = 15 pF	10	—	ns
BUSCLK High-level width	t_{BCKH}		0.5 × t_{BUSCLK} − 2.0	0.5 × t_{BUSCLK} + 2.0	ns
BUSCLK Low-level width	t_{BCKL}		0.5 × t_{BUSCLK} − 2.0	0.5 × t_{BUSCLK} + 2.0	ns
BUSCLK rising time	t_{BCKR}		—	1.2	ns
BUSCLK falling time	t_{BCKF}		—	1.2	ns
CSISCK0 and CSISCK1 output frequency	$t_{CSIMSCK}$	Master mode C_L = 15 pF	—	25	MHz
SCL0 and SCL1 output frequency	t_{SCL}	High-speed mode C_L = 30 pF	—	400	kHz
SMSCK output frequency	t_{SMSCK}	C_L = 15 pF	—	50	MHz
TRACECLK output frequency	$t_{TRACECLK}$	C_L = 15 pF	—	50	MHz



Note. For other clocks, refer to the AC characteristics of each interface.

Figure 29.5 Output Clock Timing

29.8.2 Reset Pins

Item	Symbol	Conditions	MIN	MAX	Unit
RESETZ input Low-level width	t_{WRSL}	—	Secure the time (oscillation stabilization time of the external oscillator circuit + 1 μ sec).	—	ns
HOTRESETZ input Low-level width	t_{WHRSL}	—		—	ns
PONRZ input Low-level width	t_{WPRSL}	—		—	ns
PONRZ input timing (for RESETZ ↑)	t_{SKPR}	—	0	—	ns

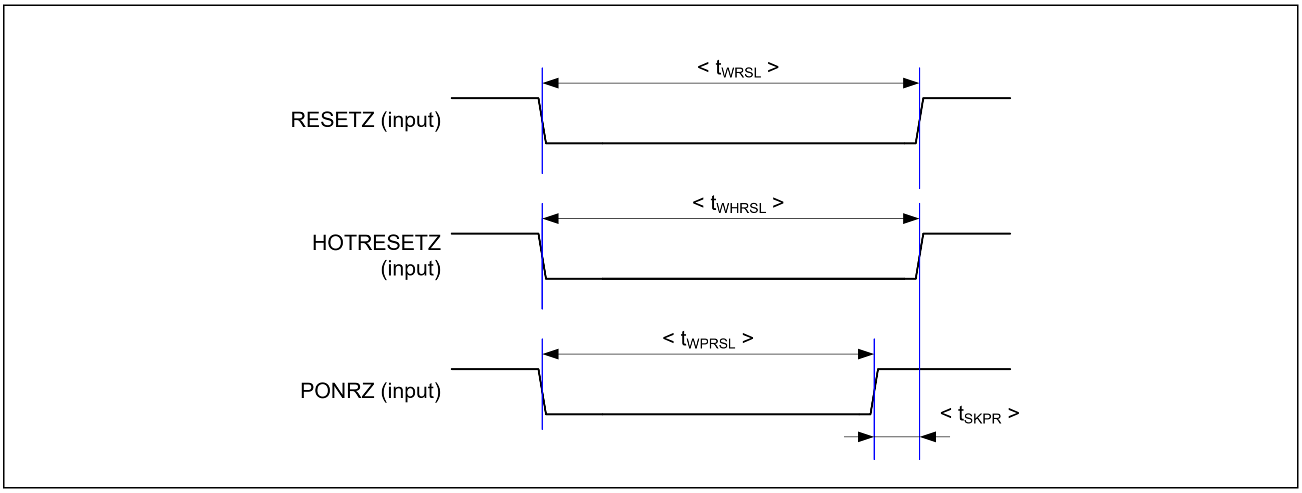


Figure 29.6 Reset Timing

29.8.3 External Memory Interface Pins

(1) How to calculate a delay value due to the external load

For the external memory interface pins of R-IN32M4-CL3, the listed values are for a load of 0 pF, but the actual loads will differ with users. Calculate the timing in accordance with the load conditions of the user. The user must also consider the wiring delay on the board.

Driving Ability	Delay Value per pF (ns)	
	MIN.	MAX.
8 mA	0.024	0.064
12 mA	0.013	0.039

Calculation Example:
When an address pin (8 mA output buffer) has a 30 pF load, the actual delay is as follows.

MIN. 1.0 ns (Minimum delay value for the load of 0 pF) + (0.024 × 30) ns = 1.72 ns
MAX. 7.0 ns (Maximum delay value for the load of 0 pF) + (0.064 × 30) ns = 8.92 ns

(2) Asynchronous SRAM controller access timing

Item	Symbol	MIN	MAX	Unit
Address and CSZ0–CSZ3 output delay time (for BUSCLK ↑)	t_{DKA}	1.0 (1.72)*	7.0 (8.92)*	ns
RDZ output delay time (for BUSCLK ↑)	t_{DKRD}	1.0 (1.72)*	7.0 (8.92)*	ns
WRZ0–WRZ3 (BENZ0–BENZ3) and WRSTBZ output delay time (for BUSCLK ↑)	t_{DKWR}	1.0 (1.72)*	7.0 (8.92)*	ns
BCYSTZ output delay time (for BUSCLK↑)	t_{DKBSL}	1.0 (1.72)*	7.0 (8.92)*	ns
WAITZ input setup time (for BUSCLK↓)	t_{SKW}	4.0	—	ns
WAITZ input hold time (for BUSCLK↓)	t_{HKW}	0	—	ns
Data input setup time (for BUSCLK↑)	t_{SKID}	4.0	—	ns
Data input hold time (for BUSCLK↑)	t_{HKID}	0	—	ns
Data output delay time (for BUSCLK↑)	t_{DKOD}	1.0 (1.72)*	7.0 (8.92)*	ns
Data float delay time (for BUSCLK↑)	t_{HKOD}	1.0 (1.72)*	7.0 (8.92)*	ns

Note. Values in parenthesis are the calculation results for the driving ability of 8 mA and the external load of 30 pF.

(a) Read timing

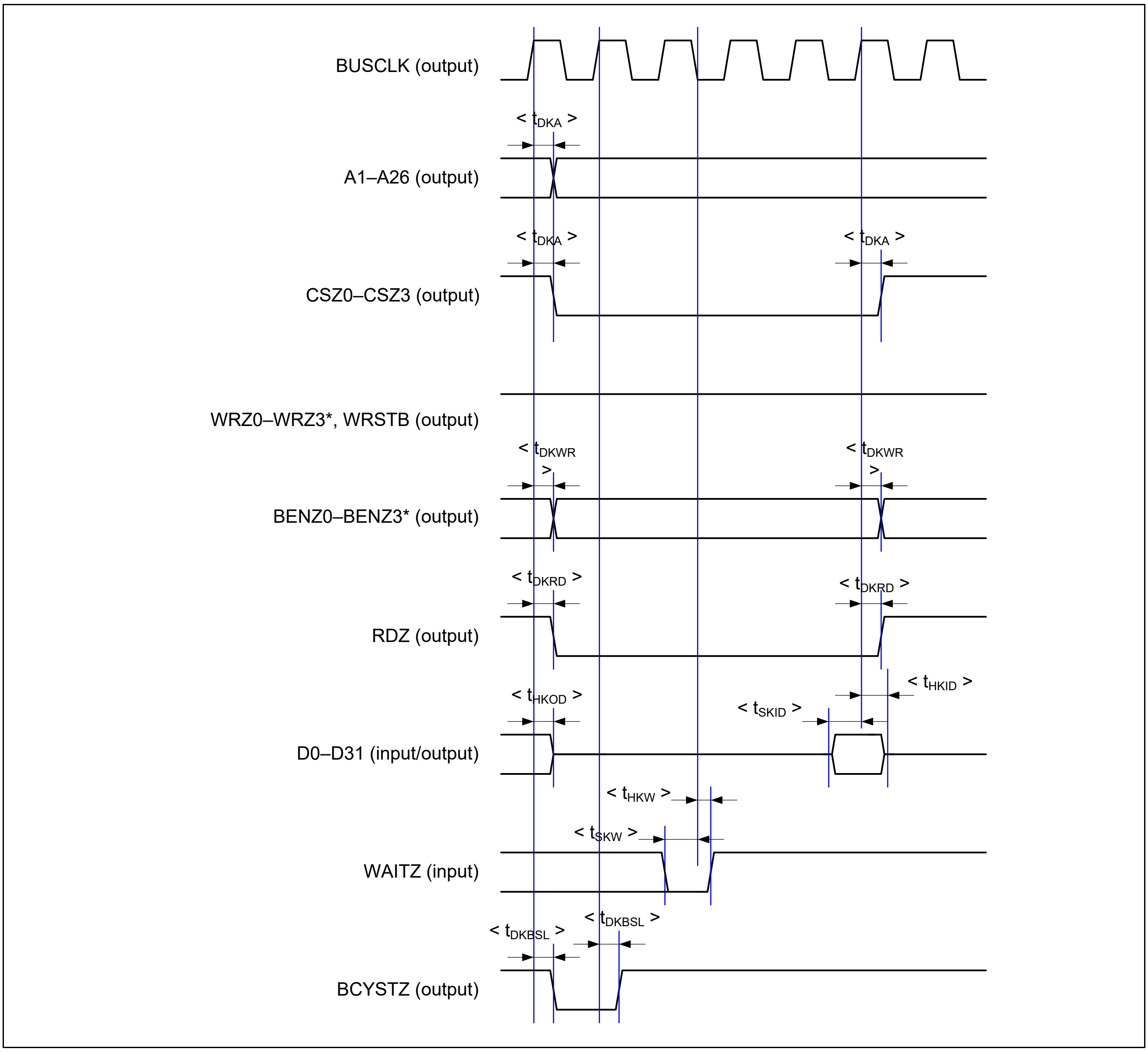


Figure 29.7 Memory Controller Read Timing (Asynchronous Memory)

Note. **The WRZ0–WRZ3 pins are multiplexed with the BENZ0–BENZ3 pin functions. The pin names are WRZ0–WRZ3.**
The WRZ0–WRZ3 pins are selected by default during a reset. Use the write enable switching register (WREN) to switch the pin functions of these pins.
For register details, refer to Section 14.3.5 "Write Enable Switching Register (WREN)".

Remark. **The above timing is for the case where the settings in the SMCn register for numbers of idle wait cycles, write recovery wait cycles, and address setting wait cycles are 0, and that for data wait cycles is 3.**

(b) Write timing

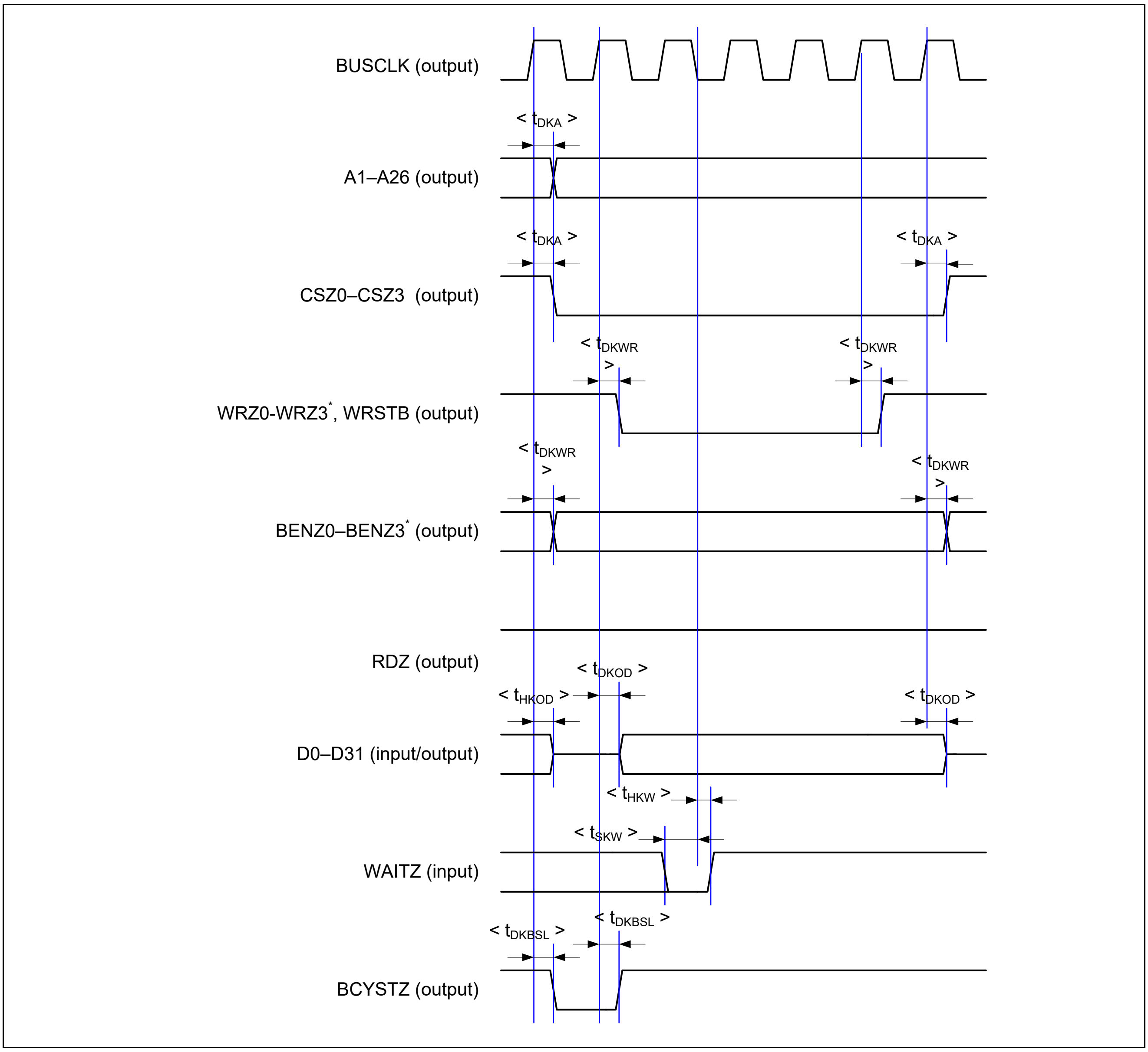


Figure 29.8 Memory Controller Write Timing (Asynchronous Memory)

Note. The WRZ0–WRZ3 pins are multiplexed with the BENZ0–BENZ3 pin functions. The pin names are WRZ0–WRZ3.
The WRZ0–WRZ3 pins are selected by default during a reset. Use the write enable switching register (WREN) to switch the pin functions of these pins.
For register details, refer to Section 14.3.5 "Write Enable Switching Register (WREN)".

Remark. The above timing is for the case where the settings in the SMCn register for numbers of idle wait cycles, write recovery wait cycles, and address setting wait cycles are 0, and that for data wait cycles is 3.

(3) Synchronous burst access memory controller access timing

Item	Symbol	MIN	MAX	Unit
BUSCLK output frequency	t_{BUSCLK}	—	50	MHz
Address and CSZ0–CSZ3 output delay time	t_{DKA}	1.0 (1.72)*	7.8 (9.72)*	ns
RDZ output delay time	t_{DKRD}	1.0 (1.72)*	7.8 (9.72)*	ns
WRZ0–WRZ3 (BENZ0–BENZ3) and WRSTBZ output delay time	t_{DKWR}	1.0 (1.72)*	7.8 (9.72)*	ns
ADVZ output delay time	t_{DKBSL}	1.0 (1.72)*	7.8 (9.72)*	ns
WAITZ and WAITZ1–WAITZ3 input setup time	t_{SKW}	5.3	—	ns
WAITZ and WAITZ1–WAITZ3 input hold time	t_{HKW}	0	—	ns
Data input setup time	t_{SKID}	5.3	—	ns
Data input hold time	t_{HKID}	0	—	ns
Data output delay time	t_{DKOD}	1.0 (1.72)*	7.8 (9.72)*	ns
Data float delay time	t_{HKOD}	1.0 (1.72)*	7.8 (9.72)*	ns

Note. Values in parenthesis are the calculation results for the driving ability of 8 mA and the external load of 30 pF.

(a) Read timing

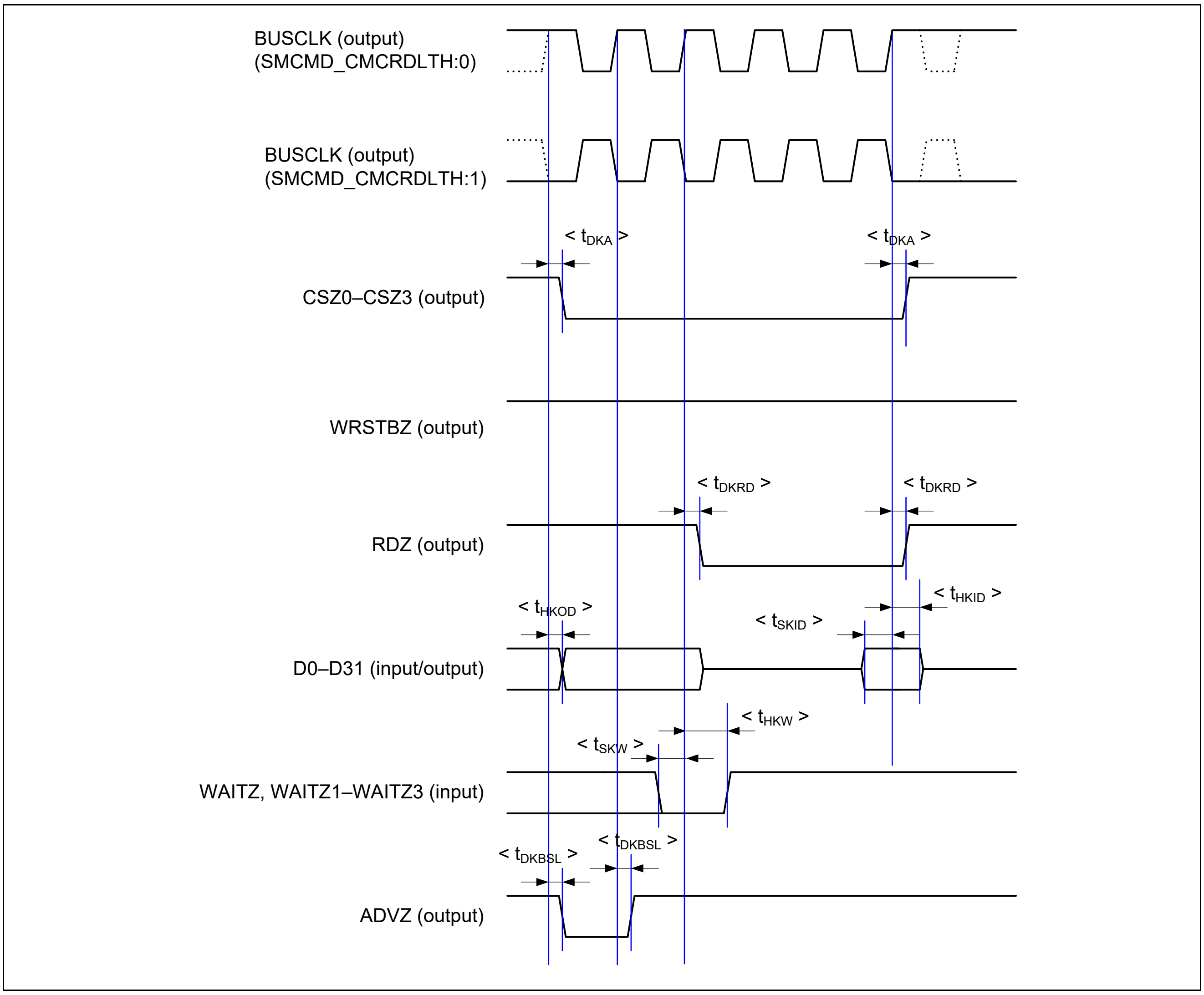


Figure 29.9 Memory Controller Read Timing (Clock Synchronous Memory)

Remark. The above timing is for the case where t_ceoe is 2 and t_rc is 4.

(b) Write timing

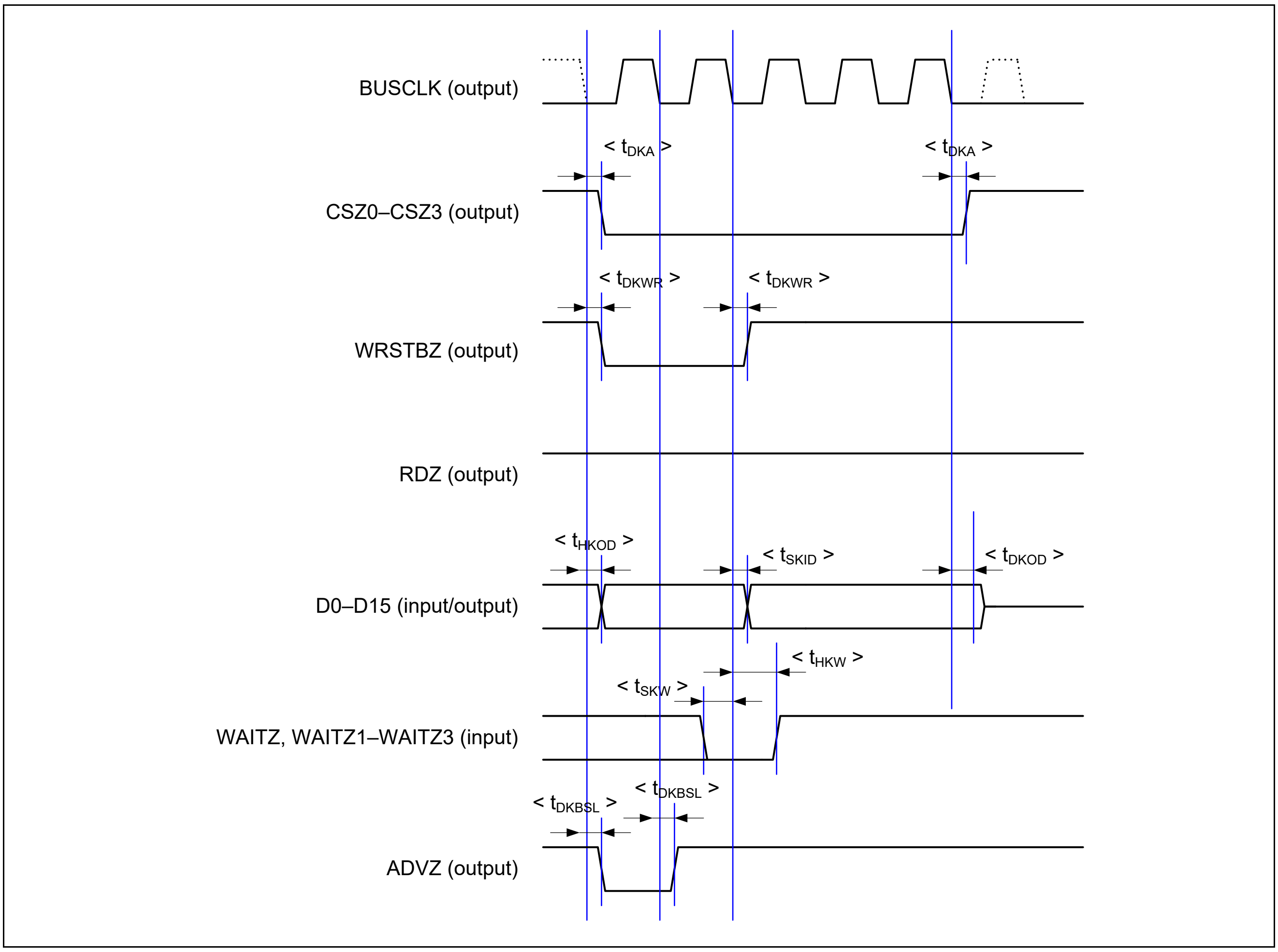


Figure 29.10 Memory Controller Write Timing (Clock Synchronous Memory)

Remark. The above timing is for the case where t_wp is 2 and t_wc is 5.

29.8.4 External MCU Interface Pins

The load condition for the external MCU interface pins: 65 pF (HD pin) and 35 pF (HWAITZ pin).

(1) Synchronous mode

No.	Item	Symbol	MIN	MAX	Unit
1	HBUSCLK High-level width	t_{HBHIGH}	$0.5 \times t_{HBUSCLK} - 2.1$	$0.5 \times t_{HBUSCLK} + 2.1$	ns
2	HBUSCLK Low-level width	t_{HBLOW}	$0.5 \times t_{HBUSCLK} - 2.1$	$0.5 \times t_{HBUSCLK} + 2.1$	ns
3	HBUSCLK input cycle	$t_{HBUSCLK}$	20.0	—	ns
4	Address, HCSZ, HPGCSZ, and HRDZ input setup time (for HBUSCLK ↑)	t_{SKHA}	4.0	—	ns
5	HBENZ0–HBENZ3 (HWRZ0–HWRZ3) and HWRSTBZ input setup time (for HBUSCLK ↑)	t_{SKHWR}	4.0	—	ns
6	Address, HCSZ, HPGCSZ, and HRDZ input hold time (for HBUSCLK ↑)	t_{HKHA}	1.0	—	ns
7	HBENZ0–HBENZ3 (HWRZ0–HWRZ3) and HWRSTBZ input hold time (for HBUSCLK ↑)	t_{HKHWR}	1.0	—	ns
8	HWRZ0–HWRZ3, HWRSTBZ recovery time (High-level width)	t_{WHWR}	35.0	—	ns
9	Data input setup time (for HBUSCLK ↑)	t_{SKIHD}	4.0	—	ns
10	Data input hold time (for HBUSCLK ↑)	t_{HKIHD}	1.0	—	ns
11	HWAITZ output delay time (for HCSZ, HPGCSZ ↓)	t_{DKHD}	2.2	—	ns
12	HWAITZ output delay time (for HWRSTBZ, HWRZ0 to HWRZ3 ↓)	t_{DKHWT}	2.2	—	ns
13	HWAITZ valid data output delay time (for HBUSCLK ↑)	t_{DKHWTV}	2.0	11.0	ns
14	HWAITZ valid data hold time (for HWRSTBZ, HWRZ0–HWRZ3 ↑)	t_{HKHWTV}	4.2	—	ns
15	HWAITZ output hold time (for HWRSTBZ, HWRZ0–HWRZ3 ↑)	t_{HKWTWR}	—	16.8	ns
16	Data and HWAITZ output hold time (for HCSZ, HPGCSZ ↑)	t_{HKWTCS}	—	16.8	ns
17	HRDZ recovery time (High-level width)	t_{WHRD}	35.0	—	ns
18	Data and HWAITZ output delay time (for HRDZ ↓)	t_{DKHDHR}	2.2	—	ns
19	Data fixing time (for HWAITZ ↑)	$t_{SKHDHWT}$	$t_{HBUSCLK} - 10.0$	—	ns
20	Data and HWAITZ valid data output hold time (for HRDZ ↑)	$t_{HKHWTHR}$	2.2	—	ns
21	Data and HWAITZ output hold time (for HRDZ ↑)	t_{HKOHD}	—	16.8	ns
22	Data and HWAITZ output delay time in on-page access (for addresses)	t_{DKPON}	4.2	15.4	ns
23	Data and HWAITZ output delay time in off-page access (for addresses) (when not crossing a 16-byte boundary)	t_{DKPOFF}	4.2	15.4	ns
	Data and HWAITZ output delay time in off-page access (for addresses) (when crossing a 16-byte boundary)	t_{DKPOFF}	4.2	49.5	ns
24	HWAITZ valid data output delay time (for HCSZ, HPGCSZ ↓)	$t_{DKWTVCS}$	—	15.4	ns

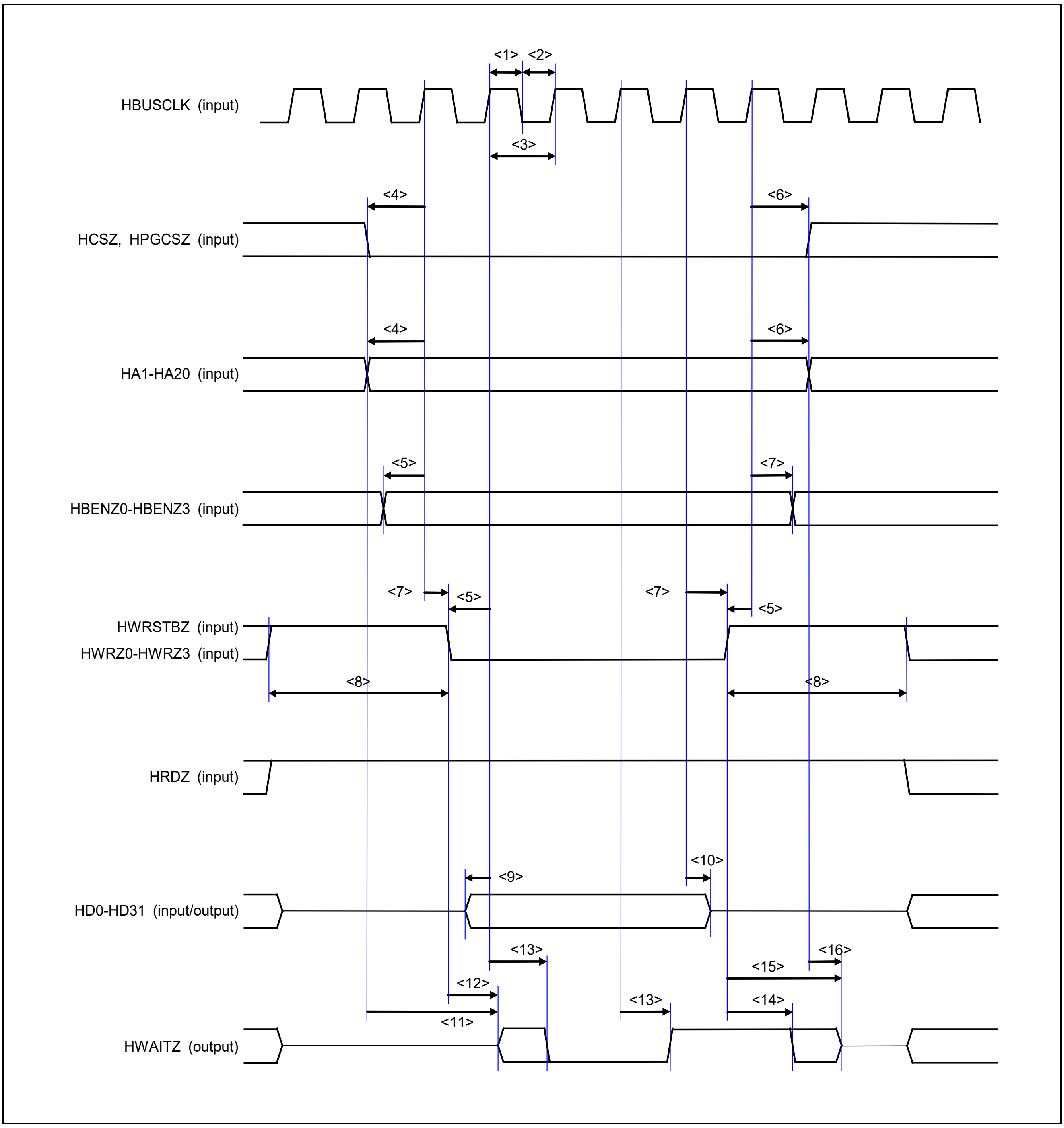


Figure 29.11 External MCU Interface Write Timing (MEMCSEL=L, HIFSYNC=H)

Note. Supply a stable signal to the address, data, and control lines during access.

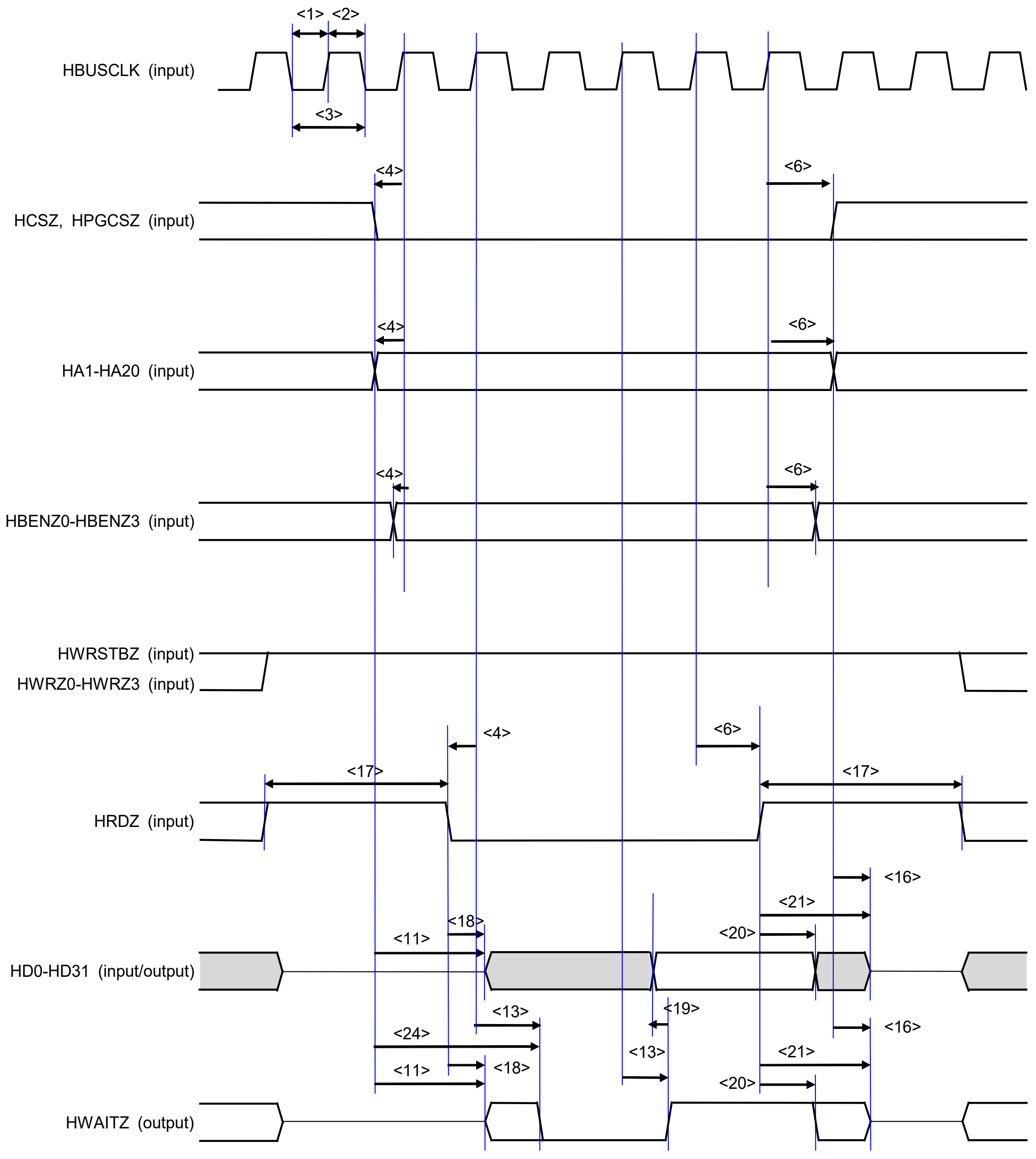


Figure 29.12 External MCU Interface Read Timing (MEMCSEL=L, HIFSYNC=H)

Note. Supply a stable signal to the address, data, and control lines during access.

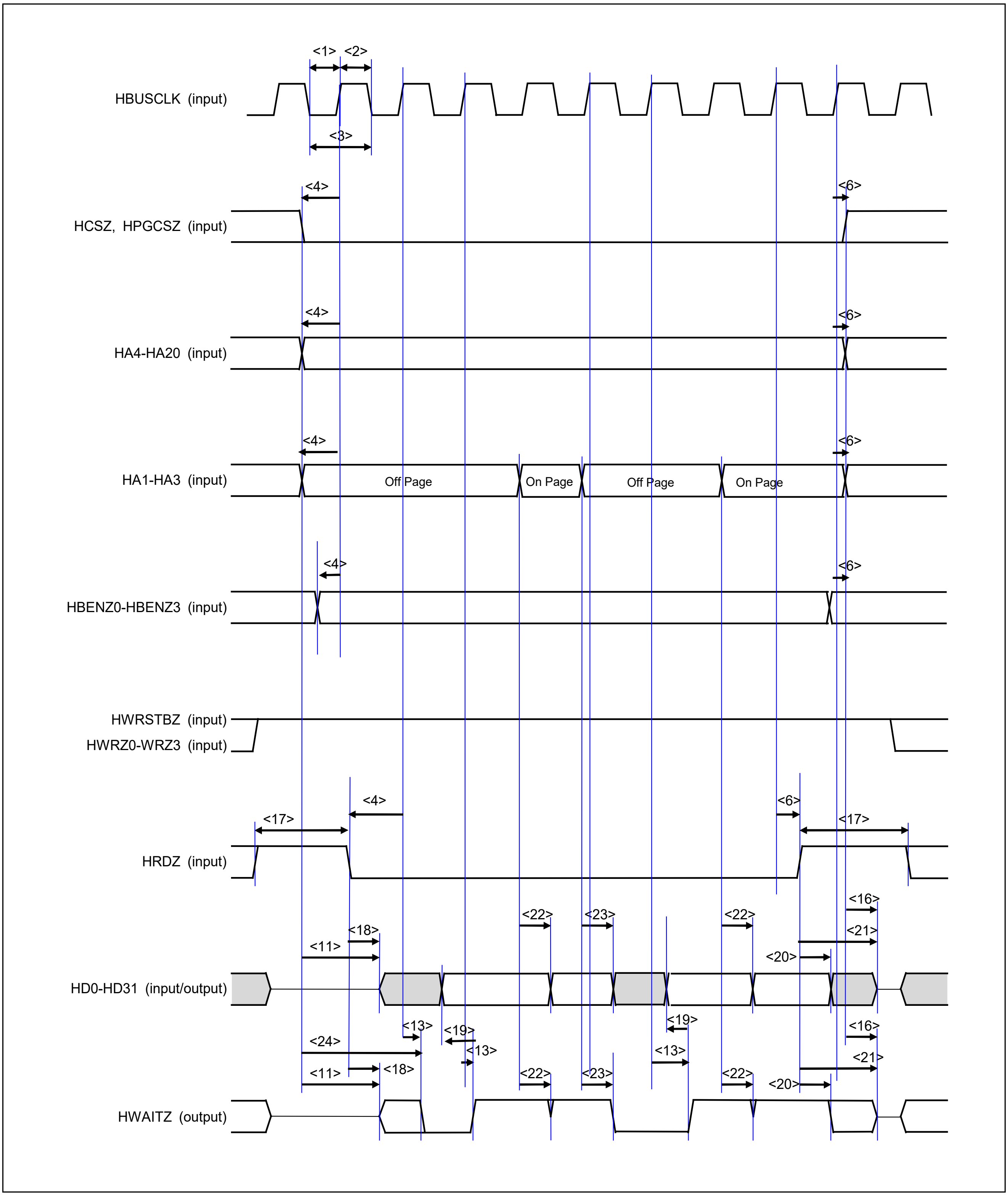


Figure 29.13 External MCU Interface Page Read Timing (MEMCSEL=L, HIFSYNC=H)

Note. Supply a stable signal to the address, data, and control lines during access.

(2) Asynchronous mode

No.	Item	Symbol	MIN	MAX	Unit
1	Address, HCSZ/HPGCSZ, and HBENZ0–HBENZ3 input setup time (for HWRSTBZ, HWRZ0–HWRZ3 ↓)	t_{ADDWRS}	7.0[*1] − 10 × n	—	ns
2	HWRZ0–HWRZ3, HWRSTBZ recovery time (High-level width)	t_{WRW}	35.0	—	ns
3	Data input setup time (for HWRSTBZ, HWRZ0–HWRZ3↓)	t_{WRS}	7.0[*1] − 10 × n	—	ns
4	Data input hold time (for HWRSTBZ, HWRZ0–HWRZ3↑)	t_{WRH}	7.0	—	ns
5	HWAITZ output delay time (for HCSZ or HPGCSZ↓)	t_{CLZ}	2.2	—	ns
6	HWAITZ output delay time (for HWRSTBZ, HWRZ0–HWRZ3↓)	t_{WAITD}	2.2	—	ns
7	HWAITZ valid data output delay time (for HWRSTBZ, HWRZ0–HWRZ3↓)	$t_{WRWAITF}$	—	15.4	ns
8	HWAITZ valid data output hold time (for HWRSTBZ, HWRZ0–HWRZ3↑)	t_{WAITVH}	4.2	—	ns
9	HWAITZ output hold time (for HWRZ0–3, HWRSTBZ↑)	t_{WAITH}	—	16.8	ns
10	Address and HWAITZ output hold time (for HCSZ, HPGCSZ↑)	t_{CHZ}	—	16.8	ns
11	Address, HCSZ, and HPGCSZ input setup time (for HRDZ↓)	t_{ADDRDS}	6.2[*2] − 10 × n	—	ns
12	Address input hold time in page access (for HRDZ↑)	t_{ADDRDH}	7.0	—	ns
13	HRDZ recovery time (High-level width)	t_{RDW}	35.0	—	ns
14	Data and HWAITZ output delay time (for HRDZ↓)	t_{RDLZ}	2.2	—	ns
15	HWAITZ valid data output delay time (for HRDZ↓)	$t_{RDWAITF}$	—	15.4	ns
16	Data fixing time (for HWAITZ↑)	t_{WAITR}	—	−6.2[*3] +10 × n	ns
17	Data and HWAITZ valid data output hold time (for HRDZ↑)	t_{DATAOH}	2.2	—	ns
18	Data and HWAITZ output hold time (for HRDZ↑)	t_{RDHZ}	—	16.8	ns
19	Data and HWAITZ output delay time in on-page access (for addresses)	$t_{PAGEOND}$	4.2	15.4	ns
20	Data and HWAITZ output delay time in off-page access (for addresses) (when not crossing a 16-byte boundary)	$t_{PAGEOFD}$	4.2	15.4	ns
	Data and HWAITZ output delay time in off-page access (for addresses) (when crossing a 16-byte boundary)	$t_{PAGEOFD}$	4.2	49.5	ns
21	HWAITZ valid data output delay time (for HCSZ, HPGCSZ↓)	t_{WAITVD}	—	15.4	ns

Note 1. When the value of WRSTD2–WRSTD0 in the HIFBTC register is 000B.
n: Setting of WRSTD2–WRSTD0
2. When the value of RDSTD1–RDSTD0 in the HIFBTC register is 00B.
n: Setting of RDSTD1–RDSTD0
3. When the value of RDDTS1–RDDTS0 in the HIFBTC register is 00B.
n: Setting of RDDTS1–RDDTS0

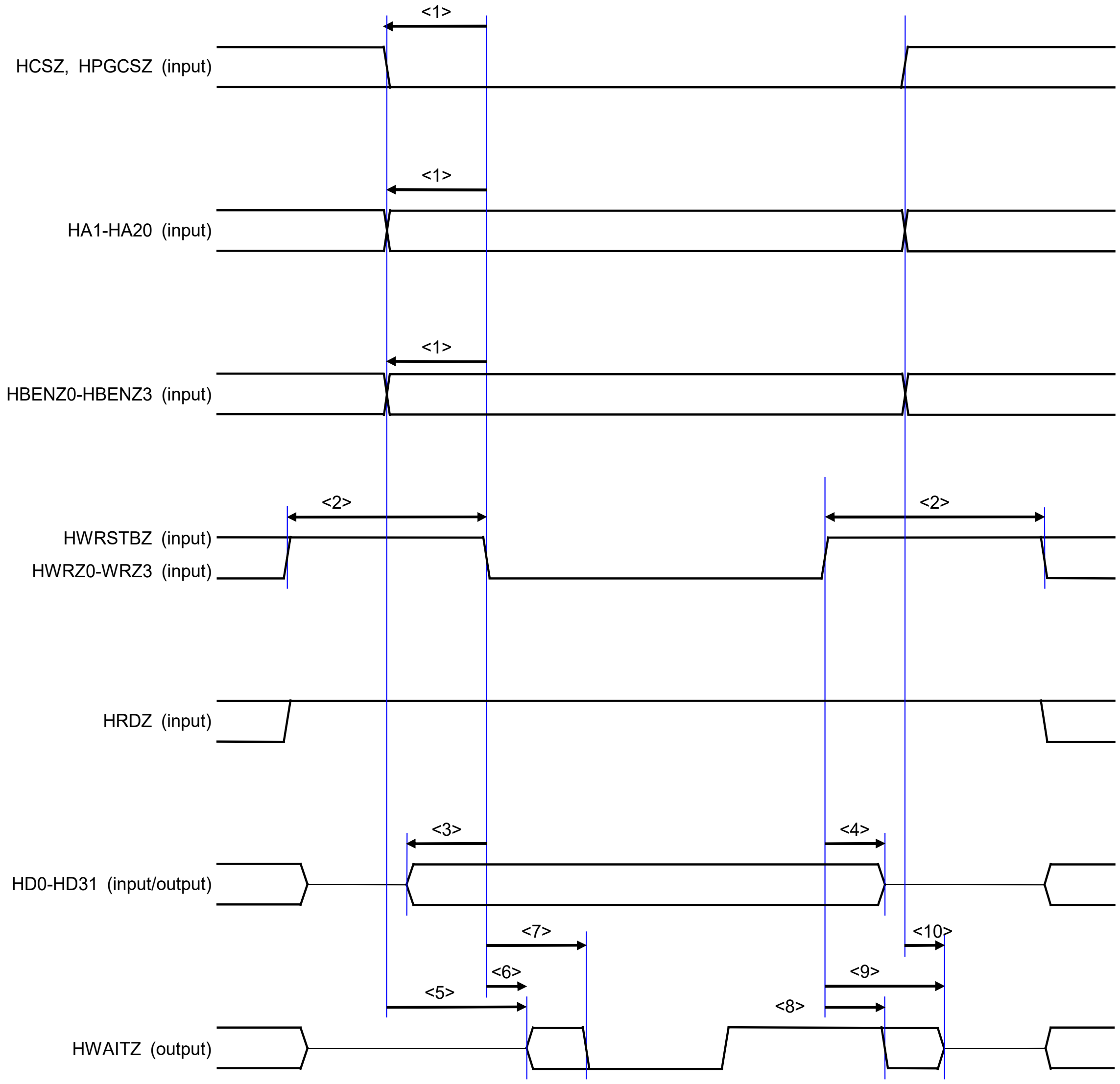


Figure 29.14 External MCU Interface Write Timing (MEMCSEL=L, HIFSYNC=L)

Note. Supply a stable signal to the address, data, and control lines during access.

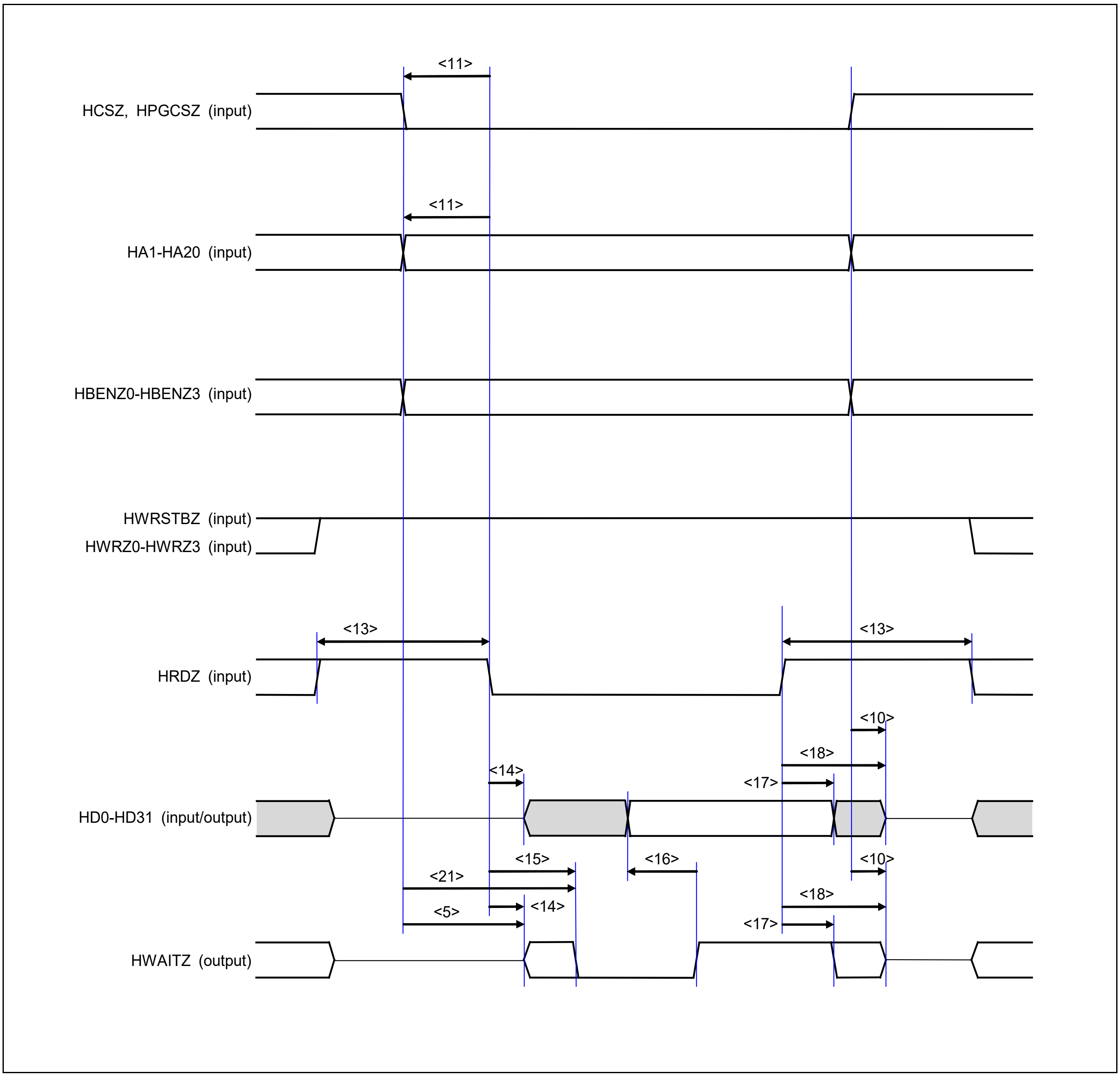


Figure 29.15 External MCU Interface Read Timing (MEMCSEL=L, HIFSYNC=L)

Note. Supply a stable signal to the address, data, and control lines during access.

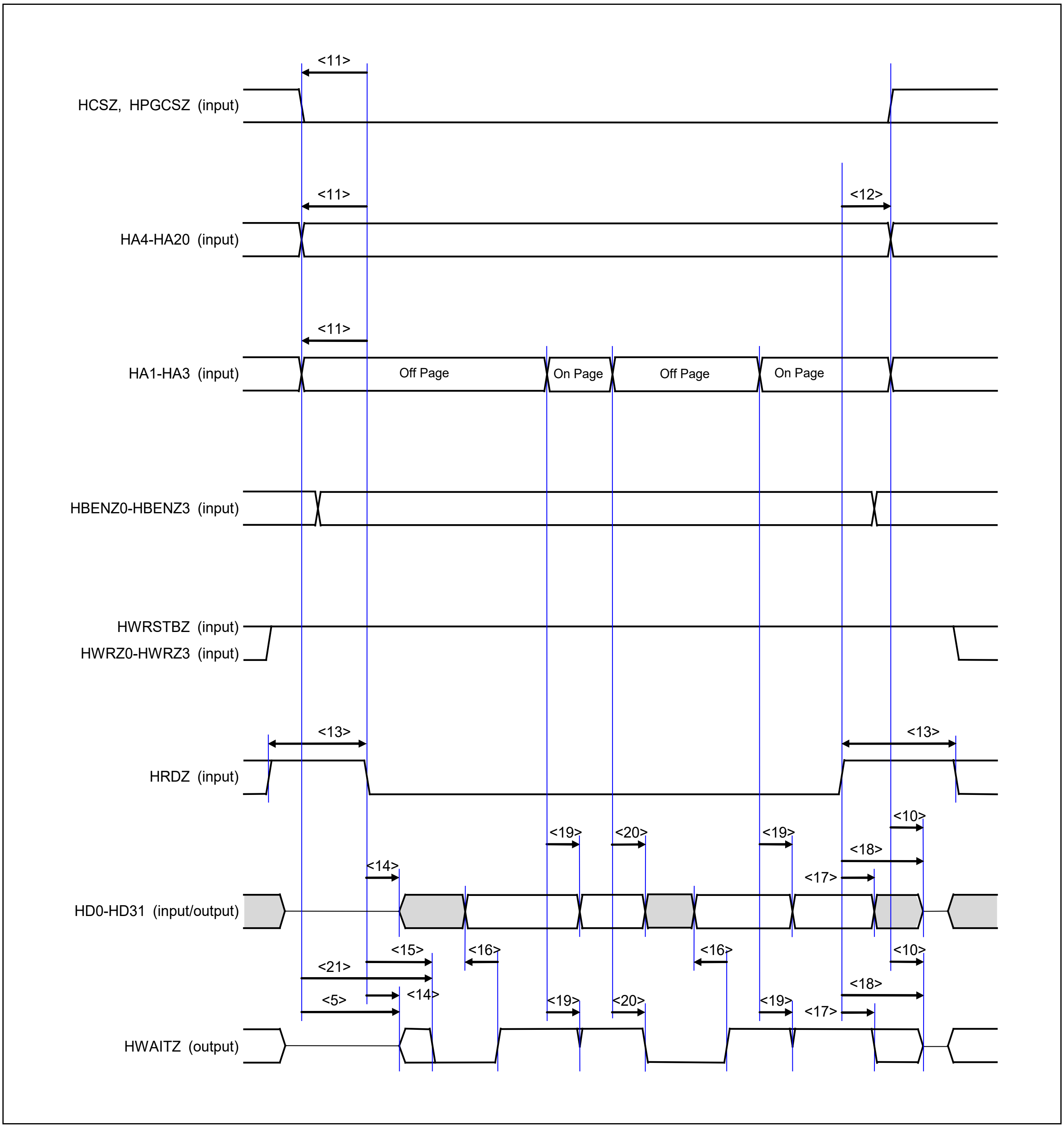


Figure 29.16 External MCU Interface Page Read Timing (MEMCSEL=L, HIFSYNC=L)

Note. Supply a stable signal to the address, data, and control lines during access.

(3) Synchronous SRAM type transfer mode

No.	Item	Symbol	MIN	MAX	Unit
1	HBUSCLK High-level width	t_{HBHIGH}	0.5 × $t_{HBUSCLK}$ − 2.1	0.5 × $t_{HBUSCLK}$ + 2.1	ns
2	HBUSCLK Low-level width	t_{HBLOW}	0.5 × $t_{HBUSCLK}$ − 2.1	0.5 × $t_{HBUSCLK}$ + 2.1	ns
3	HBUSCLK input cycle	$t_{HBUSCLK}$	20	—	ns
4	Address and HCSZ/HPGCSZ input setup time (for HBUSCLK ↑)	t_{SKPHA}	4.0	—	ns
5	Address and HCSZ/HPGCSZ input hold time (for HBUSCLK ↑)	t_{HKPCS}	1.0	—	ns
6	Address and HCSZ/HPGCSZ input setup time (for HBUSCLK ↓)	t_{SKNHA}	4.0	—	ns
7	Address, HCSZ, and HPGCSZ input hold time (for HBUSCLK ↓)	t_{HKNHA}	1.0	—	ns
8	HWRZ0–HWRZ3 input setup time (for HBUSCLK ↑)	t_{SKPHWR}	4.0	—	ns
9	HWRZ0–HWRZ3 input hold time (for HBUSCLK ↑)	t_{HKPHWR}	1.0	—	ns
10	HWRZ0–HWRZ3 input setup time (for HBUSCLK ↓)	t_{SKNHWR}	4.0	—	ns
11	HWRZ0–HWRZ3 input hold time (for HBUSCLK ↓)	t_{HKNHWR}	1.0	—	ns
12	HBCYSTZ, HWRSTBZ input setup time (for HBUSCLK ↑)	$t_{SKPHBCY}$	4.0	—	ns
13	HBCYSTZ, HWRSTBZ input hold time (for HBUSCLK ↑)	$t_{HKPHBCY}$	1.0	—	ns
14	HBCYSTZ, HWRSTBZ input setup time (for HBUSCLK ↓)	$t_{SKNHBCY}$	4.0	—	ns
15	HBCYSTZ, HWRSTBZ input hold time (for HBUSCLK ↓)	$t_{HKNHBCY}$	1.0	—	ns
16	HRDZ input setup time (for HBUSCLK ↑)	t_{SKPHRD}	4.0	—	ns
17	HRDZ input hold time (for HBUSCLK ↑)	t_{HKPHRD}	1.0	—	ns
18	HRDZ input setup time (for HBUSCLK ↓)	t_{SKNHRD}	4.0	—	ns
19	HRDZ input hold time (for HBUSCLK ↓)	t_{HKNHRD}	1.0	—	ns
20	Data input setup time (for HBUSCLK ↑)	t_{SKPHD}	4.0	—	ns
21	Data input hold time (for HBUSCLK ↑)	t_{HKPHD}	1.0	—	ns
22	Data input setup time (for HBUSCLK ↓)	t_{SKNHD}	4.0	—	ns
23	Data input hold time (for HBUSCLK ↓)	t_{HKNHD}	1.0	—	ns
24	Data output delay time (for HRDZ ↓)	t_{DKNHRD}	2.2	—	ns
25	Data output hold time (for HRDZ ↑)	t_{HKPHRD}	—	16.8	ns
26	Data output delay time (for HBUSCLK ↑)	t_{DKPHD}	2.0	10.0	ns
27	Data output delay time (for HBUSCLK ↓)	t_{DKNHD}	2.0	10.0	ns
28	HWAITZ output delay time (for HBUSCLK ↑)	t_{DKPHWT}	2.0	11.0	ns
29	HWAITZ output delay time (for HBUSCLK ↓)	t_{DKNHWT}	2.0	11.0	ns
30	Data output hold time (for HCSZ/HPGCSZ ↑)	t_{HKPHCS}	—	16.8	ns

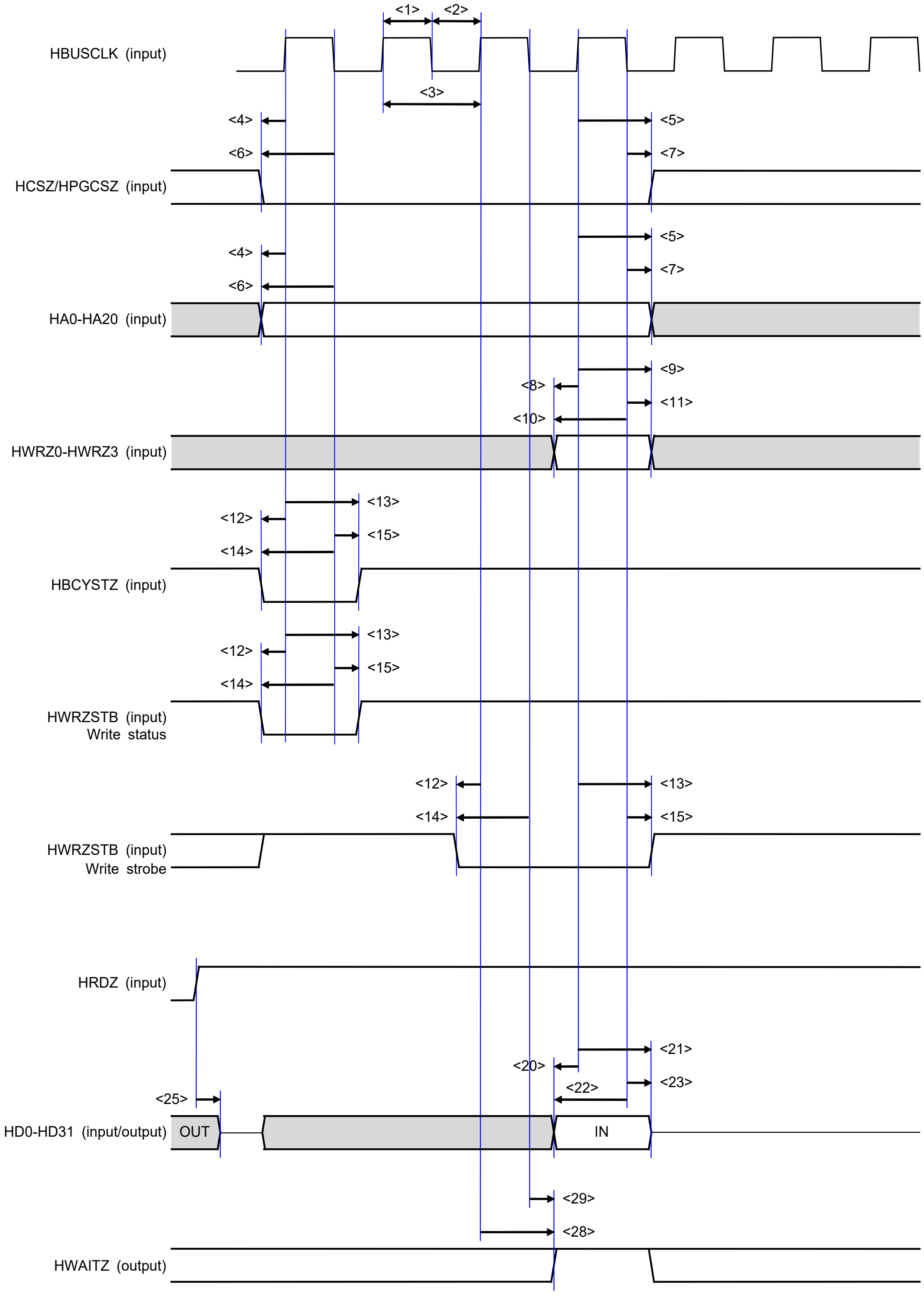


Figure 29.17 External MCU Interface Write Timing (MEMCSEL=H, ADMUXMODE=L)

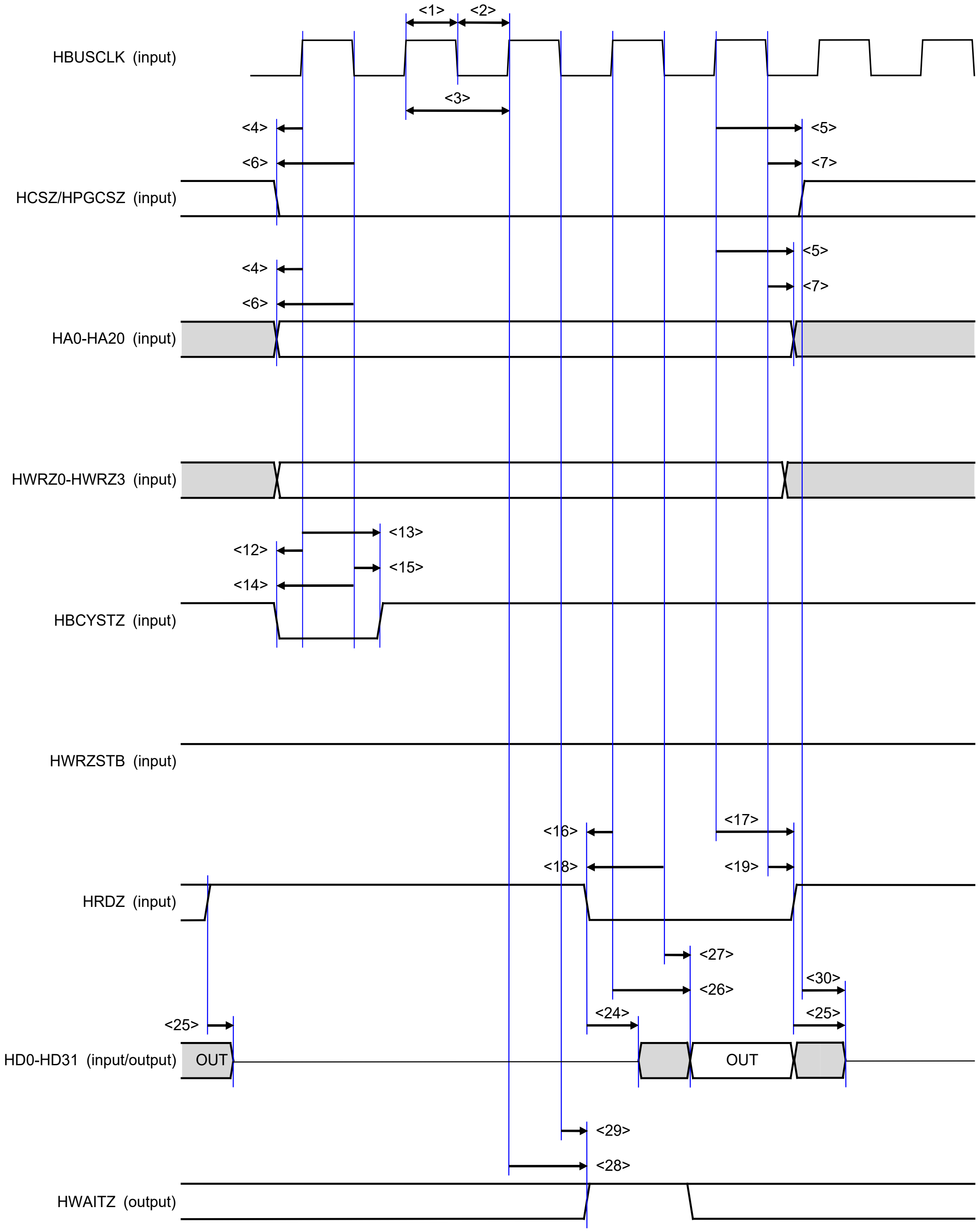


Figure 29.18 External MCU Interface Read Timing (MEMCSEL=H, ADMUXMODE=L)

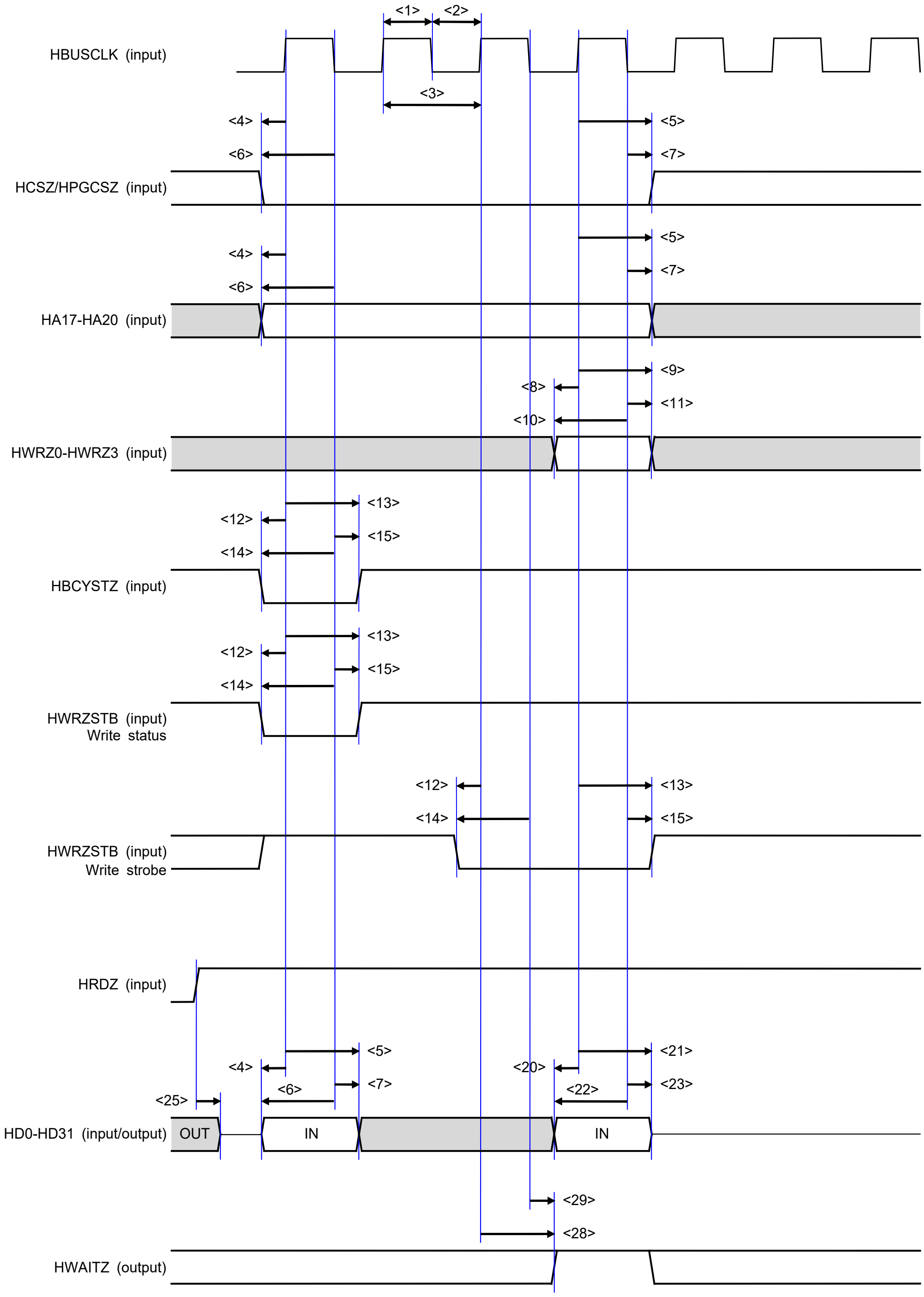


Figure 29.19 External MCU Interface Write Timing (MEMCSEL=H, ADMUXMODE=H)

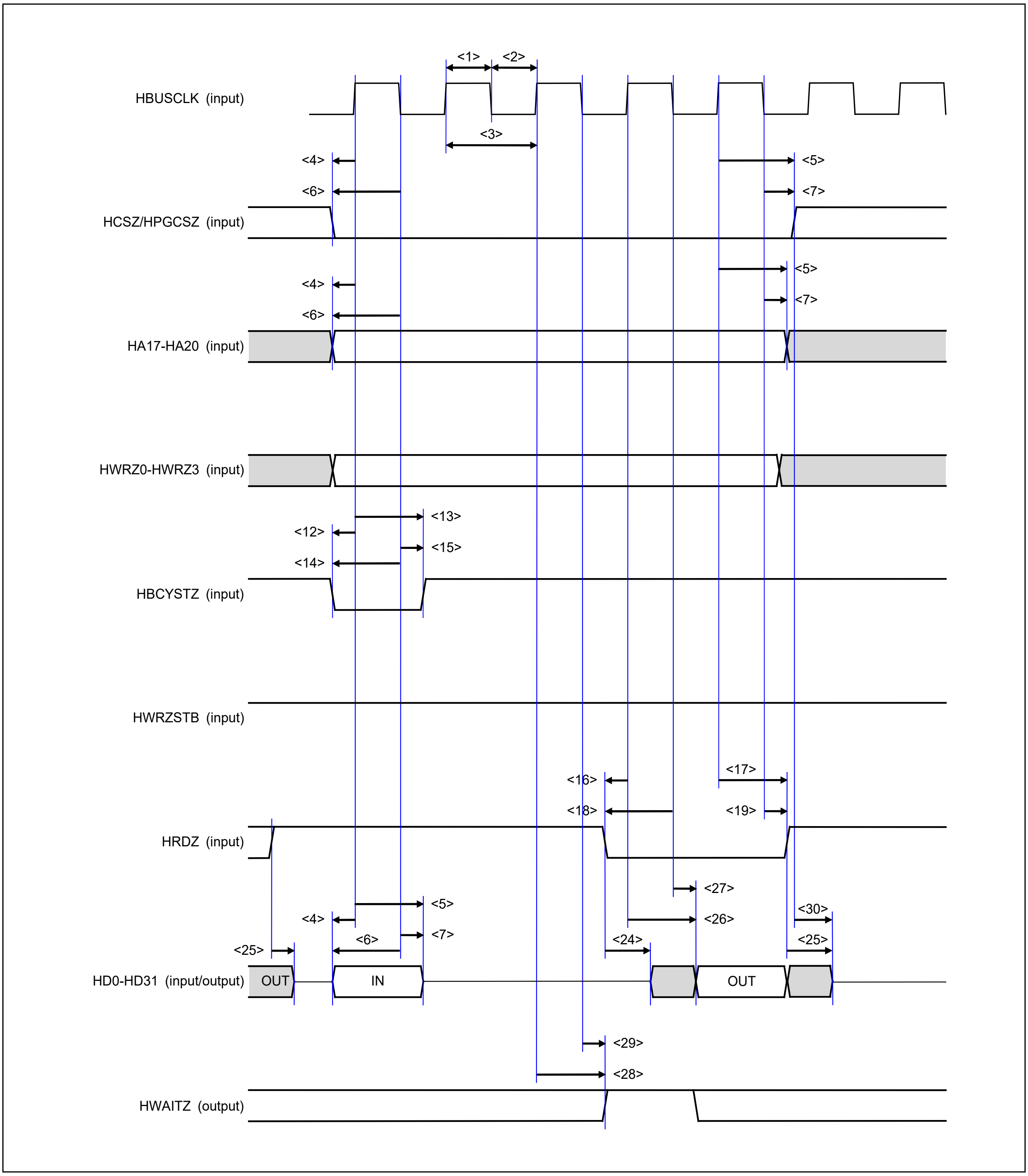


Figure 29.20 External MCU Interface Read Timing (MEMCSEL=H, ADMUXMODE=H)

29.8.5 Serial Flash ROM Interface

Item	Symbol	Conditions	MIN	MAX	Unit
SMSCK output cycle	t_{SFRCYC}	C_L = 15 pF	20	—	ns
SMSCK High-level width	t_{SMCKH}		0.5 × t_{SFRCYC} − 2.0	0.5 × t_{SFRCYC} + 2.0	ns
SMSCK Low-level width	t_{SMCKL}		0.5 × t_{SFRCYC} − 2.0	0.5 × t_{SFRCYC} + 2.0	ns
SMSCK rising time	t_{SMCKR}		—	1.9	ns
SMSCK falling time	t_{SMCKF}		—	1.9	ns
Delay time between SMCSZ falling and SMSCK rising	$t_{DSMCSCK}$	C_L = 15 pF, Freq = 50 MHz	6.0*	—	ns
Hold time from SMSCK rising to SMCSZ rising	$t_{DSMCKCS}$	C_L = 15 pF, Freq = 50 MHz	9.0*	—	ns
SMCSZ High-level width	t_{SMCSH}	C_L = 15 pF	14*	—	ns
SMIO0–SMIO3 input setup time (for SMSCK ↓)	t_{SSMIO}	—	6.0	—	ns
SMIO0–SMIO3 input hold time (for SMSCK ↓)	t_{HSMIO}	—	0	—	ns
SMIO0–SMIO3 output delay time (for SMSCK ↓)	t_{DSMIO}	C_L = 15 pF	-1.0	5.0	ns

Note. The timing can be extended by the setting of the SFMSSC register.
For details, refer to Section 17.2.2 "Chip Selection Control Register (SFMSSC)".

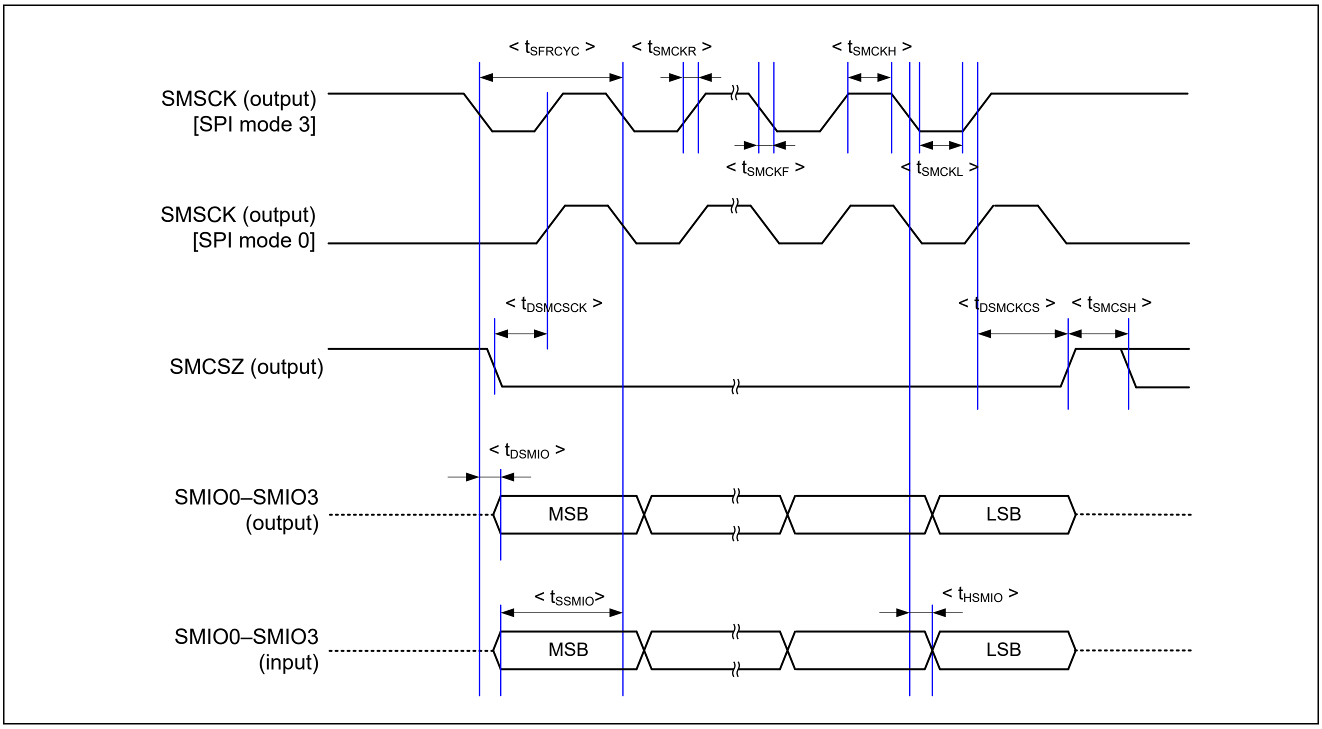


Figure 29.21 Serial Flash Memory Access Timing

29.8.6 External DMA Interface

Item	Symbol	Conditions	MIN	MAX	Unit
DMAREQZn and RTDMAREQZ input setup time (for BUSCLK ↑)	t_{SKDR}	—	7.0	—	ns
DMAREQZn and RTDMAREQZ input hold time 1	t_{HKDR1}	—	Until DMAACKZ↓, RTDMAACKZ↓	—	ns
DMAREQZn and REDMAREQZ input hold time 2 (for BUSCLK ↑)	t_{HKDR2}	—	—	t_{BUSCLK}*1 × m*2 – 7.0	ns
DMAACKZn and RTDMACKZ output delay time (for BUSCLK ↑)	t_{DKDA}	C_L = 30 pF	2.0	10.0	ns
DMAACKZ and RTDMAACKZ output Low-level width	t_{WDAL}	—	t_{BUSCLK}*1 × m*2 – 8	t_{BUSCLK}*1 × m*2 + 8	ns
DMATCZn and RTDMATCZ output delay time (for BUSCLK ↑)	t_{DKTC}	C_L = 30 pF	2.0	10.0	ns

Note 1. t_{BUSCLK} is one cycle (10 ns) of BUSCLK.

2. n = 0, 1, m = 1–31 (DMAIFC0, DMAIFC1, and RTMDAIFC registers)

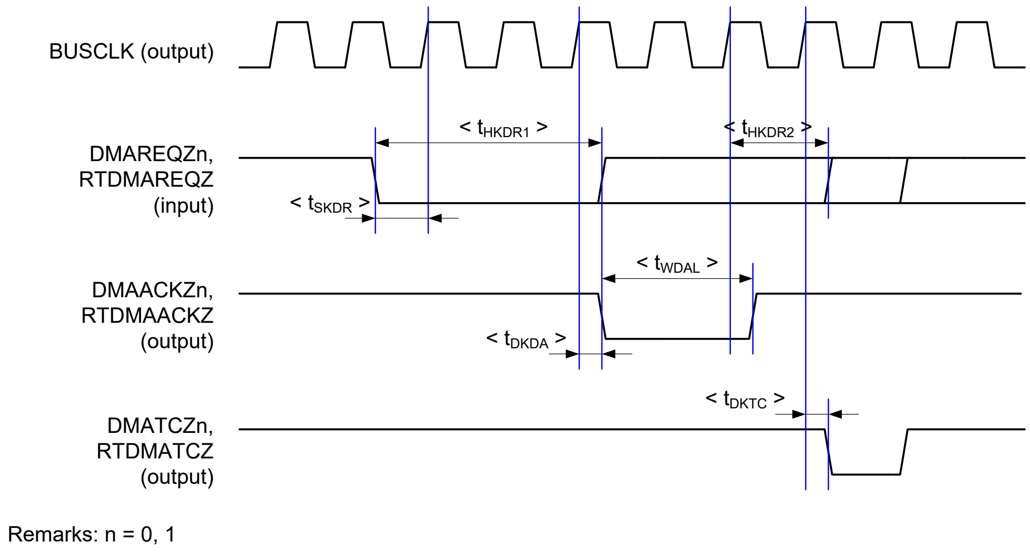


Figure 29.22 External DMA Access Timing

29.8.7 CSI Interface

The clocked serial interface (CSI) supports master mode and slave mode.

(1) Master mode

Item	Symbol	Conditions	MIN	MAX	Unit
CSISCKn output cycle	$t_{CSIMSCK}$	C_L = 15 pF	40	—	ns
CSISCKn output High-level width	t_{WSKH}	C_L = 15 pF	0.5 × $t_{CSIMSCK}$ − 5.0	—	ns
CSISCKn output Low-level width	t_{WSKL}	C_L = 15 pF	0.5 × $t_{CSIMSCK}$ − 5.0	—	ns
CSISIn input setup time (for CSISCKn ↑)	t_{SMSI}	—	8.5	—	ns
CSISIn input setup time (for CSISCKn ↓)	t_{SMSI}	—	8.5	—	ns
CSISIn input hold time (for CSISCKn ↑)	t_{HMSI}	—	7.0	—	ns
CSISIn input hold time (for CSISCKn ↓)	t_{HMSI}	—	7.0	—	ns
CSISOn output delay time (for CSISCKn ↑)	t_{DMSO}	C_L = 15 pF	—	7.0	ns
CSISOn output delay time (for CSISCKn ↓)	t_{DMSO}		—	7.0	ns
CSISOn output hold time (for CSISCKn ↑)	t_{HMSO}		0.5 × $t_{CSIMSCK}$ − 5.0	—	ns
CSISOn output hold time (for CSISCKn ↓)	t_{HMSO}		0.5 × $t_{CSIMSCK}$ − 5.0	—	ns

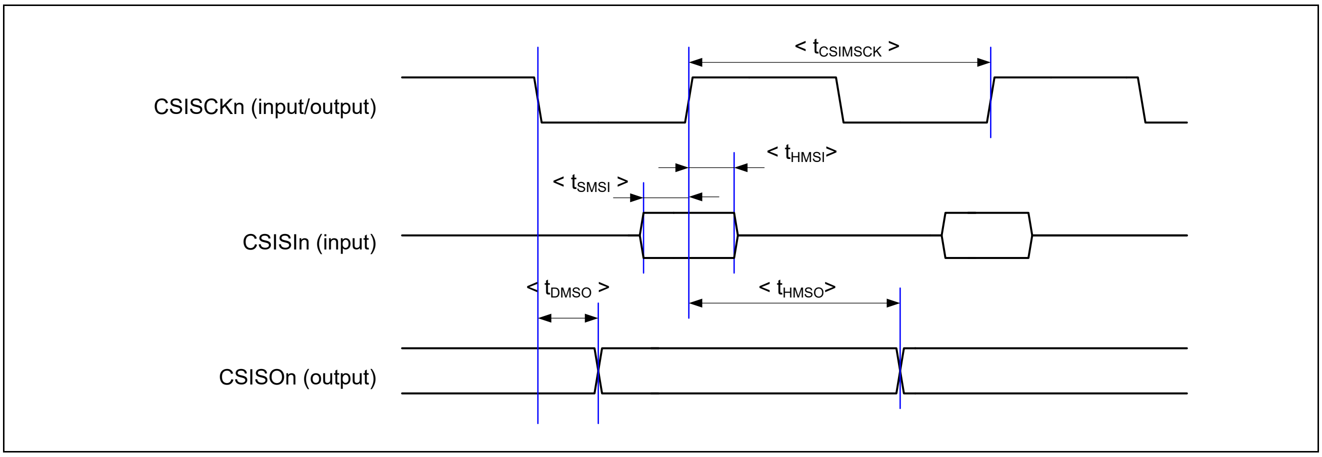


Figure 29.23 CSI Access Timing (Master Mode)

Remarks 1. n = 0, 1

2. The above is an example of the timing for data output of "for CSISCKn ↓" and data input of "for CSISCKn ↑". Read the timing for reference in accordance with the operating mode.

(2) Slave mode

Item	Symbol	Conditions	MIN	MAX	Unit
CSISCKn input cycle	$t_{CSISSCK}$	—	60	—	ns
CSISCKn input High-level width	t_{WSKH}	—	0.5 × $t_{CSISSCK}$ − 5.0	—	ns
CSISCKn input Low-level width	t_{WSKL}	—	0.5 × $t_{CSISSCK}$ − 5.0	—	ns
CSISIn input setup time (for CSISCKn ↑)	t_{SSSI}	—	10.0	—	ns
CSISIn input setup time (for CSISCKn ↓)	t_{SSSI}	—	10.0	—	ns
CSISIn input hold time (for CSISCKn ↑)	t_{HSSI}	—	15	—	ns
CSISIn input hold time (for CSISCKn ↓)	t_{HSSI}	—	15	—	ns
CSISOn output delay time (for CSISCKn ↑)	t_{DSSO}	C_L = 15 pF	—	10.0	ns
CSISOn output delay time (for CSISCKn ↓)	t_{DSSO}		—	10.0	ns
CSISOn output hold time (for CSISCKn ↑)	t_{HSSO}		0.5 × $t_{CSISSCK}$ − 5.0	—	ns
CSISOn output hold time (for CSISCKn ↓)	t_{HSSO}		0.5 × $t_{CSISSCK}$ − 5.0	—	ns

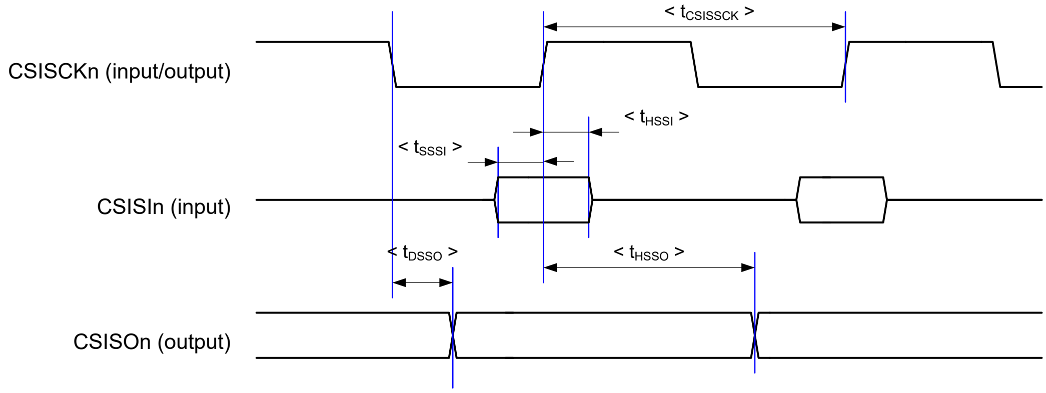


Figure 29.24 CSI Access Timing (Slave Mode)

Remarks 1. n = 0, 1

2. The above is an example of the timing for data output of "for CSISCKn ↓" and data input of "for CSISCKn ↑". Read the timing for reference in accordance with the operating mode.

29.8.8 I²C Interface

Item		Symbol	Conditions	Normal Mode		High-Speed Mode		Unit
				MIN	MAX	MIN	MAX	
SCLn input/output frequency		t_{SCL}	C_L = 30 pF	0	100	0	400	kHz
Bus-free time between the stop condition and start condition		t_{BUF}		4.7	—	1.3	—	μs
Hold time		t_{HSTA}		4.0	—	0.6	—	μs
SCLn clock Low-level width		t_{SCLL}		4.7	—	1.3	—	μs
SCLn clock High-level width		t_{SCLH}		4.0	—	0.6	—	μs
Setup time for the start and restart conditions		t_{SSTA}		4.7	—	0.6	—	μs
Data hold time	For a CBUS compatible master	t_{HDAT}		5.0	—	—	—	μs
	For an I²C bus			0	—	0	0.9	μs
Data setup time		t_{SDAT}		250	—	100	—	ns
SDAn and SCLn rising time		t_{SCLR}		—	1000	20 + 0.1 × C_b	300	ns
SDAn and SCLn falling time		t_{SCLF}		—	300	20 + 0.1 × C_b	300	ns
Setup time for the stop condition		t_{SSTO}		4.0	—	0.6	—	μs
Pulse width of spike suppressed by input filter		t_{SP}		—	—	0	50	ns
Capacitive load of each bus line		C_b	—	—	400	—	400	pF

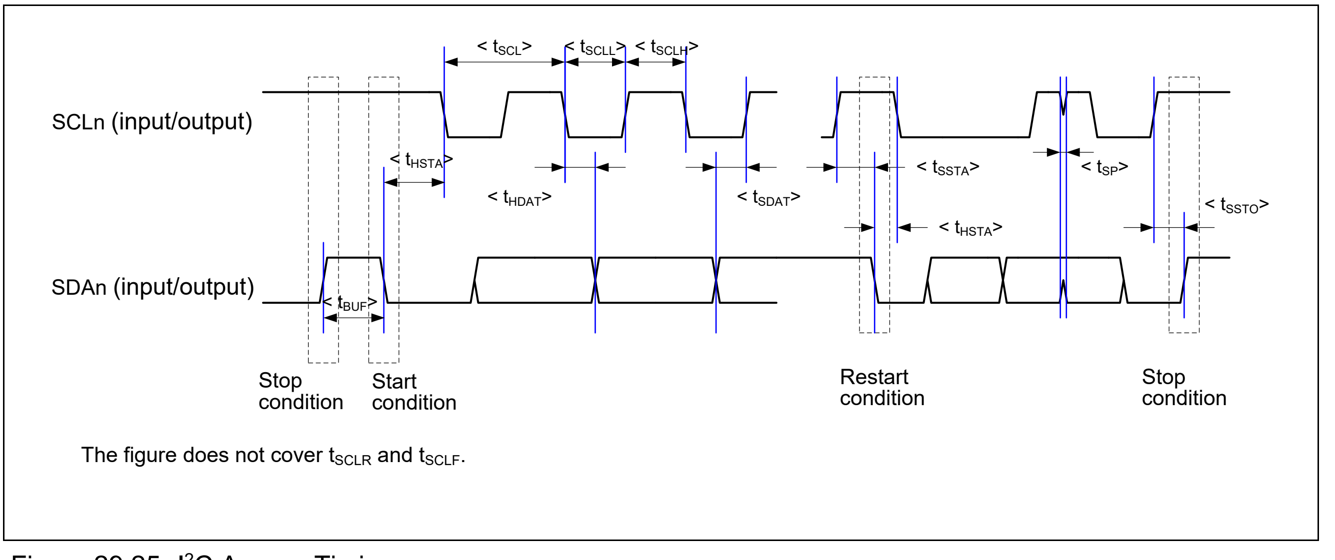


Figure 29.25 I²C Access Timing

Remark. n = 0, 1

29.8.9 CAN Interface

Item	Symbol	Conditions	MIN	MAX	Unit
Internal delay time	t_{NODE}	C_L = 30 pF	—	75	ns

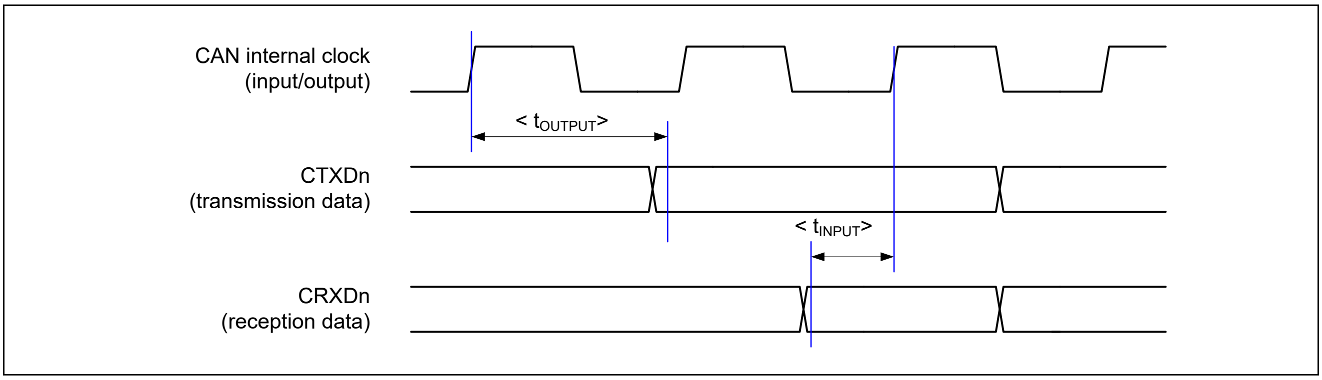


Figure 29.26 CAN Access Timing

Internal delay time (t_{NODE}) = Internal transmission delay time (t_{OUTPUT}) + Internal reception delay time (t_{INPUT})

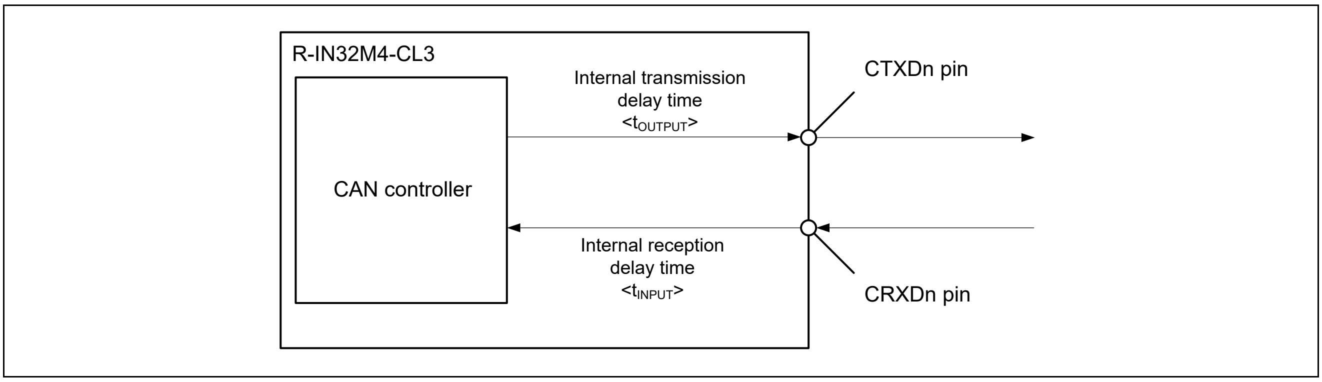


Figure 29.27 CAN Access Timing (Supplement)

Remarks 1. n = 0, 1

2. CAN internal clock (f_{CAN}): CAN baud-rate clock

29.8.10 Debugging Interface

(1) Debugging Serial Interface

Item	Symbol	Conditions	MIN	MAX	Unit
TCK input cycle	t_{TCK}	—	20	—	ns
TMS input setup time (for TCK ↑)	t_{STMS}	—	6.5	—	ns
TMS input hold time (for TCK ↑)	t_{HTMS}	—	0	—	ns
TDI input setup time (for TCK ↑)	t_{STDI}	—	6.5	—	ns
TDI input hold time (for TCK ↑)	t_{HTDI}	—	0	—	ns
TDO output delay time (for TCK ↓)	t_{DTDO}	C_L = 30 pF	3.0	13.0	ns

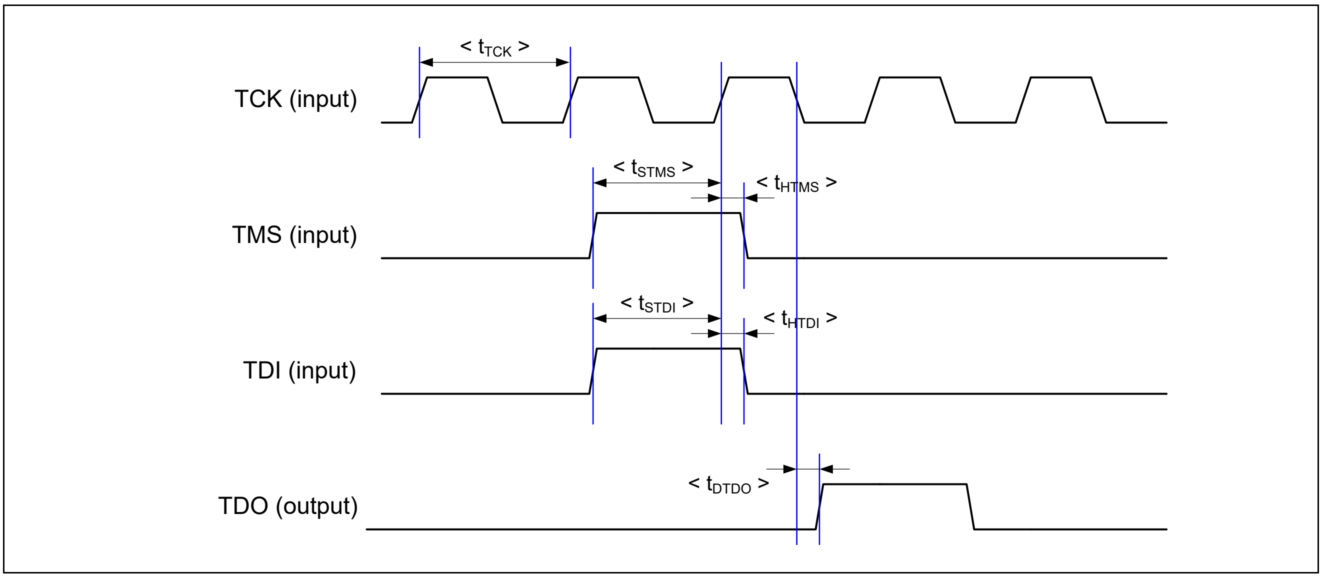


Figure 29.28 Debugging Serial Interface

(2) Trace Interface

Item	Symbol	Conditions	MIN	MAX	Unit
TRACECLK output cycle	t_{TRCCLK}	C_L = 15 pF	20	—	ns
TRACEDATAn output delay time (for TRACECLK)	$t_{DTRCDAT}$	C_L = 15 pF	0.26	8.43	ns

Remark. n = 0–3

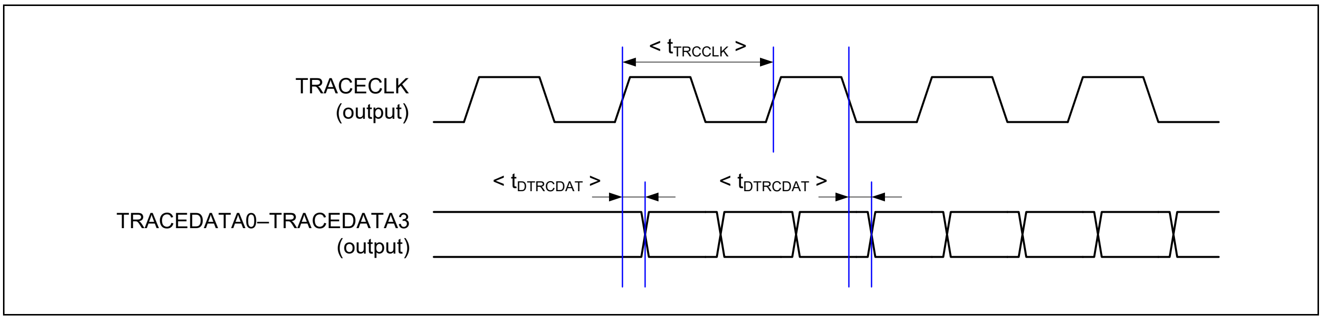


Figure 29.29 Trace Interface

29.9 2.5-V built-in Regulator Characteristics

Item	Symbol	Conditions	MIN	MAX	Unit
VDD25A rising time	$t_{VDD25AH}$	REG_EN = High	—	1*1	ms
VDD25A falling time	$t_{VDD25AL}$	REG_EN = High	—	—*2	ms

Note 1. This specification is based on the peripheral circuit configuration shown in Section 7 "2.5-V built-in Regulator Peripheral Circuit Configuration" in the "R-IN32M4-CL3 User's Manual: Board design edition".

2. There is no timing specification when the AVDDREG_33 and VDDREG_33 are falling since the power will be turned off.

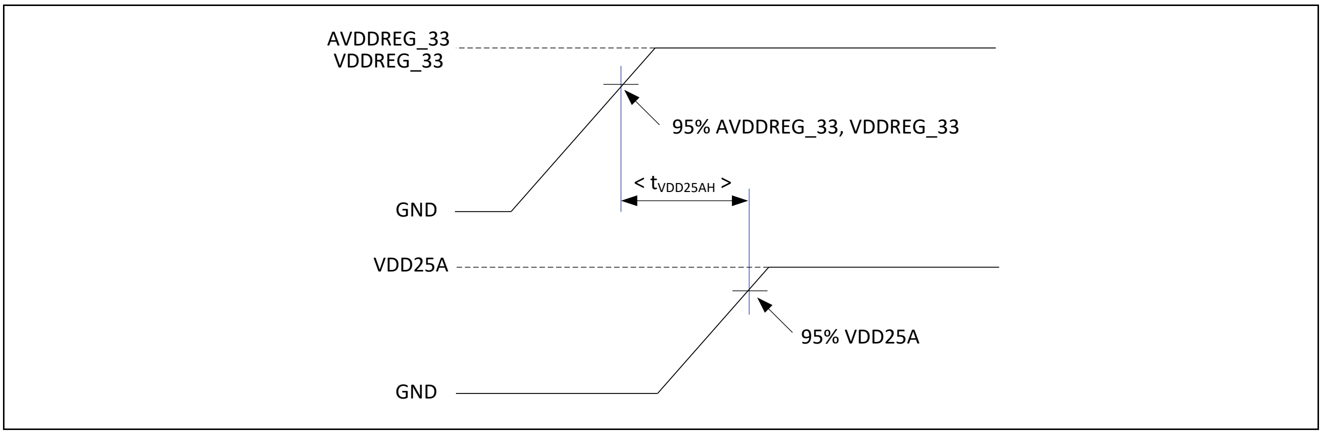
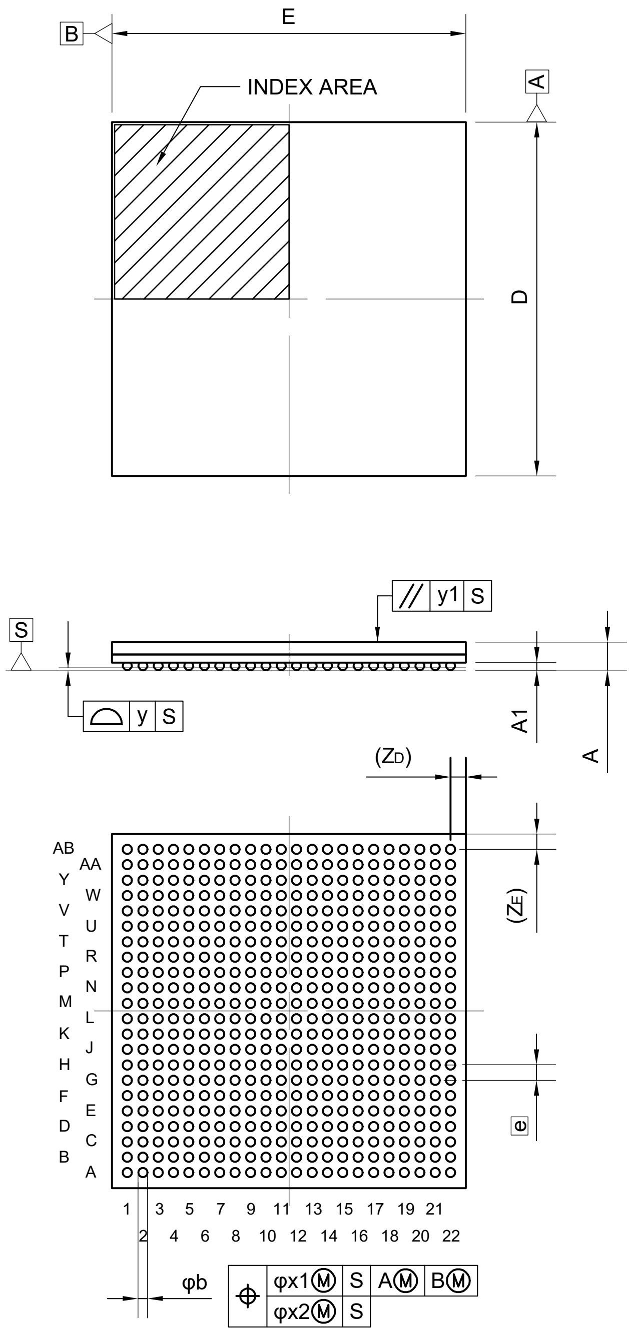


Figure 29.30 VDD25A rise timing

Appendix. Package Dimensions

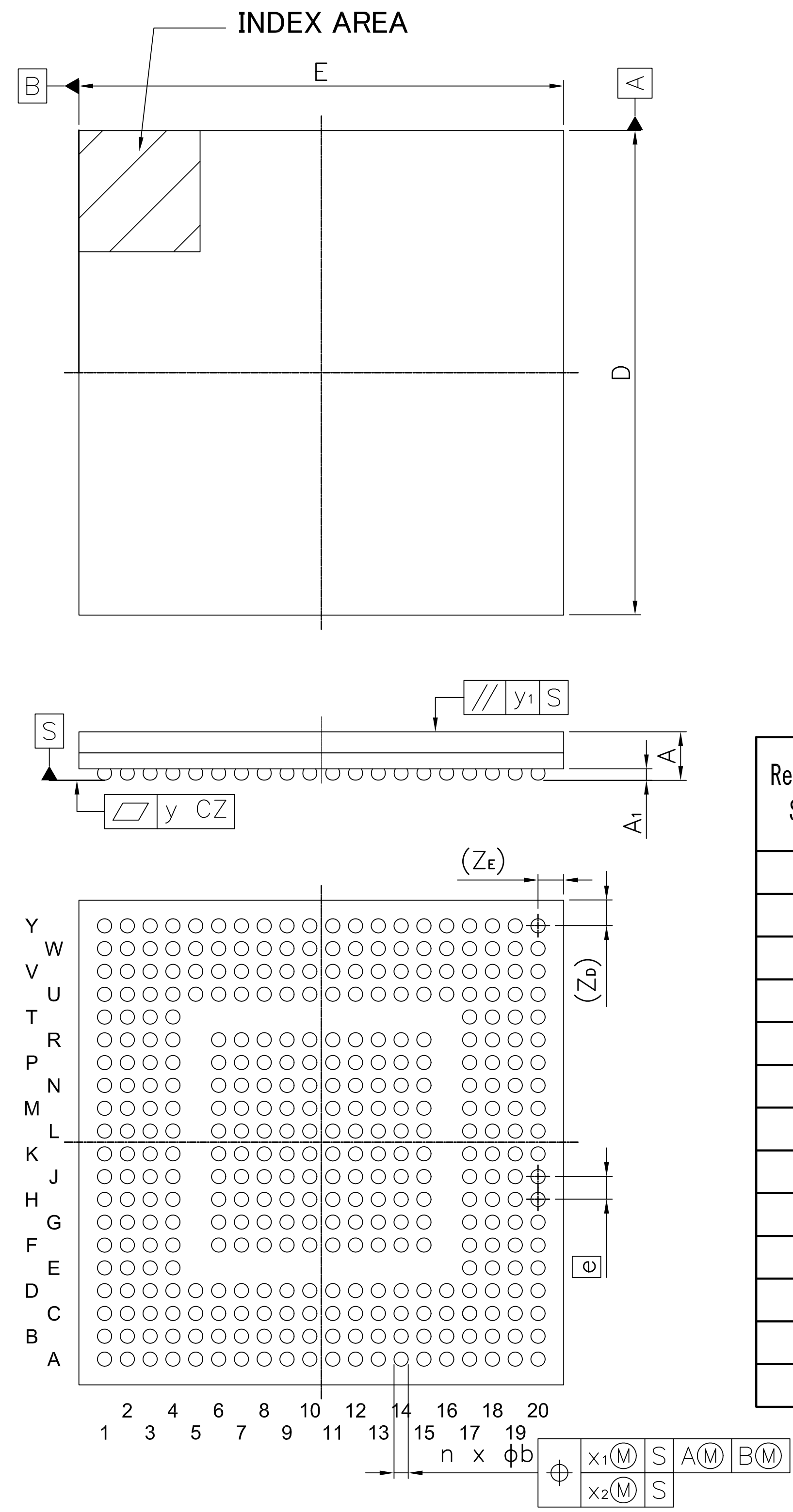
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-BGA484-23x23-1.00	PRBG0484FC-A	1.93



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	22.85	23.00	23.15
E	22.85	23.00	23.15
A	—	—	2.03
A1	0.40	0.50	0.60
e	—	1.00	—
b	0.50	0.60	0.70
x1	—	—	0.25
x2	—	—	0.10
y	—	—	0.15
y1	—	—	0.35
n	—	484	—
Z_D	—	1.00	—
Z_E	—	1.00	—

Figure A.1 23 mm Square 484-ball PBGA

JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-FBGA356-17×17-0.80	PRBG0356GA-A	1.03



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	16.90	17.00	17.10
E	16.90	17.00	17.10
A	—	—	2.00
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
X_1	—	—	0.15
X_2	—	—	0.08
y	—	—	0.10
y_1	—	—	0.20
n	—	356	—
Z_D	—	0.90	—
Z_E	—	0.90	—

Figure A.2 17 mm Square 356-ball FBGA

REVISION HISTORY	R-IN32M4-CL3 User's Manual: Hardware edition

Rev.	Date	Description	
		Page	Summary
1.00	Dec 24, 2019	—	First Edition issued
2.00	May 31, 2024	4-9	4.3.3 Types of Reset Name of the target reset signal in (1) Reset Control by Hardware corrected
		13-40	13.3.6.2 Interrupt Status/ACK Register (TSM_IRQ_STAT_ACK) Description for bit 1, IRQEVTOFF, corrected
		15-19	15.3.2 Address Output External address pin assignment corrected
		15-20	15.3.3 Address/Data Multiplexing Feature External SRAM pin assignment corrected
		15-29 to 15-45	15.4.1 Asynchronous Access Timing to 15.4.3 Wait Timing Address signals in the timing chart corrected
		16-44	16.3.6 Precautions Table 16.13 revised, and a remark added
		17-30	17.4.3 SPI Instruction Set for Use in Access to the Serial Flash ROM Write Enable, 06H, and Exit 4-byte mode, E9H, deleted from Table 17.2
		24-1 to 24-129	24. I^2C BUS (IICB) Interrupt signals, IICBTIAn and IICBTISn, renamed to INTIICBnITA and INTIICBnTIS

R-IN32M4-CL3 User's Manual: Hardware edition

Publication Date: Rev.1.00 Dec 24, 2019
Rev.2.00 May 31, 2024

Published by: Renesas Electronics Corporation

R-IN32M4-CL3 User's Manual:
Hardware edition

Renesas Electronics Corporation

R18UZ0073EJ0200