

R-IN32M3 Series

User's Manual: Peripheral Modules

R-IN32M3-EC
R-IN32M3-CL



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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

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

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

Trademarks (continued)

Examples of trademark or registered trademark used in the document of R-IN32M3;

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-CL2" (R9J03G019GBG) and design application systems using it. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

When designing an application system that includes this MCU, take all points to note into account. Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual.

Literature Literature may be preliminary versions. Note, however, that the following descriptions do not indicate "Preliminary". Some documents on cores were created when they were planned or still under development. So, they may be directed to specific customers. Last four digits of document number (described as ****) indicate version information of each document. Please download the latest document from our web site and refer to it.

Documents Related to R-IN32M3

Document Name	Document No.
R-IN32M3 Series Data Sheet	R18DS0008EJ****
R-IN32M3-EC User's Manual	R18UZ0003EJ****
R-IN32M3-CL User's Manual	R18UZ0005EJ****
R-IN32M3 Series Programming Manual: OS	R18UZ0011EJ****
R-IN32M3 Series Programming Manual: Driver	R18UZ0009EJ****
R-IN32M3 Series User's Manual: Peripheral Modules	This manual

2. Numbers and Symbols

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

Active low representation:

xxxZ (capital letter Z after pin or signal name)

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

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

Note:

Footnote for item marked with Note in the text

Caution:

Information requiring particular attention

Remark:

Supplementary information

Numeric representation:

Binary: xxxx, xxxxB or n'bxxxx (n bits)

Decimal: xxxx

Hexadecimal: xxxxH or n'hxxxx (n bits)

Prefix indicating the power 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

Half word: 16 bits

Byte: 8 bits

Contents

1. Introduction	1-1
1.1 Type Names of R-IN32M3-Series Products	1-1
1.2 Base Addresses of the System Registers Area	1-1
2. Clocks and Resets	2-1
2.1 Clock Configuration	2-1
2.1.1 Description of Internal Clocks	2-1
2.1.2 Clock Configuration Diagram	2-2
2.2 Stopping of Clock Supply	2-3
2.2.1 Overview	2-3
2.2.2 Clock Control Registers (CLKGTD0, CLKGTD1)	2-4
2.3 Reset	2-6
2.3.1 Overview	2-6
2.3.2 Features	2-6
2.3.3 Reset Control Registers	2-8
2.4 Operations for Reset	2-9
3. CPU and Internal RAMs	3-1
3.1 CPU-Core Information	3-1
3.2 CPU-Core Configuration Information	3-1
3.3 Restrictions	3-2
3.4 Internal Instruction RAM	3-3
3.4.1 Outline of Features	3-3
3.4.2 Read Buffer	3-3
3.4.3 Write Interface	3-3
3.5 Internal Data RAM	3-4
3.5.1 Outline of Features	3-4
3.6 Buffer RAM	3-5
3.6.1 Outline of Features	3-5
4. Bus Architecture	4-1
4.1 Bus Occupancy by the Cortex-M3	4-2
5. Booting Procedure	5-1
5.1 Selecting the Boot Mode	5-1

5.2	Initializing the Internal RAM	5-2
5.3	Memory Map in Each Boot Mode	5-2
5.4	Boot Sequence	5-3
5.4.1	When Booting from an External Memory	5-3
5.4.2	When Booting from the External Serial Flash ROM	5-4
5.4.3	When Downloading the Program from the External MCU and Booting the CPU	5-4
6.	Hardware Real-Time OS (HW-RTOS)	6-1
6.1	Outline of Features	6-1
6.2	Semaphores	6-1
6.3	Events	6-1
6.4	Mailboxes	6-2
6.5	Operation of HW-RTOS	6-2
7.	Gigabit Ethernet MAC	7-1
7.1	Overview	7-1
7.1.1	Ethernet Interface Architecture	7-2
7.2	Features	7-3
7.3	Control Registers	7-4
7.3.1	List of Registers	7-4
7.3.2	Ethernet Interface Select Register	7-6
7.3.3	Ethernet Interface Mode Registers	7-7
7.3.4	Gigabit Ethernet MAC Control Register	7-9
7.3.5	Hardware Function Call Register	7-26
7.4	Functions	7-31
7.4.1	Hardware Functions	7-31
7.4.2	Interrupts	7-58
7.4.3	Transmitting Ethernet Frames	7-61
7.4.4	Receiving Ethernet Frames	7-71
7.4.5	TCPIP Accelerator Function	7-80
7.5	Notes	7-82
7.5.1	Appending Padding to the MAC Header Section within the TX Frame	7-82
7.5.2	Erroneous Judgment about Checksum Validation at Specific Packet Reception	7-82
7.5.3	Error of Rx Frame Information at RX FIFO Overflow	7-83
7.5.4	Error of Rx Frame Information at Reception of the Frame more than 64 Bytes with Padding	7-86
7.5.5	Transmitting Data in Cut-Through Mode	7-87
7.5.6	Jumbo Frames	7-87

8.	Ethernet Switch	8-1
8.1	Overview	8-1
8.2	Characteristics.....	8-2
8.3	Control Registers	8-3
8.3.1	List of Registers.....	8-3
8.3.2	Operating Mode Setting Registers	8-6
8.3.3	Switch Configuration Registers	8-10
8.3.4	Learning Interface Registers.....	8-30
8.3.5	Mac Port Registers.....	8-32
8.3.6	Timer Module Registers	8-41
8.3.7	DLR Module Registers	8-51
8.4	Function Details.....	8-63
8.4.1	Switching Engine.....	8-63
8.4.2	Hub Module Supporting Cut-Through.....	8-73
8.4.3	DLR Module.....	8-79
8.4.4	IEEE 1588 Timer & Control Module	8-85
8.4.5	Management Port (Internal Port) Specific Frame Tagging.....	8-93
8.5	Overview of Control Software.....	8-96
8.5.1	Overview	8-96
8.5.2	Switch Initialization.....	8-96
8.5.3	Address Table Setting.....	8-101
9.	Asynchronous SRAM Memory Controller (ROM/SRAM)	9-1
9.1	Overview	9-1
9.2	Features.....	9-2
9.3	Bus Control.....	9-3
9.3.1	Overview of Registers	9-3
9.3.2	Bus Size Control Register (BSC).....	9-4
9.3.3	Static Memory Control Registers 0 to 3 (SMC0 to SMC3).....	9-5
9.3.4	Page ROM Control Register (PRC).....	9-8
9.3.5	Write Enable Switching Register (WREN)	9-11
9.4	Memory Connection Examples.....	9-12
9.4.1	SRAM Connection Example.....	9-12
9.4.2	Page ROM Connection Example	9-13
9.5	Procedure for Setting the Control Registers	9-14
9.6	External Wait Function.....	9-15
9.7	Memory Access Timing Examples.....	9-16

10. Synchronous Burst Access Memory Controller	10-1
10.1 Features.....	10-1
10.2 Control Registers	10-3
10.2.1 Wait Signals Selection Register (WAITZSEL)	10-4
10.2.2 Synchronous Burst Access Memory Controller Area Select Registers (SMADSEL0 to SMADSEL3)	10-6
10.2.3 Bus Clock Division Setting Register (BCLKSEL)	10-8
10.2.4 Synchronous Burst Access Memory Controller Operation Mode Setting Register (SMCMD).....	10-9
10.2.5 Synchronous Burst Access Memory Controller Direct Command Register (DIRECTCMD).....	10-10
10.2.6 Synchronous Burst Access Memory Controller Cycle Setting Register (SETCYCLES).....	10-11
10.2.7 Synchronous Burst Access Memory Controller Mode Setting Register (SETOPMODE)	10-13
10.2.8 Synchronous Burst Access Memory Controller Refresh Setting Register (REFRESH0).....	10-15
10.2.9 Synchronous Burst Access Memory Controller CSZn Cycle Setting Registers (SRAM_CYCLES0_n)	10-15
10.2.10 Synchronous Burst Access Memory Controller CSZn Mode Registers (OPMODE0_n).....	10-16
10.2.11 Register Setup Procedure.....	10-17
10.3 Function Details.....	10-18
10.3.1 Bus Clock Control	10-18
10.3.2 Address Output	10-19
10.3.3 Address/Data Multiplexing Feature	10-19
10.3.4 Write Enable Signal (WRZn) Cycle Extension	10-20
10.3.5 Controlling the Data Read Timing.....	10-21
10.3.6 Wait Signals Control.....	10-22
10.3.7 Specify the Operating Mode of the Synchronous Burst Access Memory Controller	10-25
10.3.8 Switching External Memory Area Mapping.....	10-26
10.4 Memory Access Timing Example	10-27
10.4.1 Asynchronous Access Timing	10-28
10.4.2 Synchronous Access Timing.....	10-36
10.4.3 Wait Timing.....	10-43
11. External MCU Interface	11-1
11.1 Memory Map	11-3
11.2 Synchronous/Asynchronous SRAM Interface Mode.....	11-5
11.2.1 Functional Overview.....	11-5
11.2.2 Operation	11-6
11.2.3 Basic Operation Timing of the External MCU Interface	11-9
11.2.4 Timing Adjustment of the External MCU Interface	11-12
11.2.5 Control Registers	11-16
11.2.6 Precautions.....	11-26

11.3	Synchronous SRAM Type Transfer Mode	11-27
11.3.1	Functional Overview.....	11-27
11.3.2	Selection of Operational Mode	11-27
11.3.3	Write Status Mode and Write Strobe Mode.....	11-28
11.3.4	Control Registers	11-28
11.3.5	Basic Operation Timing in Synchronous SRAM Type Transfer Mode	11-31
11.3.6	Precautions.....	11-34
12.	Serial Flash ROM Memory Controller	12-1
12.1	Features.....	12-1
12.2	Control Registers	12-2
12.2.1	Transfer Mode Control Register (SFMSMD).....	12-3
12.2.2	Chip Selection Control Register (SFMSSC).....	12-5
12.2.3	Clock Control Register (SFMSKC)	12-6
12.2.4	Status Register (SFMSST).....	12-8
12.2.5	Communications Port Register (SFMCOM).....	12-9
12.2.6	Communications Mode Control Register (SFMCMD).....	12-10
12.2.7	Communications Status Register (SFMCST)	12-11
12.3	Connection with Serial Flash ROM.....	12-12
12.4	Operation	12-13
12.4.1	SPI Bus Operation	12-13
12.4.2	SPI Bus Timing Adjustment	12-14
12.4.3	SPI Instruction Set for Use in Access to the Serial Flash ROM	12-18
12.4.4	Modifying the SPI Bus Cycle	12-21
12.4.5	Automatic Release from the Deep Power-Down State	12-24
12.4.6	Direct Communications	12-25
12.5	Example of Configuration	12-26
12.5.1	Standard Reading.....	12-27
12.5.2	Fast Read Dual I/O	12-31
13.	DMA Controllers	13-1
13.1	Features.....	13-2
13.1.1	Overview	13-2
13.2	Relation between DMA Units/Channels and DMA Triggers	13-5
13.3	Terms and Definition	13-6
13.4	DMA Controller Registers.....	13-7
13.4.1	Register Configuration.....	13-7
13.4.2	Control Register Outline.....	13-9
13.4.3	General DMA Controller Register Set.....	13-13

13.4.4	Register Set of DMA Controller for Real-Time Ports	13-47
13.4.5	DMA Transfer Interface Signal Control Registers (DMAIFC0, DMAIFC1, RTDMAIFC)	13-78
13.4.6	DMA Trigger Source Selection Registers (DTFRn, RTDTFR)	13-84
13.5	DMA Interface Pins	13-88
13.5.1	BUSCLK Synchronization	13-88
13.5.2	Transfer Request and Acknowledge	13-88
13.6	Interrupt Output	13-89
13.7	DMAC Operation Setting	13-90
13.7.1	Register Mode and Link Mode Selection	13-90
13.7.2	Register Mode.....	13-91
13.7.3	Link Mode	13-98
13.7.4	Write-Only Mode.....	13-110
13.8	DMAC Operation	13-111
13.8.1	Transfer Mode	13-111
13.8.2	DMA Unit Priority Control.....	13-114
13.8.3	DMA Transfer Request.....	13-117
13.8.4	DMA Acknowledge Output.....	13-120
13.8.5	DMA Transfer Completion Interrupt.....	13-125
13.8.6	DMA Terminal Count Output.....	13-127
13.8.7	Forced Dumping	13-129
13.8.8	DMA Error Interrupt.....	13-129
13.8.9	Interval Counting	13-129
13.8.10	Differences in Operation by Transfer Size	13-130
13.8.11	Transfer Status	13-133
13.8.12	Suspension	13-133
13.8.13	Suspending Transfer	13-134
13.9	DMA Transfer Setting Examples.....	13-135
13.9.1	Setting Example 1 (Register Mode, Single Transfer Mode, and Hardware Trigger).....	13-135
13.9.2	Setting Example 2 (Register Mode, Block Transfer Mode, and Software Trigger).....	13-138
13.9.3	Setting Example 3 (Register Mode: Continuous Execution, Block Transfer Mode, and Software Trigger)	13-141
13.9.4	Setting Example 4 (Link Mode, Block Transfer Mode, and Software Trigger)	13-144
13.10	Notes.....	13-147
14.	Timer Array Unit (TAUJ2).....	14-1
14.1	Features of TAUJ2.....	14-1
14.1.1	Functional List of Timer Operations.....	14-3
14.2	Functional Overview.....	14-4
14.2.1	Terms.....	14-4

14.2.2	Description of Blocks	14-6
14.3	Registers	14-7
14.3.1	TAUJ2 Registers Overview	14-7
14.3.2	TAUJ2 Prescaler Registers Details	14-9
14.3.3	TAUJ2 Control Registers Details	14-14
14.3.4	TAUJ2 Simultaneous Reload Registers Details.....	14-24
14.3.5	TAUJ2 Output Registers Details	14-26
14.4	General Operating Procedure.....	14-30
14.5	Overview of Synchronous Channel Operation	14-31
14.5.1	Basic Rules of Synchronous Channel Operation	14-31
14.6	Simultaneous Reloading	14-33
14.6.1	Outline of Operation	14-33
14.6.2	How to Control Simultaneous Reloading (in Case of PWM Output)	14-34
14.6.3	Other General Rules of Simultaneous Reloading	14-35
14.7	Independent Channel Operation	14-36
14.7.1	Interval Timer	14-37
14.7.2	TAUJ2TTINm Input Interval Timer	14-46
14.7.3	Delay Counting	14-52
14.7.4	TAUJ2TTINm Input Pulse Interval Measurement	14-57
14.7.5	TAUJ2TTINm Input Signal Width Measurement	14-64
14.7.6	External Event Counting.....	14-76
14.7.7	TAUJ2TTINm Input Position Detection.....	14-82
14.8	Synchronous Channel Operation	14-93
14.8.1	PWM Output.....	14-93
15.	Window Watchdog Timer A (WDTA).....	15-1
15.1	Features of WDTA	15-1
15.2	Functional Overview.....	15-2
15.3	Registers	15-3
15.3.1	Overview of WDTA Registers.....	15-3
15.3.2	Details of WDTA Registers	15-3
15.4	Functional Description.....	15-6
15.4.1	WDTA after Release from the Reset State	15-6
15.4.2	WDTA Trigger	15-8
15.4.3	Error Detection	15-8
15.4.4	Window Function	15-10
15.5	WDTOUTZ Output	15-11
15.6	Notes.....	15-11

16. Asynchronous Serial Interface J (UARTJ).....	16-1
16.1 Features of UARTJn.....	16-1
16.2 Functional Overview.....	16-3
16.3 Configuration.....	16-4
16.4 UARTJn Registers.....	16-5
16.5 Interrupt Request Signals.....	16-27
16.5.1 Transmission Interrupt Request INTUAJnTIT.....	16-27
16.5.2 Reception Interrupt Request INTUAJnTIR.....	16-29
16.5.3 Status Interrupt Request INTUAJnTIS.....	16-30
16.6 Operation.....	16-32
16.6.1 Data Formats.....	16-32
16.6.2 BF Transmission/Reception Format.....	16-34
16.6.3 BF Transmission.....	16-36
16.6.4 BF Reception.....	16-38
16.6.5 UARTJn Transmission.....	16-40
16.6.6 UARTJn Reception.....	16-43
16.6.7 Reception Errors.....	16-49
16.6.8 Parity Types and Operations.....	16-50
16.6.9 Digital Receive Data Noise Filter.....	16-51
16.7 Bit-Rate Generator.....	16-52
17. Clocked Serial Interface H (CSIH).....	17-1
17.1 Features of CSIH.....	17-1
17.2 Functional Overview.....	17-4
17.3 CSIH Control Registers.....	17-6
17.3.1 CSIH Register Details.....	17-7
17.4 Functional Description.....	17-35
17.4.1 Operating Modes (Master/Slave).....	17-36
17.4.2 Master/Slave Connections.....	17-38
17.4.3 Chip Selection (CS) Features.....	17-40
17.4.4 Chip Select Timing Details.....	17-43
17.4.5 Job Concept.....	17-46
17.4.6 Serial Clock Selection.....	17-47
17.4.7 CSIH Buffer Memory.....	17-49
17.4.8 Data Transfer Modes.....	17-51
17.4.9 Data Length Selection.....	17-52
17.4.10 Serial Data Direction Selection.....	17-55
17.4.11 Communication in Slave Mode.....	17-56

17.4.12	CSIH Interrupt Requests	17-57
17.4.13	Error Detection	17-66
17.4.14	Loop-Back Mode	17-75
17.5	Operating Procedures.....	17-76
17.5.1	Procedures in Direct Access Mode	17-76
17.5.2	Procedures in Transmit-Only Buffer Mode	17-88
17.5.3	Procedures in Dual Buffer Mode	17-100
17.5.4	Procedures in FIFO Mode.....	17-112
18.	I ² C BUS (IICB).....	18-1
18.1	Features of IICB	18-1
18.2	Functional Overview.....	18-2
18.3	Registers	18-3
18.4	IIC Bus Mode Functions.....	18-30
18.4.1	Pin Configuration.....	18-30
18.5	IIC Bus Definition	18-31
18.5.1	Start Condition.....	18-31
18.5.2	Addresses.....	18-32
18.5.3	Extension Code	18-32
18.5.4	Transfer Direction Specification.....	18-33
18.5.5	Acknowledge (ACK).....	18-33
18.5.6	Data.....	18-34
18.5.7	Stop Condition	18-34
18.5.8	Wait State	18-35
18.5.9	Arbitration	18-37
18.6	Operation	18-38
18.6.1	Single Transfer Mode	18-38
18.6.2	Continuous Transfer Mode	18-43
18.6.3	Arbitration	18-48
18.6.4	Entering and Exiting Wait State	18-49
18.6.5	Extension Code	18-54
18.7	Interrupt Request Signals.....	18-55
18.7.1	Single Transfer Mode	18-55
18.7.2	Continuous Transfer Mode	18-58
18.8	Interrupt Outputs and States	18-63
18.8.1	Single Transfer Mode (Master Device Operation).....	18-64
18.8.2	Single Transfer Mode (Slave Device Operation: during Slave Address Reception (IICBnSTR0.IICBnSSC0 bit = 1)).....	18-67

18.8.3	Single Transfer Mode (Slave Device Operation: during Extension Code Reception (IICBnSTR0.IICBnSSEX bit = 1)).....	18-71
18.8.4	Single Transfer Mode (Non-Participation in Communications).....	18-75
18.8.5	Single Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1): Operation as Slave after Arbitration Loss).....	18-76
18.8.6	Single Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1): Non-Participation in Communications after Arbitration Loss).....	18-78
18.8.7	Single Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1): Non-Participation in Communications after Arbitration Loss (during Extension Code Transfer)).....	18-84
18.8.8	Continuous Transfer Mode (Master Device Operation (Reception)).....	18-85
18.8.9	Continuous Transfer Mode (Master Device Operation (Transmission)).....	18-88
18.8.10	Continuous Transfer Mode (Slave Device Operation (Reception): during Slave Address Reception (IICBnSTR0.IICBnSSC0 bit = 1)).....	18-91
18.8.11	Continuous Transfer Mode (Slave Device Operation (Reception): during Extension Code Reception (IICBnSTR0.IICBnSSEX bit = 1)).....	18-95
18.8.12	Continuous Transfer Mode (Slave Device Operation (Transmission): during Slave Address Reception (IICBnSTR0.IICBnSSC0 bit = 1)).....	18-99
18.8.13	Continuous Transfer Mode (Slave Device Operation (Transmission): during Extension Code Reception (IICBnSTR0.IICBnSSEX bit = 1)).....	18-103
18.8.14	Continuous Transfer Mode (Non-Participation in Communications).....	18-107
18.8.15	Continuous Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1) (when address was transferred during reception): Operation as Slave after Arbitration Loss).....	18-108
18.8.16	Continuous Transfer Mode (Arbitration Loss Operation (IICBnSTR0.IICBnALDF bit = 1) (when address was transferred during reception): Non-Participation in Communications after Arbitration Loss).....	18-110
18.8.17	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)).....	18-115
18.9	Setting Procedure.....	18-117
18.9.1	Single Master Environment.....	18-117
18.9.2	Multi-Master Environment.....	18-121
19.	CAN Controller (FCN).....	19-1
19.1	Features of FCN.....	19-1
19.2	Features.....	19-4
19.2.1	Overview of Functions.....	19-5
19.2.2	Configuration.....	19-6
19.3	Internal Registers of FCN.....	19-7
19.3.1	CAN Controller Configuration.....	19-7
19.3.2	CAN Controller Registers Overview.....	19-9

19.3.3	Bit Configuration of Registers	19-13
19.4	Setting or Clearing of Bits	19-19
19.5	Control Registers	19-21
19.5.1	FCN Global Registers	19-21
19.5.2	FCN Module Registers	19-30
19.5.3	FCN Message Buffer Registers	19-55
19.6	Initialization of CAN Controller	19-67
19.6.1	Initialization of FCN Module	19-67
19.6.2	Initialization of Message Buffer	19-67
19.6.3	Redefinition of Message Buffer	19-67
19.6.4	Transition from Initialization Mode to Operation Mode	19-69
19.7	Message Reception	19-71
19.7.1	Message Reception	19-71
19.7.2	Receive Data Read	19-72
19.7.3	Receive History List Function	19-73
19.7.4	Mask Function	19-75
19.7.5	Multi-Buffer Reception Blocking	19-76
19.7.6	Remote Frame Reception	19-77
19.8	Message Transmission	19-79
19.8.1	Transmission of Messages	19-79
19.8.2	Transmit History List Function	19-81
19.8.3	Automatic Block Transmission (ABT)	19-83
19.8.4	Aborting Transmission	19-85
19.8.5	Remote Frame Transmission	19-86
19.9	Power Saving Modes	19-87
19.9.1	FCN Sleep Mode	19-87
19.9.2	FCN Stop Mode	19-90
19.9.3	Example of Using Power Saving Mode	19-91
19.10	Interrupts	19-92
19.11	Diagnosis and Special Operation Modes	19-93
19.11.1	Receive-Only Mode	19-93
19.11.2	Single-Shot Mode	19-94
19.11.3	Self-Test Mode	19-95
19.11.4	Receive/Transmit Operation in Each Operation Mode	19-96
19.12	Timestamping	19-97
19.12.1	Timestamping	19-97
19.13	Baud Rate Settings	19-99
19.13.1	Baud Rate Setting Conditions	19-99
19.13.2	Representative Examples of Baud Rate Settings	19-103

19.14	Operation of the CAN Controller	19-105
19.14.1	Initialization.....	19-105
19.14.2	Message Transmission.....	19-111
19.14.3	Message Reception	19-125
19.14.4	Power Safe Mode.....	19-131
20.	CC-Link Interface.....	20-1
20.1	Registers	20-1
20.1.1	List of Registers.....	20-1
20.1.2	CC-Link Bus Size Control Register (CCBSC)	20-2
20.1.3	CC-Link Bus Bridge Control Register 0 (CCSMC0)	20-2
20.1.4	CC-Link Bus Bridge Control Register 1 (CCSMC1)	20-3
20.1.5	CC-Link Monitor Register (CCSMON)	20-3
20.1.6	CC-Link Slave RUN LED Control Register (CCSRUN)	20-4
20.1.7	CC-Link Reset Register (CCRES).....	20-5
20.1.8	CC-Link Slave Operating Mode Setting Register (CCSMD).....	20-6
21.	System Registers (APB Peripheral Registers Area)	21-1
21.1	List of Registers.....	21-1
21.2	Operating Mode Monitor Register (MDMNT).....	21-3
21.3	IDCODE Register (IDCODE)	21-3
21.4	Version Register (RINVER).....	21-4
21.5	Watchdog Timer Input Clock Selection Register (WDTCLKCFG).....	21-5
21.6	CPURESET Register (CPURESET).....	21-6
21.7	System Protect Command Register (SYSPCMD)	21-7
21.8	HW-RTOS Reset Register (RTOS_SOFTRST)	21-8
21.9	Timer Input Selection	21-9
21.9.1	Timer Input Function Selection Register (SELCNT)	21-10
21.9.2	Timer Trigger Source Registers (TMTFR0 to TMTFR03).....	21-12
21.10	Noise Eliminator.....	21-16
21.10.1	Noise Filter Setting Registers 0 to 3 (NFC0 to NFC3).....	21-16
21.10.2	Noise Filtering Operation	21-20
21.11	External Interrupt Mode Registers 0, 1, 2 (INTM0, INTM1, INTM2).....	21-21
21.12	Trigger-Synchronous Ports	21-24
21.12.1	Trigger-Synchronous Port Control Mode Register (RPTRGMD).....	21-25
21.12.2	Trigger-Synchronous Port Source Registers (RP0TFR to RP3TFR).....	21-26
21.13	Scratch Registers (SCRATCH0 to SCRATCHD)	21-30
21.14	CPU Bus Operating Mode Register (CPUBUSMD)	21-31
21.15	SRAM Bridge Select Register (SRAMBRSEL).....	21-32

22. Debugging	22-1
22.1 JTAG Interface	22-1
22.2 SWD Interface	22-4
22.3 Trace Port Interface	22-4
22.4 SWV Interface	22-5

List of Figures

Figure 2.1	Clock Configuration Diagram.....	2-2
Figure 2.2	Timing of Reset at Power On.....	2-9
Figure 2.3	Timing of Reset at System Reset.....	2-9
Figure 6.1	Overall Structure of Hardware Real-Time OS (HW-RTOS).....	6-3
Figure 7.1	Ethernet Interface Peripheral Architecture (R-IN32M3-EC).....	7-2
Figure 7.2	Ethernet Interface Peripheral Architecture (R-IN32M3-CL).....	7-2
Figure 7.3	Schematic Block Diagram of the Hardware Functions.....	7-31
Figure 7.4	Flow of Processing for Issuing the Hardware Function.....	7-33
Figure 7.5	Method of Controlling a Buffer.....	7-34
Figure 7.6	Buffer Structure.....	7-35
Figure 7.7	Address Structure of Buffers.....	7-36
Figure 7.8	Block Diagram of the MACDMAC in Context and Interrupt Signals.....	7-41
Figure 7.9	Outline of Processing by the Reception MACDMAC.....	7-42
Figure 7.10	Conceptual Diagram of Judging Whether a Received Frame is Valid or Invalid.....	7-44
Figure 7.11	Transmission Descriptor.....	7-49
Figure 7.12	Example of Transmission as One Frame by Combining Multiple Buffers.....	7-50
Figure 7.13	TX Data Format.....	7-63
Figure 7.14	Tx Frame Control Information Format.....	7-64
Figure 7.15	Tx Ethernet Frame Data Format – TCPIPACC is enabled, without VLAN tag.....	7-67
Figure 7.16	Tx Ethernet Frame Data Format – TCPIPACC is enabled, with VLAN tag.....	7-67
Figure 7.17	Tx Ethernet Frame Data Format – TCPIPACC is disabled, without VLAN tag.....	7-68
Figure 7.18	Tx Ethernet Frame Data Format – TCPIPACC is disabled, with VLAN tag.....	7-68
Figure 7.19	Structure of the TX Descriptor.....	7-69
Figure 7.20	Rx Data Format.....	7-72
Figure 7.21	Rx frame Information.....	7-73
Figure 7.22	Destination MAC Address Field (when insertion of management tag is permitted).....	7-76
Figure 7.23	Format of Receive Ethernet Frame – TCPIPACC is enabled, without VLAN Tag, no TCP/UDP packets.....	7-77
Figure 7.24	Format of Receive Ethernet Frame – TCPIPACC is enabled, with VLAN Tag, no TCP/UDP packets.....	7-77
Figure 7.25	Format of Receive Ethernet Frame – TCPIPACC is enabled, without VLAN Tag, with TCP/UDP packets.....	7-78
Figure 7.26	Format of Receive Ethernet Frame – TCPIPACC is enabled, with VLAN Tag, with TCP/UDP packets.....	7-78
Figure 7.27	Format of Receive Ethernet Frame – TCPIPACC is disabled, without VLAN Tag.....	7-79
Figure 7.28	Format of Receive Ethernet Frame – TCPIPACC is disabled, with VLAN Tag.....	7-79
Figure 7.29	Flowchart of RX FIFO Overflow Processing Task (In case the hardware real-time OS is used).....	7-84
Figure 7.30	Flowchart of Reception Processing Task (In case the hardware real-time OS is used).....	7-85
Figure 7.31	Flowchart of RX FIFO Overflow INT Processing (In case the hardware real-time OS is not used).....	7-85
Figure 7.32	Flowchart of Reception Processing (In case the hardware real-time OS is not used).....	7-86

Figure 7.33	Flowchart of Reception Processing.....	7-87
Figure 8.1	Overview of the Ethernet Switch	8-1
Figure 8.2	Switching Engine Overview	8-63
Figure 8.3	VLAN Priority Table Overview	8-64
Figure 8.4	IP COS Tables Overview	8-65
Figure 8.5	Port Look-Up Overview	8-66
Figure 8.6	Record Types of Address Memory	8-67
Figure 8.7	Learning Interface Overview	8-68
Figure 8.8	Record Formats.....	8-68
Figure 8.9	Overview of Processing for Frame Transfer.....	8-69
Figure 8.10	Overview of Output Port Memory Controller.....	8-72
Figure 8.11	Normal Switch Mode Operation.....	8-73
Figure 8.12	Operation of the Hub when Transfer from Port 0 to Port 1 is Enabled.....	8-74
Figure 8.13	Connection between the Hub Module and the DLR Module.....	8-79
Figure 8.14	Beacon Frame Format.....	8-80
Figure 8.15	Configuration of Adjustable Timer.....	8-89
Figure 8.16	Drift Correction.....	8-90
Figure 8.17	Offset Correction (when ATIME_OFFS_CORR is not zero).....	8-90
Figure 8.18	Format of Frame with Management Tag in Internal Port	8-93
Figure 8.19	Entries of Address Table and Definition of Hash Block.....	8-101
Figure 8.20	Address Learning Flow.....	8-102
Figure 9.1	Example of Control Using the MA6 to MA3 Bits of the PRC Register	9-10
Figure 9.2	Example of Connection with 32-Bit SRAM	9-12
Figure 9.3	Example of Connection with 16-Bit SRAM.....	9-12
Figure 9.4	Example of Connection with 32-Bit Page ROM.....	9-13
Figure 9.5	Example of Connection with 16-Bit Page ROM.....	9-13
Figure 9.6	Example Procedure for Setting the Control Registers of the Memory Controller	9-14
Figure 9.7	Configuration of the WAITZ Signal Sampling Circuit.....	9-15
Figure 9.8	SRAM Read Cycles	9-17
Figure 9.9	SRAM Read Cycles (with Wait Settings).....	9-18
Figure 9.10	SRAM Read Cycles (External Wait Insertion)	9-19
Figure 9.11	SRAM Write Cycles (with No Wait).....	9-20
Figure 9.12	SRAM Write Cycles (with Wait States)	9-21
Figure 9.13	SRAM Write Cycles (External Wait Insertion)	9-22
Figure 9.14	Page ROM Read Cycles (Single Transfer)	9-23
Figure 9.15	Page ROM Read Cycles (Four Burst Transfer)	9-24
Figure 10.1	Register Setup Procedure	10-17
Figure 10.2	Clock Output Timing Example (SMCMD.SMCCLKTH = 0).....	10-18
Figure 10.3	Clock Output Timing Example (SMCMD.SMCCLKTH = 1).....	10-18
Figure 10.4	Write Enable Signal Operation	10-20
Figure 10.5	Read Data Timing Control.....	10-21
Figure 10.6	External Memory Space.....	10-26
Figure 10.7	Asynchronous SRAM, Separate Bus Mode, Read Access, ADVZ Enabled.....	10-28

Figure 10.8	Asynchronous SRAM, Separate Bus Mode, Read Access, ADVZ Disabled.....	10-29
Figure 10.9	Asynchronous Page ROM, Separate Bus Mode, Read Access, ADVZ Enabled	10-30
Figure 10.10	Asynchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled.....	10-31
Figure 10.11	Asynchronous SRAM, Separate Bus Mode, Write Access, ADVZ Disabled.....	10-32
Figure 10.12	Asynchronous SRAM, Separate Bus Mode, Write Access, ADVZ Enabled.....	10-33
Figure 10.13	Asynchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled, WE_TIME = 0 ..	10-34
Figure 10.14	Asynchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled, WE_TIME = 1 ..	10-35
Figure 10.15	Synchronous SRAM, Separate Bus Mode, Read Access, ADVZ Enabled.....	10-36
Figure 10.16	Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled	10-37
Figure 10.17	Synchronous SRAM, Multiplexed Bus Mode, Burst Read Access (4-Beat), ADVZ Enabled	10-38
Figure 10.18	Synchronous SRAM, Separate Bus Mode, Write Access, ADVZ Enabled.....	10-39
Figure 10.19	Synchronous SRAM, Separate Bus Mode, Burst Write Access (8-Beat), ADVZ Enabled	10-40
Figure 10.20	Synchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled	10-41
Figure 10.21	Synchronous SRAM, Multiplexed Bus Mode, Burst Write Access (4-Beat), ADVZ Enabled	10-42
Figure 10.22	Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled	10-43
Figure 10.23	Synchronous SRAM, Separate Bus Mode, Burst Write Access (4-Beat), ADVZ Enabled	10-44
Figure 11.1	External MCU Interface Space	11-3
Figure 11.2	Writing to the CC-Link IE Field Network Area (SRAM writing)	11-9
Figure 11.3	Reading from CC-Link IE Field Network Area (SRAM reading)	11-9
Figure 11.4	Writing to the External MCU Interface Register Area (SRAM writing)	11-10
Figure 11.5	Reading from External MCU Interface Register Area (SRAM reading)	11-10
Figure 11.6	Writing to Other Areas (SRAM writing)	11-11
Figure 11.7	Reading from Other Areas (SRAM reading)	11-11
Figure 11.8	Timing Adjustment (SRAM writing).....	11-13
Figure 11.9	Timing Adjustment (SRAM reading, page ROM reading).....	11-14
Figure 11.10	Timing Adjustment (page ROM reading).....	11-15
Figure 12.1	Connection with Serial Flash ROM.....	12-12
Figure 12.2	Basic Operation of SPI Bus	12-13
Figure 12.3	Correction of the SMSCK Signal Duty Factor by Using the SFMDTY Bit (Example of HCLK/3)	12-14
Figure 12.4	SMCSZ Signal Setup Time Adjustment by Using the SFMSLD Bit.....	12-15
Figure 12.5	SMCSZ Signal Hold Time Adjustment by Using the SFMSHD Bit	12-15
Figure 12.6	Output Enable Time Adjustment by Using the SFMOEX Bit	12-16
Figure 12.7	Serial Data Setup Time Adjustment by Using the SFMOSW Bit.....	12-16
Figure 12.8	Serial Data Hold Time Adjustment by Using the SFMOHW Bit	12-17
Figure 12.9	Bus Cycles for Standard Reading	12-18
Figure 12.10	Bus Cycles for Fast Read.....	12-19
Figure 12.11	Bus Cycles for Fast Read Dual Output	12-19
Figure 12.12	Bus Cycles for Fast Read Dual I/O.....	12-20
Figure 12.13	Bus Cycles for Release from Deep Power-Down	12-20
Figure 12.14	Continuous Data Reading by Individual Conversion.....	12-21
Figure 12.15	Continuous Data Reading by Using Prefetching	12-22
Figure 12.16	Continuous Data Reading by Using SPI Bus Cycle Extension.....	12-23

Figure 12.17	Operation for Automatic Release from the Deep Power-Down State.....	12-24
Figure 13.1	Relation between DMA Units/Channels and DMA Triggers	13-5
Figure 13.2	Name of Transfers	13-6
Figure 13.3	Register Block Diagram of DMA	13-8
Figure 13.4	Relationship between the SSKPn and SCNTn Registers in Skip Mode	13-38
Figure 13.5	Relationship between the DSKPn and DCNTn Registers in Skip Mode	13-40
Figure 13.6	Relationship between the RTSSKP and RTSCNT Registers in Skip Mode	13-70
Figure 13.7	Relationship between the RTDSKP and RTDCNT Registers in Skip Mode	13-72
Figure 13.8	DMA Pin Signals and Internal Signals (1) (DMAIFCp = 8000 0000H).....	13-81
Figure 13.9	DMA Pin Signals and Internal Signals (2) (DMAIFCp = 8000 0000H).....	13-81
Figure 13.10	DMA Pin Signals and Internal Signals (3) (DMAIFCp = 8000 0200H).....	13-82
Figure 13.11	DMA Pin Signals and Internal Signals (4) (DMAIFCp = 8000 0002H).....	13-82
Figure 13.12	DMA Pin Signals and Internal Signals (5) (DMAIFCp = 8000 0002H).....	13-83
Figure 13.13	DMA Pin Signals and Internal Signals (6) (DMAIFCp = 8000 0202H).....	13-83
Figure 13.14	Outline of the Register Mode Operation.....	13-91
Figure 13.15	Outline of Link Mode	13-98
Figure 13.16	Outline of the Descriptor Area and DMA Transfer Area.....	13-106
Figure 13.17	Link Mode Configuration Example	13-109
Figure 13.18	Single Transfer Mode Example	13-112
Figure 13.19	Block Transfer Mode Example	13-113
Figure 13.20	Fixed Priority Mode Example.....	13-115
Figure 13.21	Round Robin Mode.....	13-116
Figure 13.22	Edge Detection Mode Operation Example 1	13-118
Figure 13.23	Edge Detection Mode Operation Example 2	13-118
Figure 13.24	Level Detection Mode Operation Example 1.....	13-119
Figure 13.25	Level Detection Mode Operation Example 2.....	13-119
Figure 13.26	Pulse Output Mode Operation Example 1	13-122
Figure 13.27	Pulse Output Mode Operation Example 2	13-122
Figure 13.28	Level Output Mode Operation Example 1	13-123
Figure 13.29	Level Output Mode Operation Example 2.....	13-123
Figure 13.30	Bus Cycle Output Mode Operation Example 1.....	13-124
Figure 13.31	Bus Cycle Output Mode Operation Example 2.....	13-124
Figure 13.32	DMA Transfer Completion Interrupt Output Operation Example.....	13-126
Figure 13.33	DMA Terminal Count Output Operation Example.....	13-127
Figure 13.34	When the Transfer Size of the Source Is Smaller Than That of the Destination	13-130
Figure 13.35	When the Transfer Size of the Destination Is Smaller Than That of the Source	13-130
Figure 13.36	When the Transfer Size of the Destination Is Equal to That of the Source.....	13-131
Figure 13.37	DMA Write Access and Access Type Example	13-132
Figure 13.38	Operation Flow of Setting Example 1.....	13-137
Figure 13.39	Operation Flow of Setting Example 2.....	13-140
Figure 13.40	Operation Flow of Setting Example 3.....	13-143
Figure 13.41	Operation Flow of Setting Example 4.....	13-146
Figure 14.1	Block Diagram of the TAUJ2	14-5

Figure 14.2	General Procedure for Specifying TAUJ2TTOUTm Channel Output Mode.....	14-27
Figure 14.3	Grouping of the Channels and Assignment of Operation Clocks	14-32
Figure 14.4	Basic Procedure of Simultaneous Reloading	14-34
Figure 14.5	Block Diagram of Interval Timer.....	14-37
Figure 14.6	General Timing Diagram of Interval Timer.....	14-38
Figure 14.7	Count Clock = PCLK/2.....	14-42
Figure 14.8	Count Clock = PCLK.....	14-43
Figure 14.9	Operation Stop and Restart (TAUJ2CMORm.TAUJ2MD0 = 1).....	14-44
Figure 14.10	Forced Restart Operation, TAUJ2CMORm.TAUJ2MD0 = 1	14-45
Figure 14.11	TAUJ2TTINm Block Diagram of TAUJ2TTINm Input Interval Timer.....	14-46
Figure 14.12	TAUJ2TTINm General Timing Diagram of TAUJ2TTINm Input Interval Timer.....	14-47
Figure 14.13	Counter Triggered by Rising TAUJ2TTINm Input Edge (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B), TAUJ2CMORm.TAUJ2MD0 = 1	14-51
Figure 14.14	Block Diagram of Delay Counting	14-52
Figure 14.15	General Timing Diagram of Delay Counting.....	14-53
Figure 14.16	Block Diagram of TAUJ2TTINm Input Pulse Interval Measurement.....	14-57
Figure 14.17	General Timing Diagram of TAUJ2TTINm Input Pulse Interval Measurement	14-58
Figure 14.18	TAUJ2CMORm.TAUJ2COS[1:0] = 00B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B.....	14-62
Figure 14.19	TAUJ2CMORm.TAUJ2COS[1:0] = 10B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B.....	14-63
Figure 14.20	Block Diagram of TAUJ2TTINm Input Signal Width Measurement.....	14-64
Figure 14.21	General Timing Diagram of TAUJ2TTINm Input Signal Width Measurement	14-65
Figure 14.22	TAUJ2CMORm.TAUJ2COS[1:0] = 00B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 11B.....	14-69
Figure 14.23	TAUJ2CMORm.TAUJ2COS[1:0] = 10B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 11B.....	14-70
Figure 14.24	Block Diagram of Overflow Interrupt Output (for TAUJ2TTINm Width Measurement).....	14-71
Figure 14.25	General Timing Diagram at the Time of Overflow Interrupt Output.....	14-72
Figure 14.26	Block Diagram of External Event Counting	14-77
Figure 14.27	General Timing Diagram of External Event Counting	14-77
Figure 14.28	TAUJ2CDRm = 0000 0000H, TAUJ2CMURm.TAUJ2TIS[1:0] = 01B.....	14-80
Figure 14.29	Operation Stop and Restart (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B).....	14-80
Figure 14.30	Forced Restart (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B).....	14-81
Figure 14.31	Block Diagram of TAUJ2TTINm Input Position Detection.....	14-82
Figure 14.32	General Timing Diagram of TAUJ2TTINm Input Position Detection	14-83
Figure 14.33	Operation Stop and Restart, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B.....	14-87
Figure 14.34	Block Diagram of Overflow Interrupt Output (when TAUJ2TTINm Input Position is Detected) .	14-88
Figure 14.35	General Timing Diagram at the Time of Overflow Interrupt Output (when TAUJ2TTINm Input Position Detection is Used).....	14-89
Figure 14.36	Block Diagram of PWM Output	14-94
Figure 14.37	General Timing Diagram of PWM Output	14-95

Figure 14.38	TAUJ2CDRm (Slave) = 0000 0000H, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0).	14-101
Figure 14.39	TAUJ2CDRm (Slave) \geq TAUJ2CDRm (Master) + 1, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0)	14-102
Figure 14.40	Stopping and Restarting Operation, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0)	14-103
Figure 14.41	Simultaneous Reloading of the Master Channel	14-104
Figure 15.1	Block Diagram of WDTA	15-2
Figure 15.2	Timing Diagram of WDTA Start in Software Trigger Start Mode	15-7
Figure 15.3	Timing Diagram of WDTA NMI Request or Reset Generation	15-9
Figure 15.4	Timing Diagram of WDTA Window Function	15-10
Figure 16.1	Block Diagram of Asynchronous Serial Interface UARTJn	16-4
Figure 16.2	Transmission Interrupt Request Timing	16-28
Figure 16.3	Reception Interrupt Request Timing	16-29
Figure 16.4	Processing Flow after Interrupt Generation	16-31
Figure 16.5	LIN Transmission Outline	16-34
Figure 16.6	LIN Reception Outline	16-35
Figure 16.7	BF Transmission	16-36
Figure 16.8	Flowchart of BF Transmission	16-37
Figure 16.9	Normal BF Reception (Stop Bit after More Than 10.5 “L” Bits)	16-38
Figure 16.10	BF Reception Error (Stop Bit within 10.5 “L” Bits)	16-39
Figure 16.11	Timing Example of Data Consistency Error (No BF Reception Active, i.e. URTJnSTR0.URTJnSSBR = 0)	16-41
Figure 16.12	Flowchart of Data Transmission	16-42
Figure 16.13	UARTJn Reception	16-44
Figure 16.14	Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 0	16-45
Figure 16.15	Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 1	16-46
Figure 16.16	Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 0	16-47
Figure 16.17	Flowchart of Data Reception when URTJnSLBM = 1, URTJnSSBR = 1	16-48
Figure 16.18	Configuration of Bit-Rate Generator	16-52
Figure 17.1	CSIH Block Diagram	17-5
Figure 17.2	Transmission/Reception in Master Mode	17-36
Figure 17.3	Transmission/Reception in Slave Mode	17-37
Figure 17.4	Direct Master/Slave Connection	17-38
Figure 17.5	Connection between One Master and Two Slaves	17-38
Figure 17.6	Chip Select Timings	17-41
Figure 17.7	Chip Select and RCB Example	17-42
Figure 17.8	Clock Phase Timing (in the case of PCLK/4, $T_{hold0} = T_{setup1} = 0.5SCK$, $T_{idle0} = 0.5SCK$, $CKP0 = 0$ (CSIHnTCSS0) \rightarrow $CKP1 = 1$ (CSIHnTCSS1))	17-43
Figure 17.9	Clock Phase Timing (in the case of PCLK/4, $T_{hold0} = T_{setup1} = 0.5SCK$, $T_{idle0} = 1.0SCK$, $CKP0 = 0$ (CSIHnTCSS0) \rightarrow $CKP1 = 1$ (CSIHnTCSS1))	17-43
Figure 17.10	Clock Phase Timing (in the case of PCLK/4, $T_{hold0} = T_{setup1} = 0.5SCK$, $T_{idle0} = 0.5SCK$, $CKP0 = 0$ (CSIHnTCSS0) \rightarrow $CKP1 = 0$ (CSIHnTCSS1))	17-44
Figure 17.11	Data Phase Timing with CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0 and CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0	17-44

Figure 17.12 Data Phase Timing with CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 1 and CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 1	17-44
Figure 17.13 Job Examples	17-46
Figure 17.14 Baud Rate Generator Block Diagram	17-47
Figure 17.15 16-Bit Data, MSB First	17-52
Figure 17.16 14 Bit-Data, MSB First	17-52
Figure 17.17 EDL Timing Chart	17-54
Figure 17.18 Serial Data Direction Select Function — MSB First (CSIHnDIR = 0)	17-55
Figure 17.19 Serial Data Direction Select Function — LSB First (CSIHnDIR = 1)	17-55
Figure 17.20 Transmit/Receive Communication Timing in Slave Mode	17-56
Figure 17.21 Generation of CSIHnTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)	17-58
Figure 17.22 Immediate Generation of CSIHnTIC (CSIHnCTL1.CSIHnSLIT = 1)	17-58
Figure 17.23 Generation of CSIHnTIC in FIFO Memory Mode	17-59
Figure 17.24 Generation of CSIHnTIC in Job Mode	17-60
Figure 17.25 Generation of CSIHnTIR in Direct Access Memory Mode	17-62
Figure 17.26 CSIHnTIR Generation in Dual Buffer Mode	17-63
Figure 17.27 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)	17-65
Figure 17.28 Block Diagram of Data Consistency Checking	17-67
Figure 17.29 Parity Check Example	17-68
Figure 17.30 Timeout Error Check Functional Timing Chart	17-69
Figure 17.31 FIFO Overflow	17-70
Figure 17.32 FIFO Overflow Timing	17-71
Figure 17.33 Overrun Error Detection in Direct Access and Transmit-Only Buffer Mode	17-72
Figure 17.34 Overrun Error Detection in FIFO Mode (FIFO Full)	17-73
Figure 17.35 Overrun Error Detection in FIFO Mode (No Data)	17-74
Figure 17.36 Normal Operation (CSIHnCTL1.CSIHnLBM = 0)	17-75
Figure 17.37 Loop-Back Operation (CSIHnCTL1.CSIHnLBM = 1)	17-75
Figure 17.38 Direct Access Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)	17-76
Figure 17.39 Direct Access Mode (for Reception in Master Mode, and when Job Mode is Disabled)	17-78
Figure 17.40 Direct Access Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)	17-80
Figure 17.41 Direct Access Mode (for Reception in Slave Mode, and when Job Mode is Disabled)	17-82
Figure 17.42 Direct Access Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)	17-84
Figure 17.43 Direct Access Mode (for Reception in Master Mode, and when Job Mode is Enabled)	17-86
Figure 17.44 Transmit-Only Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)	17-88
Figure 17.45 Transmit-Only Buffer Mode (for Reception in Master Mode, and when Job Mode is Disabled)	17-90
Figure 17.46 Transmit-Only Buffer Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)	17-92
Figure 17.47 Transmit-Only Buffer Mode (for Reception in Slave Mode, and when Job Mode is Disabled)	17-94

Figure 17.48 Transmit-Only Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled).....	17-96
Figure 17.49 Transmit-Only Buffer Mode (for Reception in Master Mode, and when Job Mode is Enabled)....	17-98
Figure 17.50 Dual Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)	17-100
Figure 17.51 Dual Buffer Mode (for Reception in Master Mode, and when Job Mode is Disabled)	17-102
Figure 17.52 Dual Buffer Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)	17-104
Figure 17.53 Dual Buffer Mode (for Reception in Slave Mode, and when Job Mode is Disabled).....	17-106
Figure 17.54 Dual Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)	17-108
Figure 17.55 Dual Buffer Mode (for Reception in Master Mode, and when Job Mode is Enabled).....	17-110
Figure 17.56 FIFO Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)	17-113
Figure 17.57 FIFO Mode (for Reception in Master Mode, and when Job Mode is Disabled).....	17-116
Figure 17.58 FIFO Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled).....	17-119
Figure 17.59 FIFO Mode (for Reception in Slave Mode, and when Job Mode is Disabled).....	17-122
Figure 17.60 FIFO Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled).....	17-124
Figure 17.61 FIFO Mode (for Reception in Master Mode, and when Job Mode is Enabled).....	17-126
Figure 18.1 Block Diagram of IICBn.....	18-2
Figure 18.2 Pin Configuration Diagram.....	18-30
Figure 18.3 IIC Bus Serial Data Transfer Timing.....	18-31
Figure 18.4 Start Condition.....	18-31
Figure 18.5 Address	18-32
Figure 18.6 Transfer Direction Specification.....	18-33
Figure 18.7 Acknowledge (\overline{ACK}).....	18-33
Figure 18.8 Stop Condition	18-34
Figure 18.9 Wait State (1/2).....	18-35
Figure 18.10 Arbitration Timing Example.....	18-37
Figure 18.11 Valid Times to Write to IICBnDAT Register	18-50
Figure 18.12 INTIICBnTIA Signal Output Timing (Reception in Continuous Transfer Mode)	18-58
Figure 18.13 INTIICBnTIA Signal Output Timing (Transmission in Continuous Transfer Mode)	18-59
Figure 18.14 Master Operation Setting Procedure during Single Transfer Mode (Single Master Environment)	18-117
Figure 18.15 Slave Operation Setting Procedure during Single Transfer Mode (Single Master Environment).	18-118
Figure 18.16 Master Operation Setting Procedure during Continuous Transfer Mode (Single Master Environment)	18-119
Figure 18.17 Slave Operation Setting Procedure during Continuous Transfer Mode (Single Master Environment)	18-120
Figure 18.18 Single Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (IICBnCTL1.IICBnSLRS bit = 0) (Multi-Master Environment) (1/2)	18-121
Figure 18.19 Single Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (1/2)	18-123

Figure 18.20	Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (ICBnCTL1.ICBnSLRS bit = 0) (Multi-Master Environment) (1/2)	18-125
Figure 18.21	Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (ICBnCTL1.ICBnSLRS bit = 1) (Multi-Master Environment) (1/2)	18-127
Figure 19.1	Block Diagram of the CAN Controller	19-6
Figure 19.2	FCN Module Clock.....	19-44
Figure 19.3	Data Bit Time	19-45
Figure 19.4	Setting Transmission Request (FCNnMmCTL.FCNnMmTRQF) to Transmit Message Buffer after Redefinition	19-68
Figure 19.5	Transition to Operation Mode.....	19-69
Figure 19.6	Reception Timing	19-72
Figure 19.7	Receive History List	19-74
Figure 19.8	Message Processing Example	19-79
Figure 19.9	Transmit History List.....	19-82
Figure 19.10	FCN Module Terminal Connection in Receive-Only Mode.....	19-93
Figure 19.11	FCN Module Terminal Connection in Self-Test Mode	19-95
Figure 19.12	Timing Diagram of Capture Signal TSOUT	19-97
Figure 19.13	Initialization	19-105
Figure 19.14	Re-initialization without Using the Software Reset.....	19-106
Figure 19.15	Re-Initialization with Software Reset.....	19-107
Figure 19.16	Message Buffer Initialization.....	19-108
Figure 19.17	Message Buffer Redefinition during Reception.....	19-109
Figure 19.18	Message Buffer Redefinition during Transmission	19-110
Figure 19.19	Message Transmit Processing.....	19-111
Figure 19.20	ABT Message Transmit Processing.....	19-112
Figure 19.21	Transmission via Interrupt (Using FCNnCMLOSTR Register)	19-113
Figure 19.22	Transmission via Interrupt (Using FCNnCMTGTX Register)	19-114
Figure 19.23	Transmission via Software Polling	19-116
Figure 19.24	Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed)	19-118
Figure 19.25	Transmission Abort Processing (in Normal Operation Mode with ABT) – Repeat Option for Aborted Message	19-119
Figure 19.26	ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1)	19-120
Figure 19.27	ABT Transmission Request Abort Processing (In Normal Operation Mode with ABT) (2).....	19-121
Figure 19.28	ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) with Transmission Complete Flag	19-122
Figure 19.29	Transmission Abort Processing with Transmission Abort Interrupt and Transmission Complete Flag	19-123
Figure 19.30	Transmission Abort Processing with Transmission Complete Flag	19-124
Figure 19.31	Reception via Interrupt (Using FCNnCMLISTR Register)	19-125
Figure 19.32	Reception via Interrupt (Using FCNnCMRGRX Register)	19-126
Figure 19.33	Another Way of Reception via Interrupt (Using FCNnCMRGRX Register)	19-128
Figure 19.34	Reception via Software Polling	19-129

Figure 19.35	Setting FCN Sleep Mode/Stop Mode.....	19-131
Figure 19.36	Release from FCN Sleep/Stop Mode	19-132
Figure 19.37	Recovery from Bus-Off	19-133
Figure 19.38	Normal Shutdown Processing.....	19-134
Figure 19.39	Forced Shutdown Processing.....	19-134
Figure 19.40	Error Handling	19-135
Figure 19.41	Setting CPU Standby (from FCN Sleep Mode).....	19-136
Figure 19.42	Setting CPU Standby (from FCN Stop Mode).....	19-137
Figure 20.1	Configuration of the CCSRUN Register and CC-Link (Intelligent Device Station and Remote Device Station) RUN Signals.....	20-4
Figure 21.1	Protection Release Sequence	21-7
Figure 21.2	Configuration of Timer Input Selection.....	21-9
Figure 21.3	Digital Noise Filtering on Interrupt Signals (when the signal edge is used as the trigger)	21-20
Figure 21.4	Configuration of Trigger-Synchronous Ports.....	21-24
Figure 22.1	JTAG Interface Connection Example (20-Pin Half Pitch without Trace)	22-1
Figure 22.2	JTAG Interface Connection Example (20-Pin Half Pitch with Trace).....	22-2
Figure 22.3	JTAG Interface Connection Example (20-Pin Full Pitch)	22-3
Figure 22.4	SWD Interface Connection Example.....	22-4
Figure 22.5	Trace Port Interface Connection Example	22-4
Figure 22.6	SWV Interface Connection Example.....	22-5

List of Tables

Table 1.1	Type Names of R-IN32M3-Series Products	1-1
Table 2.1	Reset Sources and Targets to be Reset.....	2-7
Table 3.1	Interrupt from Internal Instruction RAM and Request for Peripheral Modules.....	3-3
Table 3.2	Interrupt from Internal Data RAM and Request for Peripheral Modules	3-4
Table 3.3	Interrupt from Buffer RAM and Request for Peripheral Modules	3-5
Table 4.1	AHB Internal Buses of an R-IN32M3	4-1
Table 5.1	Selecting the Boot Mode.....	5-1
Table 7.1	Number of Buffers that can be Acquired	7-35
Table 7.2	HWFNC_LongBuffer_Get	7-37
Table 7.3	HWFNC_ShortBuffer_Get	7-38
Table 7.4	HWFNC_Buffer_Release	7-39
Table 7.5	HWFNC_Buffer_Return.....	7-40
Table 7.6	HWFNC_MACDMA_RX_Enable	7-46
Table 7.7	HWFNC_MACDMA_RX_Disable	7-47
Table 7.8	HWFNC_MACDMA_RX_Control.....	7-48
Table 7.9	HWFNC_MACDMA_RX_Errstat.....	7-48
Table 7.10	HWFNC_MACDMA_TX_Start.....	7-51
Table 7.11	HWFNC_MACDMA_TX_Errstat.....	7-52
Table 7.12	HWFNC_Direct_Memory_Transfer.....	7-54
Table 7.13	HWFNC_Direct_Memory_Replace.....	7-55
Table 7.14	HWFNC_INTBUFF_DMA_Start.....	7-56
Table 7.15	HWFNC_INTBUFF_DMA_Start (Descriptor)	7-57
Table 7.16	Interrupts Related to Operations for Transmission	7-58
Table 7.17	Interrupts Related to Operations for Reception	7-59
Table 7.18	Interrupts Related to Other Operations	7-60
Table 7.19	GMAC_ACC Register Settings and Operation of the Tx TCPIP Accelerator.....	7-80
Table 7.20	GMAC_ACC Register Settings and Operation of the Rx TCPIP Accelerator.....	7-81
Table 8.1	Operation of the Hub and Switch by Filter Setting.....	8-75
Table 8.2	PTPv2 Multicast Domains: Layer 2.....	8-76
Table 8.3	PTP Multicast Domains: UDP/IP	8-76
Table 8.4	Management Frame Domains	8-76
Table 8.5	Switch Management Frame Domains	8-76
Table 8.6	DLR Multicast Domains	8-76
Table 8.7	Typical Hub MAC Filter Setup.....	8-77
Table 8.8	Definitions of Beacon Frame Fields	8-81
Table 8.9	UDP/IP Multicast Domains	8-85
Table 8.10	UDP Port Numbers	8-85
Table 8.11	PTPv2 Multicast Domains	8-85
Table 8.12	Common PTPv1 Message Header	8-86
Table 8.13	PTPv1 Message Type Identification	8-86

Table 8.14	Common PTPv2 Message Header	8-87
Table 8.15	PTPv2 Message Type Identification	8-87
Table 8.16	PTPv2 Message Flags Field Definitions	8-88
Table 8.17	Management Frame Tag (in transfer from the switch to the internal Ethernet MAC)	8-94
Table 8.18	Management Frame Tag (in transfer from the internal Ethernet MAC to the switch)	8-94
Table 8.19	Examples of Initial Settings of the Address Table	8-96
Table 8.20	Examples of Initial Settings of the Switch Engine	8-97
Table 8.21	Examples of Initial Settings of the MAC	8-98
Table 8.22	Initial Settings of the Hub	8-99
Table 8.23	Examples of Initial Settings of the Timer Module	8-99
Table 8.24	Examples of Initial Settings of the DLR Module	8-100
Table 9.1	Overview of Bus Control Registers	9-3
Table 9.2	Memory Access Timing Examples	9-16
Table 10.1	Synchronous Burst Access Memory Controller Control Registers	10-3
Table 10.2	Memory Access Timing Examples	10-27
Table 11.1	Mode of the External MCU Interface Selected by the Level on the Operating Mode Setting Pin....	11-1
Table 11.2	Method of Transfer	11-6
Table 11.3	Bus Sizing	11-6
Table 11.4	Synchronous Relationship of External MCU interface Signals	11-7
Table 11.5	Timing Adjustment of the External MCU Interface	11-12
Table 11.6	Write Strobe Signal	11-13
Table 11.7	Address Range for which Advance Reading and Page ROM Reading are Selectable	11-18
Table 11.8	Register Settings for Each Area and Method of Access	11-26
Table 11.9	Operating Mode Settings	11-27
Table 11.10	Synchronous SRAM Control Registers of the External MCU Interface	11-28
Table 11.11	Register Settings and Method of Access Using the HCSZ Pin	11-34
Table 11.12	Register Settings and Method of Access Using the HPGCSZ Pin	11-34
Table 12.1	Control Registers of the Serial Flash ROM Memory Controller	12-2
Table 12.2	SPI Instruction Set to Be Generated Automatically	12-18
Table 12.3	SFMSMD Register Settings for Standard Reading	12-28
Table 12.4	SFMSSC Register Settings for Standard Reading	12-29
Table 12.5	SFMSKC Register Settings for Standard Reading	12-30
Table 12.6	SFMSMD Register Settings for Fast Read Dual I/O	12-32
Table 12.7	SFMSSC Register Settings for Fast Read Dual I/O	12-33
Table 12.8	SFMSKC Register Settings for Fast Read Dual I/O	12-34
Table 13.1	R-IN32M3 DMA Controllers	13-1
Table 13.2	Slaves as Targets for Transfer by the DMA Controller	13-3
Table 13.3	Relation between DMA Units/Channels and External DMA Interface Pins	13-4
Table 13.4	Definition of the Terms Used for the DMA Controller	13-6
Table 13.5	DMA Controller Register Configuration	13-7
Table 13.6	DMA Controller Control Registers	13-9
Table 13.7	Correspondence between DMA End Status Registers and Interrupt Signals	13-44
Table 13.8	Correspondence between DMA End Status Register and Interrupt Signal	13-76

Table 13.9	General DMA Controller Interrupt Output	13-89
Table 13.10	Interrupt Output of DMA Controller for Real-Time Ports.....	13-89
Table 13.11	Register Mode and Link Mode	13-90
Table 13.12	Register Mode Setting	13-94
Table 13.13	INTDMAn Operation Selection.....	13-94
Table 13.14	Terminal Count Output (DMATCZp) Mask Setting.....	13-94
Table 13.15	Continuous Execution Set.....	13-95
Table 13.16	Automatic Register Set Switch Setting.....	13-95
Table 13.17	Link Mode Selection.....	13-101
Table 13.18	Link Address Register Set	13-101
Table 13.19	Descriptor Format	13-102
Table 13.20	Description of Each Field of the Descriptor	13-103
Table 13.21	Correspondence between the Descriptors Other Than the Header and the DMAC Internal Registers	13-105
Table 13.22	Setting for Write-Only Mode.....	13-110
Table 13.23	DMA Transfer Mode Selection.....	13-111
Table 13.24	DMA Channel Priority Control Selection.....	13-114
Table 13.25	Specification of the Detection for Each DMA Transfer Request Source.....	13-117
Table 13.26	DMA Transfer Request Signal Detection Method.....	13-117
Table 13.27	Specification of the Acknowledge Signal Mode for Each DMA Transfer Request Source.....	13-121
Table 13.28	DMA Acknowledge Signal (DMAACKZp) Output Mode.....	13-121
Table 13.29	Relationship between DMA Transfer Completion Interrupts and Units/Channels.....	13-125
Table 13.30	DMA Transfer Completion Interrupt Asserting Conditions	13-126
Table 13.31	DMA Terminal Count Output Setting	13-128
Table 13.32	Conditions for Transfer in DMA Transfer Setting Examples	13-135
Table 13.33	DMA Transfer Setting Example 1	13-135
Table 13.34	Register Settings of Setting Example 1.....	13-135
Table 13.35	Channel Configuration Register (CHCFG1) Settings of Setting Example 1	13-136
Table 13.36	DMA Transfer Setting Example 2	13-138
Table 13.37	Register Settings of Setting Example 2.....	13-138
Table 13.38	Channel Configuration Register (CHCFG2) Settings of Setting Example 2	13-139
Table 13.39	DMA Transfer Setting Example 3	13-141
Table 13.40	Register Settings of Setting Example 3.....	13-141
Table 13.41	Channel Configuration Register (CHCFG1) Settings of Setting Example 3	13-142
Table 13.42	DMA Transfer Setting Example 4	13-144
Table 13.43	Descriptor 1 Settings of DMA Transfer Setting Example 4	13-144
Table 13.44	Descriptor 2 Settings of DMA Transfer Setting Example 4	13-145
Table 13.45	Descriptor 3 Settings of DMA Transfer Setting Example 4	13-145
Table 13.46	Register Settings of Setting Example 4.....	13-146
Table 13.47	Descriptor Settings of Setting Example 4	13-146
Table 14.1	TAUJ2 I/O Signals.....	14-1
Table 14.2	TAUJ2 Interrupt Signals and Requests for Peripheral Modules	14-2
Table 14.3	TAUJ2 Operations	14-3

Table 14.4	TAUJ2 Registers Overview	14-7
Table 14.5	TAUJ2CMORm Settings for Interval Timer	14-39
Table 14.6	TAUJ2CMURm Settings for Interval Timer	14-39
Table 14.7	Simultaneous Reload Settings for Interval Timer	14-40
Table 14.8	Control Bit Settings for Independent Channel Output	14-40
Table 14.9	Operating Procedure	14-41
Table 14.10	TAUJ2CMORm Settings	14-48
Table 14.11	TAUJ2CMURm Settings for TAUJ2TTINm Input Interval Timer	14-48
Table 14.12	Simultaneous Reload Settings for TAUJ2TTINm Input Interval Timer	14-49
Table 14.13	Control Bit Settings for Channel Output.....	14-49
Table 14.14	Operating Procedure	14-50
Table 14.15	TAUJ2CMORm Settings for Delay Counting	14-54
Table 14.16	TAUJ2CMURm Settings for Delay Counting	14-54
Table 14.17	Simultaneous Reload Settings for Delay Counting.....	14-55
Table 14.18	Control Bit Settings for Independent Channel Output	14-55
Table 14.19	Operating Procedure	14-56
Table 14.20	TAUJ2CMORm Settings for TAUJ2TTINm Input Pulse Interval Measurement.....	14-59
Table 14.21	TAUJ2CMORm Settings for TAUJ2TTINm Input Pulse Interval Measurement.....	14-60
Table 14.22	Simultaneous Reload Settings for TAUJ2TTINm Input Pulse Interval Measurement	14-60
Table 14.23	Control Bit Settings for Independent Channel Output	14-60
Table 14.24	Operating Procedure	14-61
Table 14.25	TAUJ2CMORm Settings for TAUJ2TTINm Input Signal Width Measurement.....	14-66
Table 14.26	TAUJ2CMURm Settings for TAUJ2TTINm Input Signal Width Measurement.....	14-67
Table 14.27	Simultaneous Reload Settings for TAUJ2TTINm Input Signal Width Measurement	14-67
Table 14.28	Control Bit Settings for Independent Channel Output Mode.....	14-67
Table 14.29	Operating Procedure	14-68
Table 14.30	TAUJ2CMORm Settings	14-73
Table 14.31	TAUJ2CMURm Settings for TAUJ2TTINm Input Signal Width Measurement.....	14-74
Table 14.32	Simultaneous Reload Settings for TAUJ2TTINm Input Signal Width Measurement	14-74
Table 14.33	Control Bit Settings for Independent Channel Output	14-74
Table 14.34	Operating Procedure	14-75
Table 14.35	Contents of the TAUJ2CMORm Register for External Event Counting	14-78
Table 14.36	Contents of the TAUJ2CMURm Register for External Event Counting	14-78
Table 14.37	Simultaneous Reload Settings for External Event Counting.....	14-79
Table 14.38	Operating Procedure for External Event Counting	14-79
Table 14.39	TAUJ2CMORm Settings for TAUJ2TTINm Input Position Detection.....	14-84
Table 14.40	TAUJ2CMURm Settings for TAUJ2TTINm Input Position Detection.....	14-84
Table 14.41	Simultaneous Reload Settings for Delay Counting.....	14-85
Table 14.42	Control Bit Settings for Independent Channel Output	14-85
Table 14.43	Operating Procedure	14-86
Table 14.44	TAUJ2CMORm Settings	14-90
Table 14.45	TAUJ2CMURm Settings	14-90
Table 14.46	Simultaneous Reload Settings for TAUJ2TTINm Input Signal Width Measurement	14-91

Table 14.47	Control Bit Settings for Independent Channel Output	14-91
Table 14.48	Operating Procedure	14-92
Table 14.49	TAUJ2CMORm Settings for the Master Channel of PWM Output.....	14-96
Table 14.50	TAUJ2CMURm Settings for the Master Channel for PWM Output	14-97
Table 14.51	Simultaneous Reload Settings.....	14-97
Table 14.52	Control Bit Settings for Independent Channel Output.....	14-97
Table 14.53	TAUJ2CMORm Settings for the Slave Channel for PWM Output	14-98
Table 14.54	TAUJ2CMURm Settings for the Slave Channel(s) for PWM Output	14-98
Table 14.55	Simultaneous Reload Settings.....	14-99
Table 14.56	Control Bit Settings in Independent Channel Output Mode 1	14-99
Table 14.57	Operating Procedure for PWM Output	14-100
Table 15.1	Channels of WDTA	15-1
Table 15.2	WDTA Interrupts and Reset Outputs.....	15-1
Table 15.3	Overview of WDTA Registers.....	15-3
Table 15.4	Trigger Register and Activation Code	15-8
Table 16.1	Channels of UARTJn.....	16-1
Table 16.2	UARTJn I/O Signals	16-1
Table 16.3	UARTJn Interrupts	16-2
Table 16.4	UARTJn Registers	16-5
Table 16.5	Data Format Specification	16-32
Table 16.6	Reception Error Causes and Indicators	16-49
Table 16.7	Bit-Rate Generator Clocks Output.....	16-52
Table 16.8	Allowable Scope of Error in Bit Rate	16-53
Table 16.9	Example of Bit-Rate Generator Settings (PCLK = 100 MHz).....	16-54
Table 17.1	Channels of CSIH	17-1
Table 17.2	Number of Chip Select Signals of CSIH	17-1
Table 17.3	Maximum Transfer Speed (Baud Rate) of CSIH.....	17-1
Table 17.4	CSIHn Interrupts and Requests to Peripheral Modules	17-2
Table 17.5	CSIHn I/O Signals	17-3
Table 17.6	CSIH0 Register Overview	17-6
Table 17.7	CSIH1 Register Overview	17-6
Table 17.8	Memory Mode Operation	17-16
Table 17.9	Generation of CSIHnTIC in Job Mode	17-61
Table 17.10	CSIHnTIR Interrupt Generation	17-62
Table 17.11	Data Error Types.....	17-64
Table 17.12	CSIHnTIJC Interrupt Generation.....	17-64
Table 18.1	Channels of I ² CB	18-1
Table 18.2	IICBn Interrupts and Requests for Peripheral Modules	18-1
Table 18.3	I ² C Register.....	18-3
Table 18.4	Conditions for Generating Serial Output Timing.....	18-13
Table 18.5	Extension Code Bit Definitions	18-32
Table 18.6	Wait State Transit Timings	18-49
Table 18.7	Wait State Exit Conditions.....	18-53

Table 18.8	Interrupt Request Signal Output Timing (Single Transfer Mode)	18-55
Table 18.9	Interrupt Request Signal Output Conditions and Interrupt Request Signals Output during Address Transfer (Single Transfer Mode)	18-56
Table 18.10	Interrupt Request Signal Output Conditions and Interrupt Request Signals Output during Data Transfer (Single Transfer Mode)	18-57
Table 18.11	INTIICBnTIS Signal Output Timing	18-60
Table 18.12	INTIICBnTIS Signal Output Conditions during Address Transfer (Continuous Transfer Mode)..	18-61
Table 18.13	INTIICBnTIS Signal Output Conditions during Data Transfer (Continuous Transfer Mode)	18-62
Table 19.1	Channels of FCN	19-1
Table 19.2	Message Buffers of FCN Channels.....	19-1
Table 19.3	FCNn Interrupts and Requests for Peripheral Modules	19-2
Table 19.4	FCN I/O Signals.....	19-3
Table 19.5	Overview of Functions.....	19-5
Table 19.6	List of FCN Registers (1/2).....	19-7
Table 19.7	FCN0 Global and Module Registers (1/2)	19-9
Table 19.8	FCN1 Global and Module Registers (1/2)	19-11
Table 19.9	Bit Configuration of FCN Global Registers.....	19-13
Table 19.10	Bit Configuration of FCN Module Mask Control 16-Bit Registers	19-14
Table 19.11	Bit Configuration of FCN Module Mask Control 32-Bit Registers	19-15
Table 19.12	Bit Configuration of FCN Module Registers	19-16
Table 19.13	Bit Configuration of FCN Message Buffer Registers (1/2)	19-17
Table 19.14	Bit Set/Clear Operation.....	19-20
Table 19.15	Multi-Buffer Receive Block (MBRB) Priorities	19-71
Table 19.16	List of FCN Module Interrupt Sources	19-92
Table 19.17	Outline of the Receive/Transmit in Each Operation Mode	19-96
Table 19.18	Combinations of Available Bit Rate Settings (1/3).....	19-100
Table 19.19	Representative Examples of Baud Rate Settings ($f_{CANMOD} = 20$ MHz) (1/2).....	19-103
Table 20.1	CC-Link Outline Specifications.....	20-1
Table 21.1	Signals Subject to Noise Elimination.....	21-16

1. Introduction

This document describes the internal peripheral modules of the R-IN32M3 series of industrial Ethernet network LSI chips.

1.1 Type Names of R-IN32M3-Series Products

In the type names of R-IN32M3-series (R-IN32M3-EC and R-IN32M3-CL) products, those of the current products include “B” while those of the old products do not.

Unless specially stated otherwise, the products mentioned in this manual are the current products.

Table 1.1 Type Names of R-IN32M3-Series Products

Product Name	Designation	Type Name	Features
R-IN32M3-EC	Current products	MC-10287BF1-HN4-A MC-10287BF1-HN4-M1-A	EtherCAT [®] slave controller supported version
	Old products	MC-10287F1-HN4-A MC-10287F1-HN4-M1-A	
R-IN32M3-CL	Current products	UPD60510BF1-HN4-A UPD60510BF1-HN4-M1-A	CC-Link IE intelligent device station supported version
	Old products	UPD60510F1-HN4-A UPD60510F1-HN4-M1-A	

Caution: In the current products, the ETHSW10HDEN and CPUBUSMD registers have been newly added and the initial value of the RINVER register has been changed from that in the old products. For details, see the description of the following sections.

8.3.2.4, Ethernet Switch 10-Mbps/Half-Duplex Mode Setting Register (ETHSW10HDEN)

21.4, Version Register (RINVER)

21.14, CPU Bus Operating Mode Register (CPUBUSMD)

1.2 Base Addresses of the System Registers Area

The addresses of registers given in the subsequent sections 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 microcontroller interface
BASE = D_0000H

2. Clocks and Resets

2.1 Clock Configuration

2.1.1 Description of Internal Clocks

An R-IN32M3 uses various clocks.

The following lists the major clock signals covered in this document.

Clock Signal	Application
OSCCLK	This is the clock before passing through the internal PLL. It is a 25-MHz clock with no frequency multiplication.
FCLK	This is the clock signal of the internal system bus. This clock signal runs even in standby mode. This clock is the base clock signal for use in access to the CPU and Ethernet MAC. If the setting for the clock signal generated by the VCO in the PLL is 500 MHz, this clock signal runs at 100 MHz. The duty of this clock signal is 50%.
HCLK	This is the clock signal of the internal system bus. This clock signal stops in standby mode. This clock is the base clock signal for use in access to the HW-RTOS, DMA controller, and memory controller. If the setting for the clock signal generated by the VCO in the PLL is 500 MHz, this clock signal runs at 100 MHz. The duty of this clock signal is 50%. The watchdog timer uses this clock as is, or with its frequency divided.
PCLK	This is the clock signal for internal peripheral macros. This clock is the base clock used for accessing peripheral circuits such as the timers, serial interfaces, and I2C. The frequency of PCLK is the same as that of HCLK. If the frequency of HCLK is 100 MHz, the frequency of PCLK is also 100 MHz. The duty of this clock signal is 50%.
BUSCLK ^{Note1}	This is the 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 ^{Note2}	This is the clock signal for the external MCU interface. This clock signal is used in clock-synchronous access by an external microcontroller.
CLKOUT25M0-1 ^{Note3}	This is the PHY clock output pin. This pin outputs the 25 MHz oscillation clock as is.

Notes 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 you use asynchronous mode, input a low level to the HBUSCLK pin.

3. The CLKOUT25M0-1 pins are only provided in the R-IN32M3-CL.

2.1.2 Clock Configuration Diagram

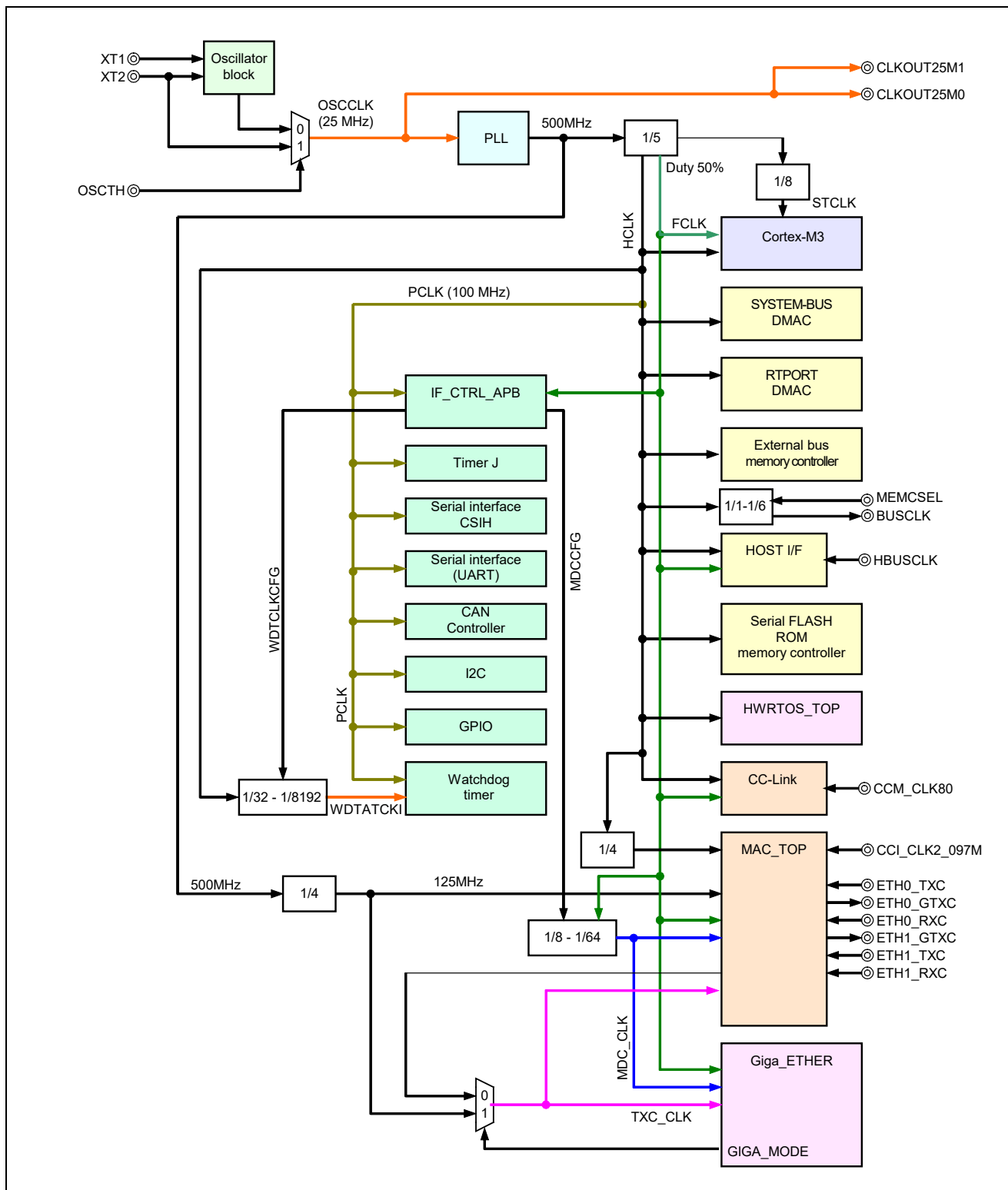


Figure 2.1 Clock Configuration Diagram

2.2 Stopping of Clock Supply

2.2.1 Overview

An R-IN32M3 is capable of stopping clock supply to unused modules. Once supply of the clock signal to a module is stopped by using the CLKGTD register, it cannot be resumed. To supply the clock signal again, reset the system.

2.2.2 Clock Control Registers (CLKGTD0, CLKGTD1)

These registers are used to stop clock supply to unused modules to save power.

These registers can be read or written in 32- or 16-bit units.

- Access These registers can be read or written in 32- or 16-bit units.

Cautions 1. These registers are write-protected and can only be written after they have been released from protection with a special instruction sequence by using the system protection command register (SYSPCMD). For how to unlock protection, see the description of the system protection command register (SYSPCMD). No special instruction sequence is required for reading the value of this 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 system.

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

	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		
CLKGTD0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	GCBCLK	GCI2C1	GCI2C0	GCUART1	GCUART0	GCTIMER	GCRTOS	GCHOSTIF	1	1	1	1	GCMEMC	GCSROM	GCDMACRT	GCDMAC	Address BASE+ 01A0H Initial value 0000 FFFFH
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	1	1	1	1	R/W	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	GCBCLK	BUSCLK output (1: Operating, 0: Stopped)
14	GCI2C1	I2C bus (I2C1) (1: Operating, 0: Stopped)
13	GCI2C0	I2C bus (I2C0) (1: Operating, 0: Stopped)
12	GCUART1	Asynchronous serial interface 1 (UART1) (1: Operating, 0: Stopped)
11	GCUART0	Asynchronous serial interface 0 (UART0) (1: Operating, 0: Stopped)
10	GCTIMER	Timer array unit (TAUJ2) (1: Operating, 0: Stopped)
9	GCRTOS	HW-RTOS (including Gigabit Ethernet controller) (1: Operating, 0: Stopped)
8	GCHOSTIF	External MCU interface (1: Operating, 0: Stopped)
7 to 4	—	Reserved. When writing to these bits, write 1. When read, 1 is returned.
3	GCMEMC ^{Note}	Memory controller (asynchronous/synchronous burst access) module (1: Operating, 0: Stopped)
2	GCSROM	Serial flash ROM (1: Operating, 0: Stopped)
1	GCDMACRT	DMAC for real-time ports (1: Operating, 0: Stopped)
0	GCDMAC	General-purpose DMAC (1: Operating, 0: Stopped)

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

	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	
CLKGTD1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	GCIE	GCCC	0	1	1	1	1	1	GCESW	GCCAT	GCWDT	GCFCN1	GCFCN0	GCCS11	GCCS10	BASE+01A4H
																		Initial value										Note3						
R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	0	1	1	1	1	1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit Position	Bit Name	Description
31 to 15	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
14	GCIE ^{Note1}	CC-Link IE Field Network (slave) (1: Operating, 0: Stopped)
13	GCCC	CC-Link (1: Operating, 0: Stopped)
12	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.
11 to 7	—	Reserved. When writing to these bits, write 1. When read, 1 is returned.
6	GCESW	Ethernet switch (1: Operating, 0: Stopped)
5	GCCAT ^{Note2}	EtherCAT module (1: Operating, 0: Stopped)
4	GCWDT	Watchdog timer (1: Operating, 0: Stopped)
3	GCFCN1	CAN1 module (1: Operating, 0: Stopped)
2	GCFCN0	CAN0 module (1: Operating, 0: Stopped)
1	GCCS11	CSI1 module (1: Operating, 0: Stopped)
0	GCCS10	CSI0 module (1: Operating, 0: Stopped)

Notes 1. This function is only available in the R-IN32M3-CL. When using a product other than the R-IN32M3-CL, write 0 to this bit. When read, 0 is returned.

2. This function is only available in the R-IN32M3-EC. When using a product other than the R-IN32M3-EC, write 0 to this bit. When read, 0 is returned.

3. The initial value depends on the product you are using.

R-IN32M3-CL: 0000 6FDFH

R-IN32M3-EC: 0000 2FFFH

Cautions 1. Once the clock supply is stopped by using the CLKGTD register, it cannot be resumed. To supply the clock signal again, reset the system.

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

2.3 Reset

2.3.1 Overview

- Reset by signal input from the RESETZ pin
- Power-on reset by signal input from the PONRZ pin (including initialization of internal RAM of an R-IN32M3)
- Reset by signal input from the HOTRESETZ pin ^{Note}
- Noise elimination for external reset pins based on analog delay (Applicable pins: RESETZ, PONRZ, HOTRESETZ ^{Note}, and TRSTZ)
- Reset control by using software
- Reset control by using watchdog timer (WDT)
- Reset output

Note: The HOTRESETZ pin is only provided in the R-IN32M3-CL.

2.3.2 Features

(1) Reset by signal input from a pin

When the reset signal PONRZ, RESETZ, or HOTRESETZ ^{Note} is input, the CPU core and internal peripheral modules are initialized. Note that the input of HOTRESETZ ^{Note} does not reset the internal PLL.

The width at low level of each reset signal must be at least 1 μ s. However, in order for oscillation to produce the external oscillator clock (25 MHz) to become stable, the oscillation stabilization time must be included in the width at low level for the signal on each reset pin.

TRSTZ is only connected to the Cortex-M3 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. When using a half pitch connector as the ICE connector, it is recommended that TRSTZ should be open. For an example of the connection, see section 22 Debugging.

(2) Noise elimination for input reset signals

Noise on the external reset pins PONRZ, RESETZ, HOTRESETZ ^{Note}, and TRSTZ is eliminated based on analog delay. This is capable of eliminating transient signals shorter than 100 ns as noise.

(3) Reset by using software

An R-IN32M3 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.

(4) Reset by using the watchdog timer (WDT)

When a reset request is generated by the on-chip watchdog timer (WDT), 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 RESETZ pin. The internal RAM is not initialized.

Note: The HOTRESETZ pin is only provided in the R-IN32M3-CL.

(5) Reset output (RSTOUTZ output)

When a reset is generated in an R-IN32M3, a low-level signal is output from the RSTOUTZ pin. This reset can be used as a general reset for external devices.

Table 2.1 Reset Sources and Targets to be Reset

Reset Source	Target to be Reset						
	Instruction RAM Data RAM Buffer RAM	PLL	CC-Link IE Field Network ^{Note1} Power on reset	CC-Link	EtherCAT ^{Note2}	CPU's debugging unit	Other peripheral circuits
PONRZ pin	✓	✓	✓	✓	✓	—	✓
RESETZ pin	—	✓	✓	✓	✓	—	✓
HOTRESETZ pin ^{Note1}	—	—	—	✓	—	—	✓
TRSTZ pin	—	—	—	—	—	✓	—
Watchdog timer	—	—	—	✓	✓	—	✓
SYSRESET register	—	—	—	✓	✓	—	✓
CPU reset	—	—	—	✓	✓	—	✓
CCRES register	—	—	—	✓	—	—	—
CATRESET register ^{Note2}	—	—	—	—	✓	—	—

✓: Applicable; —: Not applicable

Notes 1. This is only provided in the R-IN32M3-CL.

2. This is only provided in the R-IN32M3-EC.

2.3.3 Reset Control Registers

(1) System reset register (SYSRESET)

This register resets an R-IN32M3 (equivalent to the HOTRESETZ input pin). The register for the PONRZ pin is not reset. This register is used to release an R-IN32M3 from the reset state.

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

Caution: This register is write-protected and can only be written after it has been released from protection with a special instruction sequence by using the system protection command register (SYSPCMD). For how to unlock protection, see the description of the system protection command register (SYSPCMD). No special instruction sequence is required for reading the value of this 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	0	BASE+01C0H	0001H
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
0	SYSRSET	This is for resetting an R-IN32M3. 0: An R-IN32M3 is in the reset state. 1: An R-IN32M3 has been released from the reset state.

2.4 Operations for Reset

The charts below show the timing of the reset at power on and when a system reset is issued for the R-IN32M3 series.

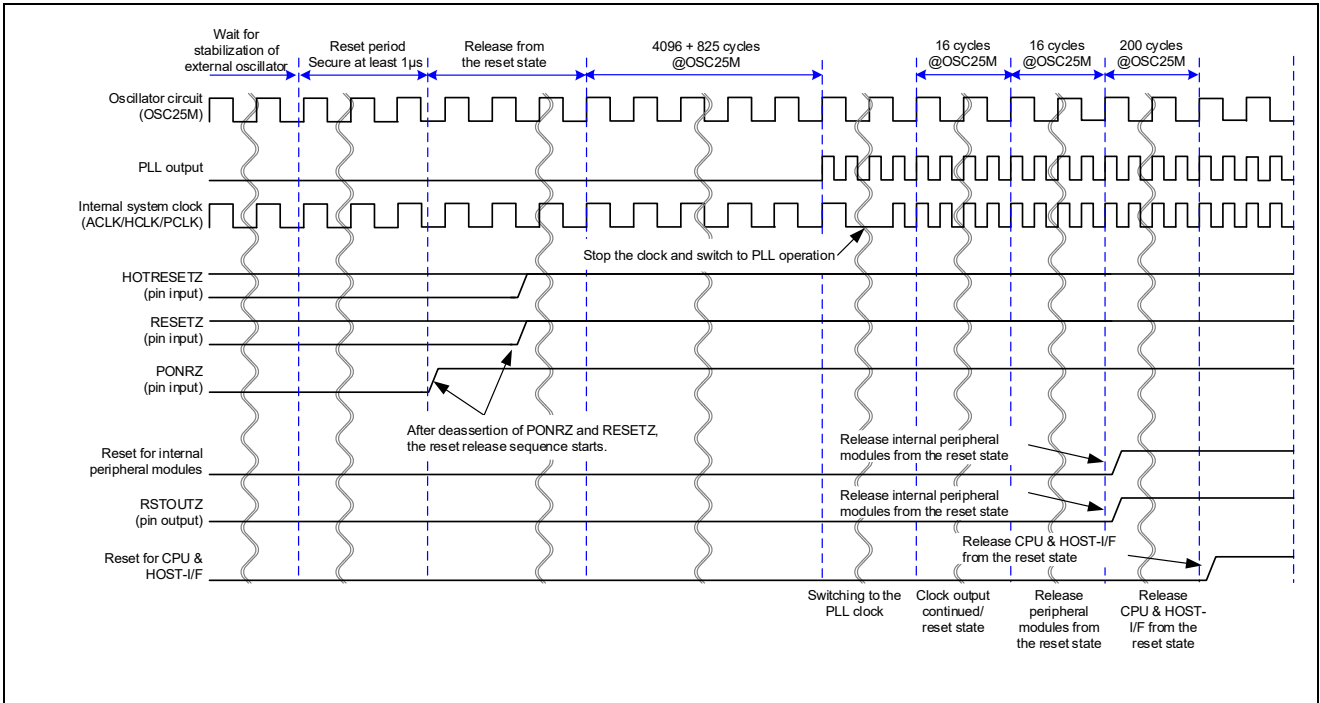


Figure 2.2 Timing of Reset at Power On

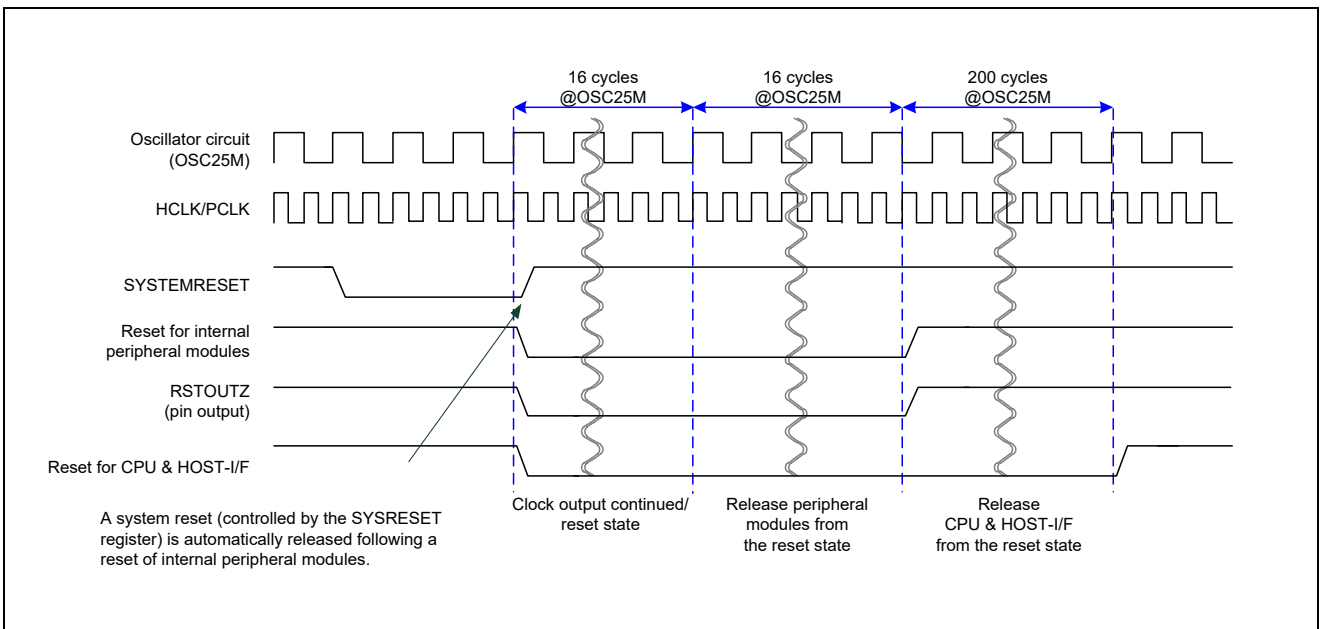


Figure 2.3 Timing of Reset at System Reset

(1) Software reset register (SFTRES1)

The watchdog timer can be reset by software. To apply a reset, be sure to clear the corresponding bit in the register to 0. Setting it to 1 leads to release from the reset state.

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

Caution: This register is write-protected and can only be written after it has been released from protection with a special instruction sequence by using the system protection command register (SYSPCMD). For how to unlock protection, see the description of the system protection command register (SYSPCMD). No special instruction sequence is required for reading the value of this register.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
SFTRES1	1	1	1	1	1	1	1	1	1	1	1	RSW DT	1	1	1	1	BASE+01B4H	FFFFH
R/W	1	1	1	1	1	1	1	1	1	1	1	R/W	1	1	1	1		

Bit Position	Bit Name	Description
15 to 5	—	Not used (fixed to 1)
4	RSWDT	Software reset for watchdog timer
3 to 0	—	Not used (fixed to 1)

3. CPU and Internal RAMs

An R-IN32M3 device incorporates a high-performance 32-bit processor (Arm® Cortex®-M3 core).

This section describes an overview of the CPU and internal RAMs of an R-IN32M3.

3.1 CPU-Core Information

The revision of the Cortex-M3 core currently used in an R-IN32M3 is Cortex-M3 r2p1.

Refer to the following URL of Arm Ltd. for details of the CPU core section, architecture, etc.

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

3.2 CPU-Core Configuration Information

The Cortex-M3 of an R-IN32M3 has the following configurations.

Category	Configuration Item	Setting	Remark
Process	Process	TBD	Arbitrary values are listed
Interrupt	NUM_IRQ	128	The number of IRQ interrupts to be input: 1 to 240 (NMI interrupts are counted separately)
Interrupt Priority	LVL_WIDTH	4	Priority bit number 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 (Settings in R-IN32M3)
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 (Settings in R-IN32M3)
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-IN32M3 products do not support SLEEPDEEP-mode. Do not set the SLEEPDEEP bit of the SCR register to 1.

3.3 Restrictions

The Cortex-M3 core of an R-IN32M3 has restrictions. Only the following one restriction applies to this product.

752419: Interrupts during loading to the SP can cause erroneous behavior

Outline:

When an interrupt occurs between the data phases of the single-word read access to the SP register, an incorrect address is entered in the SP register.

Countermeasures:

A countermeasure in software is required.

Change the instruction which loads a value to SP register directly to a group of instructions which load the value to a general register once, then move it to the SP register.

For details, refer to the Arm Ltd. website.

3.4 Internal Instruction RAM

The internal instruction RAM is a 768 Kbyte-RAM. It is accessible by I-Code AHB, D-Code AHB, DMAC, and host CPU.

3.4.1 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
- Burst transfer: single burst transfer, burst transfer of the required length, burst transfer of the fixed length (INCR4/8/16, WRAP4/8/16)
- Little endian fixed
- Support for ECC: 1-bit error correction, 2-bit error detection

Table 3.1 Interrupt from Internal Instruction RAM and Request for Peripheral Modules

Internal Instruction RAM Interrupt Signal	Function	Connected To
IRAMECCSEC	Instruction RAM 1-bit ECC error correction interrupt	- Interrupt controller
IRAMECCDED	Instruction RAM 2-bit ECC error detection interrupt	- Interrupt controller

3.4.2 Read Buffer

- 128-bit (32 bits x 4) read buffer
- Response to the AHB involves no waiting 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 is handled as an ECC error interrupt is generated.

3.4.3 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).

3.5 Internal Data RAM

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

3.5.1 Outline of Features

- AHB latency: latency is 1 in read and write access (latency is 2 in read access following write access).
- Communication-bus latency: 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
- Burst transfer: single burst transfer, burst transfer of the required length, burst transfer of the fixed length (INCR4/8/16, WRAP4/8/16)
- Little endian fixed
- Support for ECC: 1-bit error correction, 2-bit error detection

Table 3.2 Interrupt from Internal Data RAM and Request for Peripheral Modules

Internal Data RAM Interrupt Signal	Function	Connected To
DRAMECCSEC	Data RAM 1-bit ECC error correction interrupt	- Interrupt controller
DRAMECCDED	Data RAM 2-bit ECC error detection interrupt	- Interrupt controller

3.6 Buffer RAM

The buffer RAM is a 64-Kbyte RAM. It is accessible by the communication bus.

3.6.1 Outline of Features

- Communication-bus latency: latency is 1 in read and write access
- Arbitration of access when contention arises: Fixed priority (the communication bus is given priority)
- 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 3.3 Interrupt from Buffer RAM and Request for Peripheral Modules

Buffer RAM Interrupt Signal	Function	Connected To
BRAMECCSEC	Buffer RAM 1-bit ECC error correction interrupt	- Interrupt controller
BRAMECCDED	Buffer RAM 2-bit ECC error detection interrupt	- Interrupt controller

4. Bus Architecture

A multi-layered configuration is used for the internal AHB buses of an R-IN32M3, and a bus layer is provided for every six 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 4.1 AHB Internal Buses of an R-IN32M3

Master Slave	1.High ← (Default Priority) → Low						Priority Decision System
	DMAC for real-time ports	Host CPU	Cortex-M3 CPU D code bus	Cortex-M3 CPU system bus	General DMAC	Cortex-M3 CPU I code bus	
Data RAM	A	✓	—	✓	✓	—	Round robin (alternate) ^{Note3}
Instruction RAM	✓	✓	✓	—	✓	✓	Fixed priority
Buffer RAM	—	—	✓	—	✓	—	Round robin (fair) ^{Note4}
External memory	A	—	✓	✓	✓	✓	Round robin (alternate) ^{Note3}
Serial flash ROM	—	—	✓	✓	✓	✓	Round robin (fair) ^{Note4}
Ethernet MAC ^{Note6}	A	✓	—	✓	✓	—	Round robin (alternate) ^{Note3}
CC-Link	A	✓	—	✓	✓	—	Round robin (alternate) ^{Note3}
APB internal peripheral modules ^{Note1}	A	✓	—	✓	✓	—	Round robin (alternate) ^{Note3}
Real-time ports	A	✓	—	✓	✓	—	Round robin (alternate) ^{Note3}
General ports	A	✓	—	✓	✓	—	Round robin (fair) ^{Note4}
HW-RTOS ^{Note2}	—	—	—	✓	—	—	—
DMAC for real-time ports ^{Note5}	—	—	—	✓	—	—	—
General DMAC ^{Note5}	—	—	—	✓	—	—	—
Synchronous burst access memory controller	—	—	—	✓	—	—	—

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

- Notes**
- 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. RR (fair): Round robin.**
 - 5. The registers areas of the respective DMA controllers**
 - 6. The target module differs with the given product of the R-IN32M3 series.
R-IN32M3-EC: EtherCAT
R-IN32M3-CL: CC-Link IE Field Network**

4.1 Bus Occupancy by the Cortex-M3

By default, when the Cortex-M3 of an R-IN32M3-series device successively copies data within a given memory, it proceeds with burst transfer of the required length and occupies the bus over that period, making other masters wait for access to the memory. The period over which another master must wait for access thus depends on the period of continuous access by the Cortex-M3.

To reduce the period over which other masters are kept waiting for access, set the CPU bus operation mode register (CPUBUSMD) to change the form of transfer by the Cortex-M3 to single transfer. This allows access by other masters even during continuous access by the Cortex-M3.

5. Booting Procedure

An R-IN32M3 allows three locations from which to boot the CPU, which can be selected by using external pins BOOT0 and BOOT1: an external memory, the serial flash ROM, and the instruction RAM. The reset vector and interrupt vector can be switched by register settings. This section describes booting from an external memory, copying the program to the instruction RAM, and then specifying exception vectors to the instruction RAM.

5.1 Selecting the Boot Mode

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

Table 5.1 Selecting the Boot Mode

BOOT1	BOOT0	Boot Mode	Boot Area
0	0	External memory boot	Memory connected to the CSZ0 pin of the external bus interface
0	1	External serial flash ROM boot	Serial flash ROM
1	0	External MCU boot	Instruction RAM
1	1	Instruction RAM boot (debugger used ONLY)	Instruction RAM

(1) External memory boot mode

The CPU is booted from the external memory connected to the CSZ0 pin of the external bus interface.

(2) External serial flash ROM boot

The CPU is booted from the external 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 an R-IN32M3 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 period is terminated by using the CPURESET register.

(4) Instruction RAM boot (for debugging only)

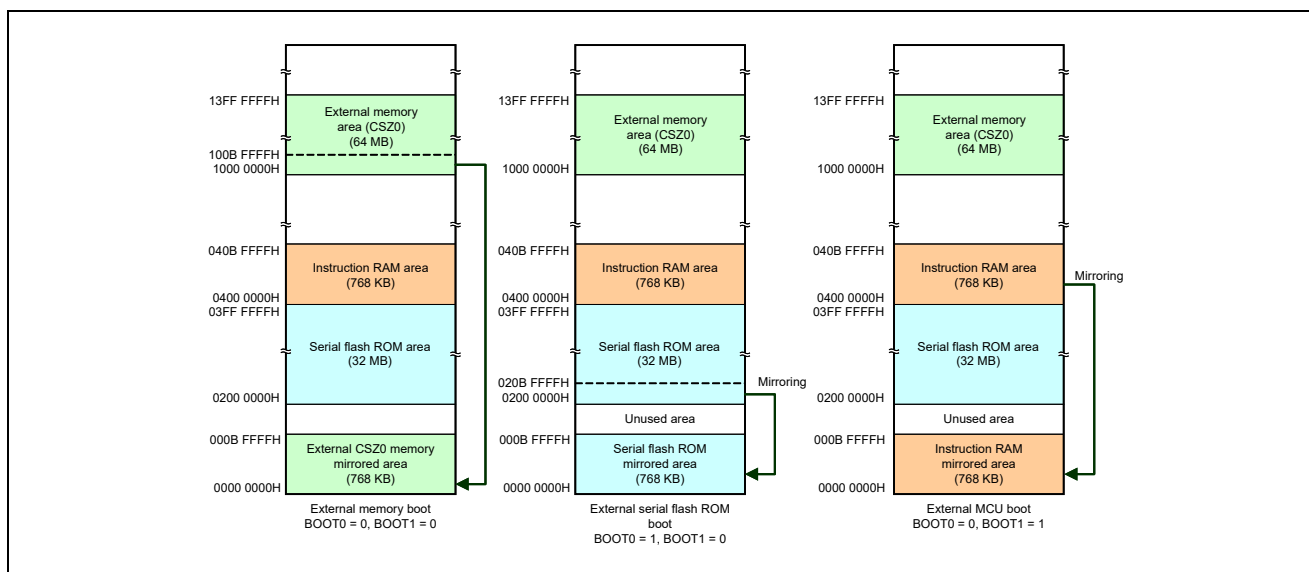
This mode is for directly downloading a program from a debugger to the instruction RAM to run during software development.

5.2 Initializing the Internal RAM

An R-IN32M3 incorporates a large-capacity instruction RAM, data RAM, and buffer 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 PLL lock-in wait time. This significantly reduces the time required by the program to initialize the internal RAM.

5.3 Memory Map in Each Boot Mode

In an R-IN32M3, the memory map from addresses 0000_0000H to 000B_FFFFH differs depending on the selected boot mode.



5.4 Booting Sequence

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

5.4.1 When Booting from an 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 768 KB 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. Perform this step also when a reset is executed by using the SYSRESET register.

(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 the main routine (regular operation)

Fetching the program from the instruction RAM starts.

5.4.2 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 768 KB 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. Perform this step also when a reset is executed by using the SYSRESET register.

(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 the main routine (regular operation)

Fetching the program from the instruction RAM starts.

5.4.3 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 an R-IN32M3 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 an R-IN32M3 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.

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

The hardware real-time OS supports 30 system calls for elements such as events, semaphores, and mailboxes.

6.1 Outline of Features

- μ ITRON-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
- Internal DMA controller
- Buffer allocator
- Header EnDec
- Gigabit Ethernet MAC (with built-in MAC DMAC)

Remark: The hardware real-time OS can be controlled by using the μ ITRON system calls provided by the sample driver. For how to use the driver, see the R-IN32M3 Series Programming Manual: OS.

6.2 Semaphores

The semaphores handled by HW-RTOS in an R-IN32M3 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. 128 semaphore identifiers can be handled.

6.3 Events

The event flags handled by HW-RTOS in an R-IN32M3 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.

6.4 Mailboxes

The mailboxes handled by HW-RTOS in an R-IN32M3 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.

6.5 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-IN32M3 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.

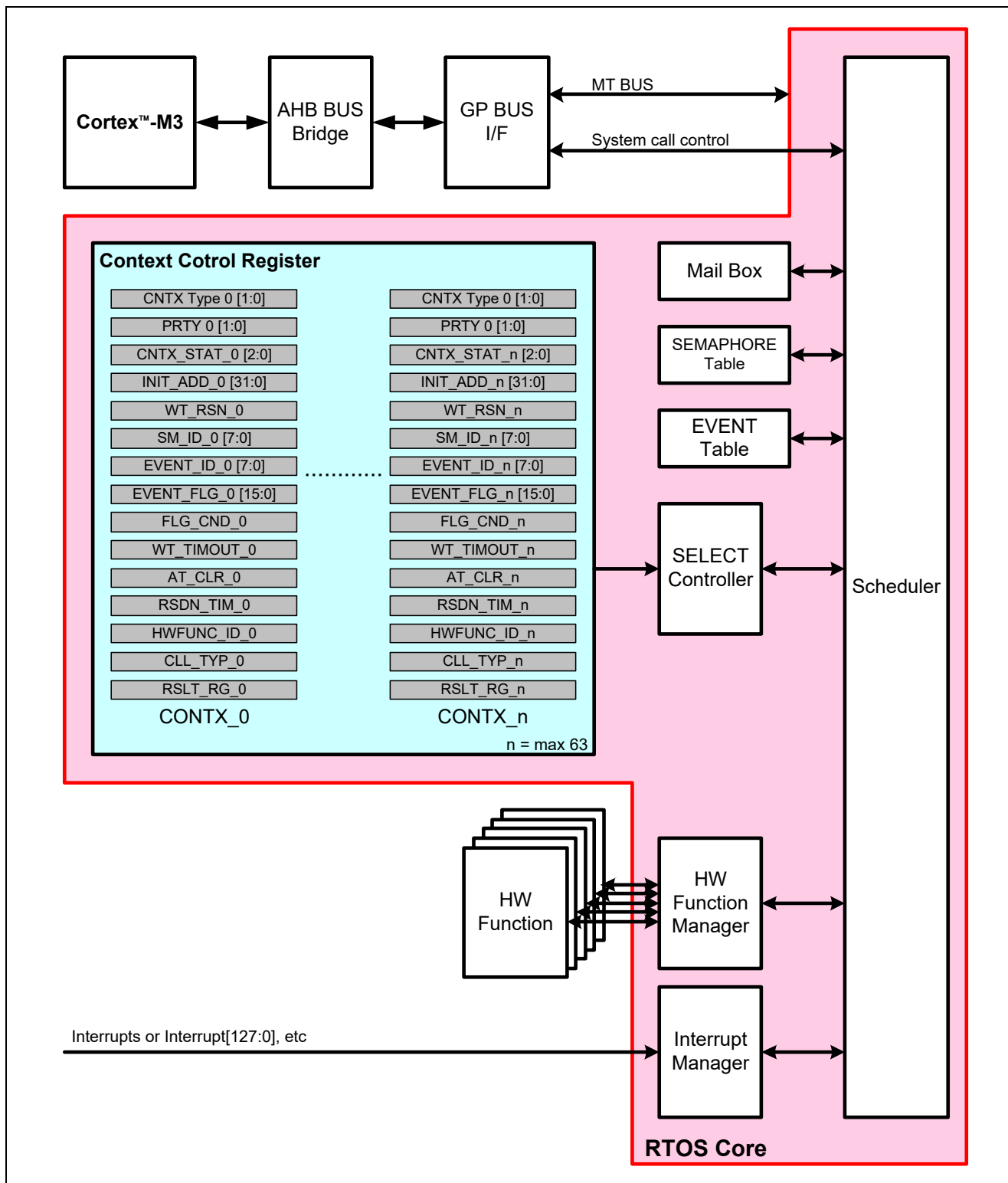


Figure 6.1 Overall Structure of Hardware Real-Time OS (HW-RTOS)

7. Gigabit Ethernet MAC

This section explains the Ethernet MAC module in products of the R-IN32M3 series. The Ethernet switch is explained in the next section.

Please refer to "R-IN32M3-EC User's Manual" and "R-IN32M3-CL User's Manual" respectively about the EtherCAT slave function of "R-IN32M3-EC" and the CC-Link IE Field function of "R-IN32M3-CL".

7.1 Overview

Products of the R-IN32M3 series can also be used for a general Gigabit Ethernet interface as well as for an Ethernet interface compliant with the industrial Ethernet protocol (EtherCAT, CC-Link IE Field).

Switching of the Ethernet interface and mode settings are controlled by using a register.

Caution: The R-IN32M3-EC has a 10/100M Ethernet PHY layer. Accordingly, it cannot operate at 1 Gbps.

7.1.1 Ethernet Interface Architecture

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

The structure differs between "R-IN32M3-EC" and "R-IN32M3-CL". The control registers in the figure are described later.

For details of the Ethernet PHY-related registers, refer the R-IN32M3-EC User's Manual.

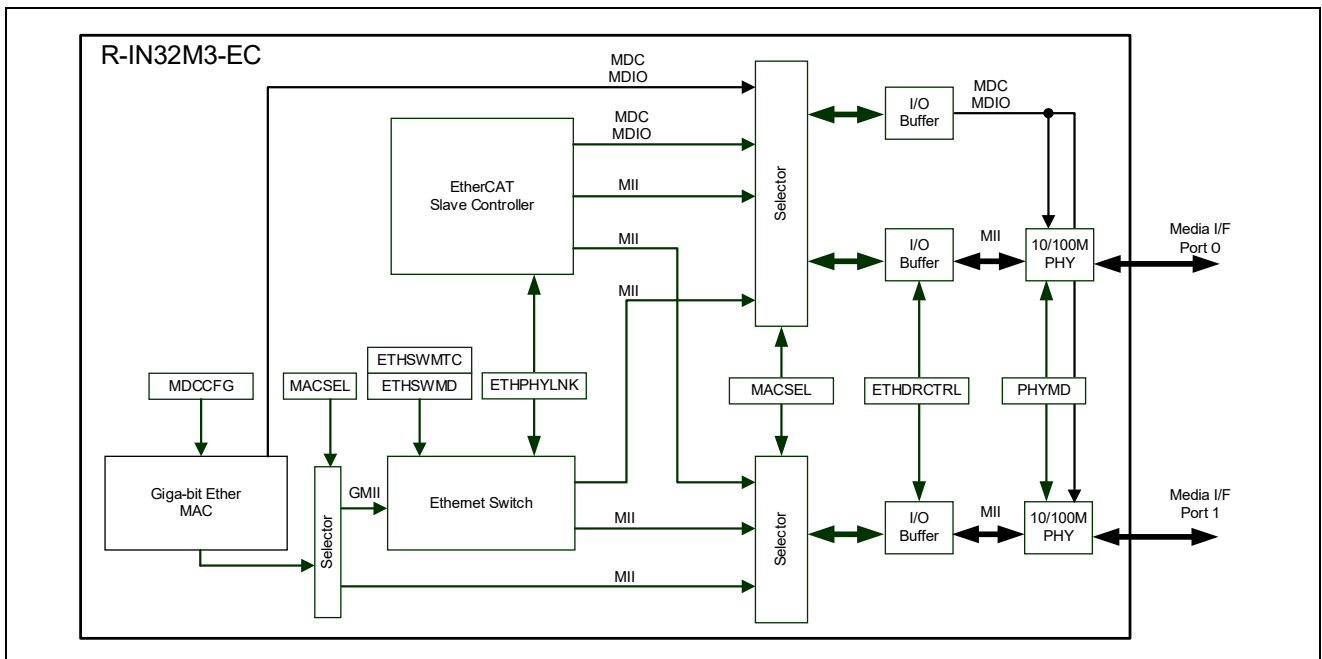


Figure 7.1 Ethernet Interface Peripheral Architecture (R-IN32M3-EC)

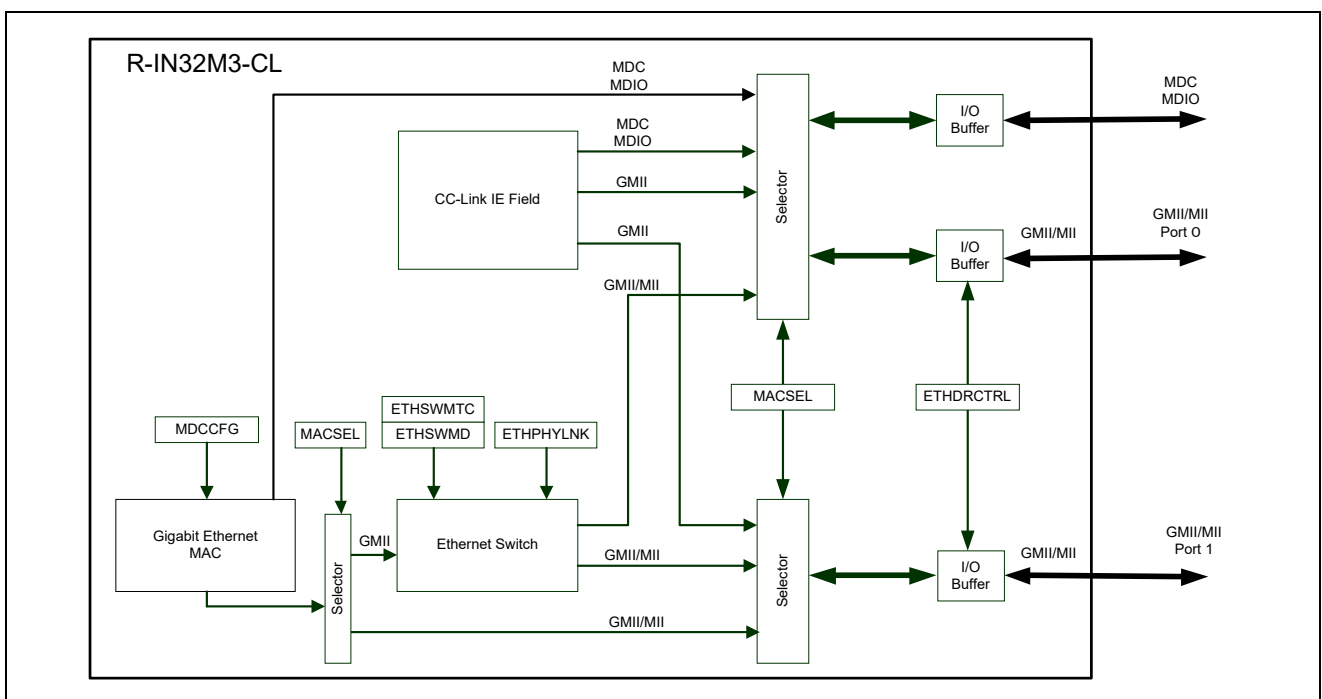


Figure 7.2 Ethernet Interface Peripheral Architecture (R-IN32M3-CL)

7.2 Features

The functions of the Ethernet interface of the R-IN32M3 series (when the Ethernet switch is in use) are given below.

- 1 Port
- Support for IEEE802.3
- 10BASE, 100BASE
- 1000BASE ^{Note}
- 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 ^{Note}

Refer to the R-IN32M3-EC User's Manual for the functions of Ethernet PHY in the R-IN32M3-EC.

Note: For the R-IN32M3-CL only

7.3 Control Registers

7.3.1 List of Registers

(1) Ethernet interface select register

Register Name	Symbol	Address
MAC select register	MACSEL	BASE + 0600H

(2) Ethernet interface mode registers

Register Name	Symbol	Address
MDC clock select register	MDCCFG	BASE + 0604H
Ethernet interface buffer switching register	ETHDRCTRL	BASE + 0610H

(3) Gigabit Ethernet MAC control registers

(1/2)

Register Name	Symbol	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 1A	GMAC_ADR1A	4009 0100H
MAC address register 1B	GMAC_ADR1B	4009 0104H
MAC address register 2A	GMAC_ADR2A	4009 0108H
MAC address register 2B	GMAC_ADR2B	4009 010cH
MAC address register 3A	GMAC_ADR3A	4009 0110H
MAC address register 3B	GMAC_ADR3B	4009 0114H
MAC address register 4A	GMAC_ADR4A	4009 0118H
MAC address register 4B	GMAC_ADR4B	4009 011cH
MAC address register 5A	GMAC_ADR5A	4009 0120H
MAC address register 5B	GMAC_ADR5B	4009 0124H
MAC address register 6A	GMAC_ADR6A	4009 0128H
MAC address register 6B	GMAC_ADR6B	4009 012CH

(2/2)

Register Name	Symbol	Address
MAC address register 7A	GMAC_ADR7A	4009 0130H
MAC address register 7B	GMAC_ADR7B	4009 0134H
MAC address register 8A	GMAC_ADR8A	4009 0138H
MAC address register 8B	GMAC_ADR8B	4009 013CH
MAC address register 9A	GMAC_ADR9A	4009 0140H
MAC address register 9B	GMAC_ADR9B	4009 0144H
MAC address register 10A	GMAC_ADR10A	4009 0148H
MAC address register 10B	GMAC_ADR10B	4009 014CH
MAC address register 11A	GMAC_ADR11A	4009 0150H
MAC address register 11B	GMAC_ADR11B	4009 0154H
MAC address register 12A	GMAC_ADR12A	4009 0158H
MAC address register 12B	GMAC_ADR12B	4009 015cH
MAC address register 13A	GMAC_ADR13A	4009 0160H
MAC address register 13B	GMAC_ADR13B	4009 0164H
MAC address register 14A	GMAC_ADR14A	4009 0168H
MAC address register 14B	GMAC_ADR14B	4009 016CH
MAC address register 15A	GMAC_ADR15A	4009 0170H
MAC address register 15B	GMAC_ADR15B	4009 0174H
MAC address register 16A	GMAC_ADR16A	4009 0178H
MAC address register 16B	GMAC_ADR16B	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	Symbol	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

7.3.2 Ethernet Interface Select Register

7.3.2.1 MAC Select Register (MACSEL)

This register selects the function of the Ethernet interface.

Using a general-purpose Ethernet interface requires setting of this register. After changing the register value, reset the PHY, and then set the MAC.

- This register can be read or written in 32- or 16-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 how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.

2. When changing the value of this register, do so while the Ethernet MAC 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		
MACSEL	<table border="1" style="width: 100%; height: 40px; border-collapse: collapse;"> <tr> <td style="width: 100%;">0 0</td> </tr> </table>	0 0	BASE+ 0600H	
0 0				
		Initial value		
		<small>Note1</small>		
R/W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 100%;">0 0</td> </tr> </table>	0 0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 100%;">R/W R/W R/W</td> </tr> </table>	R/W R/W R/W
0 0				
R/W R/W R/W				

Bit Position	Bit Name	Description																		
2 to 0	MAC2 to MAC0	Select the function of the MAC interface to be used. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 25%;">MAC2 to MAC0</th> <th style="width: 25%;">Ethernet Port 0</th> <th style="width: 50%;">Ethernet Port 1</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>General-purpose Ethernet port 0 (with Ethernet switch)</td> <td>General-purpose Ethernet port 1 (with Ethernet switch)</td> </tr> <tr> <td>001 <small>Note2</small></td> <td>EtherCAT slave port 0</td> <td>EtherCAT slave port 1</td> </tr> <tr> <td>011 <small>Note3</small></td> <td>N/A</td> <td>General-purpose Ethernet port 1 (without Ethernet switch)</td> </tr> <tr> <td>100 <small>Note4</small></td> <td>CC-Link IE Field port 0</td> <td>CC-Link IE Field port 1</td> </tr> <tr> <td>Others</td> <td colspan="2">Setting prohibited</td> </tr> </tbody> </table>	MAC2 to MAC0	Ethernet Port 0	Ethernet Port 1	000	General-purpose Ethernet port 0 (with Ethernet switch)	General-purpose Ethernet port 1 (with Ethernet switch)	001 <small>Note2</small>	EtherCAT slave port 0	EtherCAT slave port 1	011 <small>Note3</small>	N/A	General-purpose Ethernet port 1 (without Ethernet switch)	100 <small>Note4</small>	CC-Link IE Field port 0	CC-Link IE Field port 1	Others	Setting prohibited	
MAC2 to MAC0	Ethernet Port 0	Ethernet Port 1																		
000	General-purpose Ethernet port 0 (with Ethernet switch)	General-purpose Ethernet port 1 (with Ethernet switch)																		
001 <small>Note2</small>	EtherCAT slave port 0	EtherCAT slave port 1																		
011 <small>Note3</small>	N/A	General-purpose Ethernet port 1 (without Ethernet switch)																		
100 <small>Note4</small>	CC-Link IE Field port 0	CC-Link IE Field port 1																		
Others	Setting prohibited																			

Notes 1. The initial value depends on the product used.
R-IN32M3-EC: 0000 0001H
R-IN32M3-CL: 0000 0004H

2. This is only supported in the R-IN32M3-EC.

3. When not using the Ethernet switch, Ethernet port 0 cannot be used.

4. This is only supported in the R-IN32M3-CL.

7.3.3 Ethernet Interface Mode Registers

7.3.3.1 MDC Clock Select Register (MDCCFG)

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

- This register can be read or written in 32- or 16-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 how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.

2. The setting of this register is only effective while the value of the MACSEL register is "0000 0000H" or "0000 0003H". When EtherCAT or the CC-Link IE Field is selected, MDC is controlled according to the selected function.

MDCCFG	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
	0 0	BASE+ 0604H
	0 0	Initial value
	0 0	0000 0000H
R/W	0 0	R/W R/W

Bit Position	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

7.3.3.2 Ethernet Interface Buffer Switching Register (ETHDRCTRL)

This register changes the output buffers of the Ethernet interface in the R-IN32M3-CL.

With the R-IN32M3-EC, do not change the initial value (MII buffer).

- This register can be read or written in 32- or 16-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 how to release protection, see the description of the system protection 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
ETHDRCTRL	0 0	BASE+ 0610H
	ETHIOL11 ETHIOL10 ETHIOL01 ETHIOL00	Initial value 0000 0005H
R/W	0 0	
		R/W R/W R/W R/W

Bit Position	Bit Name	Description
3, 2	ETHIOL11, ETHIOL10	Change the output buffers of Ethernet interface 1 (ETH1). 01: 3.3V MII 10: 3.3V GMII Others: Setting prohibited
1, 0	ETHIOL01, ETHIOL00	Change the output buffers of Ethernet interface 0 (ETH0). 01: 3.3V MII 10: 3.3V GMII Others: Setting prohibited

7.3.4 Gigabit Ethernet MAC Control Register

7.3.4.1 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 the RWDV bit, PHY address to the PHYADDR4 to 0 bits, PHY register address to the REGADDR4 to 0 bits, and write data to the 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 the RWDV bit, PHY address to the PHYADDR4 to 0 bits, and PHY register address to the 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	Address
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	4009 00A0H Initial value 0400 0000H <small>Note</small>
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	R/W

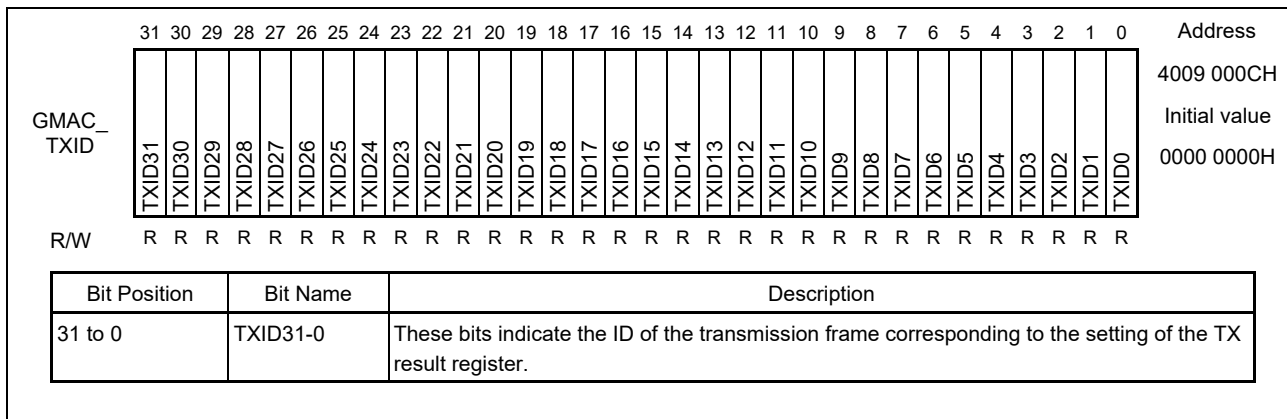
Bit Position	Bit Name	Description
26	RWDV	Read/write operation starts by writing the following value to this bit. Set other associated bits at the same time. 1: Write operation starts. 0: Read operation starts. The state of the operation can be confirmed by reading the following value from this bit.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

7.3.4.2 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.

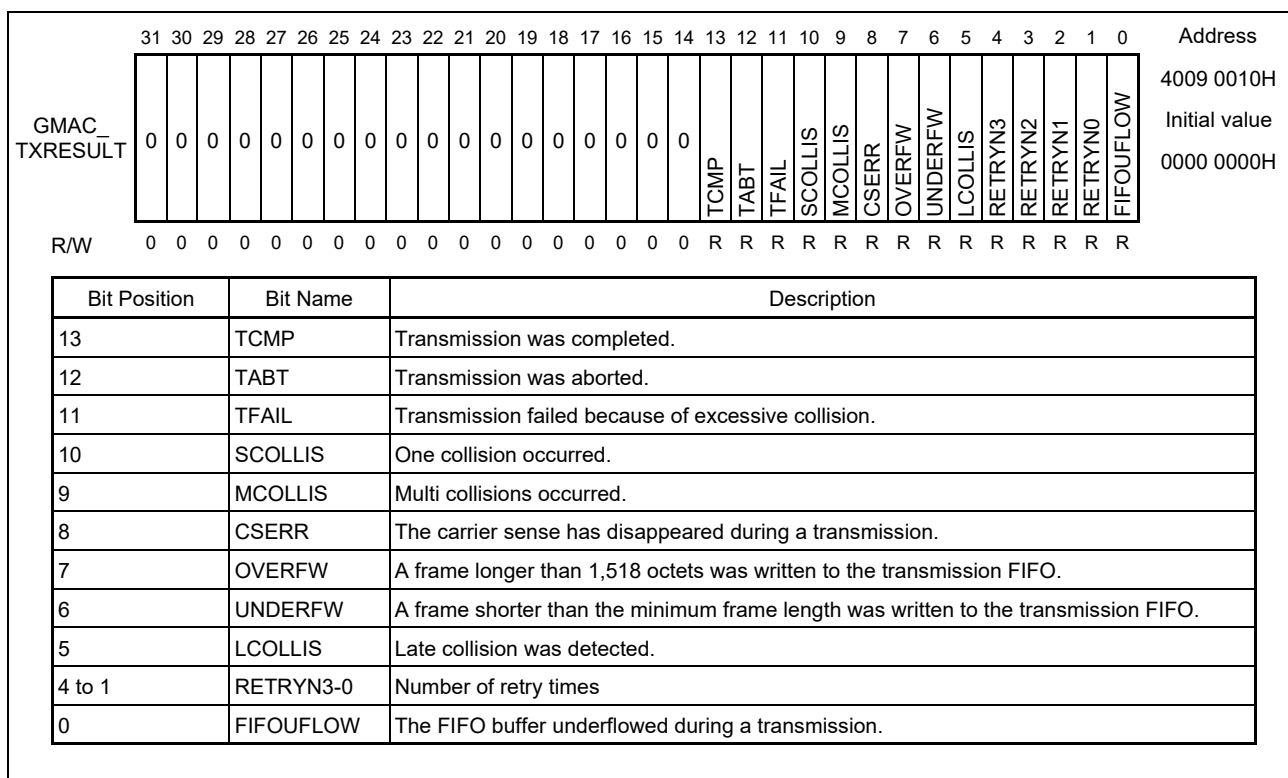


7.3.4.3 TX Result Register (GMAC_TXRESULT)

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.



7.3.4.4 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	Address
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	4009 0020H
																		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	

Bit Position	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.

7.3.4.5 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	Address
GMAC_RXMODE	AFILLTEREN	MFILLTEREN	SFRXFIFO	RAMASKEN	0	0	0	0	0	0	0	0	0	0	0	0	REMPH1	REMPH0	RFULLTH1	RFULLTH0	RRTTH2	RRTTH1	RRTTH0	0	0	0	0	0	0	0	0	0	4009 0024H
																		Initial value 2000 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	

Bit Position	Bit Name	Description
31	AFILLTEREN	Address Filtering Enable 1: Enable address filtering. ^{Note} 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 = 1 to 16). 0: Acquire all frames with multicast addresses.

Bit Position	Bit Name	Description
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.
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 = 1 to 16) 0: Disable the above function
15, 14	REMPH1-0	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 to 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.

7.3.4.6 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.

	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_TXMODE	RTRANSDEN	LPTXEN	SF	SPTXEN	RTRANSLC	0	0	0	0	0	0	0	0	0	0	0	0	0	TEMPH2	TEMPH1	TEMPH0	TFULLTH1	TFULLTH0	0	TRBMODE1	TRBMODE0	0	0	0	0	0	0	0	0	Address 4009 0028H Initial value 0000 0000H
R/W	R	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W		

Bit Position	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. <small>Note 1</small> 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. <small>Note 2</small>
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 7.5.5, Transmitting Data in Cut-Through Mode.

Bit Position	Bit Name	Description
13 to 11	TEMPH2-0	<p>Transmit Almost Empty Threshold</p> <p>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.</p> <p>000: 4 words 001: 8 words 010: 16 words 011: 32 words 100: 64 words 101: 128 words 110: 256 words 111: 512 words</p>
10, 9	TFULLTH1-0	<p>Transmit Almost Full Threshold</p> <p>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.</p> <p>00: 4 words 01: 8 words 10: 16 words 11: 32 words</p>
7, 6	TRBMODE1-0	<p>Transmission Result Buffer Mode</p> <p>Control how to write the transmission result to the GMAC_TXRESULT register.</p> <p>00: Always writing 01: Writing only proceeds when an error occurs. 10: Writing does not proceed 11: Setting prohibited</p>

7.3.4.7 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

	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_RESET	ALLRST	0	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	0	0	4009 0030H Initial value 0000 0000H
RW	W	0	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	0	0	

Bit Position	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.

7.3.4.8 Pause Packet Data Register (GMAC_PAUSEn)

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

GMAC_PAUSE1	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
	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	4009 0080H
	Initial value	0000 0000H
R/W	R/W/R/W	
Bit Position	Bit Name	Description
31 to 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	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
	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	4009 0084H
	Initial value	0000 0000H
R/W	R/W/R/W	
Bit Position	Bit Name	Description
31 to 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	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
	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	4009 0088H
	Initial value	0000 0000H
R/W	R/W/R/W	
Bit Position	Bit Name	Description
31 to 0	PPDATA31-0	The 12th to 9th bytes of a pause packet for transmission are designated. Data are transmitted in order from the LSB.

	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_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	4009 008CH
																														Initial value			
																													0000 0000H				
R/W																													R/RW				

Bit Position	Bit Name	Description
31 to 0	PPDATA31-0	The 16th to 13th bytes of a pause packet for transmission are designated. Data are transmitted in order from the LSB.

	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_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	40090090H
																														Initial value			
																													0000 0000H				
R/W																													R/RW				

Bit Position	Bit Name	Description
31 to 0	PPDATA31-0	The 20th to 17th bytes of a pause packet for transmission are designated. Data are transmitted in order from the LSB.

7.3.4.9 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.

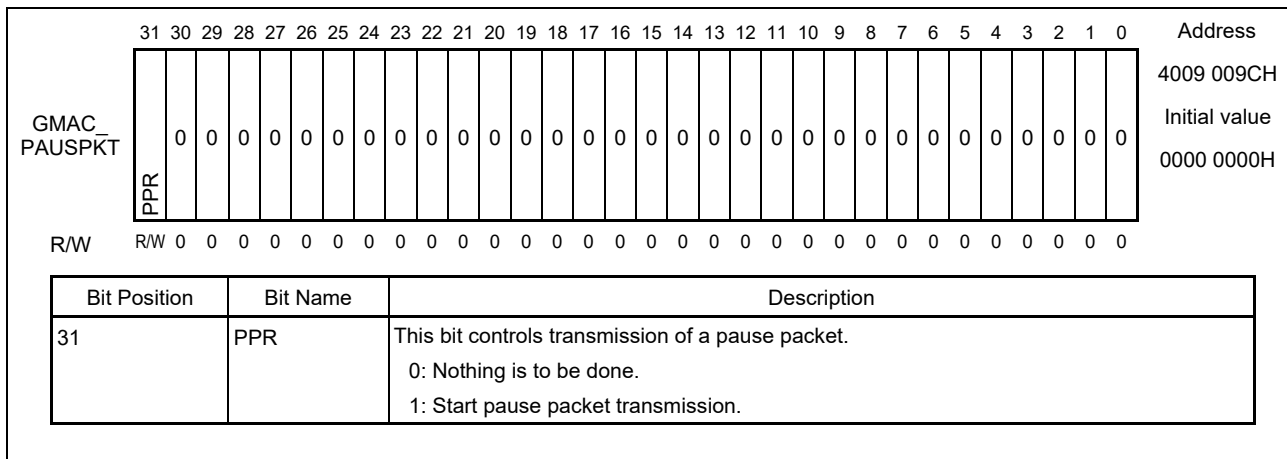
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
GMAC_FLWCTL	PPRXEN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40090098H
																														Initial value			
																													0000 0000H				
R/W																													R/W				

Bit Position	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.

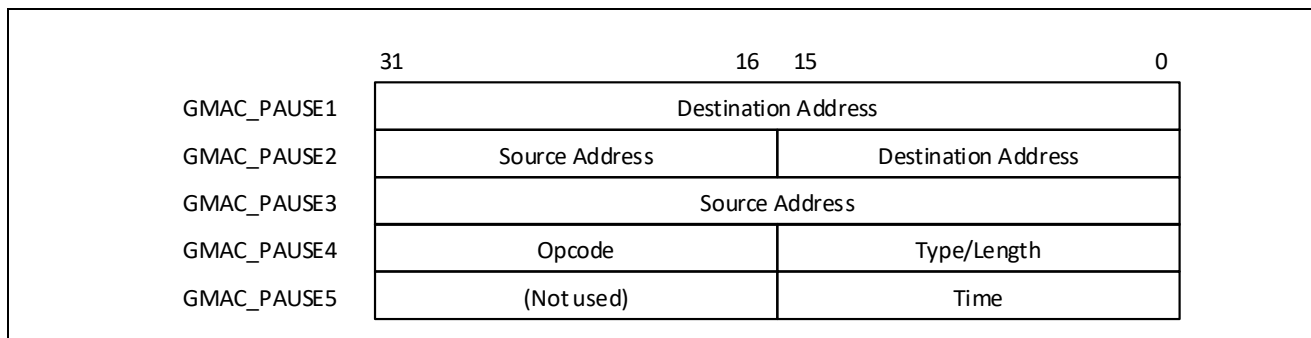
7.3.4.10 Pause Packet Register (GMAC_PAUSPKT)

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.



The transmission packet format is shown below.



7.3.4.11 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 = 1, 2, ..., 16).

GMAC_ADRnA

	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
	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																											4009 0100H +8H*n Initial value 0000 0000H					
R/W	R/W/R/W																																

Bit Position	Bit Name	Description
31 to 24	MADDR4B7-0	Indicate the 4th byte from the beginning of the MAC address to be acquired.
23 to 16	MADDR3B7-0	Indicate the 3rd byte from the beginning of the MAC address to be acquired.
15 to 8	MADDR2B7-0	Indicate the 2nd byte from the beginning of the MAC address to be acquired.
7 to 0	MADDR1B7-0	Indicate the 1st byte of the MAC address to be acquired.

GMAC_ADRnB

	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
	0 0 0 0 0 0 0 0 0 BITMSK7 BITMSK6 BITMSK5 BITMSK4 BITMSK3 BITMSK2 BITMSK1 BITMSK0 MADDR6B7 MADDR6B6 MADDR6B5 MADDR6B4 MADDR&B3 MADDR6B2 MADDR6B1 MADDR6B0 MADDR5B7 MADDR5B6 MADDR5B5 MADDR5B4 MADDR5B3 MADDR5B2 MADDR5B1 MADDR5B0																											4009 0104H +8H*n Initial value 00FF 0000H					
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																																

Bit Position	Bit Name	Description
23 to 16	BITMSK7-0	<p>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.</p> <p>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.</p>
15 to 8	MADDR6B7-0	Indicate the 1st byte to 6th byte of the MAC address to be acquired.
7 to 0	MADDR5B7-0	Indicate the 1st byte to 5th byte of the MAC address to be acquired.

7.3.4.12 RX FIFO Status Register (GMAC_RXFIFO)

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

GMAC_RXFIFO	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 4009 0200H Initial value 4000 0000H
	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		

Bit Position	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 to 17	RSW11-0	Stored Words in RX FIFO Indicate the number of words of the data in the RX FIFO buffer.

7.3.4.13 TX FIFO Status Register (GMAC_TXFIFO)

This register is a status register which indicates the state of the transmission 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	Address
GMAC_TXFIFO	TFULL	TEMP	TSTATUS2	TSTATUS1	TSTATUS0	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	4009 0204H Initial value 6000 0000H	
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			

Bit Position	Bit Name	Descriptor
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 to 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 to 24	TRBFR2-0	The number of frames in the transmission result buffer.

7.3.4.14 TCPIPACC Register (GMAC_ACC)

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

GMAC_ACC	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		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4009 0208H	
																																			Initial value
																																			0000 0003H
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	0	R/W/R/W/R/W	

Bit Position	Bit Name	Description
2	RTCPACC	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	TTCPDEN	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	RTCPDEN	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.

7.3.4.15 RX MAC Enable Register (GMAC_RXMAC_ENA)

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

GMAC_RXMAC_ENA	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		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4009 0220H	
																																			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	0	R/W		

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

7.3.4.16 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.

	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_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	0	4009 0224H
																																		Initial value 0000 0000H
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 Position	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.

7.3.4.17 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.

	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_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	4009 0228H	
																																		Initial value 0000 080FH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	Descriptor
31 to 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 to 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 effective while the MACSEL register value is 0000 0003H.

7.3.4.18 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 MACDMA 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).

	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
BUFID	NOEMP	0	0	VALID	WORD11	WORD10	WORD9	WORD8	WORD7	WORD6	WORD5	WORD4	WORD3	WORD2	WORD1	WORD0	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	4009 1100H Initial value 0000 0000H
R/W	R	0	0	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	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 to 16	WORD11-0	Number of words of received data (including the received MAC information). A word unit is 32 bits.
15 to 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).

7.3.5 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 7.4.1, Hardware Functions.

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

7.3.5.1 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		SYSC15 SYSC14 SYSC13 SYSC12 SYSC11 SYSC10 SYSC9 SYSC8 SYSC7 SYSC6 SYSC5 SYSC4 SYSC3 SYSC2 SYSC1 SYSC0		Address 4008 F000H Initial value -
SYSC	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	R/W	R/W/R/W/R/W/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 to 0	SYSC15-0	Select the hardware function. The following functions are available. <table border="1" style="width: 100%; margin-top: 10px; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">SYSC15-0</th> <th style="width: 70%;">Function</th> </tr> </thead> <tbody> <tr> <td>0x5000</td> <td>Acquires a long buffer.</td> </tr> <tr> <td>0x5006</td> <td>Acquires a short buffer.</td> </tr> <tr> <td>0x5001</td> <td>Releases the whole area of the buffer.</td> </tr> <tr> <td>0x5002</td> <td>Releases the part of the buffer.</td> </tr> <tr> <td>0x5101</td> <td>Enables DMA for the reception MAC.</td> </tr> <tr> <td>0x5102</td> <td>Disables DMA for the reception MAC.</td> </tr> <tr> <td>0x510B</td> <td>Controls interrupts for the reception MACDMAC.</td> </tr> <tr> <td>0x510D</td> <td>Obtains error sources for the reception MACDMAC.</td> </tr> <tr> <td>0x5100</td> <td>Starts transfer by the transmission MACDMAC.</td> </tr> <tr> <td>0x510C</td> <td>Obtains error sources in the transmission MACDMAC.</td> </tr> <tr> <td>0x5211</td> <td>Starts DMA transfer between the buffer RAM and data RAM.</td> </tr> <tr> <td>0x5212</td> <td>Starts replacing data in the buffer RAM or data RAM.</td> </tr> <tr> <td>0x5104</td> <td>Starts DMA transfer between the buffer RAMs.</td> </tr> <tr> <td>0x5114</td> <td>Starts DMA transfer between the buffer RAMs (descriptor method).</td> </tr> <tr> <td>Others</td> <td>Setting prohibited</td> </tr> </tbody> </table>	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
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																																	

7.3.5.2 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 7.4, Functions.

R4	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 4008 F004H				
	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	Initial value -				
R/W	R/W/R/W																																				
Bit Position		Bit Name		Description																																	
31 to 0		R4B31-0		Specify the argument for transfer to the hardware function.																																	

R5	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 4008 F008H				
	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	Initial value -				
R/W	R/W/R/W																																				
Bit Position		Bit Name		Description																																	
31 to 0		R5B31-0		Specify the argument for transfer to the hardware function.																																	

R6	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 4008 F00CH				
	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	Initial value -				
R/W	R/W/R/W																																				
Bit Position		Bit Name		Description																																	
31 to 0		R6B31-0		Specify the argument for transfer to 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	Address					
	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	4008 F010H					
																																		Initial value				
R7																																	-					
R/W	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R					
	<table border="1"> <thead> <tr> <th>Bit Position</th> <th>Bit Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>31 to 0</td> <td>R7B31-0</td> <td>Specify the argument for transfer to the hardware function.</td> </tr> </tbody> </table>																																Bit Position	Bit Name	Description	31 to 0	R7B31-0	Specify the argument for transfer to the hardware function.
Bit Position	Bit Name	Description																																				
31 to 0	R7B31-0	Specify the argument for transfer to the hardware function.																																				

7.3.5.3 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	Address					
	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	4008 F014H					
																																	-					
CMD																																						
R/W	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R	R	W	R					
	<table border="1"> <thead> <tr> <th>Bit Position</th> <th>Bit Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>31 to 0</td> <td>CMDB31-0</td> <td>These bits control the operating mode of the hardware functions. 0x0000 8004: Start the hardware function and hardware real-time OS.</td> </tr> </tbody> </table>																																Bit Position	Bit Name	Description	31 to 0	CMDB31-0	These bits control the operating mode of the hardware functions. 0x0000 8004: Start the hardware function and hardware real-time OS.
Bit Position	Bit Name	Description																																				
31 to 0	CMDB31-0	These bits control the operating mode of the hardware functions. 0x0000 8004: Start the hardware function and hardware real-time OS.																																				

7.3.5.4 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 7.4, Functions.

R0	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				
	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	4008 F020H				
	Initial value																																-				
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	R				
Bit Position		Bit Name		Description																																	
31 to 0		R0B31-0		Hold the returned value from the hardware function.																																	

R1	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				
	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	4008 F024H				
	Initial value																																-				
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		R1B31-0		Hold the value returned from the hardware function.																																	

7.3.5.5 Hardware Function Type Register (CNTX_TYPE0)

This register is for setting the type of hardware function.

CNTX_TYPE0	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				
	CNTX_TYPE031	CNTX_TYPE030	CNTX_TYPE029	CNTX_TYPE028	CNTX_TYPE027	CNTX_TYPE026	CNTX_TYPE025	CNTX_TYPE024	CNTX_TYPE023	CNTX_TYPE022	CNTX_TYPE021	CNTX_TYPE020	CNTX_TYPE019	CNTX_TYPE018	CNTX_TYPE017	CNTX_TYPE016	CNTX_TYPE015	CNTX_TYPE014	CNTX_TYPE013	CNTX_TYPE012	CNTX_TYPE011	CNTX_TYPE010	CNTX_TYPE009	CNTX_TYPE008	CNTX_TYPE007	CNTX_TYPE006	CNTX_TYPE005	CNTX_TYPE004	CNTX_TYPE003	CNTX_TYPE002	CNTX_TYPE001	CNTX_TYPE000	4008 0000H				
	Initial value																																-				
R/W	R	W	R	W	R	W	R	W	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 0		CNTX_TYPE031-0		These bits set the type of hardware function.																																	

7.4 Functions

7.4.1 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.

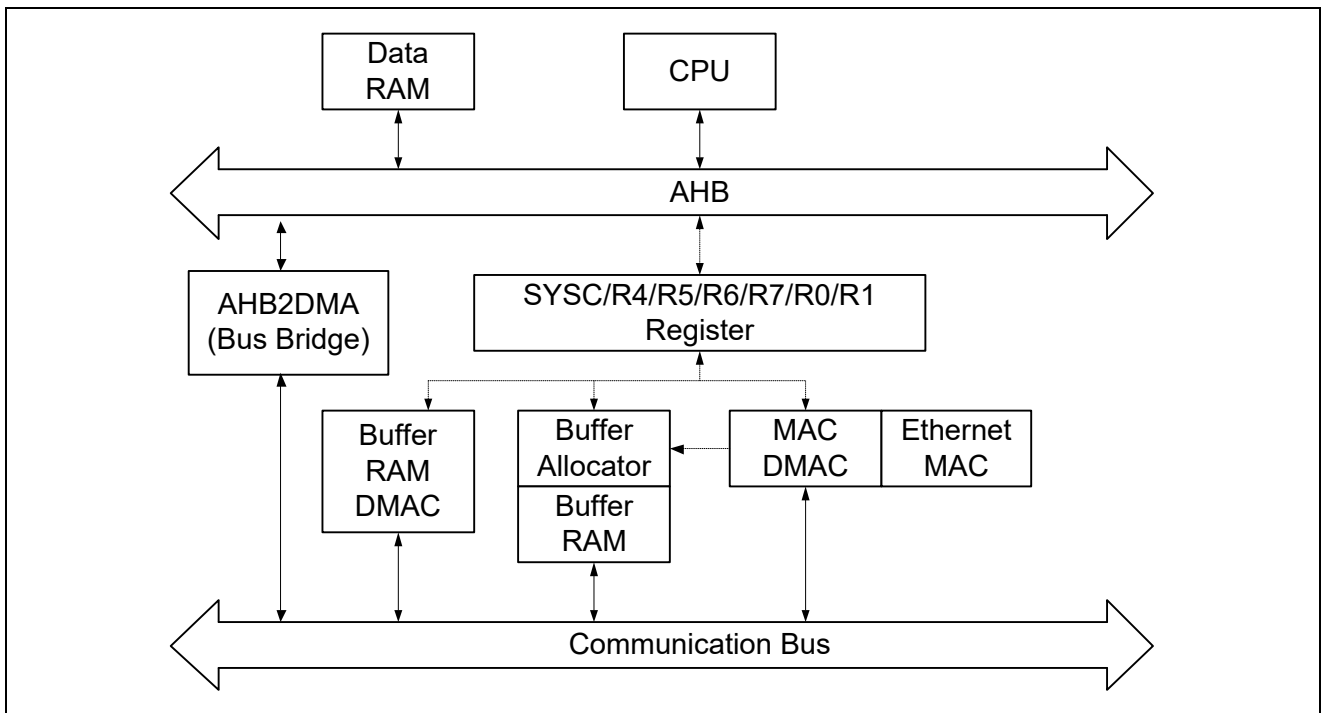


Figure 7.3 Schematic Block Diagram of the Hardware Functions

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.

7.4.1.1 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 (→ 7.3.4.11)
- TX MODE register (→ 7.3.4.6)
- RX MODE register (→ 7.3.4.5)

7.4.1.2 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.

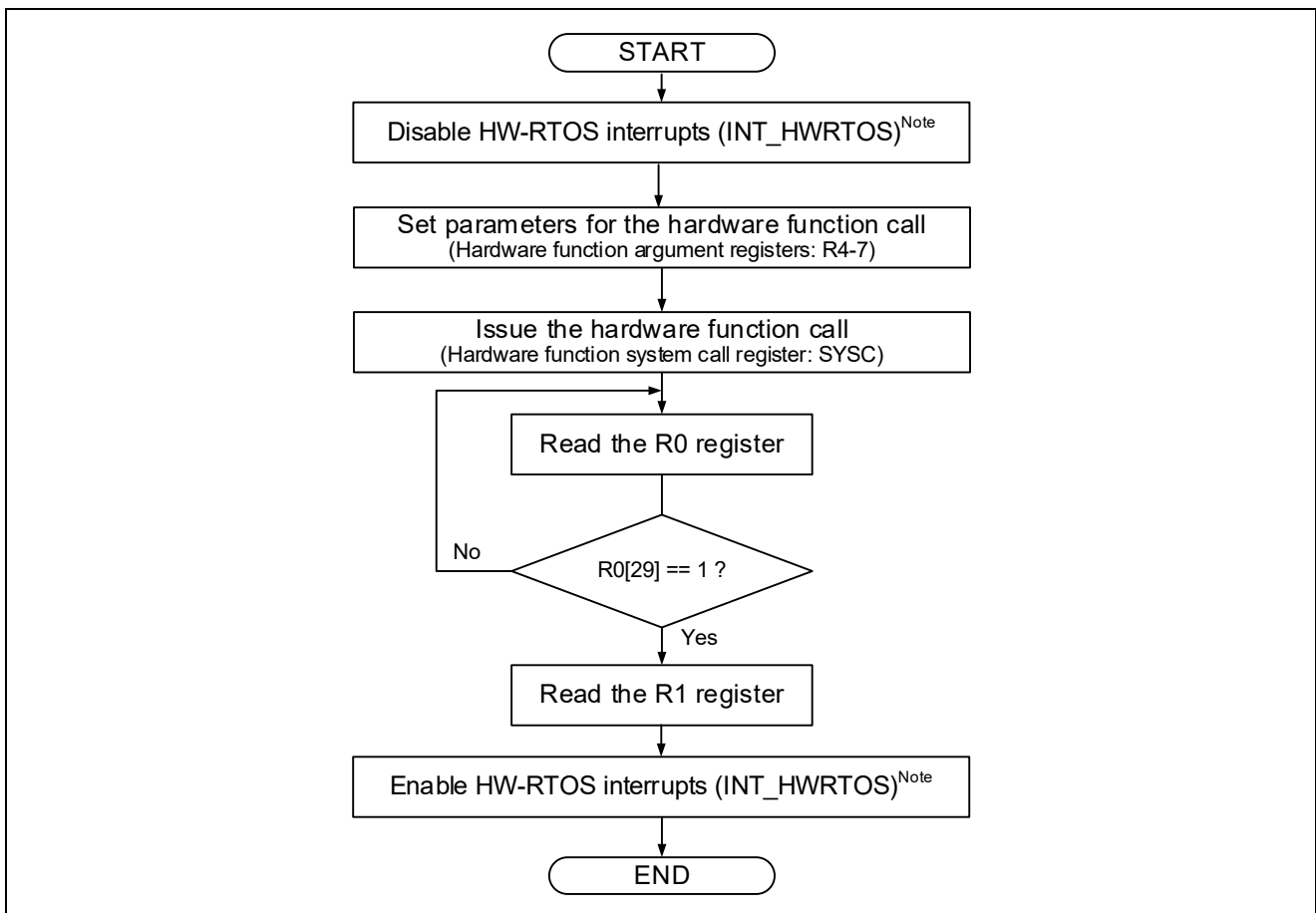


Figure 7.4 Flow of Processing for Issuing the Hardware Function

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

7.4.1.3 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.

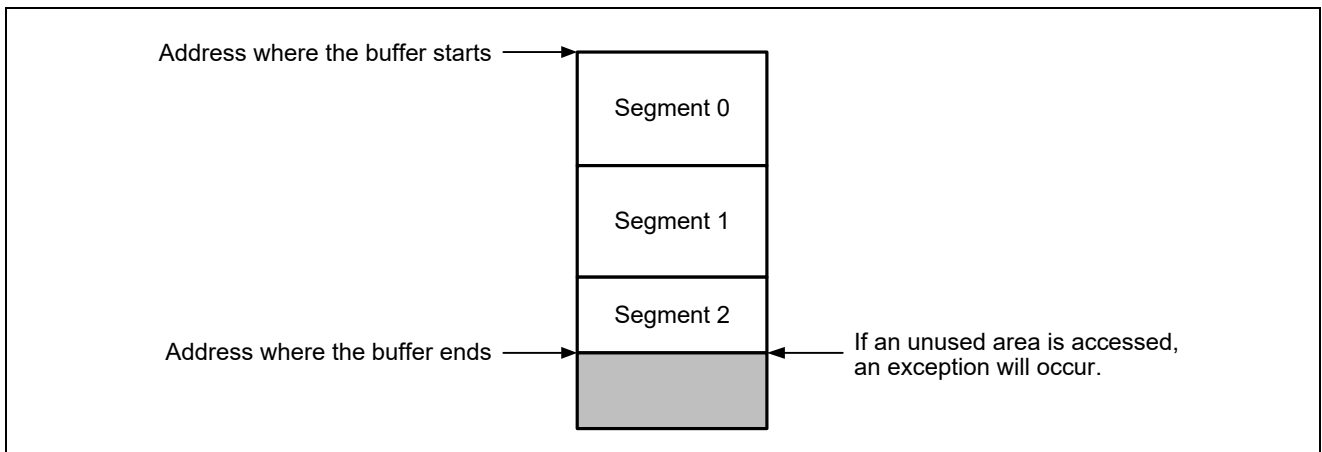


Figure 7.5 Method of Controlling a Buffer

(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.

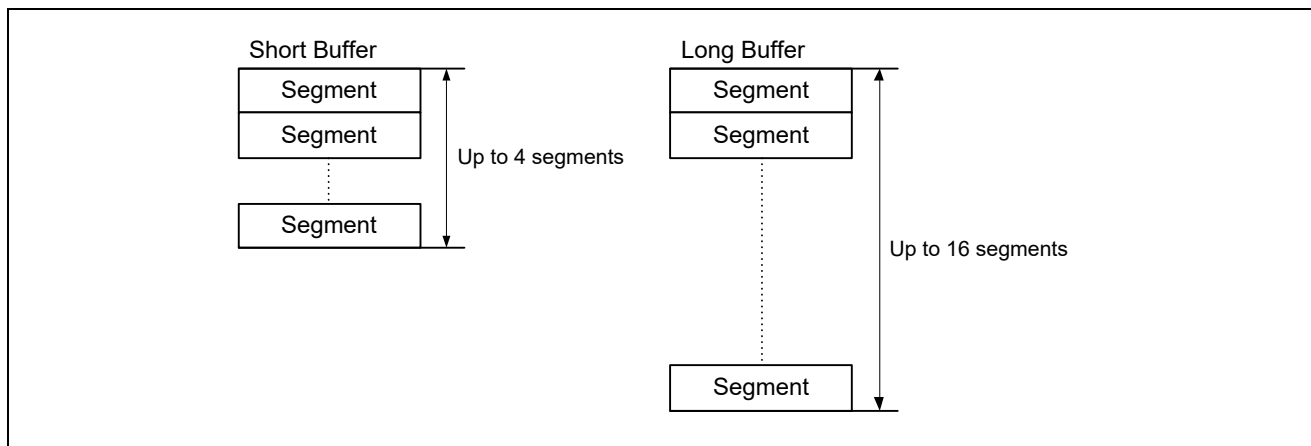


Figure 7.6 Buffer Structure

(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 7.1. 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 7.1 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.

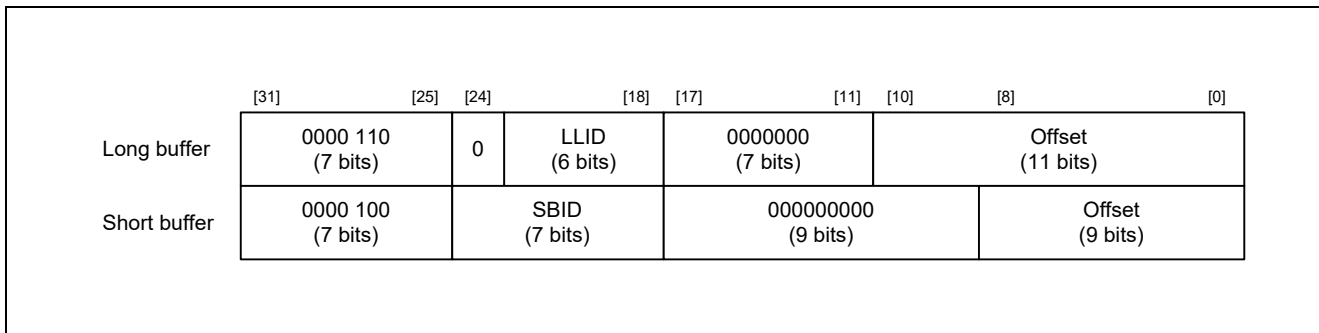


Figure 7.7 Address Structure of Buffers

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 7.2 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 7.3 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 7.4 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 7.5 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

7.4.1.4 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 7.8 is a block diagram of the MACDMA in context and the respective interrupt signals.

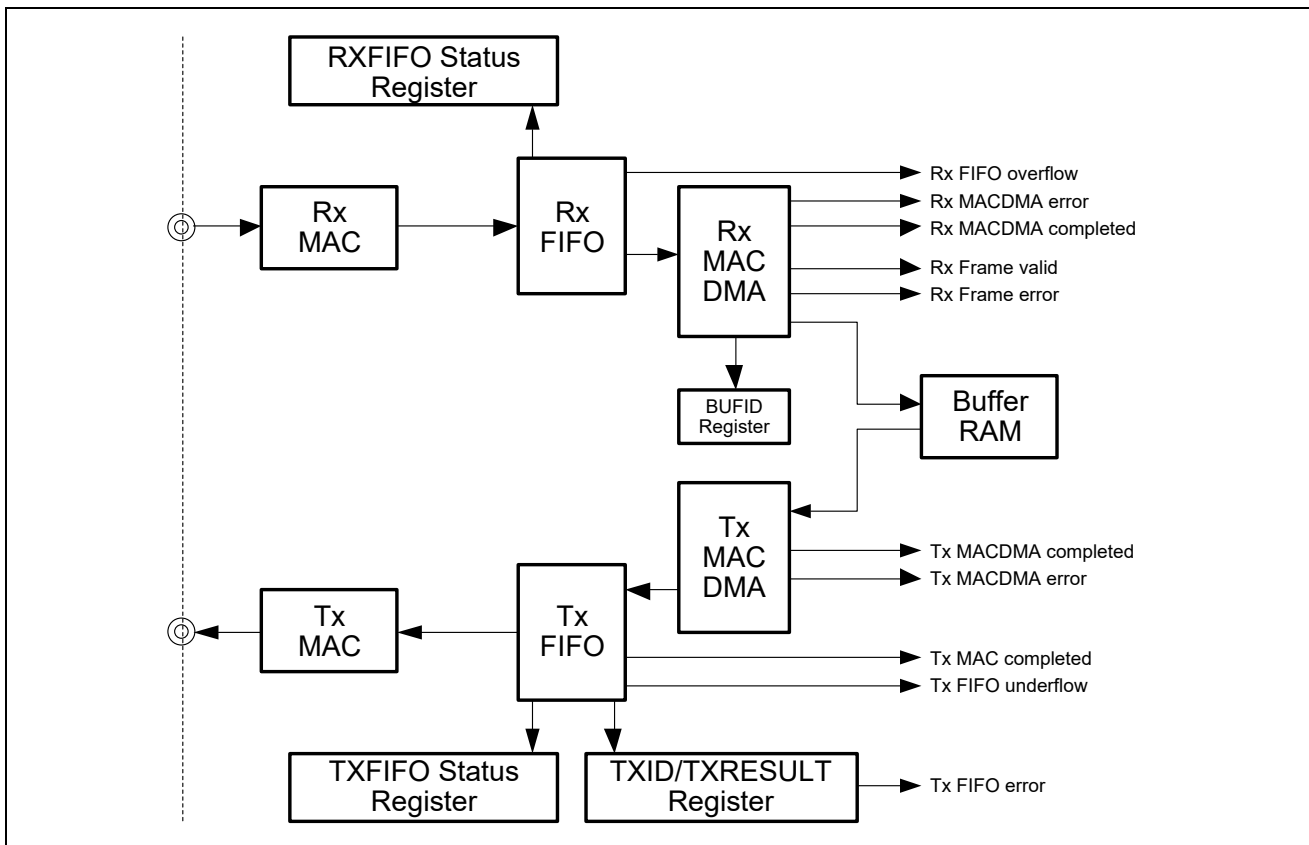


Figure 7.8 Block Diagram of the MACDMA in Context and Interrupt Signals

(2) DMA for the Reception MAC

Figure 7.9 shows an outline of processing by the reception MACDMAC. A hardware function call (HWFUNC_MACDMA_RX_Enable) must be issued to enable operation of the reception MACDMAC. The reception MACDMAC remains active until HWFUNC_MACDMA_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 7.3.4.18, Receive Buffer Information Register (BUFID). The BUFID can be read by the CPU and is capable of holding up to 32 pieces of information.

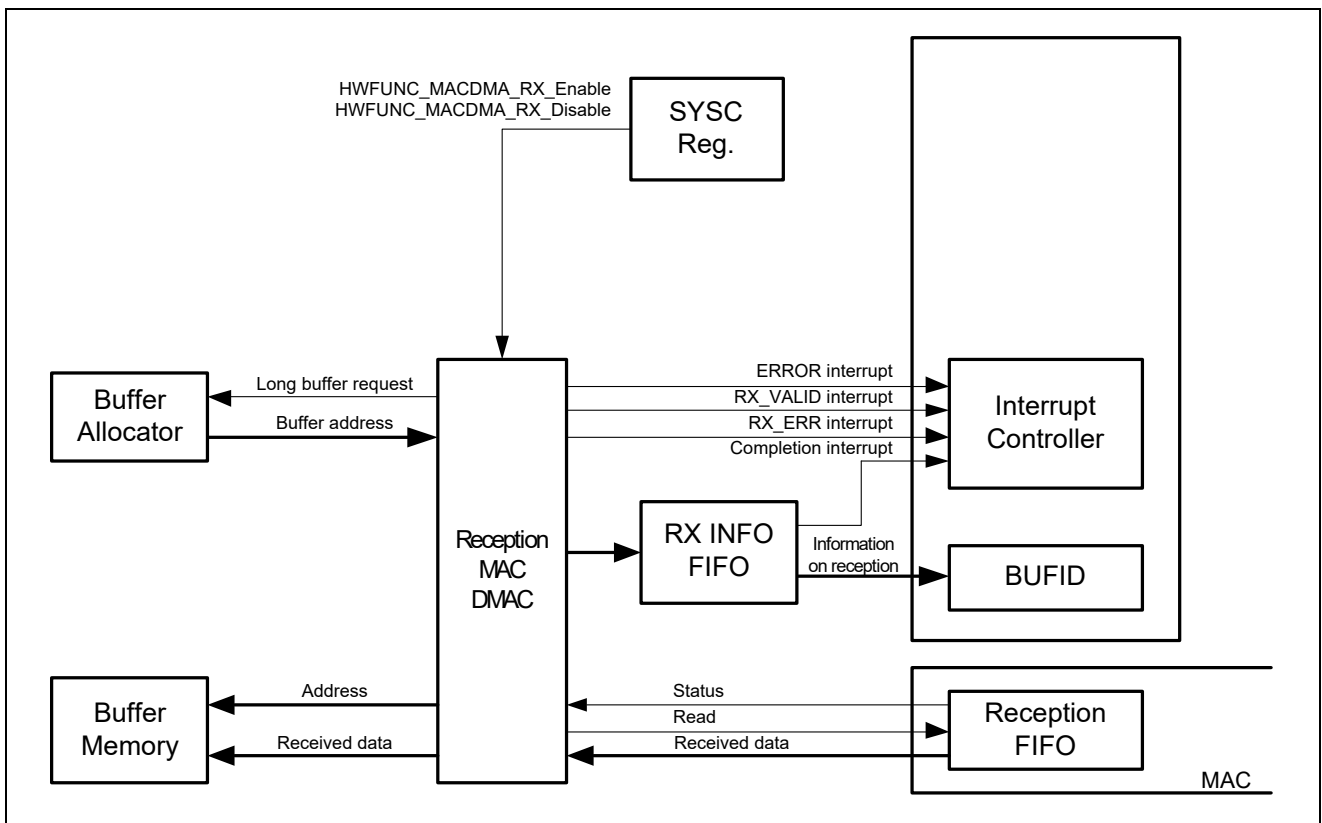


Figure 7.9 Outline of Processing by the Reception MACDMAC

(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 7.4.1.4(2) 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.

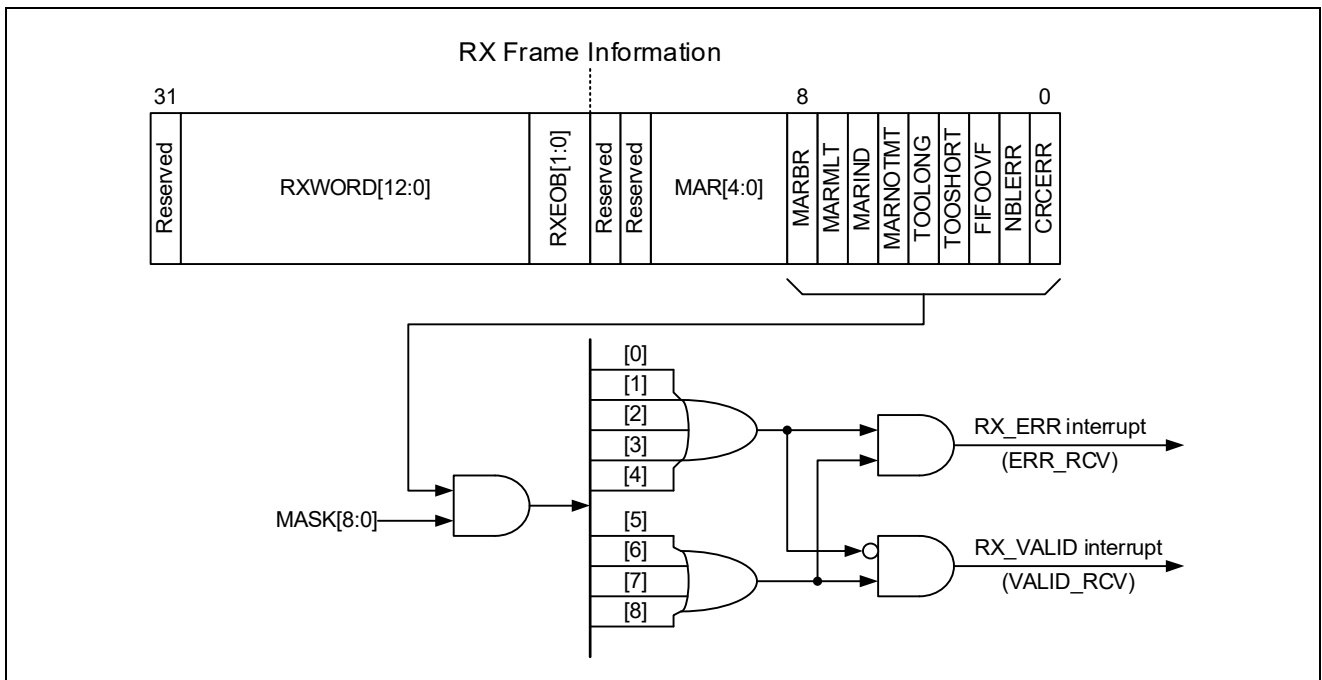


Figure 7.10 Conceptual Diagram of Judging Whether a Received Frame is Valid or Invalid

(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 7.6 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 ^{Note}
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 7.7 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 7.8 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 7.9 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 MACDMA 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 7.11. 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 MACDMA 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.

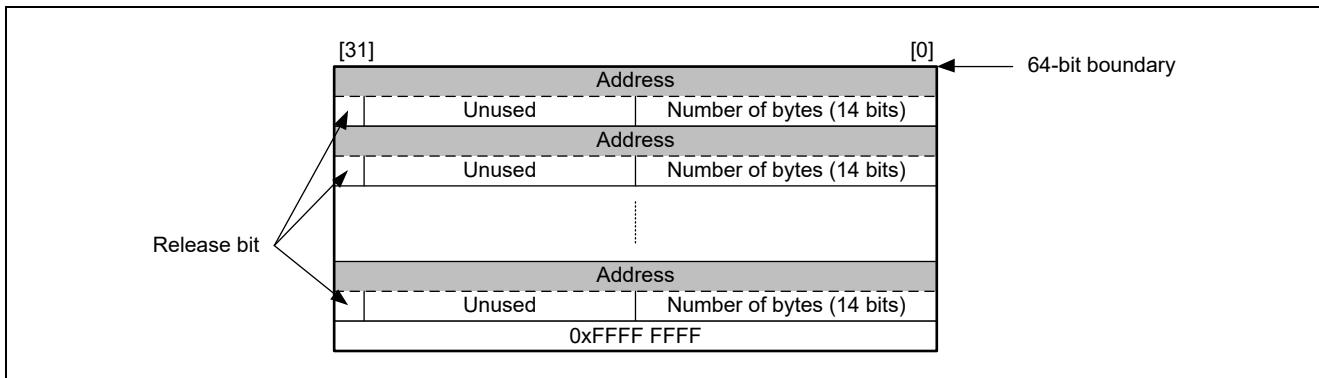


Figure 7.11 Transmission Descriptor

(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 MACDMA 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 7.12 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.

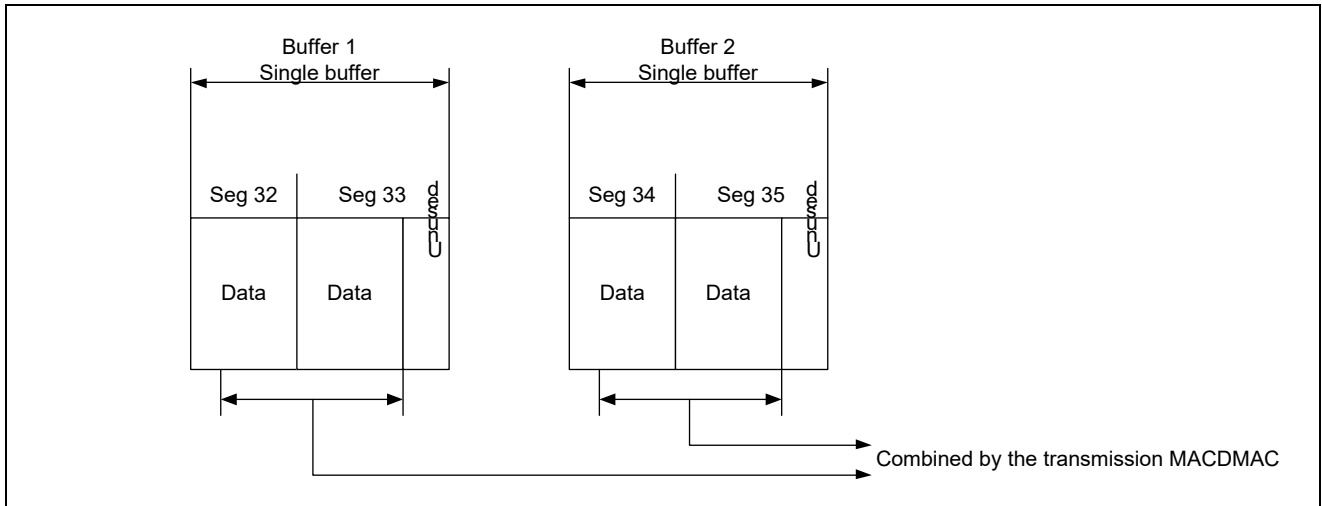


Figure 7.12 Example of Transmission as One Frame by Combining Multiple Buffers

(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 7.10 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 7.11 HWFNC_MACDMA_TX_Errstat

Name	HWFNC_MACDMA_TX_Errstat	
Function	Obtains error interrupt sources for the transmission MACDMA.	
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 <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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

7.4.1.5 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 7.12 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 7.13 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 7.14 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 7.15 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.

7.4.2 Interrupts

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

Table 7.16 Interrupts Related to Operations for Transmission

Interrupt Name	Symbol	Conditions for Asserting and De-asserting Interrupts
TX FIFO underflow interrupt	INTETHTXFIFO	<p>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.</p> <p>Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.</p>
TX-FIFO error interrupt	INTETHTXFIFOERR	<p>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.</p> <p>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.</p>
MACDMA transmission error interrupt	INTETHTXDERR	<p>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.</p> <p>Modify the settings of the transmission descriptor for retransmission.</p> <p>Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.</p>
Ethernet MACDMA transmission complete interrupt	INTETHTXDMA	<p>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.</p> <p>Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.</p>
Ethernet transmission complete interrupt	INTETHTXCMP	<p>This interrupt occurs when operations for communications by the transmission MAC are completed.</p> <p>Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.</p>

Table 7.17 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 7.18 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.

7.4.3 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 (→ 7.4.1.1).
2. Acquire a TX buffer (→ 7.4.3.1).
3. Create Tx frame control information (→ 7.4.3.2(1)).
4. Create Ethernet transmit frame data (→ 7.4.3.2(2)).
5. Create TX descriptors (→ 7.4.3.3).
6. Execute the DMAC activation command (→ 7.4.3.4).
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 (→ 7.4.3.5).
11. Release the TX buffer (optional)

The details of the above steps are described below.

7.4.3.1 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

7.4.3.2 Creating TX Data

Figure 7.13 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.

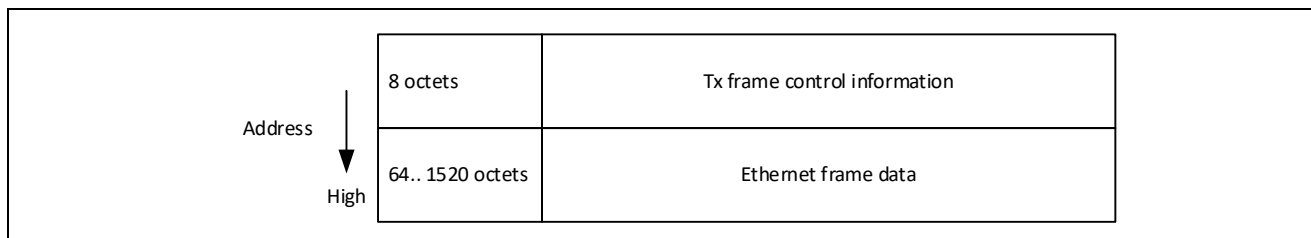


Figure 7.13 TX Data Format

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.

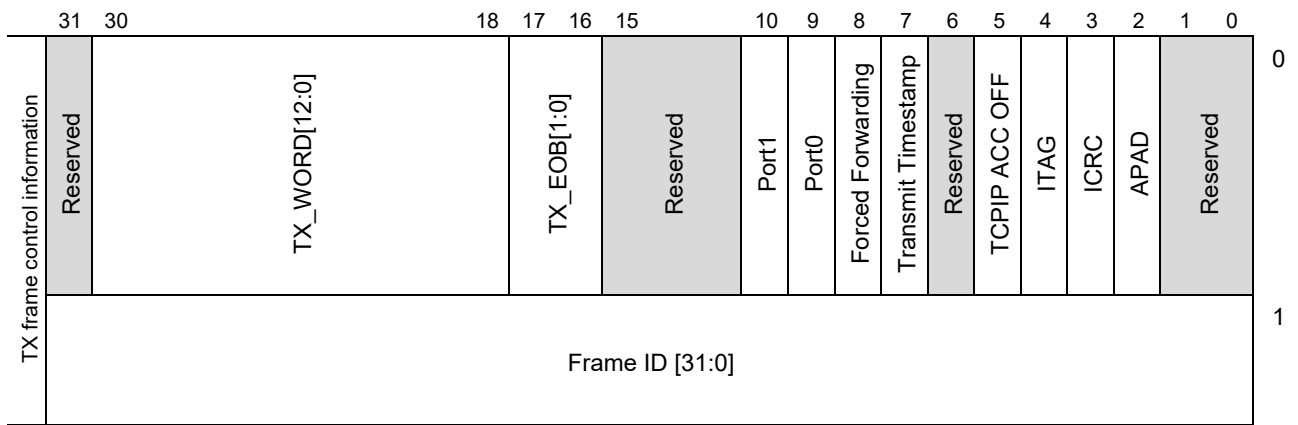


Figure 7.14 Tx Frame Control Information Format

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 ^{Note1}	Port 1 is used to enable forced forwarding of the Ethernet switch.
Port 0 ^{Note1}	Port 0 is used to enable forced forwarding of the Ethernet switch.
Forced Forwarding ^{Note1}	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 ^{Note1}	Enables timestamping of transmission frames when the Ethernet switch is in use.
TCPIP ACC OFF ^{Note2}	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.

Notes1: These functions 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_LENGTH [14:0] (15 bits), TX_LENGTH [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.TCPIPEN = 1) and 0 when it is disabled.

$$\text{TX_LENGTH [14:0]} = (\text{TX frame size} - \text{TCPIPACC Pad Size} + 3) \text{ (bytes)}$$

(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.

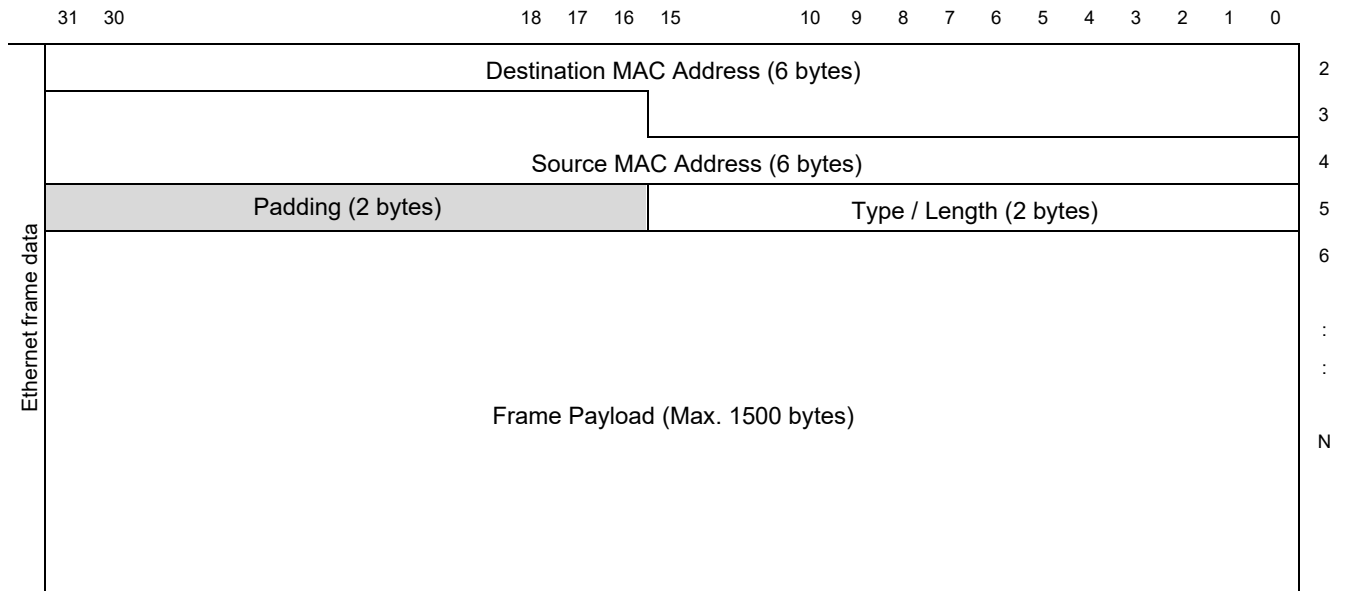


Figure 7.15 Tx Ethernet Frame Data Format – TCPIPACC is enabled, without VLAN tag

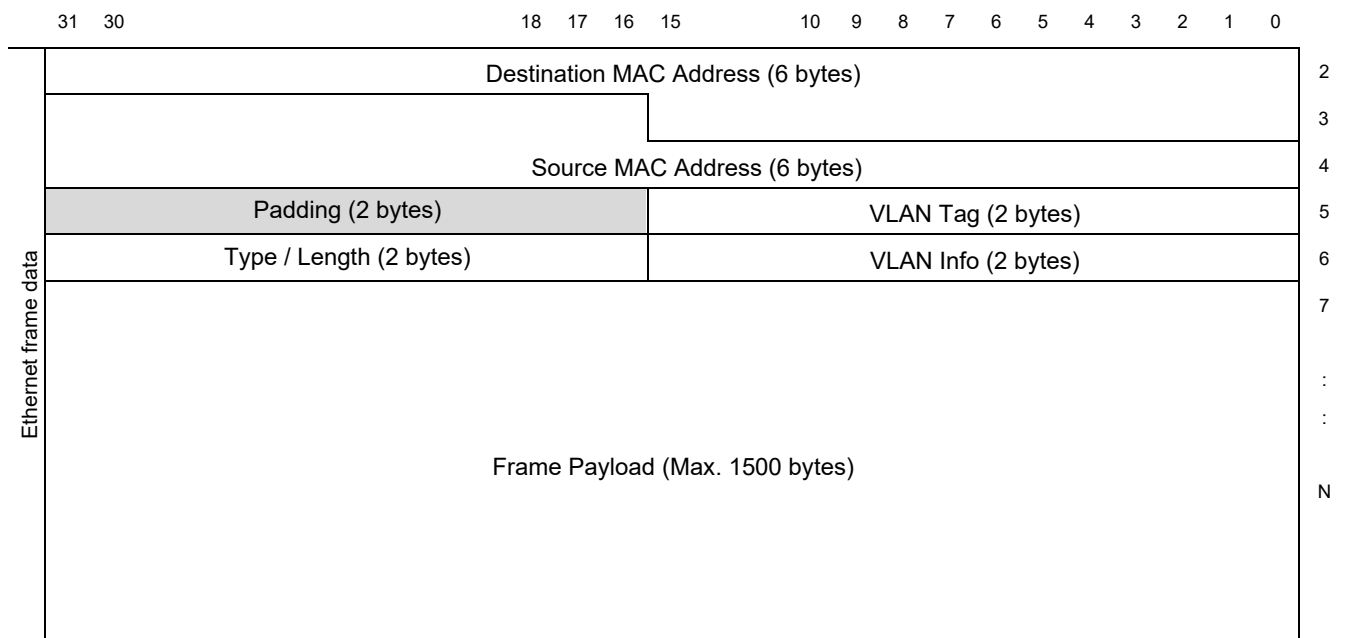


Figure 7.16 Tx Ethernet Frame Data Format – TCPIPACC is enabled, with VLAN tag

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.

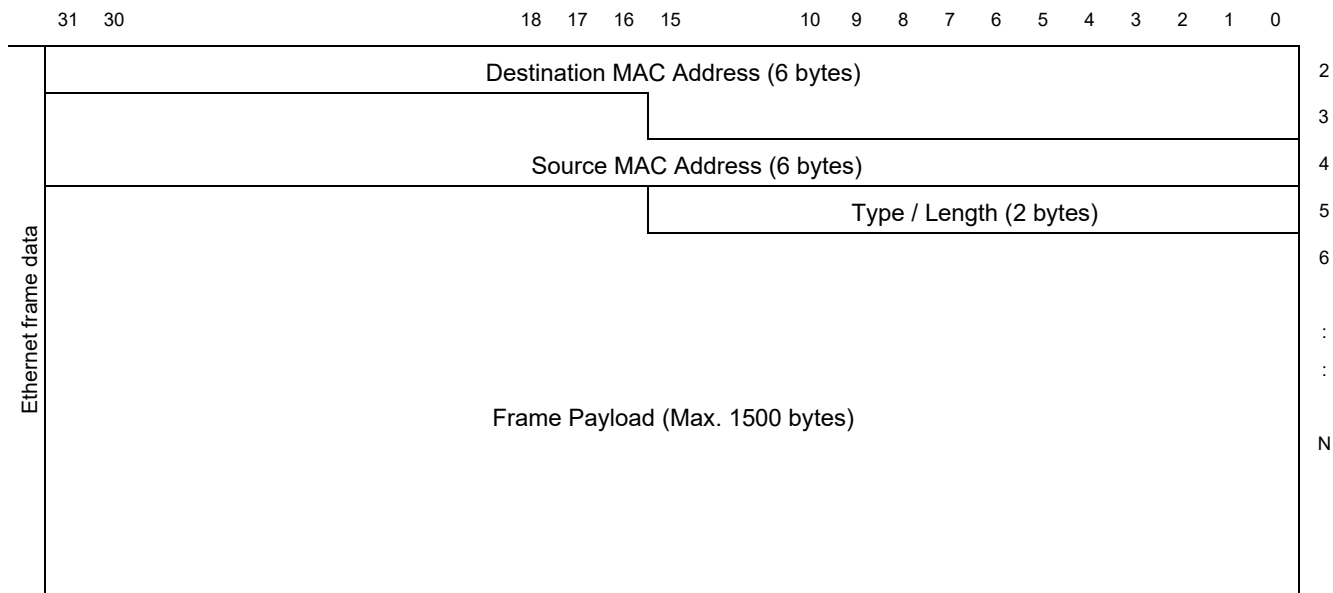


Figure 7.17 Tx Ethernet Frame Data Format – TCPIPACC is disabled, without VLAN tag

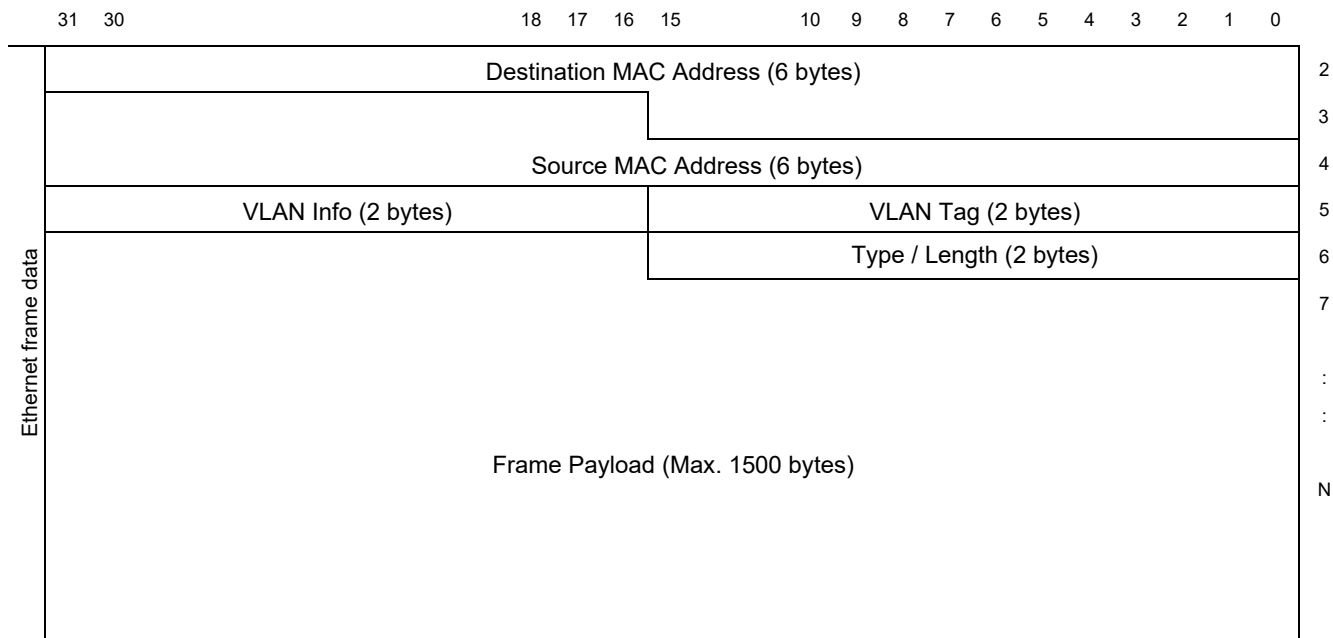


Figure 7.18 Tx Ethernet Frame Data Format – TCPIPACC is disabled, with VLAN tag

7.4.3.3 Creating TX Descriptors

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

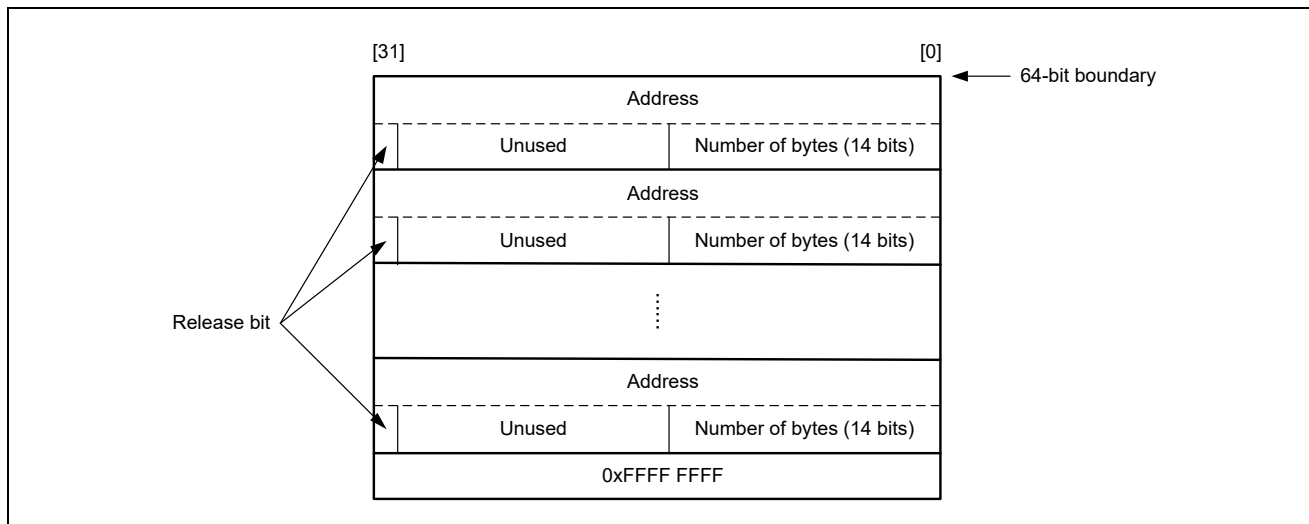


Figure 7.19 Structure of the TX Descriptor

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 MACDMAC 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 MACDMAC after the end of transmission. When a release bit is 0, the buffer is not released.

7.4.3.4 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

7.4.3.5 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.

7.4.4 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 (→ 7.4.1.1)
2. Enabling the Rx MAC (→7.4.4.1)
3. Activating the Rx DMAC (→7.4.4.2)
4. Receiving a frame and acquiring the buffer (→7.4.4.3)
5. The reception completed interrupt occurs.
6. Acquiring the Rx buffer information (→7.4.4.4)
7. Checking the status of frames (→7.4.4.5(1))
8. Acquiring the Ethernet frame data (→ 7.4.4.5(2))
9. Releasing the Rx buffer

7.4.4.1 Enabling the Rx MAC

Set 1 to the reception enable register (GMAC_RXMAC_ENA → 7.3.4.15) to enable the reception MAC.

7.4.4.2 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

7.4.4.3 Receiving a Frame and Acquiring the Buffer

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

7.4.4.4 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 7.4.4.5 for the format of received data.

7.4.4.5 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.

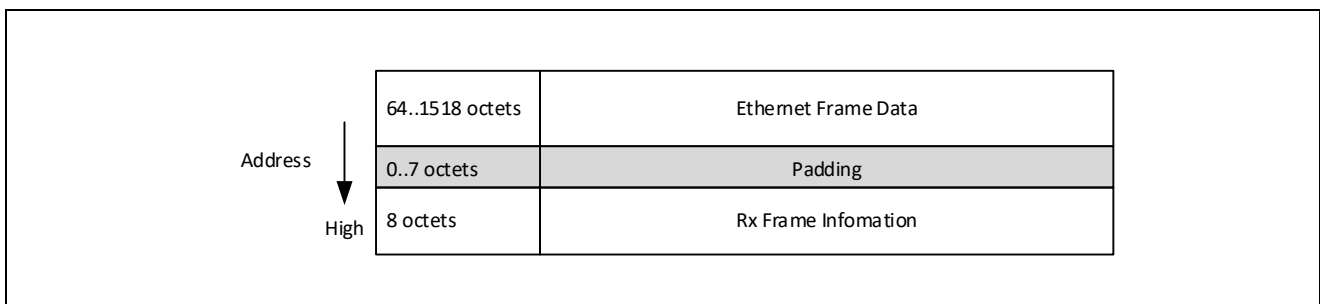


Figure 7.20 Rx Data Format

(1) Rx Frame Information

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

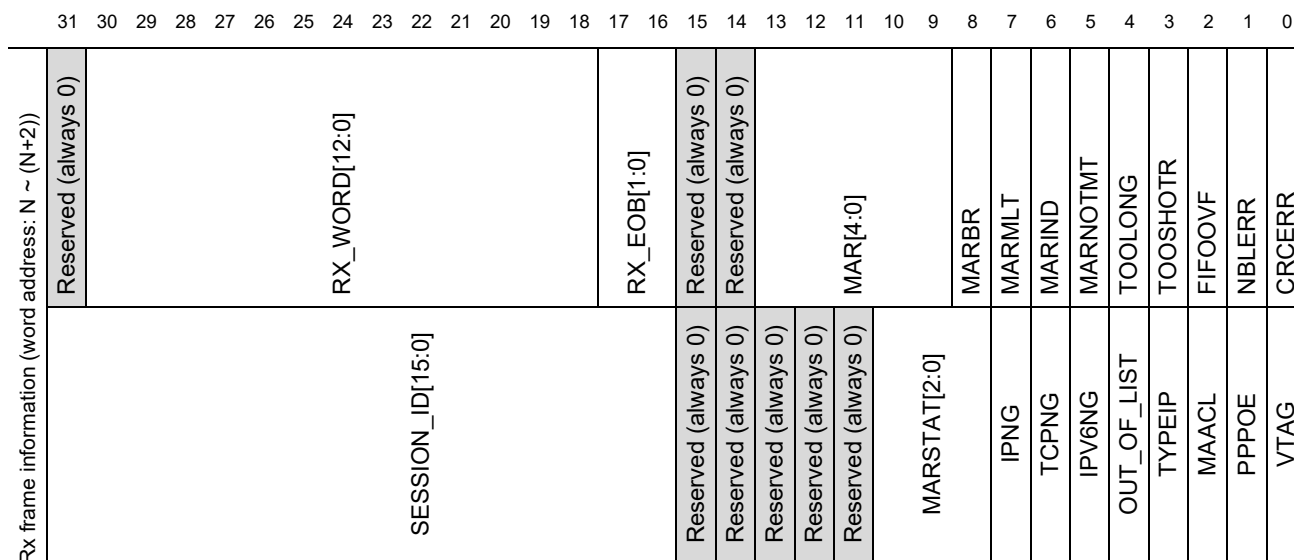


Figure 7.21 Rx frame Information

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 <small>Note2</small>	1: The checksum of the IPv4 Header does not match the calculation result of the TCPIP accelerator.
TCPNG <small>Note2</small>	1: The checksum of the TCP or UDP header does not match the calculation result of the TCPIP accelerator.
IPV6NG <small>Note2</small>	1: The IPv6 expansion header is Routing, Hop-by-Hop, or Destination Opt, and also the header length field is invalid.
OUT_OF_LIST <small>Note2</small>	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 <small>Note2</small>	1: IP packet
MAACL <small>Note2</small>	1: 802.3 (LLC/SNAP) packet

Field Name	Description
PPPOE ^{Note2}	1: PPPoE packet
VTAG ^{Note2}	1: Packet with VLAN TAG
RX_WORD[12:0]	Number of words of Ethernet frame ^{Note1}
RX_EOB[1:0]	Indicate valid bytes in the last word of this frame ^{Note1} 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

Notes 1: 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.

$$(\text{Number of received bytes in the Ethernet frame}) = \text{RX_LENGTH} [14:0] - 3$$

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.

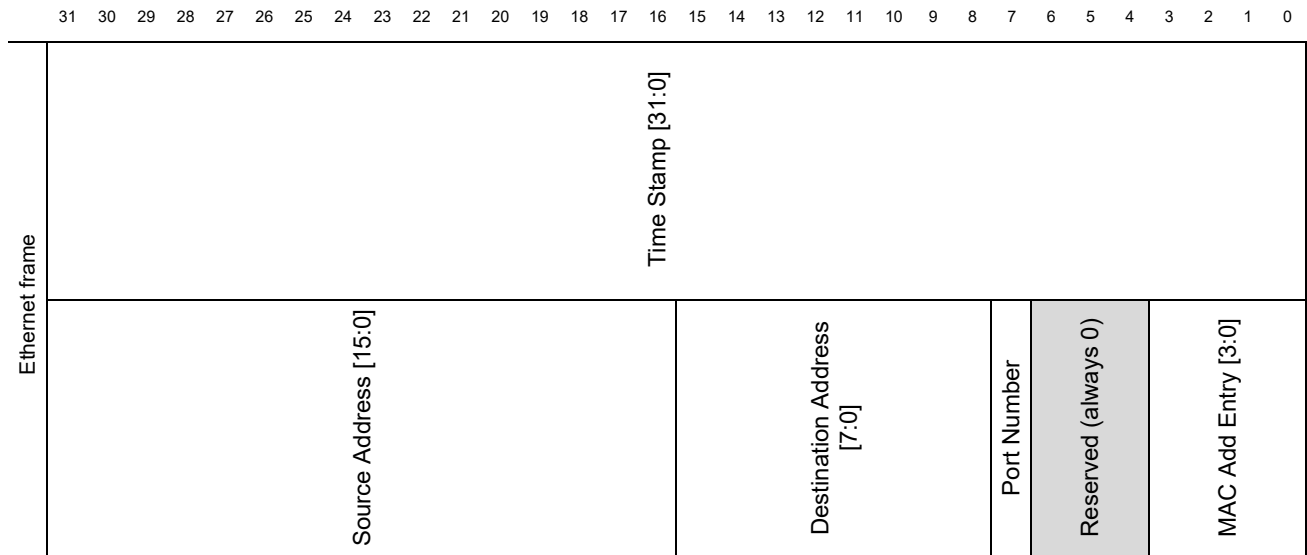


Figure 7.22 Destination MAC Address Field (when insertion of management tag is permitted)

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 AFILTEREN 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

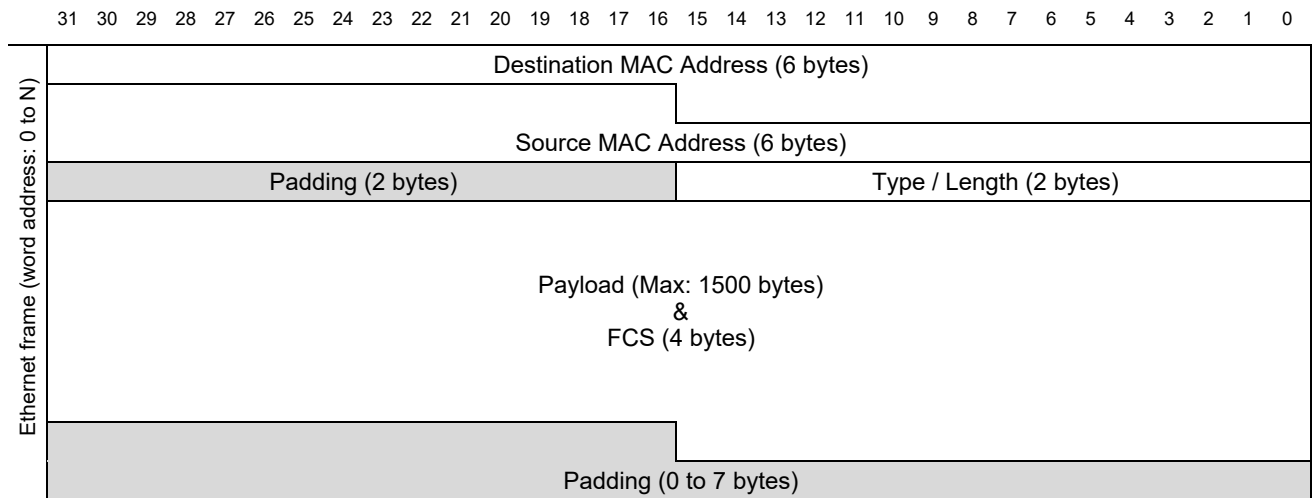


Figure 7.23 Format of Receive Ethernet Frame – TCPIPACC is enabled, without VLAN Tag, no TCP/UDP packets

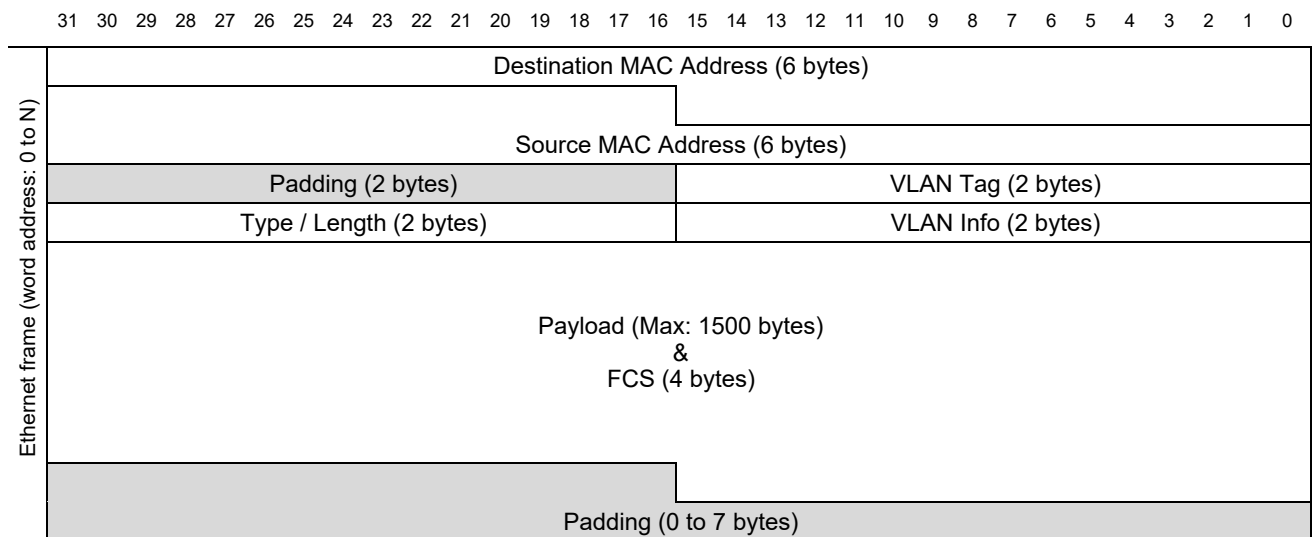


Figure 7.24 Format of Receive Ethernet Frame – TCPIPACC is enabled, with VLAN Tag, no TCP/UDP packets

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

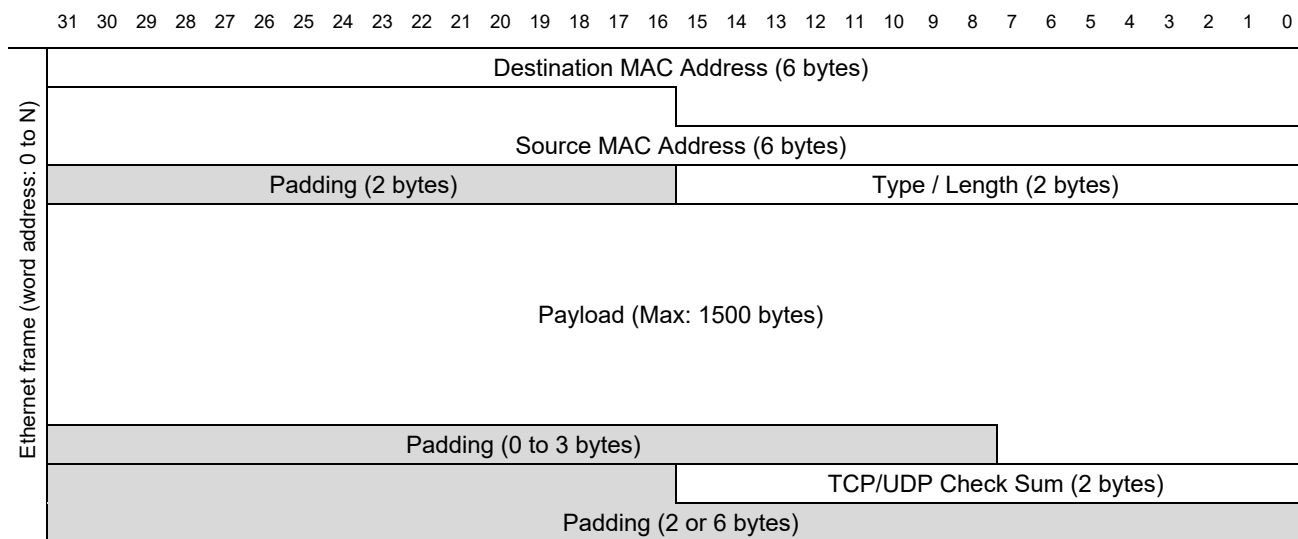


Figure 7.25 Format of Receive Ethernet Frame – TCPIPACC is enabled, without VLAN Tag, with TCP/UDP packets

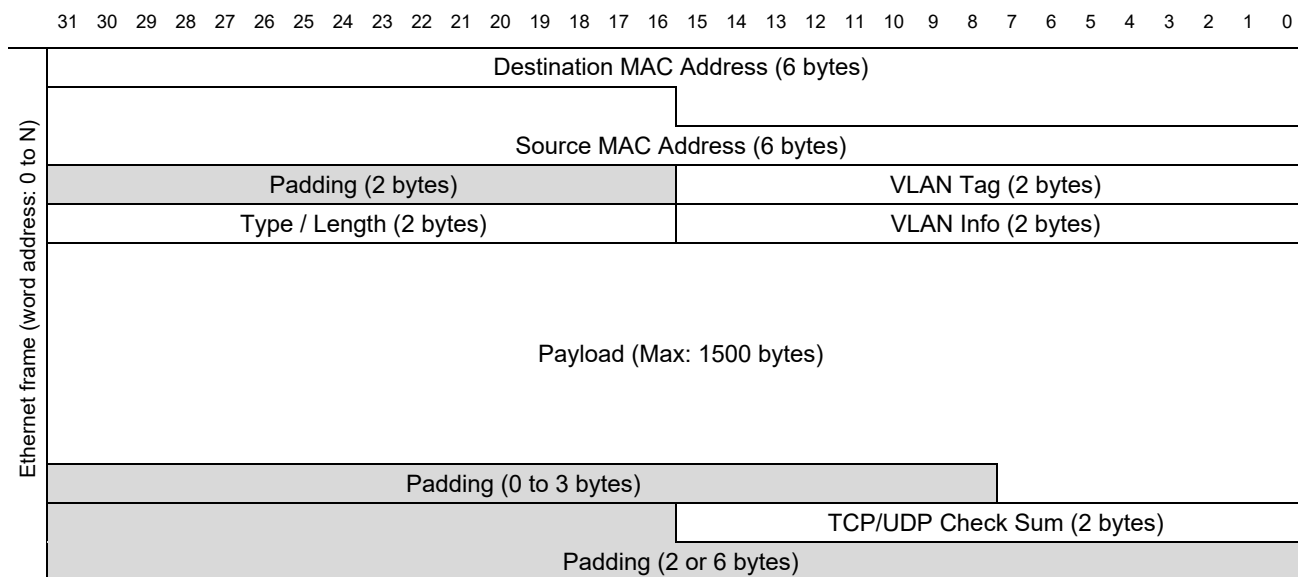


Figure 7.26 Format of Receive Ethernet Frame – TCPIPACC is enabled, with VLAN Tag, with TCP/UDP packets

(c) When Rx TCPIP accelerator is disabled

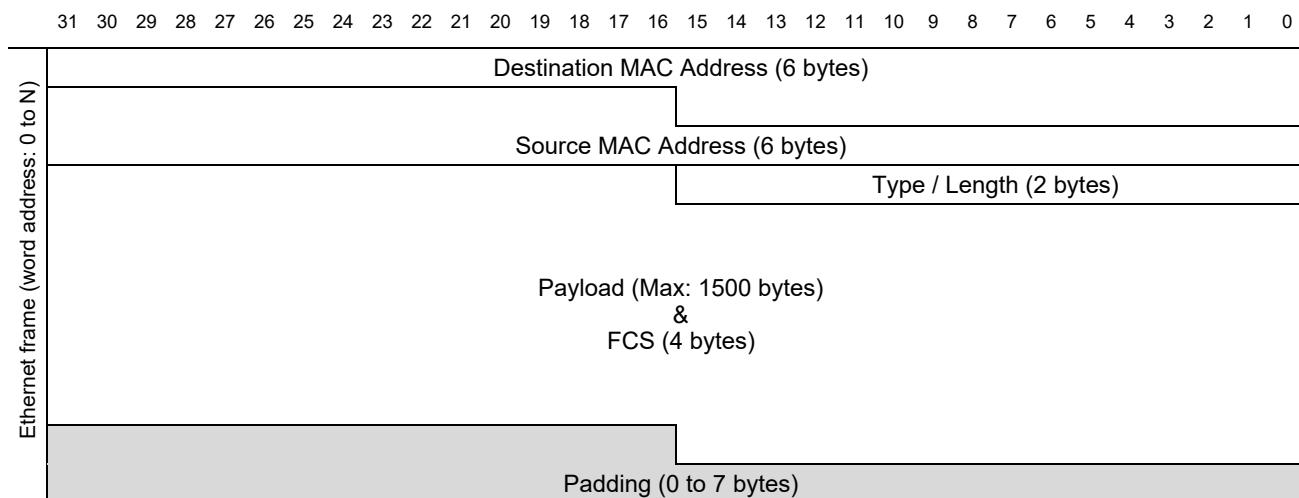


Figure 7.27 Format of Receive Ethernet Frame – TCPIPACC is disabled, without VLAN Tag

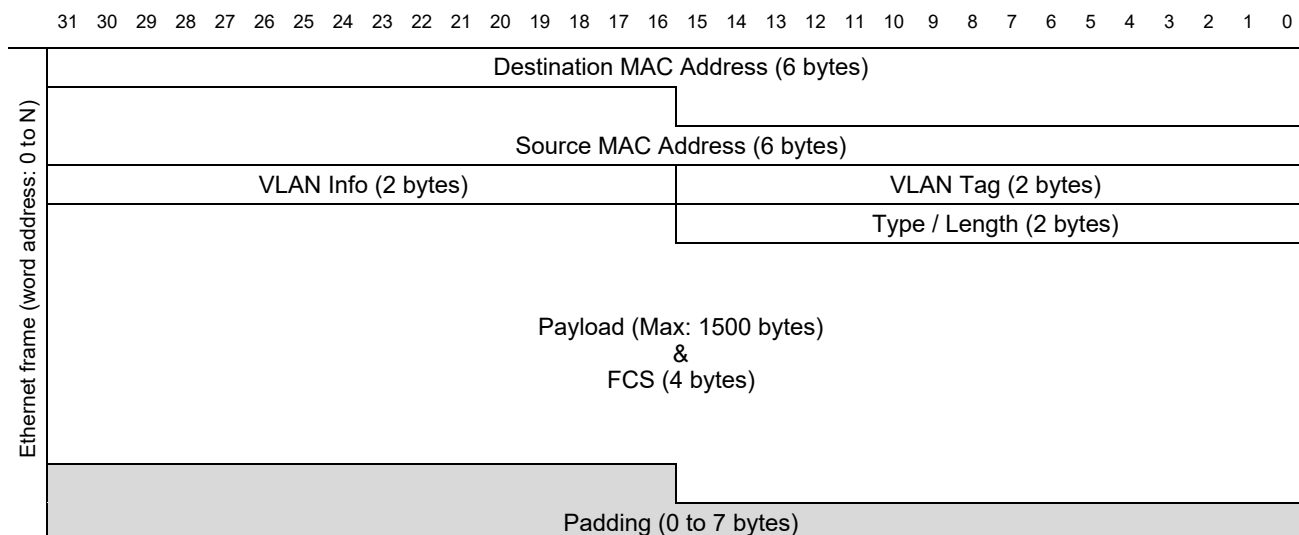


Figure 7.28 Format of Receive Ethernet Frame – TCPIPACC is disabled, with VLAN Tag

7.4.5 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.

7.4.5.1 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 TCPIP accelerator for transmission is disabled.

Table 7.19 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.

7.4.5.2 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 7.20 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.

7.5 Notes

Note the following when using the Gigabit Ethernet MAC.

7.5.1 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 7.4.5.1, Transmission Using the TCPIP Accelerator, for detail.

7.5.2 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 of 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.

7.5.3 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 7.29 to Figure 7.32 are the flowcharts of workaround (B) described above.

- In case the hardware real-time OS is used

Figure 7.29: Flowchart of RX FIFO overflow processing task

Figure 7.30: 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 7.31: Flowchart of RX FIFO overflow processing

Figure 7.32: 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

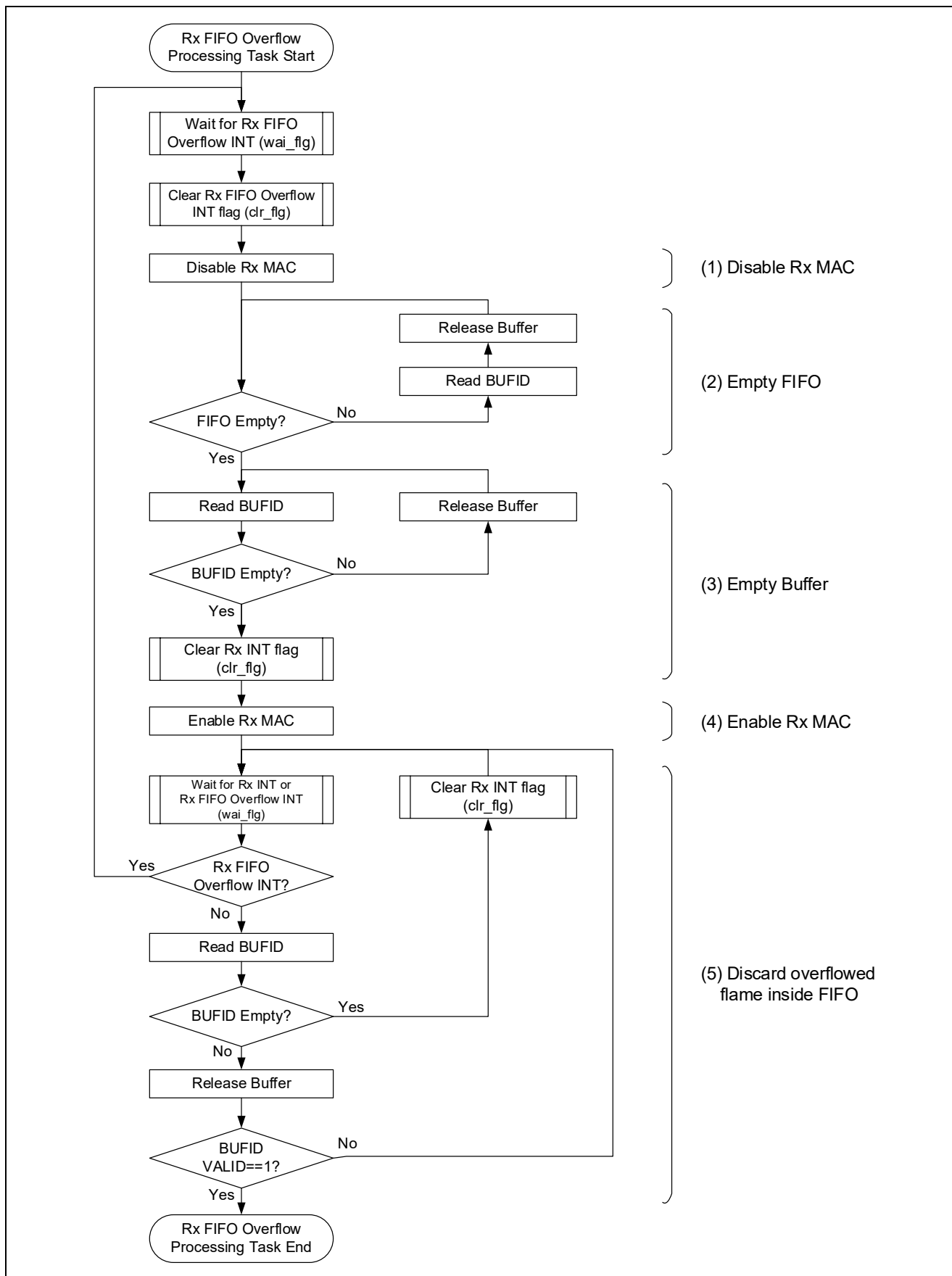


Figure 7.29 Flowchart of RX FIFO Overflow Processing Task (In case the hardware real-time OS is used)

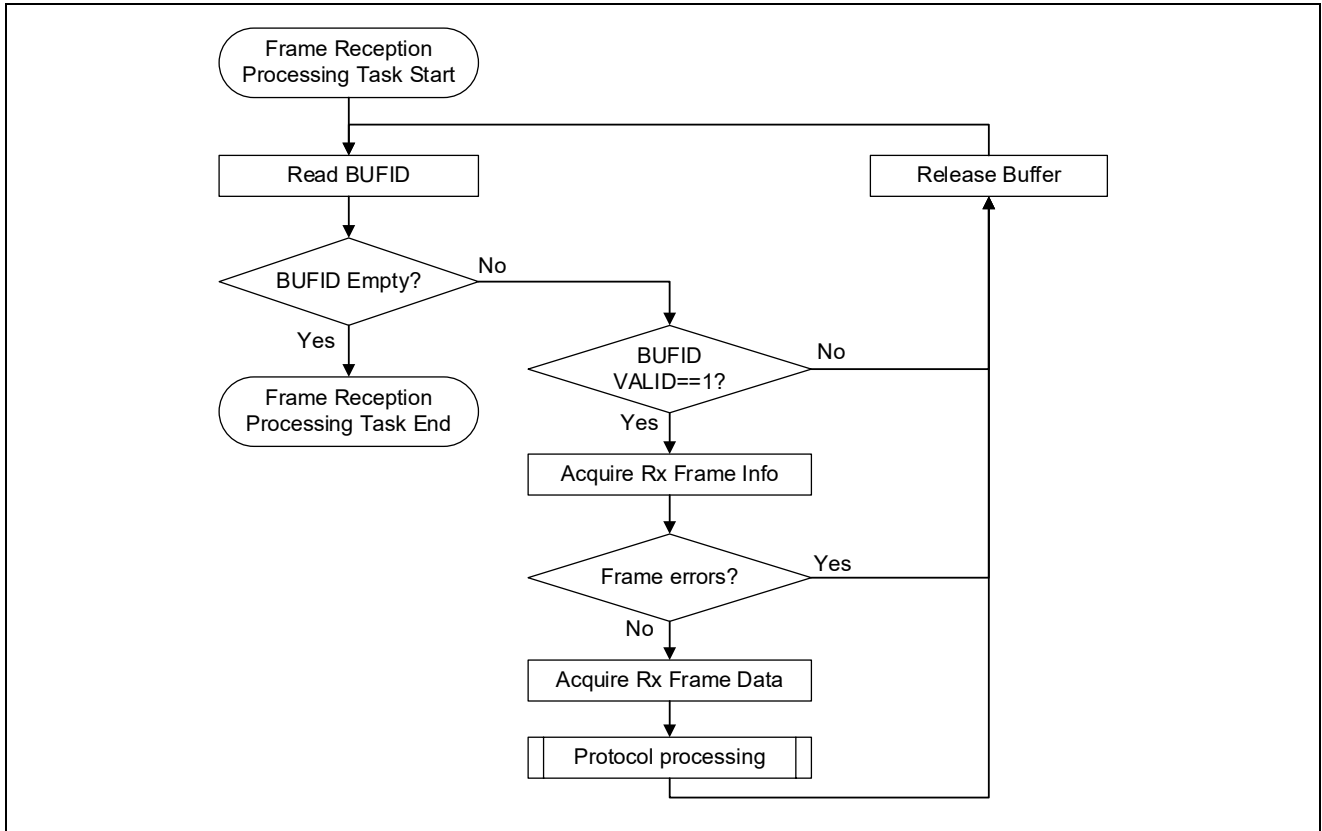


Figure 7.30 Flowchart of Reception Processing Task (In case the hardware real-time OS is used)

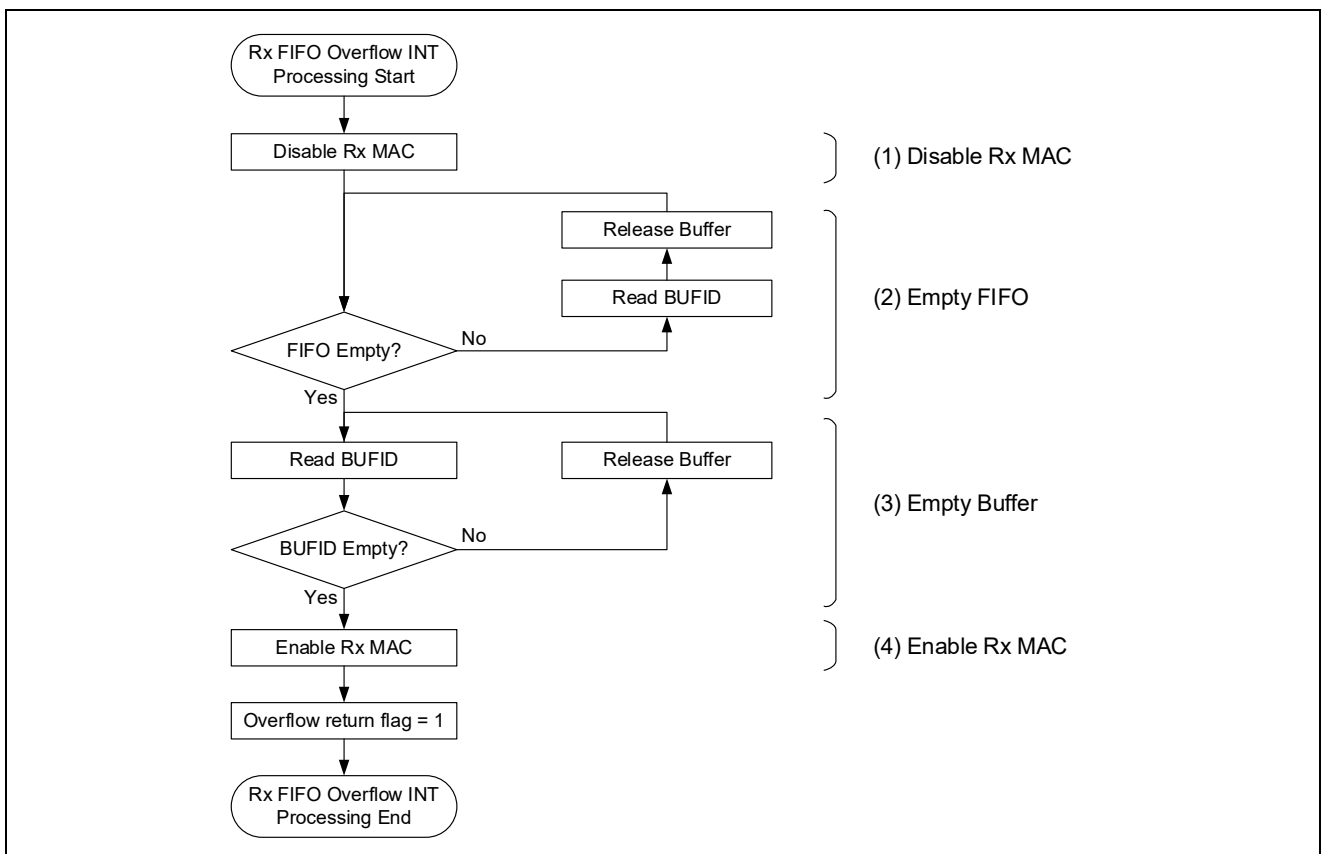


Figure 7.31 Flowchart of RX FIFO Overflow INT Processing (In case the hardware real-time OS is not used)

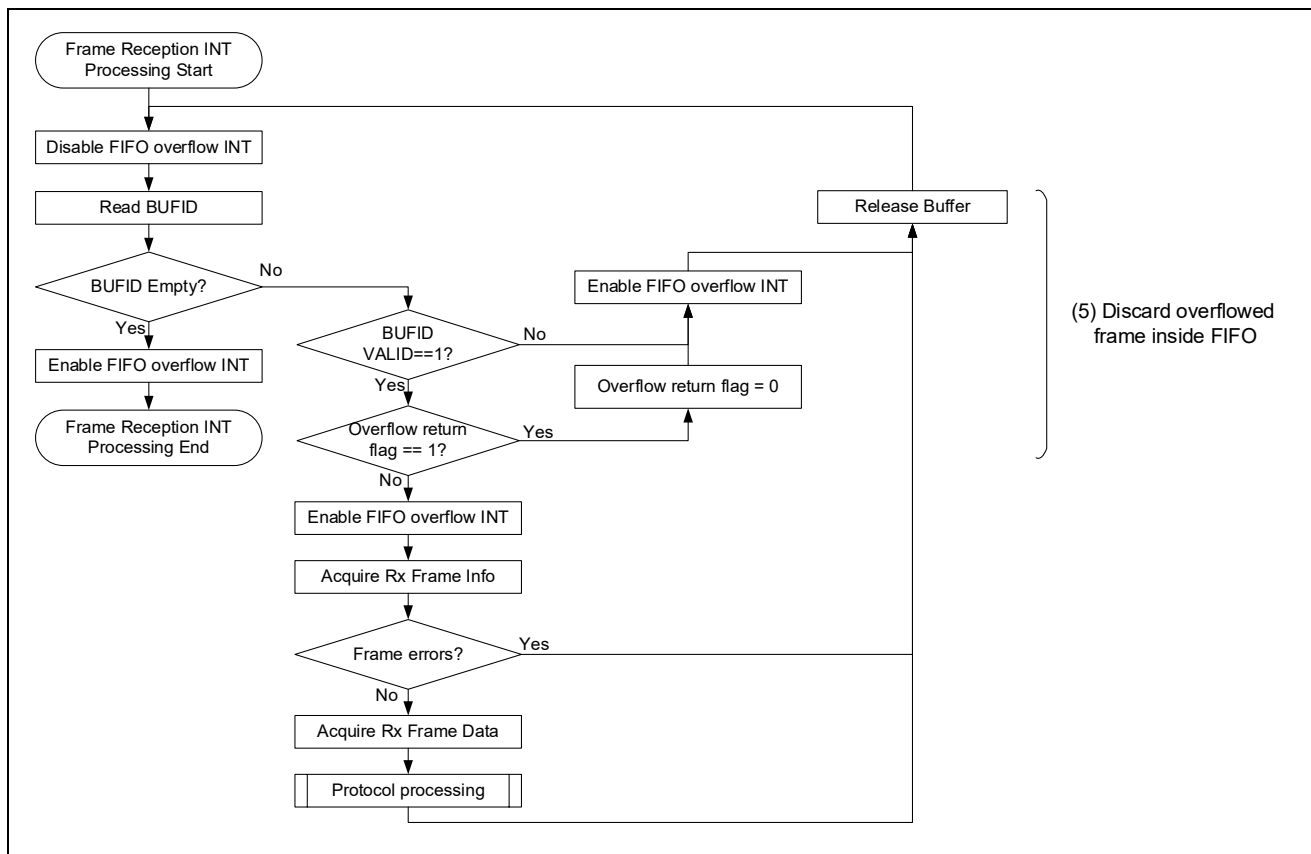


Figure 7.32 Flowchart of Reception Processing (In case the hardware real-time OS is not used)

7.5.4 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

- Disable Rx TCPIP accelerator. Specifically, clear bit0 or set bit2 of GMAC_ACC register.
- 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 7.33 is the flowchart of this workaround.

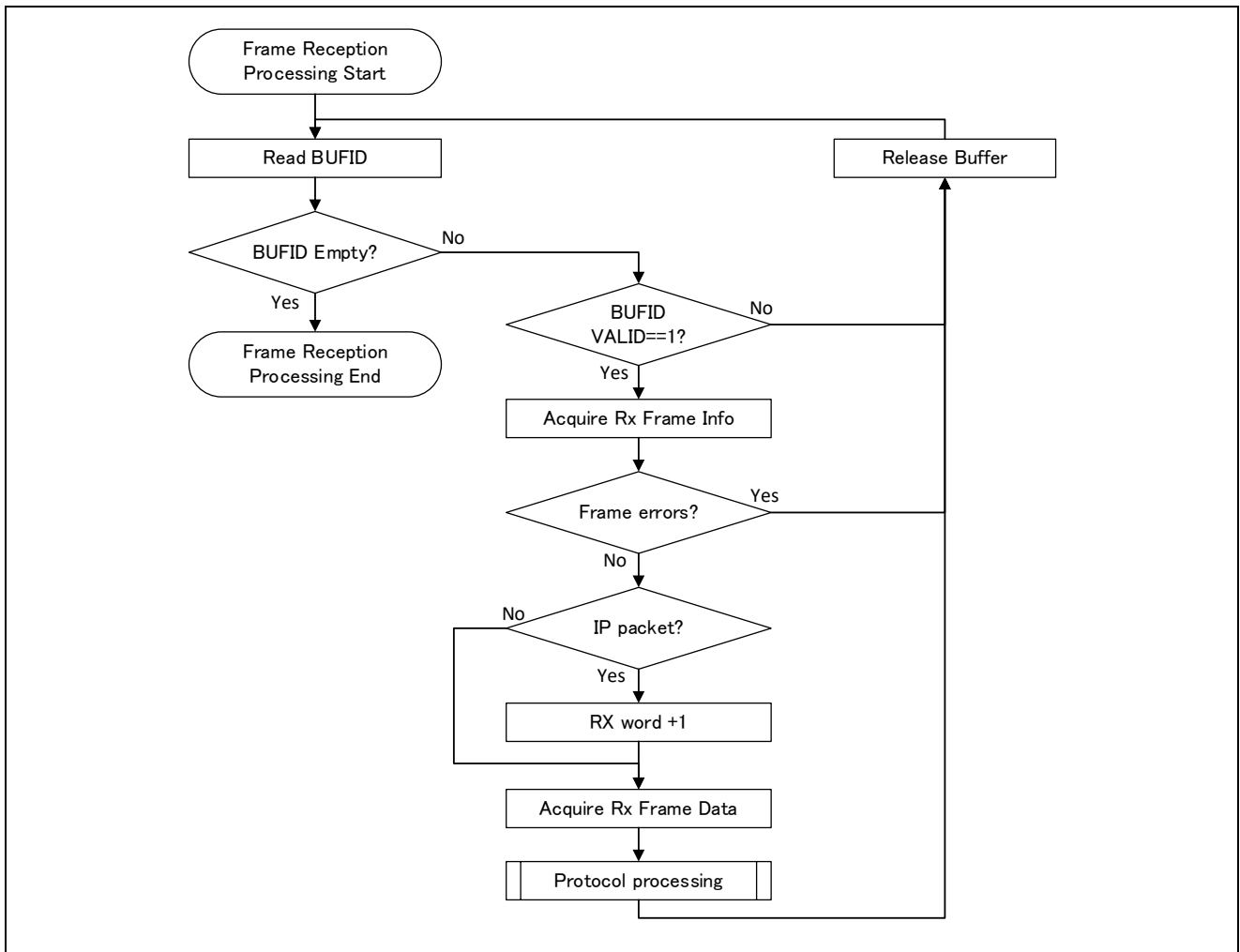


Figure 7.33 Flowchart of Reception Processing

7.5.5 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).

7.5.6 Jumbo Frames

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

8. Ethernet Switch

This section describes the Ethernet switch of an R-IN32M3.

8.1 Overview

Each R-IN32M3 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.

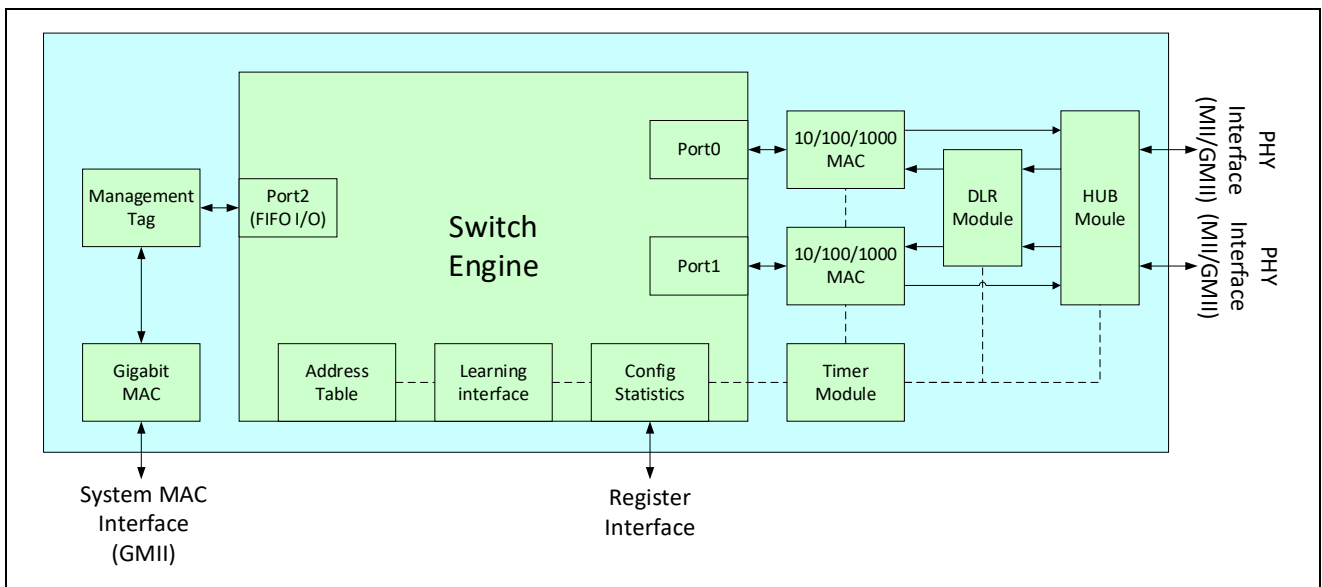


Figure 8.1 Overview of the Ethernet Switch

8.2 Characteristics

The Ethernet switch of an R-IN32M3 has the following features:

- Two-port PHY interface
- For IEEE802.3
- 10 BASE, 100 BASE
- 1000 BASE-T ^{Note1}
- Full- and half-duplex communications ^{Note2}
- 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 ^{Note1}

For details of the Ethernet PHY of the R-IN32M3-EC, refer to the R-IN32M3-EC User's Manual.

- Notes 1. Only provided in the R-IN32M3-CL.
Half-duplex communications are not supported.**
- 2. Half-duplex communications are not supported for 1000 BASE.**

Interrupt Signals of Ethernet Switch

Exception No.	Name	Interrupt Source	Connected to				
			NVIC	HW-RTOS	DMAC	Real-Time Port	Timer
54	INTETHSW	Ether SWITCH Timer interrupt	○	○	○	○	○
55	INTETHSWDLR	Ether SWITCH DLR interrupt	○	○	○	○	○
56	INTETHSWSEC	Ether SWITCH SEC interrupt	○	○	○	○	○

I/O Signals of Ethernet Switch (Excluding MII Pins)

Pin Name	I/O	Function	Shared Port	Active
ETHSWSECOUT	O	EtherSwitch event output per second	P24	High

8.3 Control Registers

8.3.1 List of Registers

(1) Operating Mode Registers

Register name	Symbol	Address
Ethernet PHY LINK mode register	ETHPHYLNK	BASE + 0614H
Ethernet switch management TAG control register	ETHSWMTC	BASE + 0680H
Ethernet switch operating mode setting register	ETHSWMD	BASE + 0684H
Ethernet switch 10-Mbps half-duplex mode setting register	ETHSW10HDEN	BASE + 060CH

(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	VLAN_PRIORITY _n	4007 0100H + 0004H*n
IP priority register	IP_PRIORITY _n	4007 0140H + 0004H*n
PRIORITY configuration register	PRIORITY_CFG _n	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) (n=0..6)		
Hub input filter MAC address low register n	HUB_FLT_MAC _{nlo}	4007 01C8H + 0008H*n
Hub input filter MAC address high register n	HUB_FLT_MAC _{nhi}	4007 01CCH + 0008H*n
• Statistics registers		
Switch statistics registers	Refer to 8.3.3.22	Refer to 8.3.3.22

(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)	Refer to section 8.3.5.3	Refer to section 8.3.5.3
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	Refer to section 8.3.5.6	Refer to section 8.3.5.6

(5) Timer Module Registers

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_CORR_OFFS	4007 C13CH

(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	Refer to 8.3.7.17	Refer to 8.3.7.17

8.3.2 Operating Mode Setting Registers

8.3.2.1 Ethernet PHY LINK Mode Register (ETHPHYLNK)

This register is used to specify the active level of the Ethernet PHY LINK signals. Set this register according to the active level of the PHYLINK signal from an external PHY to be connected. This register can be read or written in 32- or 16-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 how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.**
 - 2. Since the Ethernet PHY of the R-IN32M3-EC operates according to the setting of the initial value, leave the value at the initial value.**

																																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																																
																																ETHPHYLNK																																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BASE + 0614H
																																																																																														Initial value		
																																																																																														0000 000CH		
																																																																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			

Bit Position	Bit Name	Description
3	CATLINK1	This bit sets the active level of the PHYLINK signal for port 1 of the EtherCAT interface. 0: The PHYLINK signal is active high. 1: The PHYLINK signal is active low (initial value).
2	CATLINK0	This bit sets the active level of the PHYLINK signal for port 0 of the EtherCAT interface. 0: The PHYLINK signal is active high. 1: The PHYLINK signal is active low (initial value).
1	SWLINK1	This bit sets the active level of the PHYLINK signal for port 1 of the Ethernet switch interface. 0: The PHYLINK signal is active low (initial value) 1: The PHYLINK signal is active high.
0	SWLINK0	This bit sets the active level of the PHYLINK signal for port 0 of the Ethernet switch interface. 0: The PHYLINK signal is active low (initial value) 1: The PHYLINK signal is active high.

8.3.2.2 Ethernet Switch Management TAG Control Register (ETHSWMTC)

This register is used to specify management tag information when Ethernet switching is used. 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 how to release protection, see the description of the system protection 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
ETHSWMTC	SWTAGEN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SWTAGTYP15	SWTAGTYP14	SWTAGTYP13	SWTAGTYP12	SWTAGTYP11	SWTAGTYP10	SWTAGTYP9	SWTAGTYP8	SWTAGTYP7	SWTAGTYP6	SWTAGTYP5	SWTAGTYP4	SWTAGTYP3	SWTAGTYP2	SWTAGTYP1	SWTAGTYP0	BASE + 0680H
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	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
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0000 E001H

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.

8.3.2.3 Ethernet Switch Operating Mode Setting Register (ETHSWMD)

This register is used to specify the operating mode when Ethernet switching is used. 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 how to release protection, see the description of the system protection 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	
ETHSWMD	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 100%;">0 0</td> </tr> </table>	0 0	BASE + 0684H Initial value 0000 0000H
0 0			
R/W	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 100%;">0 0</td> </tr> </table>	0 0	R/W R/W R/W R/W
0 0			
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	

8.3.2.4 Ethernet Switch 10-Mbps/Half-Duplex Mode Setting Register (ETHSW10HDEN)

This register is for disabling the looping back of received frames by the Ethernet PHY layer during transmission when the Ethernet switch is used for 10-Mbps half-duplex communications.

The Ethernet PHY layer of the R-IN32M3-EC loops back transmitted data to received data in 10-Mbps half-duplex transfer. If Ethernet switching by this LSI chip is used to set up a loopback between two ports, transfer is mutually repeated between these ports, and Ethernet transfer does not proceed successfully.

When using 10-Mbps half-duplex communications, be sure to set the corresponding bit of this register to 1.

- Access This register can be read or written in 32- or 16-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 how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.

2. Only select disabling of received frames while the Ethernet PHY layer is linked in half-duplex communications at 10 Mbps. If this is done in another linked state, some correct received frames may also be disabled.

	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
	ETHSW10HDEN	BASE + 060CH
	0 0	Initial value
	R/W	0000 0000H
		SW10HDEN1 SW10HDEN0
		R/W R/W

Bit Position	Bit Name	Description
31 to 2	—	Reserved. When writing to these bits, write 0. If read, 0 is returned.
1	SW10HDEN1	Disables looping back of received frames during transmission from Ethernet port 1. 0: Disabling is not selected (this should be set when operation is other than for 10-Mbps half-duplex communications). 1: Disabling is selected (this should only be set when operation is for 10-Mbps half-duplex communications).
0	SW10HDEN0	Disables looping back of received frames during transmission from Ethernet port 0. 0: Disabling is not selected (this should be set when operation is other than for 10-Mbps half-duplex communications). 1: Disabling is selected (this should only be set when operation is for 10-Mbps half-duplex communications).

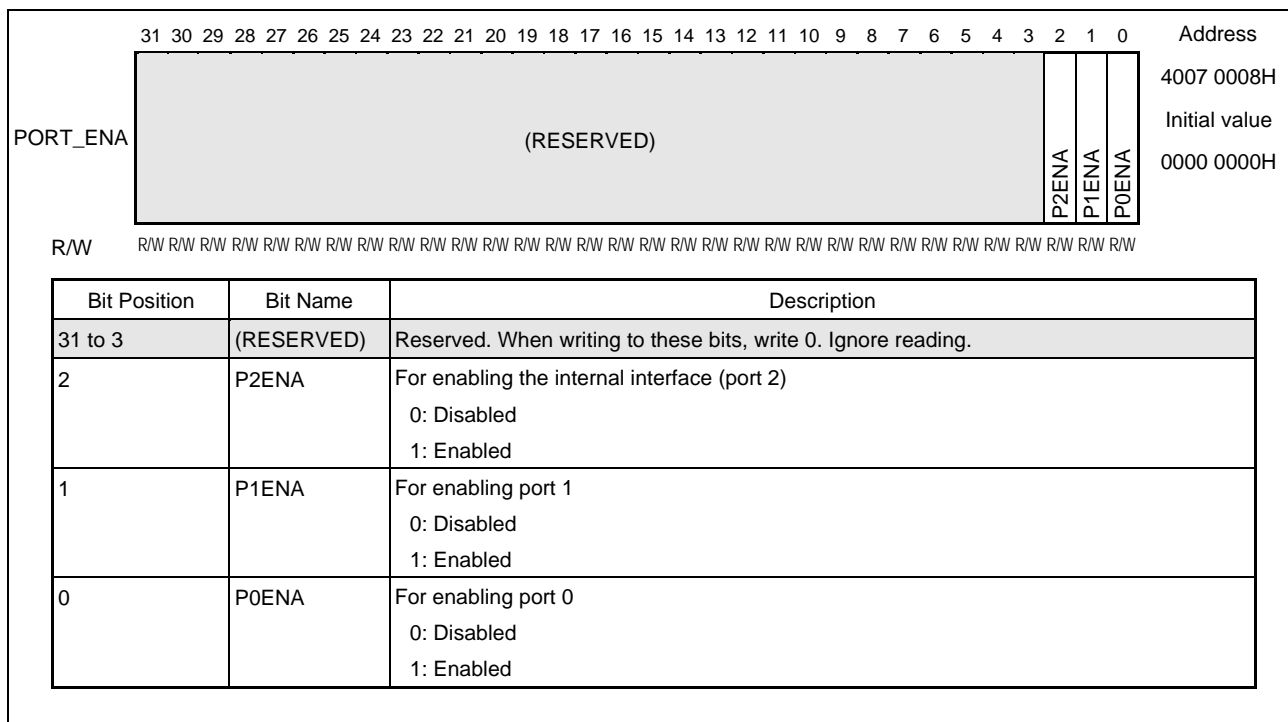
Notes 1. Only the R-IN32M3-EC supports this register.

2. The old products do not support this register.
For details of the old products, see section 1.1, Type Names of R-IN32M3-Series Products.

8.3.3 Switch Configuration Registers

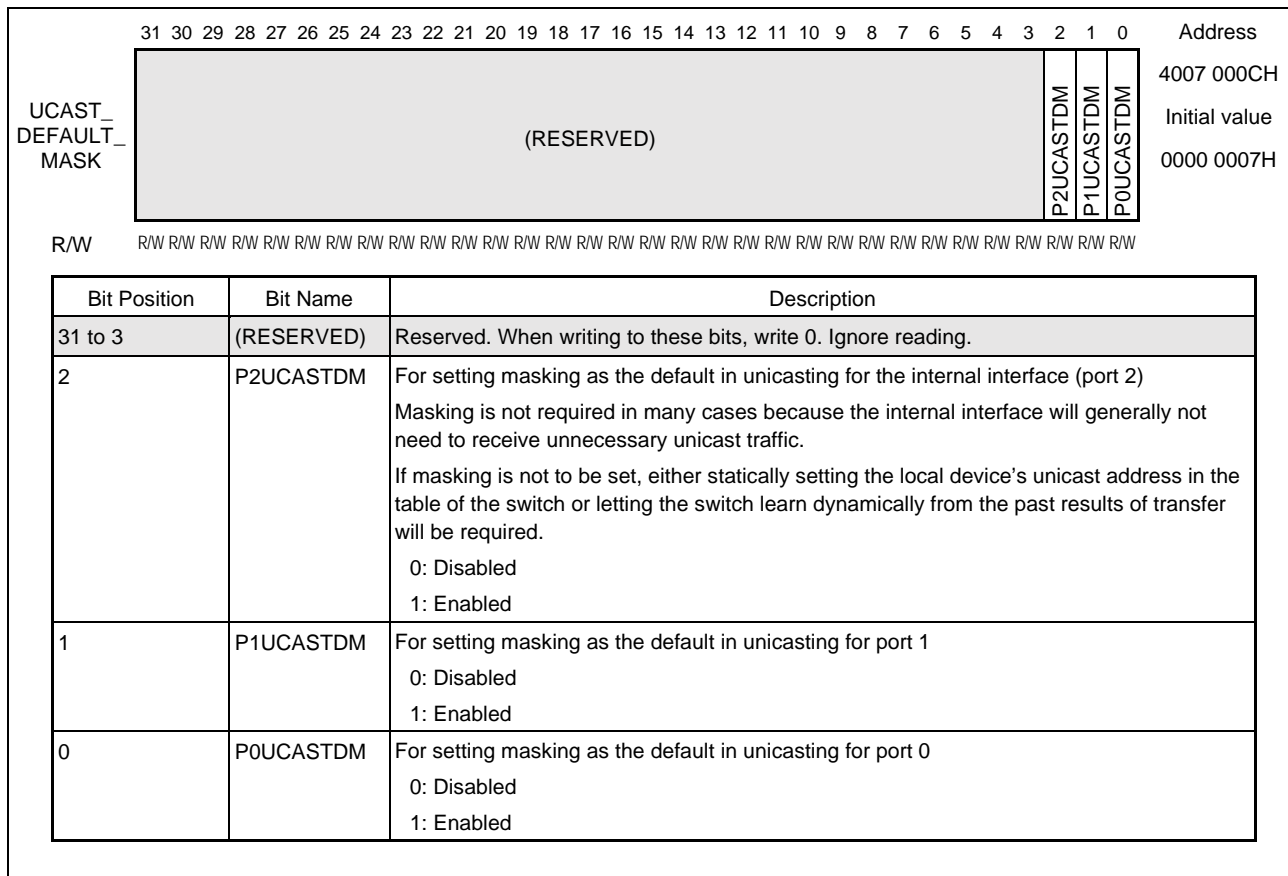
8.3.3.1 Port Enable Register (PORT_ENA)

This register enables or disables each port of the Ethernet switch. A disabled port does not transmit frames, but it can receive frames. This register can be read or written in 32-bit units.



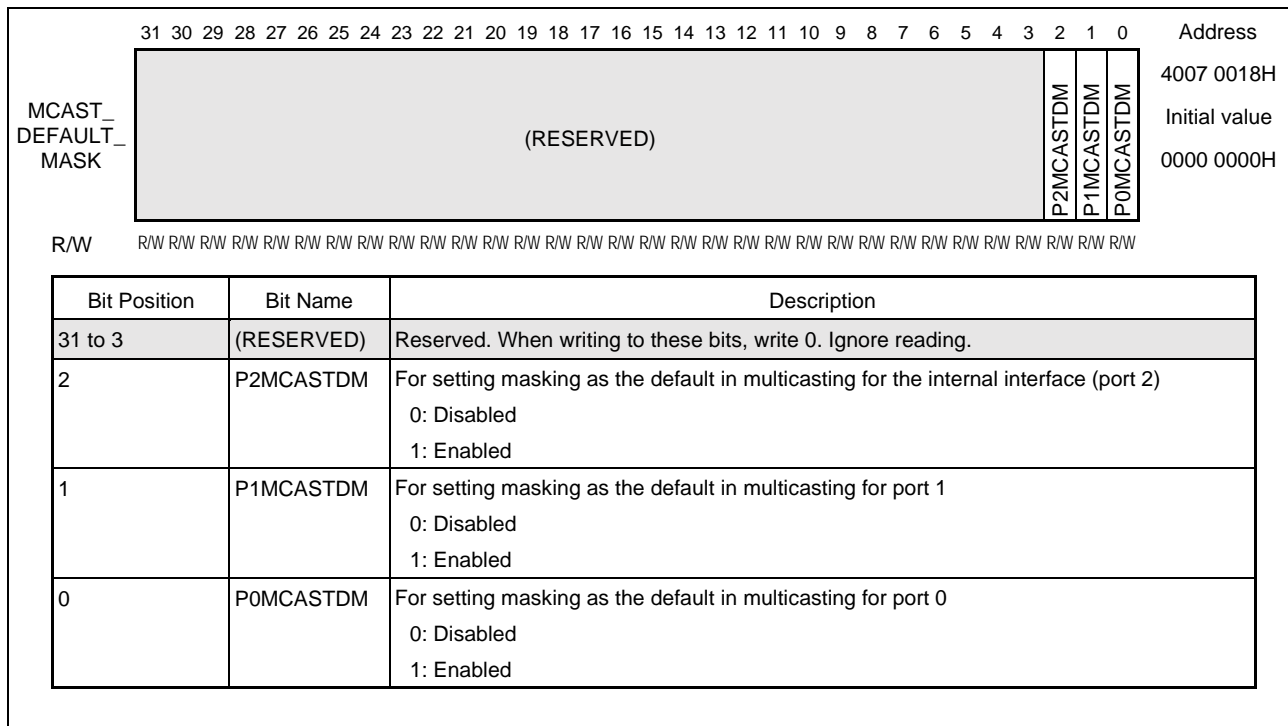
8.3.3.2 Unicast Default Mask Register (UCAST_DEFAULT_MASK)

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



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

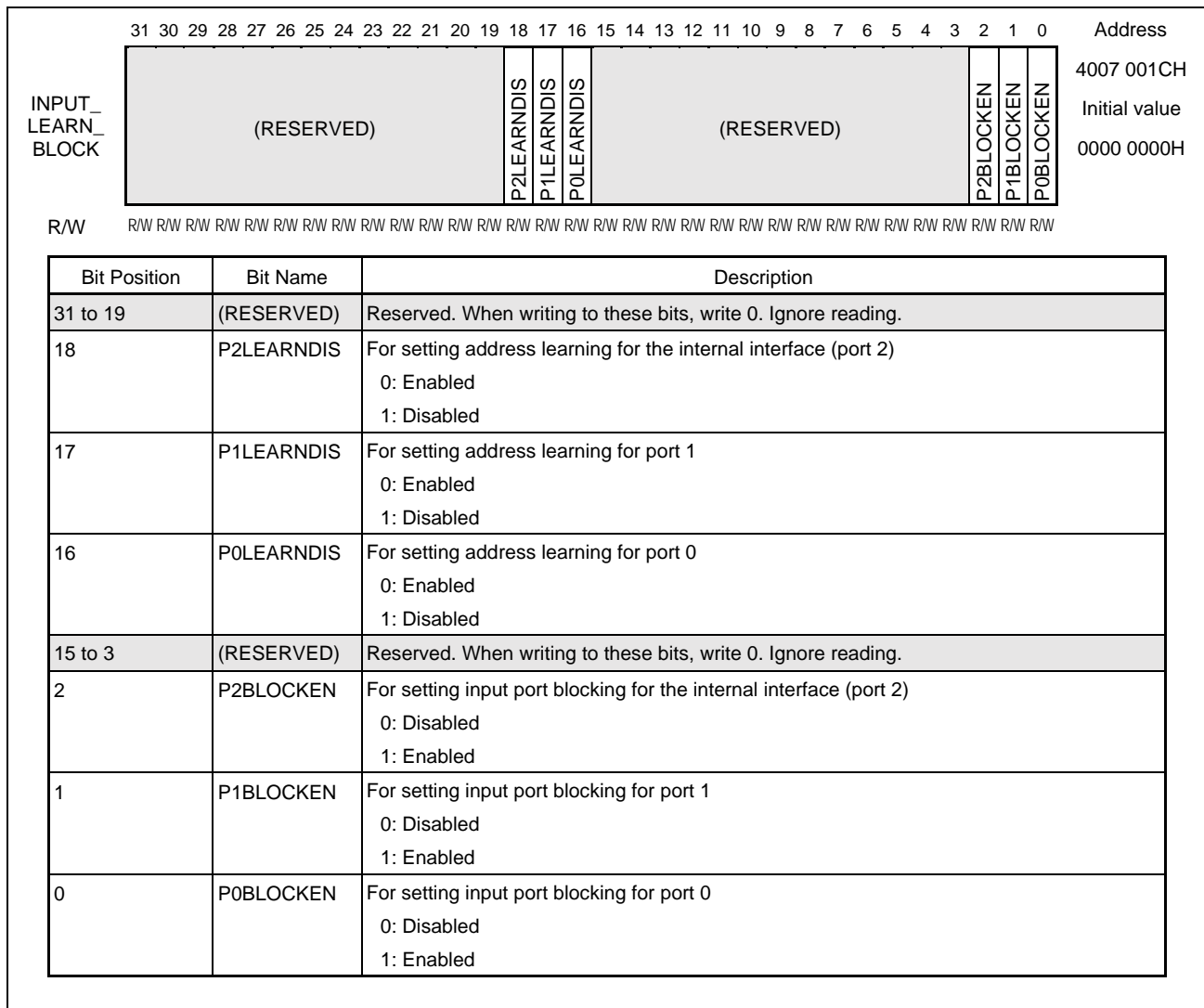


8.3.3.5 Input Learning Blocking Register (INPUT_LEARN_BLOCK)

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



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

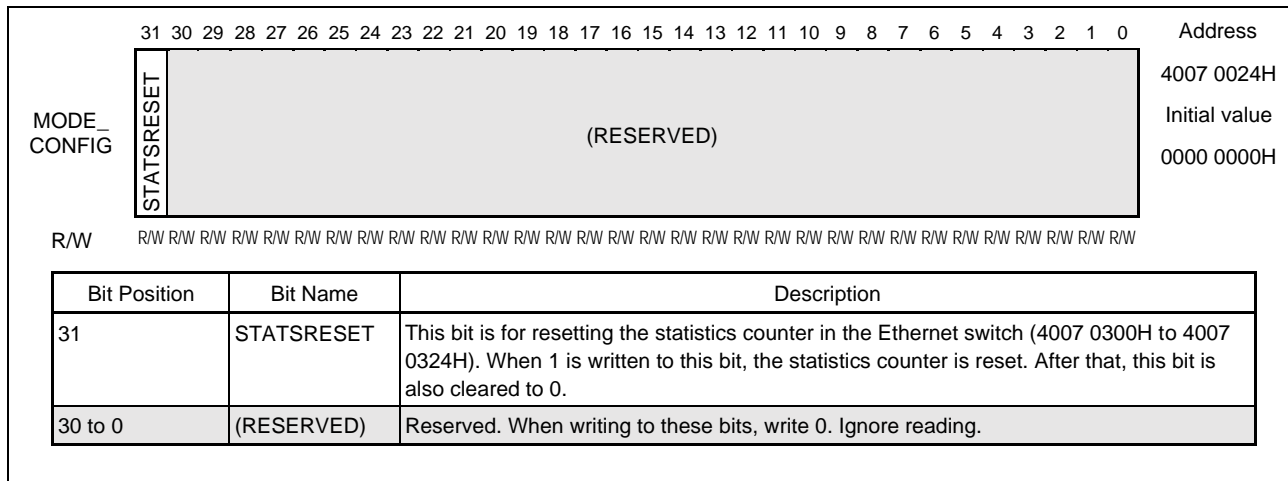
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address		
MGMT_CONFIG	(RESERVED)																P1PORTMASK	P0PORTMASK	PRIORITY	(RESERVED)					DISCARD	ENABLE	MSGTRANS	(RESERVED)			PORT	4007 0020H Initial value 0000 0000H			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	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 <small>Note</small>	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 <small>Note</small>	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. 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.

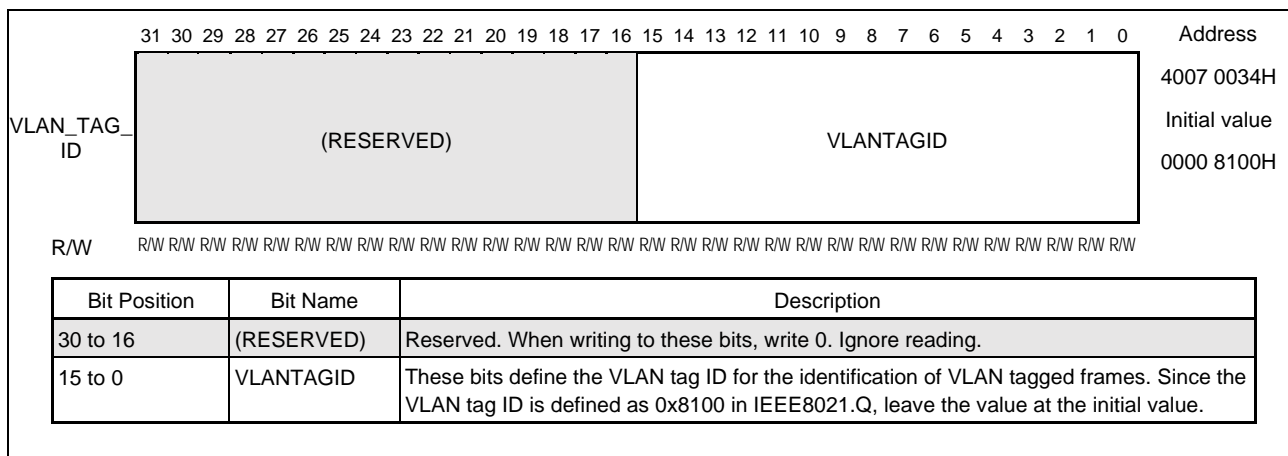
8.3.3.7 Mode Configuration Register (MODE_CONFIG)

This register is used to reset the statistics counter in the Ethernet switch. This register can be read or written in 32-bit units.



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



8.3.3.9 Output Queue Management Status Register (OQMGR_STATUS)

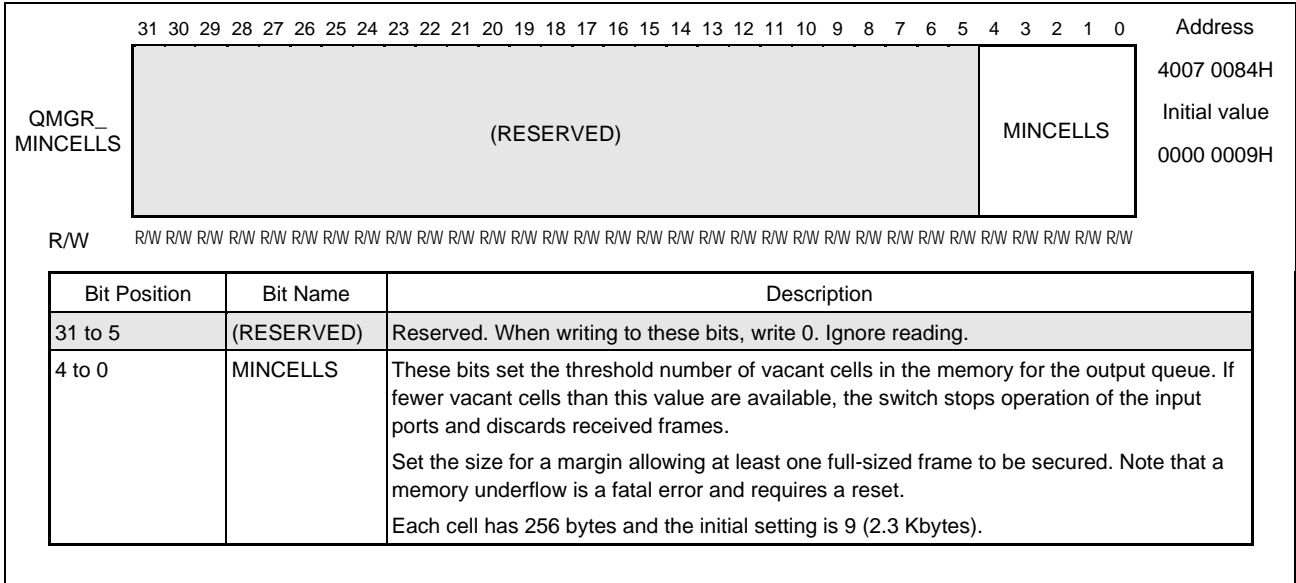
This register indicates the state of the output queue of the Ethernet switch. The latched values of bits 1 and 3 can be cleared by writing any value to the register. This register can be read or written in 32-bit units.

	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		
OQMGR_STATUS	CELLAVILABLE																(RESERVED)										DEQUEGRANT	(RESERVED)	MEMFULL_LT	MEMFULL	NOCELL	BUSYINIT	4007 0080H Initial value 0020 004AH		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	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.

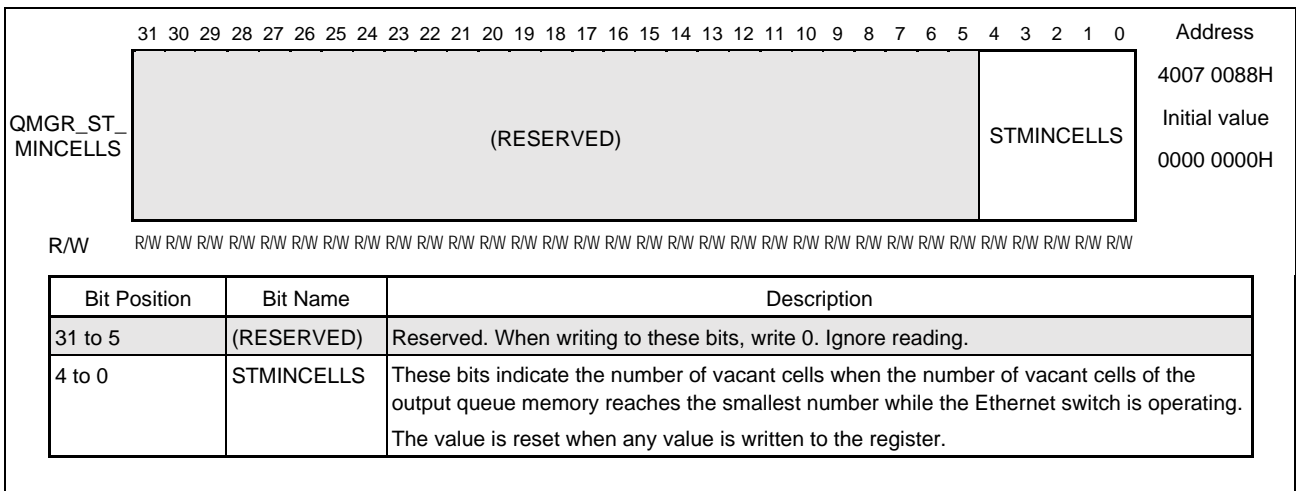
8.3.3.10 Output Queue Minimum Memory Register (QMGR_MINCELLS)

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



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



8.3.3.12 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. 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	
QMGR_CGS_STAT	(RESERVED)																												P2CGS	P1CGS	P0CGS	4007 008CH		
																																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	R	

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

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

QMGR_IFACE_STAT	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
	(RESERVED) P2RXFIFOAV P1RXFIFOAV P0RXFIFOAV (RESERVED) P2TXFIFOST P1TXFIFOST P0TXFIFOST	4007 0090H Initial value 0000 0007H
R/W	R R	

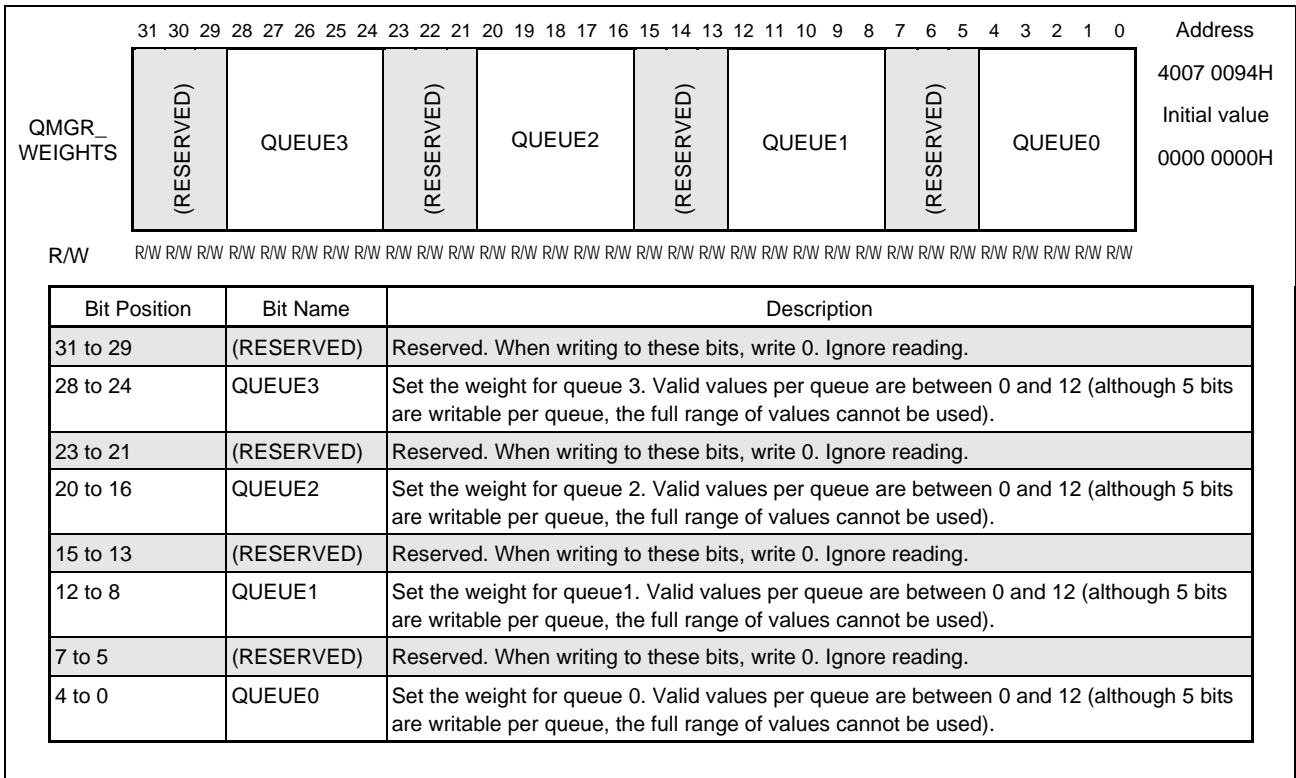
Bit Position	Bit Name	Description
31 to 19	(RESERVED)	Reserved. Ignore reading.
18	P2RXFIFOAV	Indicates whether RX FIFO data for the internal interface port (port 2) are available 0: Not available 1: Available
17	P1RXFIFOAV	Indicates whether RX FIFO data for port 1 are available 0: Not available 1: Available
16	P0RXFIFOAV	Indicates whether RX FIFO data for port 0 are available 0: Not available 1: Available
15 to 3	(RESERVED)	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

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



8.3.3.15 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. 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.

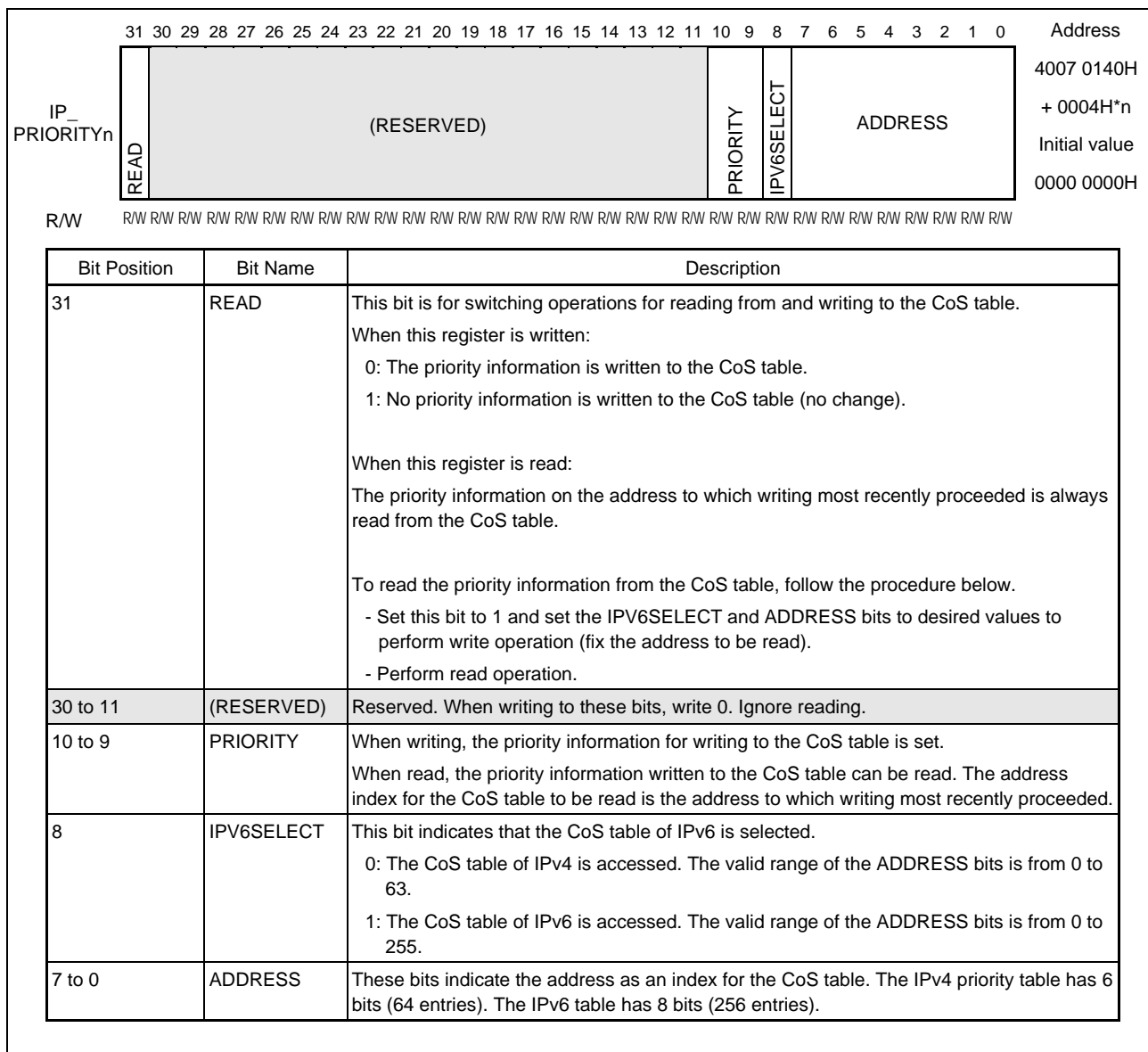
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	
VLAN_PRIORITYn	(RESERVED)									PRIORITY7	PRIORITY6	PRIORITY5	PRIORITY4	PRIORITY3	PRIORITY2	PRIORITY1	PRIORITY0										4007 0100H + 0004H*n Initial value 0000 0000H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	(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)

8.3.3.16 IP Priority Register (IP_PRIORITY_n)

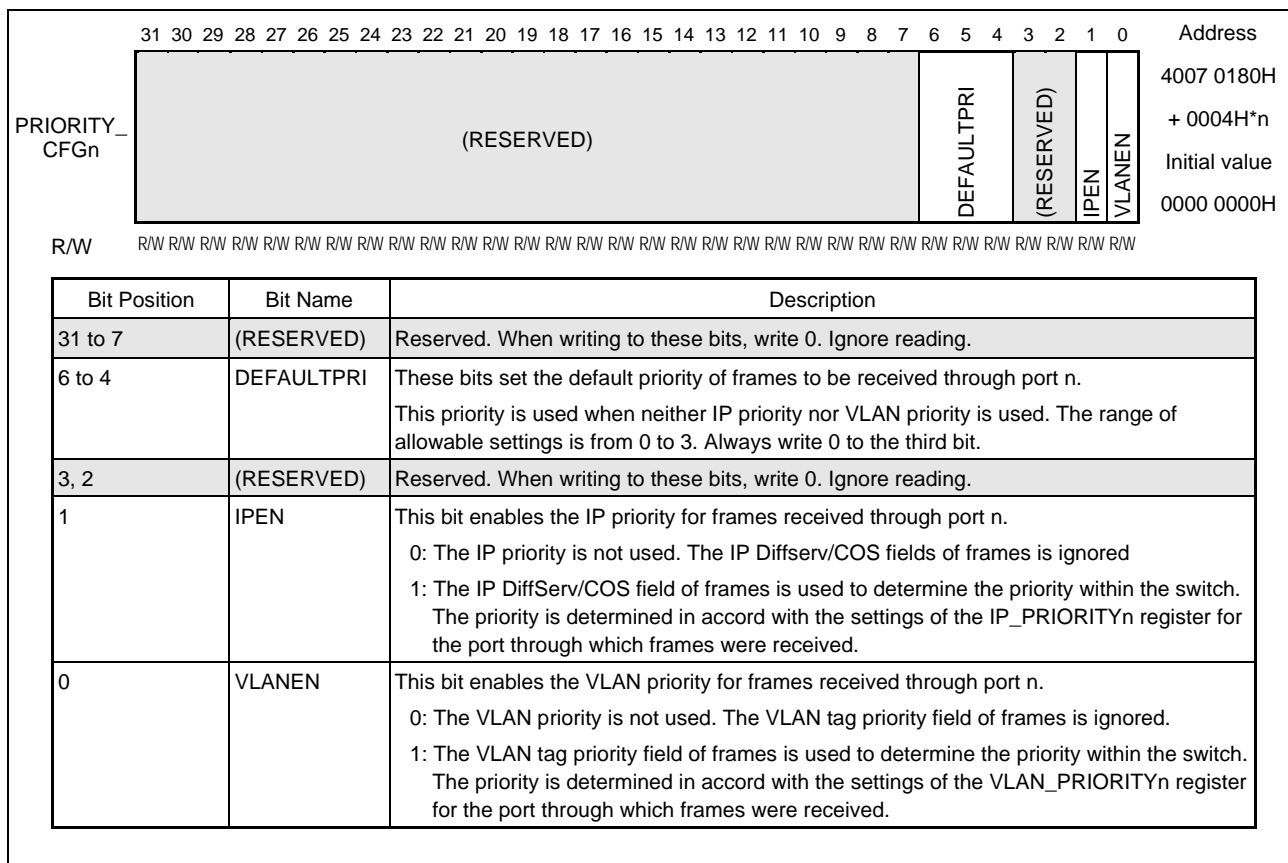
The Ethernet switch has a CoS (class of service) table of IPv4 and IPv6 for each port of the switch. On the IPv4 CoS table, the 6-bit DiffServ field included in frames is used as an index for the lookup table and the 2-bit priority information can be set. On the IPv6 CoS table, the 8-bit CoS field included 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. This register can be read or written in 32-bit units.



Remark: n = 0 to 2
n = 0: Port 0, n=1: Port 1, n = 2: Internal interface port (port 2)

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



Remark: n = 0 to 2
n = 0: Port 0, n=1: Port 1, n = 2: Internal interface port (port 2)

8.3.3.18 HUB Control Register (HUB_CONTROL)

This register selects the hub operation. Enabling the hub allows cut-through transfer at high speed. This register can be read or written in 32-bit units.

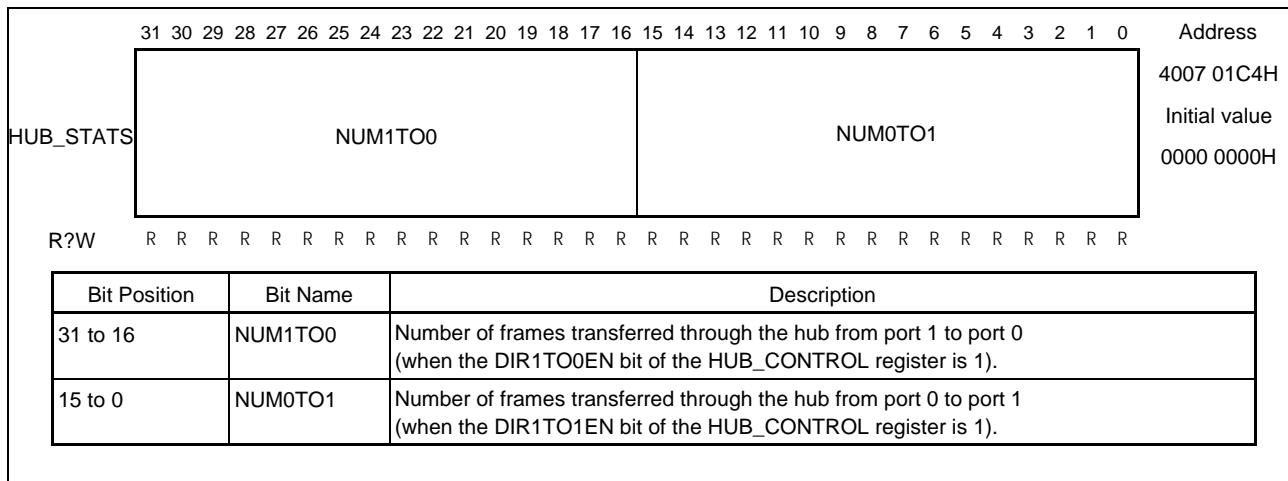
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address		
HUB_ CONTROL	(RESERVED)																								HUBIPG				BROCAFILEN	DIR1TO0EN	DIR0TO1EN	HUBEN	4007 01C0H		
																																	Initial value		
																																	0000 00A0H		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 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.

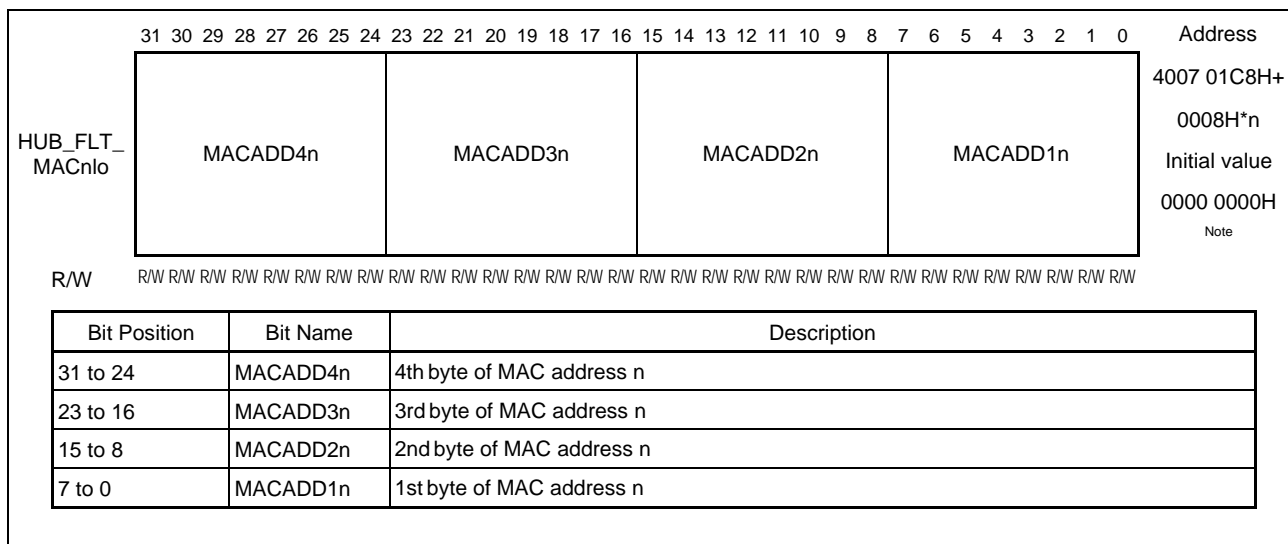
8.3.3.19 HUB Status Register (HUB_STATS)

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



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



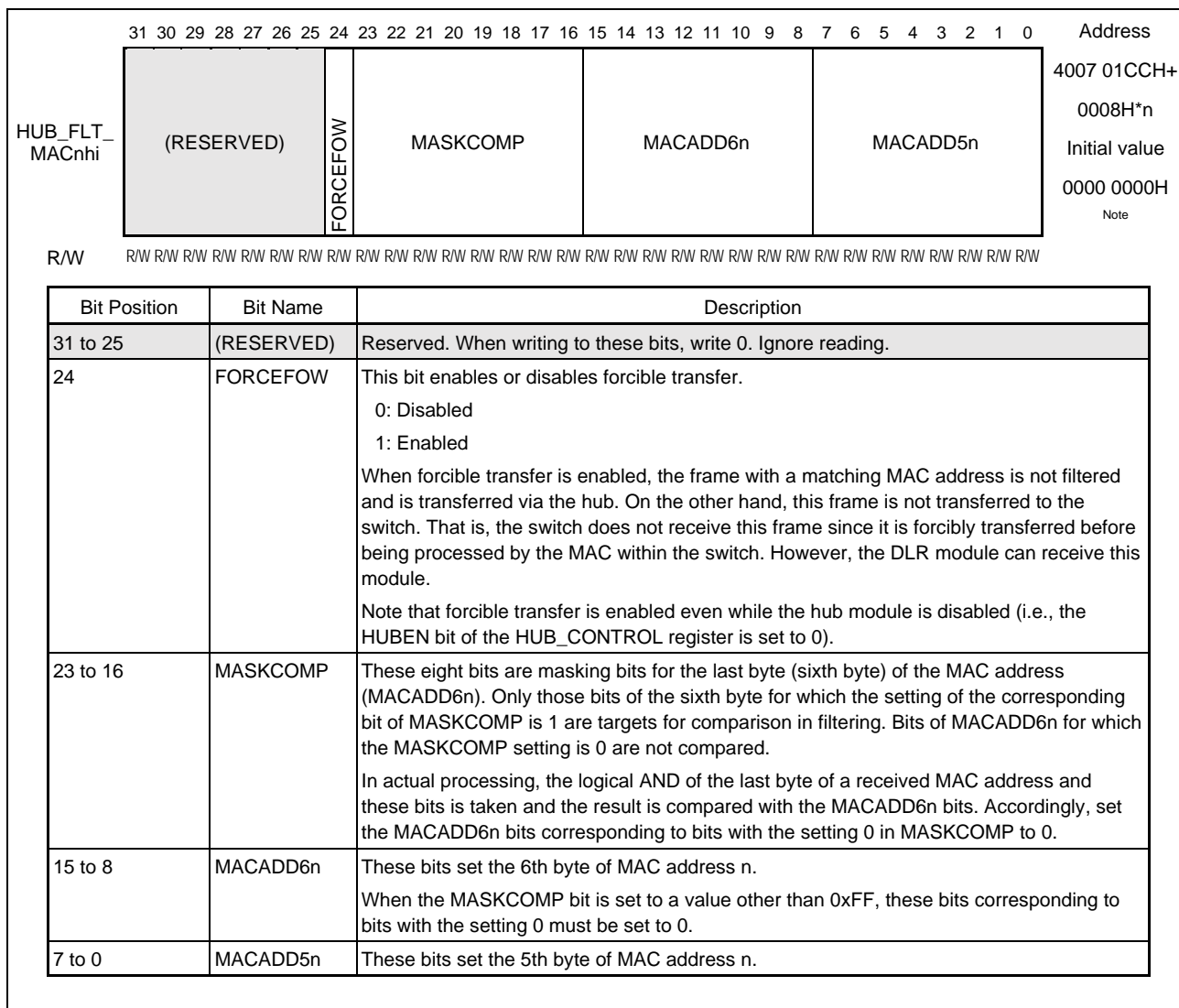
Remark: n = 0 to 6

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.

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



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.

8.3.3.22 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

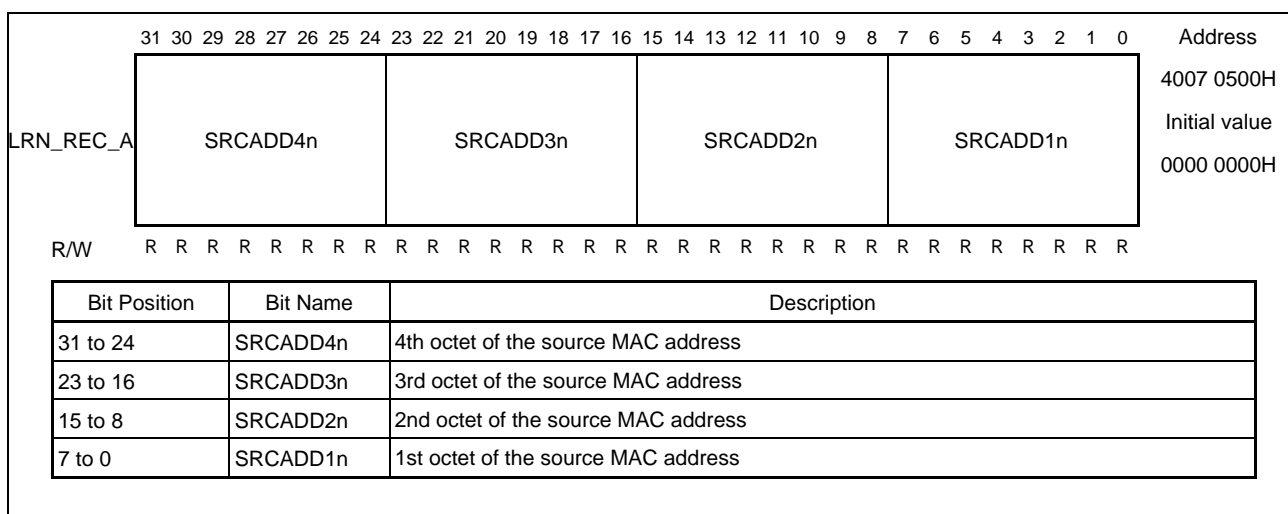
n = 0: Port 0, n=1: Port 1, n = 2: Internal interface port (port 2)

8.3.4 Learning Interface Registers

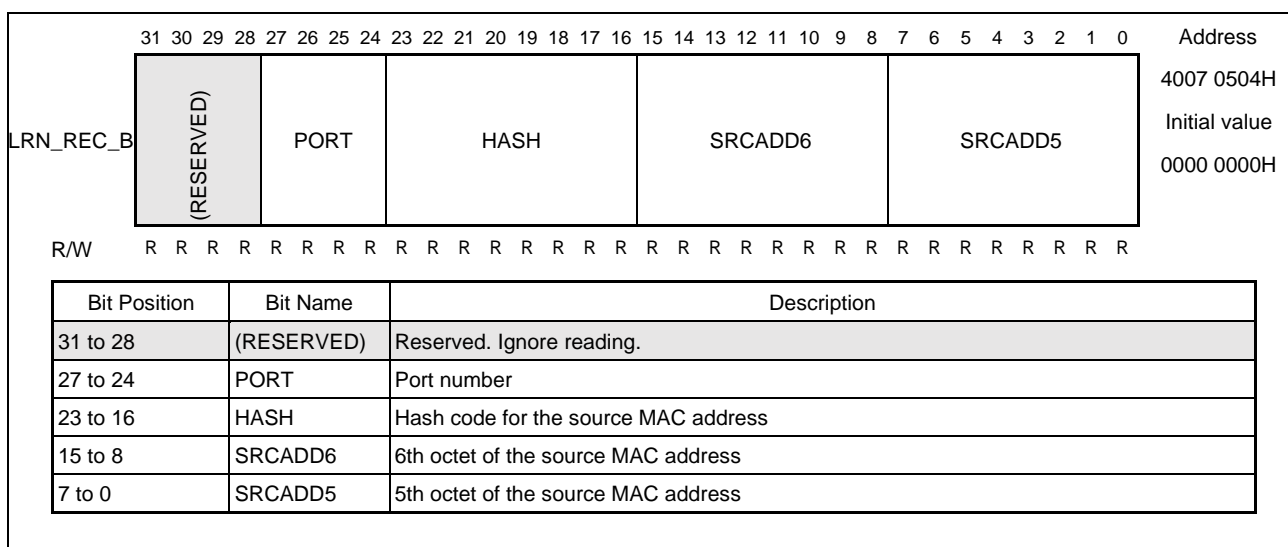
The source address and port information which the Ethernet switch has learned can be obtained through the learning interface. The information is used to construct a lookup table. The information can be obtained from the two registers, but the LRN_REC_A register must be read before the LRN_REC_B register.

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. These registers are readable in 32-bit units.

8.3.4.1 Learning Record A Register (LRN_REC_A)

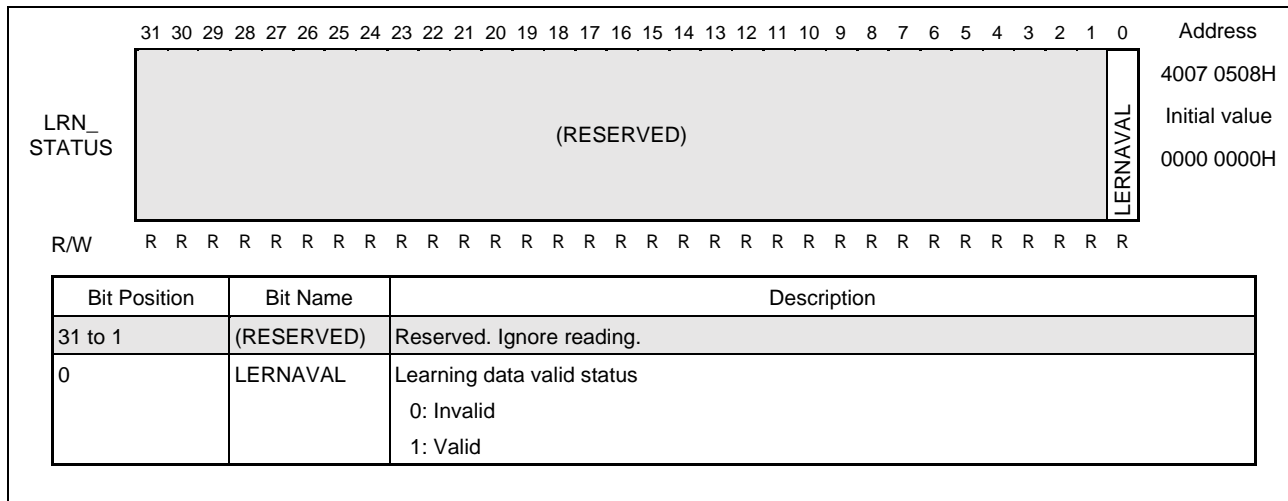


8.3.4.2 Learning Record B Register (LRN_REC_B)



8.3.4.3 Learning Data Status Register (LRN_STATUS)

This register indicates whether the values of the LRN_REC_A and LRN_REC_B registers are valid. This register is readable in 32-bit units.



8.3.4.4 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 8.4.1.4(3) and 8.5.3.

8.3.5 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.

8.3.5.1 Command Configuration Register n (COMMAND_CONFIGn)

These registers are used to set and reset the MAC. These register can be read or written in 32-bit units.

(1/2)

COMMAND_CONFIGn	31	CNTRESET	(RESERVED)	26	RXERRDISC	(RESERVED)	24	NOLGTHCHK	(RESERVED)	23	CNTRLREMEN	(RESERVED)	13	SWRESET	(RESERVED)	6	(RESERVED)	5	(RESERVED)	4	(RESERVED)	3	(RESERVED)	2	RXENA	1	TXENA	0		Address
	4007 8008H+	2000H*n		Initial value	0000 0010H																									
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	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 this bit, 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.

(2/2)

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

8.3.5.2 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. 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

4007 8014H+

2000H*n

Initial value

0000 05F2H

(RESERVED)

FRMLEN

R/W R/W

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

8.3.5.3 FIFO Buffer Threshold Register n (Shared)

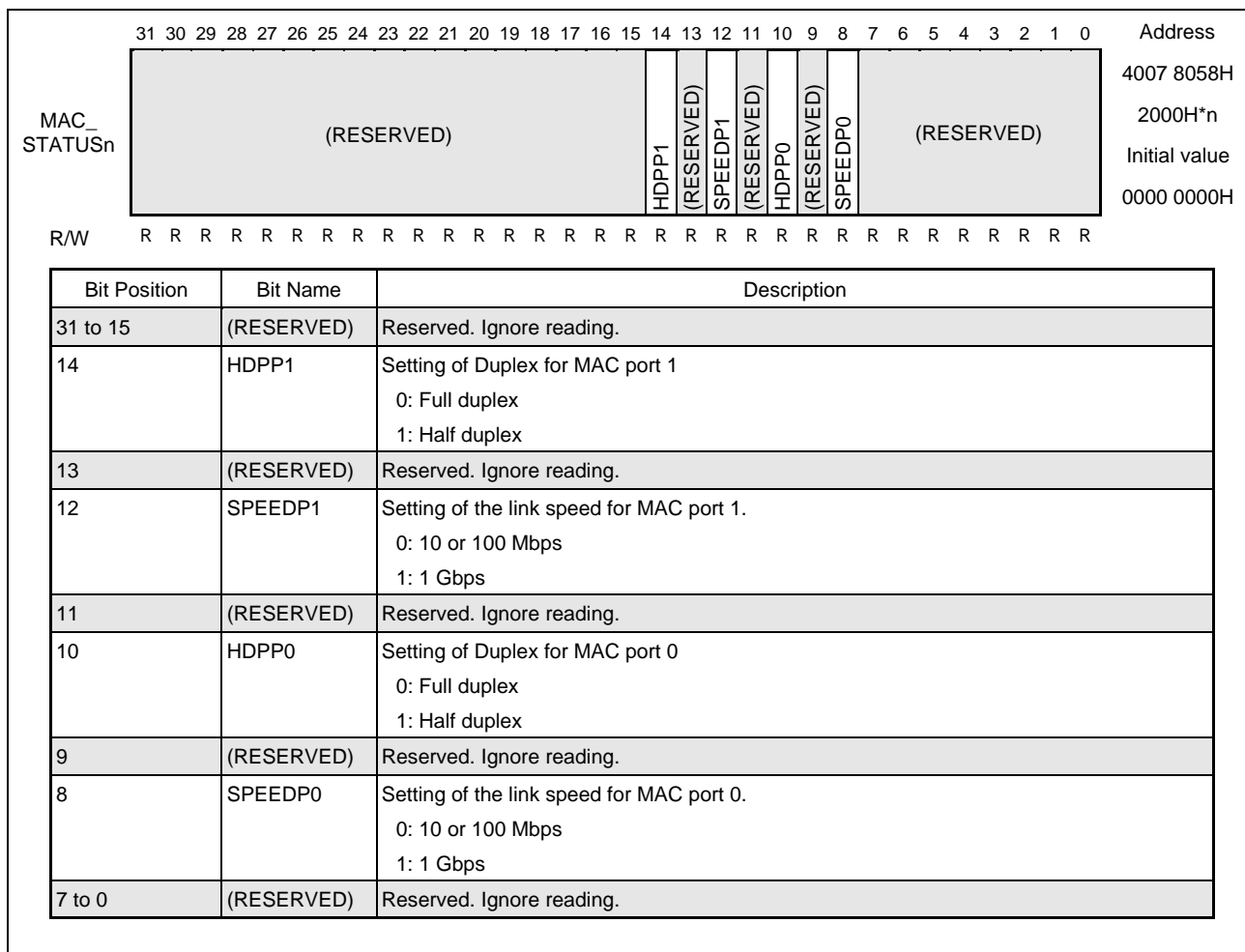
These registers set the threshold of the FIFO buffer of the MAC and manage overflow and underflow. Basically, there is no need to change the initial value.

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

8.3.5.4 MAC Status Register (MAC_STATUSn) (Shared)

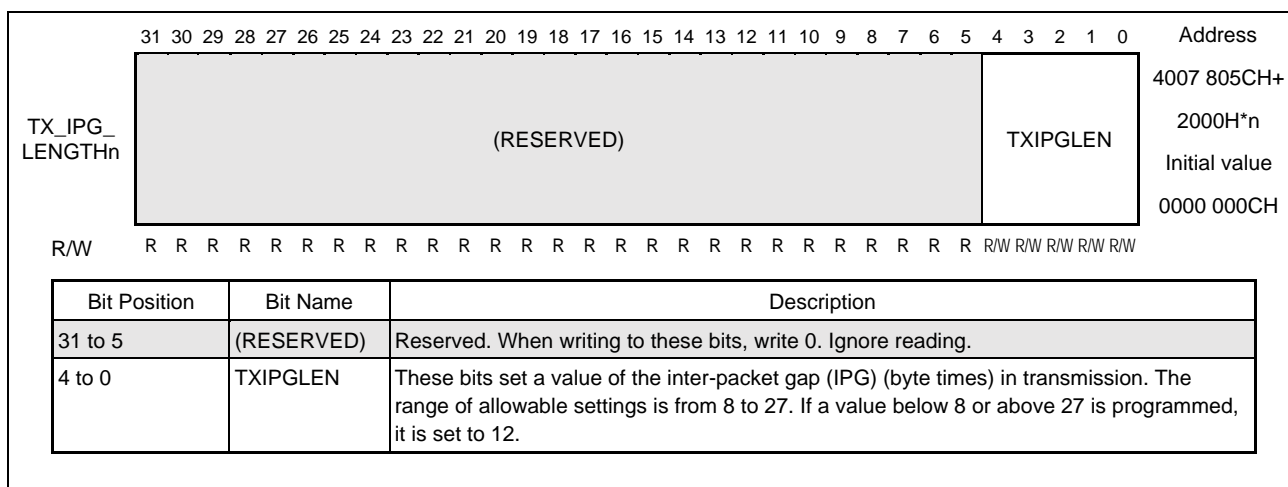
These registers indicate the communications settings for the MAC. These registers can be read or written in 32-bit units.



Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

8.3.5.5 Transmit IPG Length Register n (TX_IPG_LENGTHn) (Shared)

These registers set the inter-packet gap (IPG) in transmission. These registers can be read or written in 32-bit units.



Remark: n = 0, 1
 n = 0: MAC port 0, n = 1: MAC port 1

8.3.5.6 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. These registers can be read or written in 32-bit units.

(a) MAC RX Statistic Counters

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 iflnOctets 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

(2/3)

Address	Symbol	Description
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.
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_LENTHn)
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_LENTHn) and a normal CRC

(3/3)

Address	Symbol	Description
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

(b) 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.

(2/2)

Address	Symbol	Description
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
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_LENTHn)
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_LENTHn) 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_LENTHn) 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_LENTHn) (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

8.3.6 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.

8.3.6.1 Timer Module Configuration Register (TSM_CONFIG)

This register controls generation of interrupts in response to the event that was generated in timestamping. This register can be read or written in 32-bit units.

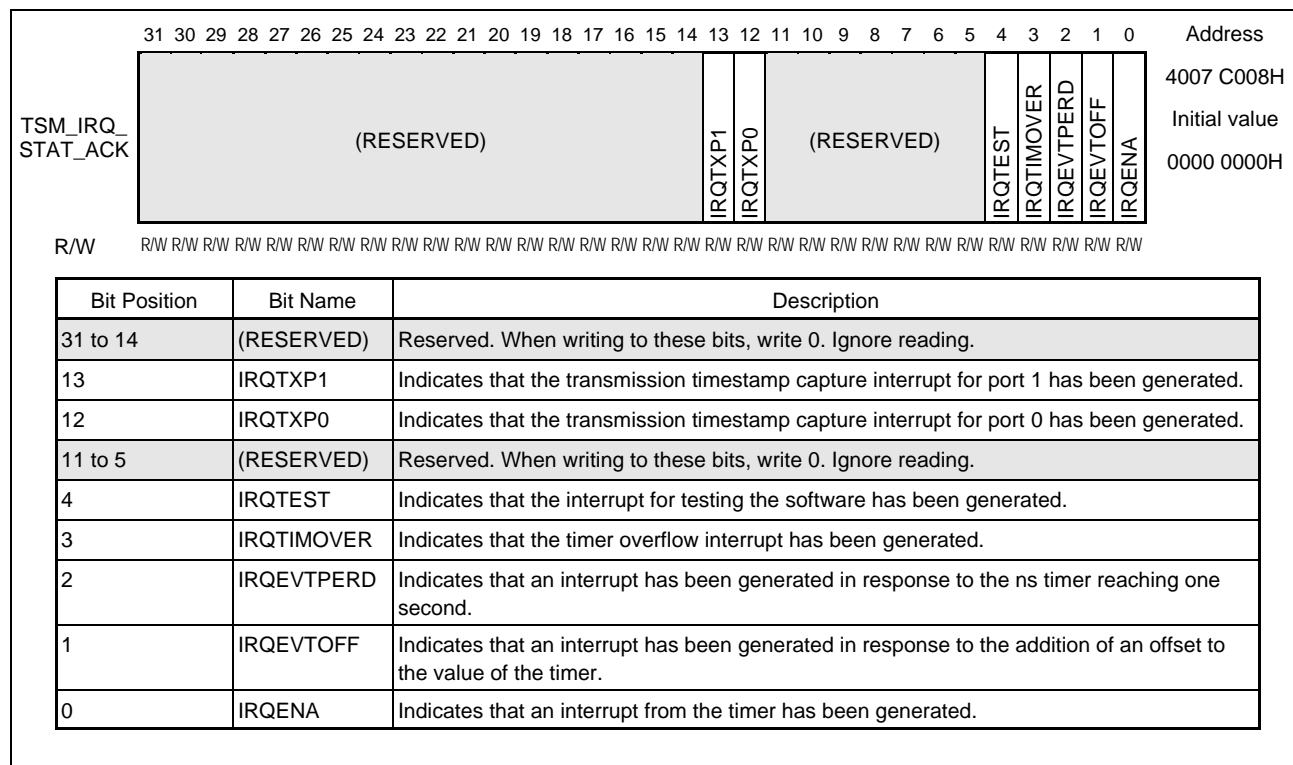
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
TSM_CONFIG		(RESERVED)																		IRQTXENAP1	IRQTXENAP0	(RESERVED)					IRQTEST	IRQTIMOVER	IRQEVTPERD	IRQEVTOFF	IRQENA	4007 C004H		
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 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																																

8.3.6.2 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.

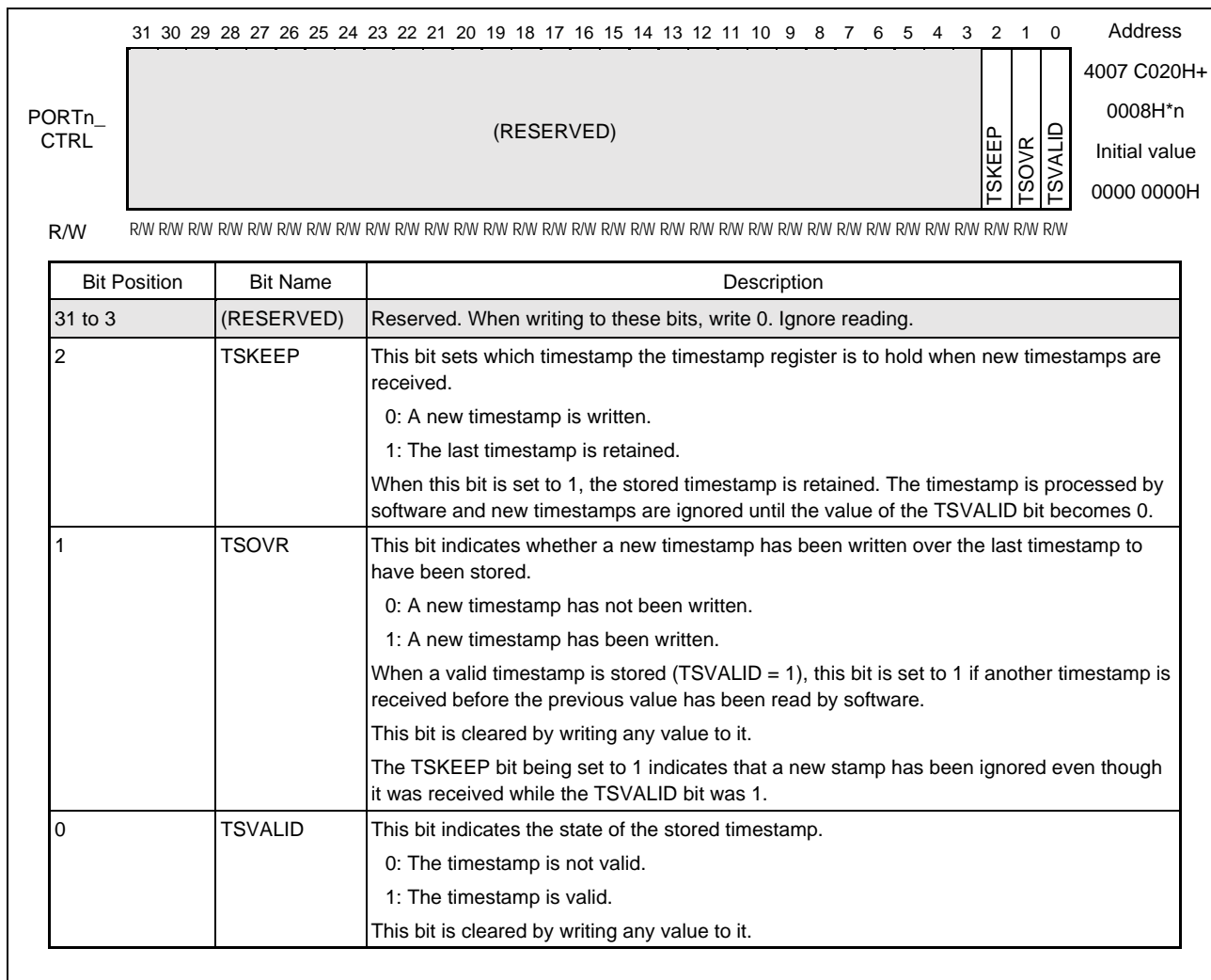
The state is confirmed by reading the value of this register. A value of 1 means that the interrupt has been generated and 0 means that the interrupt has not been generated.

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



8.3.6.3 Port Timestamp Control/Status Register (PORTn_CTRL)

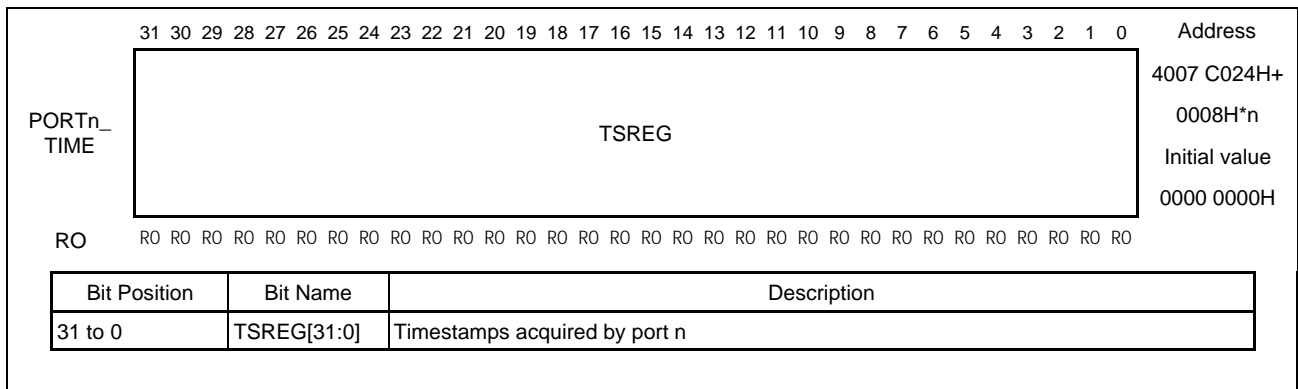
This register sets the method of storing timestamps acquired by port n in the timestamp register and indicates the state of the stored timestamp. This register can be read or written in 32-bit units.



Remark: n = 0, 1
n = 0: MAC port 0, n = 1: MAC port 1

8.3.6.4 Port Timestamp Register (PORTn_TIME)

This register holds timestamps acquired by port n.



Remark: n = 0, 1
 n = 0: MAC port 0, n = 1: MAC port 1

8.3.6.5 Timer Control Register (ATIME_CTRL)

This register is used to set timer interrupt events and control timers. This register can be read or written in 32-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			
ATIME_CTRL	(RESERVED)																				PLUS1	CAPTR	(RESERVED)	RST	(RESERVED)	PINPERIENA	(RESERVED)	EVTPERIRST	EVTPERIENA	(RESERVED)	EVTOFFENA	(RESERVED)	TMENA			4007 C120H 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	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	(RESERVED)	Reserved. When writing to this bit, write 0. Ignore reading.
7	PINPERIENA	This bit enables or disables assertion of the signal on the ETHSWSECOUT pin in response to periodic events. 0: Disabled 1: Enabled Usually set this bit to 1.
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. An EtherSWITCH SEC interrupt is generated and the signal on the ETHSWSECOUT pin is asserted. 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.

(2/2)

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**
- 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.
 - The timer value cannot be captured accurately while the timer is stopped. The captured value is invalid.

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

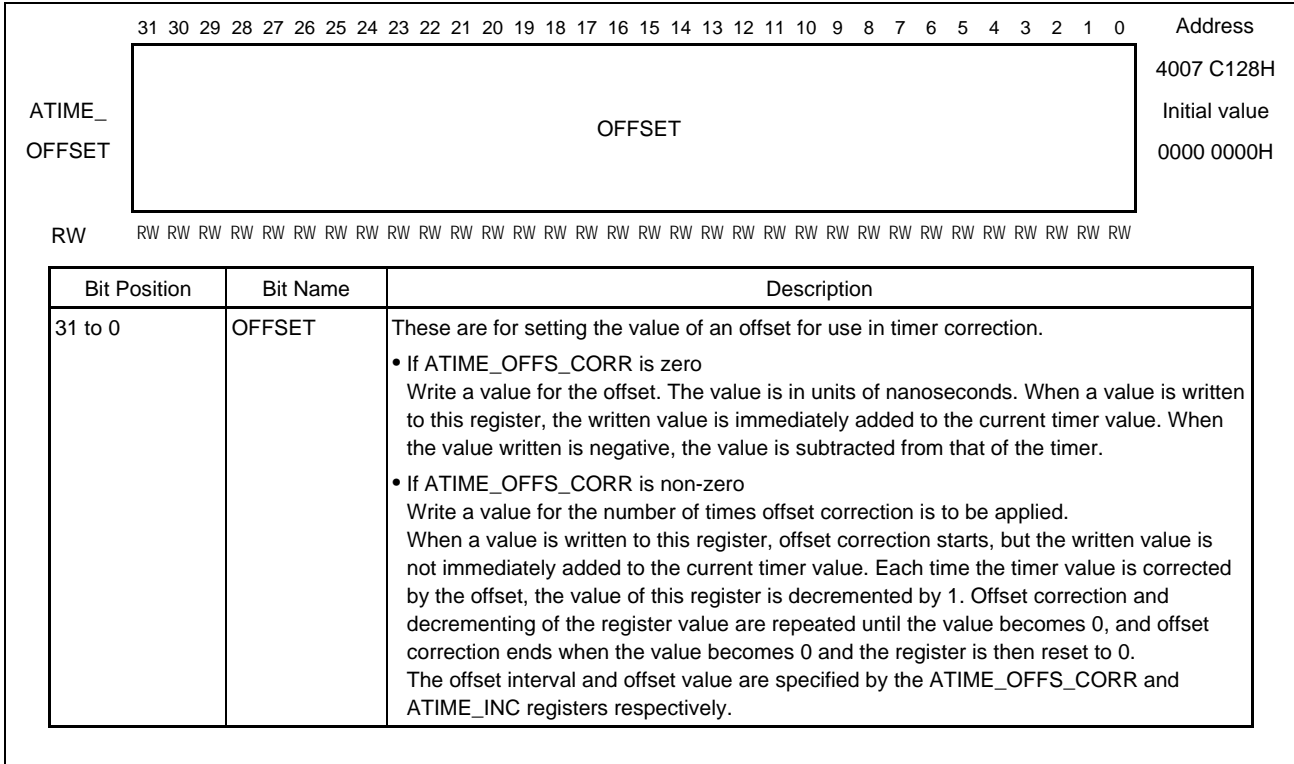
	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Address
		4007 C124H
		Initial value
		0000 0000H
ATIME	TMR	
RW	RW RW	
Bit Position	Bit Name	Description
31 to 0	TMR[31:0]	These bits represent the setting of the nanoseconds timer and the value read. When this register 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.

8.3.6.7 Timer Offset Correction Register (ATIME_OFFSET)

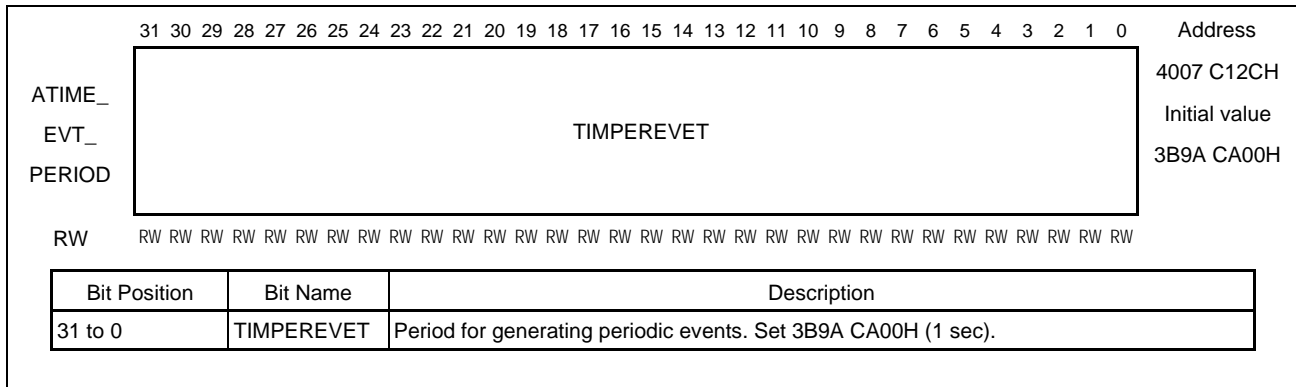
This register sets the value of an offset for use in timer correction. There are two methods of offset correction and the value to be set in this register will differ with the method.

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



8.3.6.8 Generate Timer Periodic Event Register (ATIME_EVT_PERIOD)

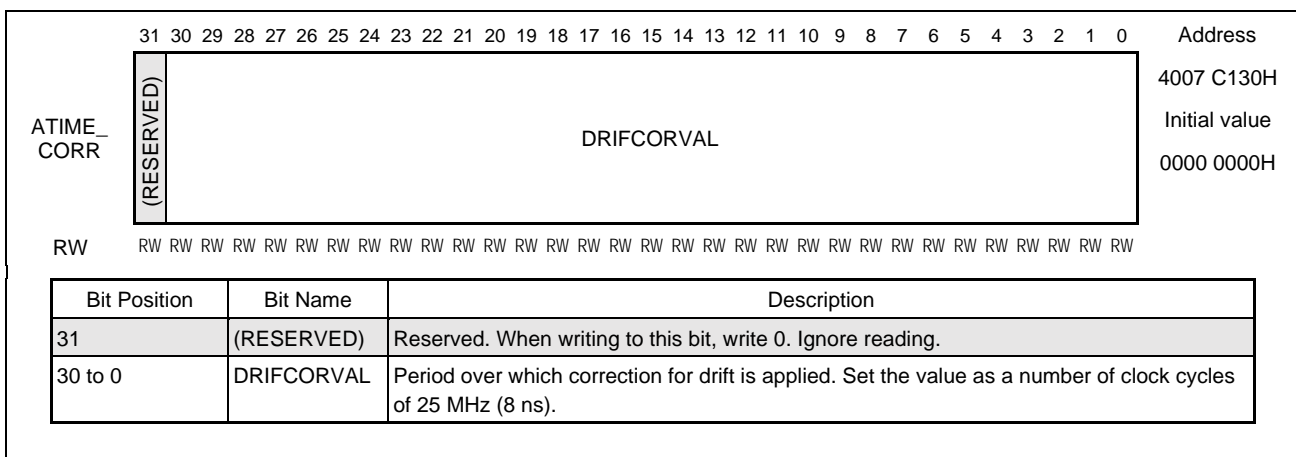
This register sets the period for generating periodic events. Each time the nanoseconds timer has reached this time, the period event occurs and the nanoseconds timer restarts. The value is in units of nanoseconds (nsec). The initial value is 10^9 [nsec]=1 [sec]. This register can be read or written in 32-bit units.



Caution: The value of periodic events is fixed to one second and cannot be changed. If changed, the timer will not operate normally.

8.3.6.9 Timer Drift Correction Register (ATIME_CORR)

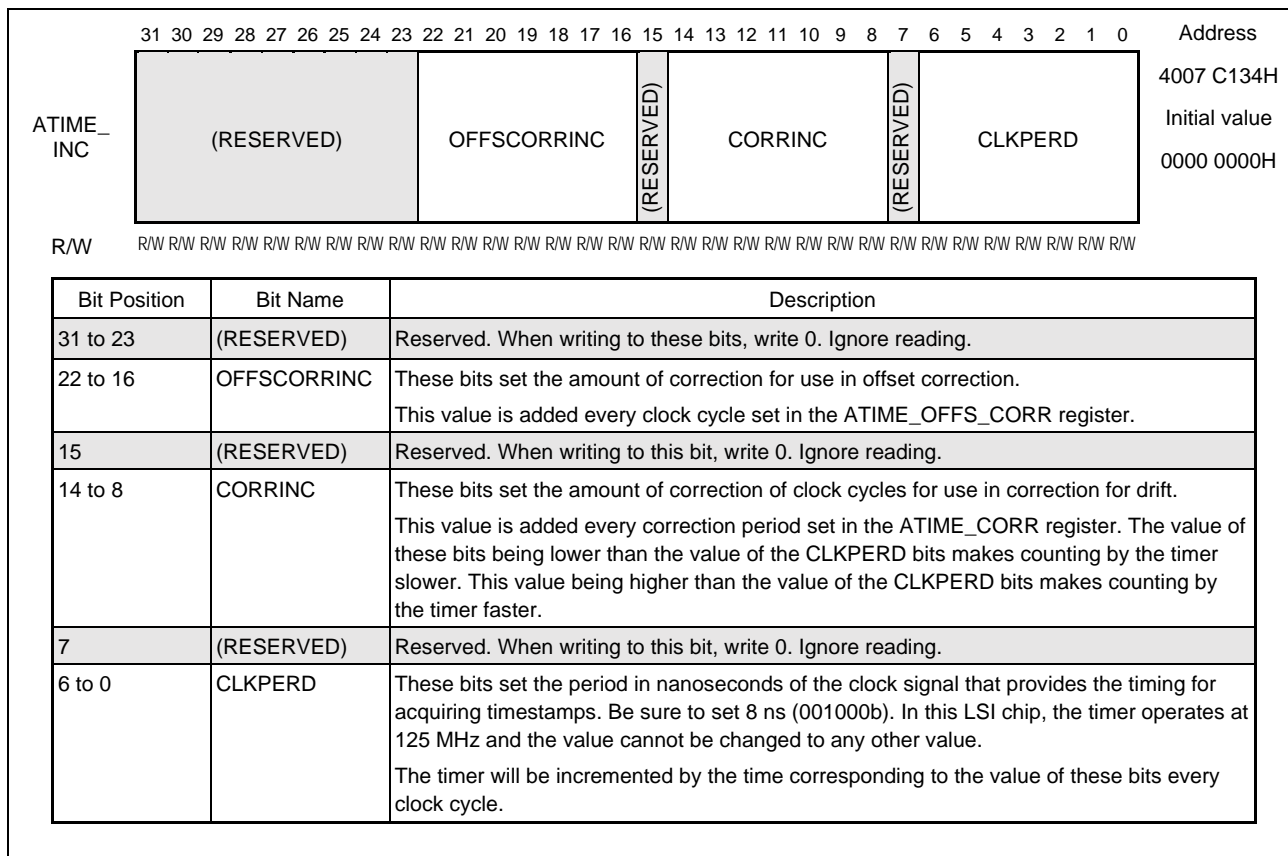
This register sets the correction period over which correction for drift is applied as a number of clock cycles. Use the ATIME_INC register to specify the amount of correction. This register can be read or written in 32-bit units.



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.

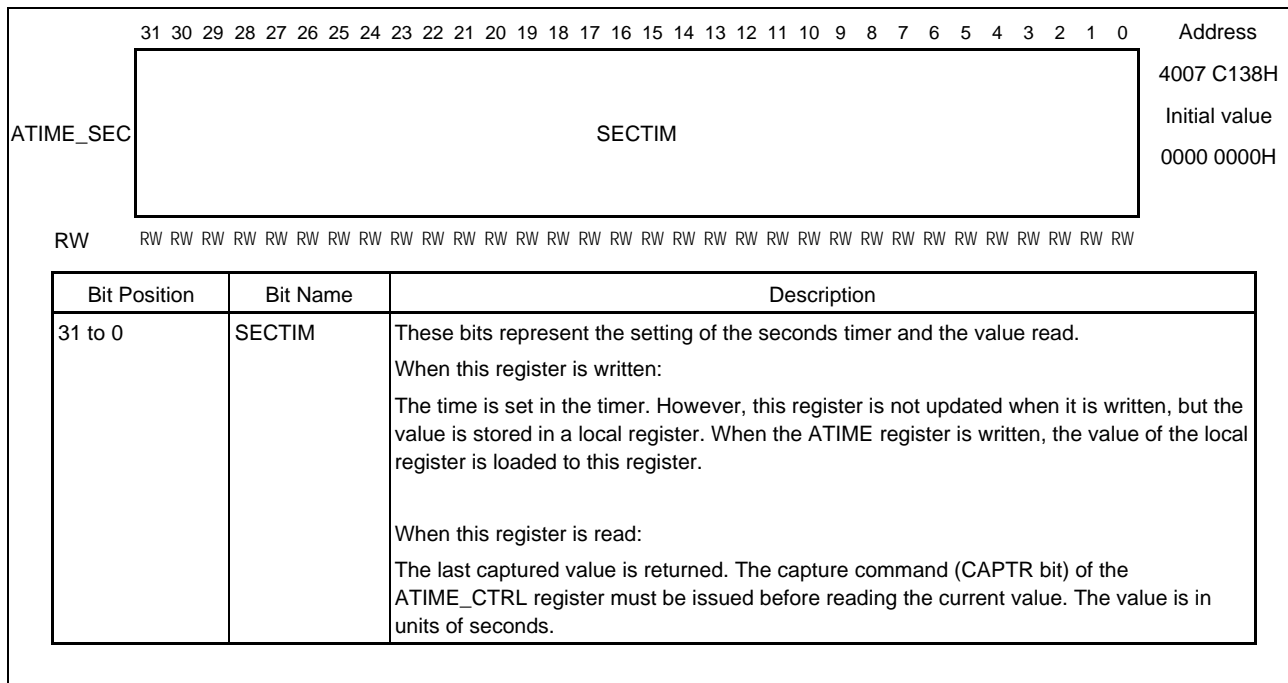
8.3.6.10 Timer Increment Register (ATIME_INC)

This register sets the amount of correction for used in correction of the timer value by the offset and correction for drift. This register can be read or written in 32-bit units.



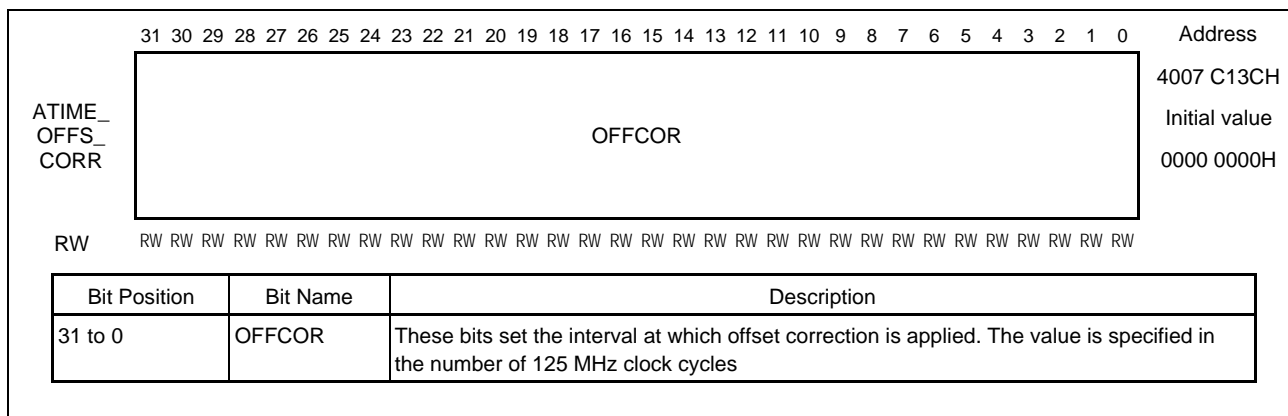
8.3.6.11 Timer Second Register (ATIME_SEC)

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 10⁹ [nsec]. This register can be read or written in 32-bit units.



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



8.3.7 DLR Module Registers

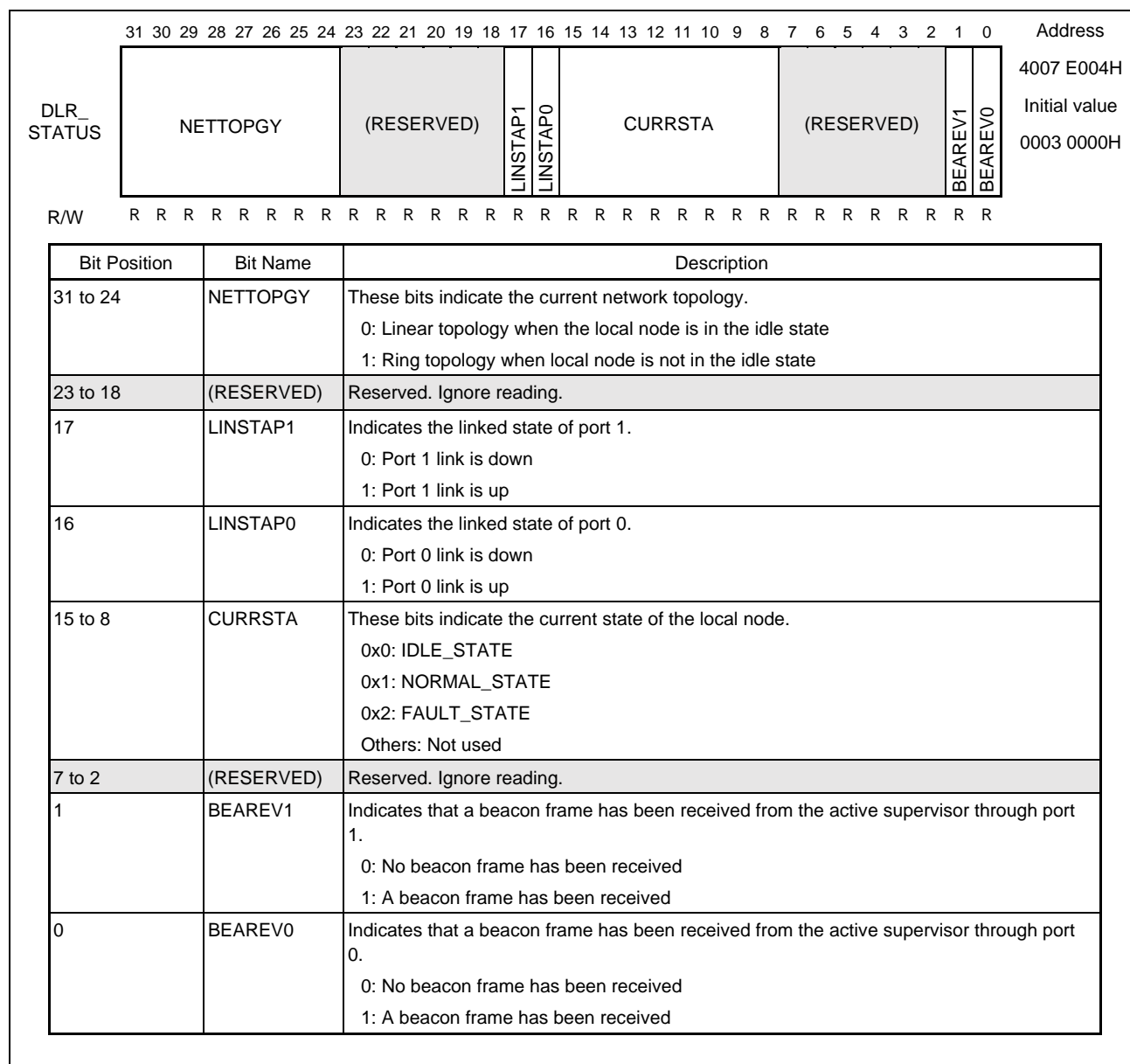
8.3.7.1 DLR Control Register (DLR_CONTROL)

This register is used to make settings for the DLR operation. 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	
DLR_CONTROL	(RESERVED)																CYCMCLK				(RESERVED)	BECTIMOUT	(RESERVED)	DLRENA	4007 E000H Initial value 0000 3200H									
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	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																																

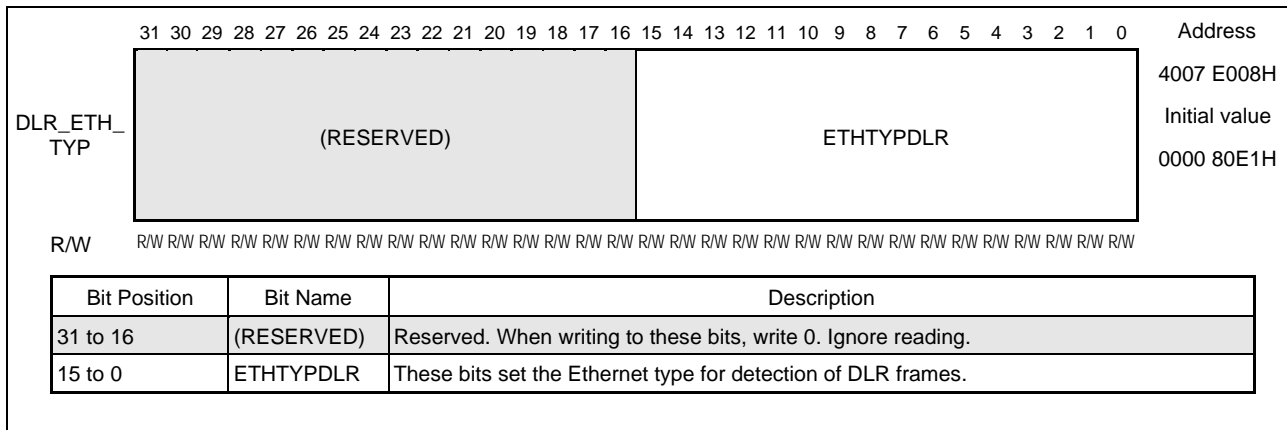
8.3.7.2 DLR Status Register (DLR_STATUS)

This register indicates the state of the DLR ring node. This register is only readable in 32-bit units.



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



8.3.7.4 DLR Interrupt Control Register (DLR_IRQ_CTRL)

(1/2)

DLR_IRQ_CTRL	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																			
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">ATOMICAND</td> <td style="width: 20px; text-align: center;">ATOMICOR</td> <td style="width: 20px; text-align: center;">(RESERVED)</td> <td style="width: 20px; text-align: center;">(RESERVED)</td> <td style="width: 20px; text-align: center;">IRQFRMDSP1</td> <td style="width: 20px; text-align: center;">IRQFRMDSP0</td> <td style="width: 20px; text-align: center;">IRQBECENA1</td> <td style="width: 20px; text-align: center;">IRQBECENA0</td> <td style="width: 20px; text-align: center;">IRQINVTMREN</td> <td style="width: 20px; text-align: center;">IRQIPADDREN</td> <td style="width: 20px; text-align: center;">IRQSUPIGENA</td> <td style="width: 20px; text-align: center;">IRQLINKENA1</td> <td style="width: 20px; text-align: center;">IRQLINKENA0</td> <td style="width: 20px; text-align: center;">IRQSUFENA</td> <td style="width: 20px; text-align: center;">IRQBECENA1</td> <td style="width: 20px; text-align: center;">IRQBECENA0</td> <td style="width: 20px; text-align: center;">IRQSTOPP1</td> <td style="width: 20px; text-align: center;">IRQSTOPP0</td> <td style="width: 20px; text-align: center;">IRQFLUENA</td> <td style="width: 20px; text-align: center;">IRQCHGENA</td> </tr> </table>	ATOMICAND	ATOMICOR	(RESERVED)	(RESERVED)	IRQFRMDSP1	IRQFRMDSP0	IRQBECENA1	IRQBECENA0	IRQINVTMREN	IRQIPADDREN	IRQSUPIGENA	IRQLINKENA1	IRQLINKENA0	IRQSUFENA	IRQBECENA1	IRQBECENA0	IRQSTOPP1	IRQSTOPP0	IRQFLUENA	IRQCHGENA
ATOMICAND	ATOMICOR	(RESERVED)	(RESERVED)	IRQFRMDSP1	IRQFRMDSP0	IRQBECENA1	IRQBECENA0	IRQINVTMREN	IRQIPADDREN	IRQSUPIGENA	IRQLINKENA1	IRQLINKENA0	IRQSUFENA	IRQBECENA1	IRQBECENA0	IRQSTOPP1	IRQSTOPP0	IRQFLUENA	IRQCHGENA		
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.																			

(2/2)

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.

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

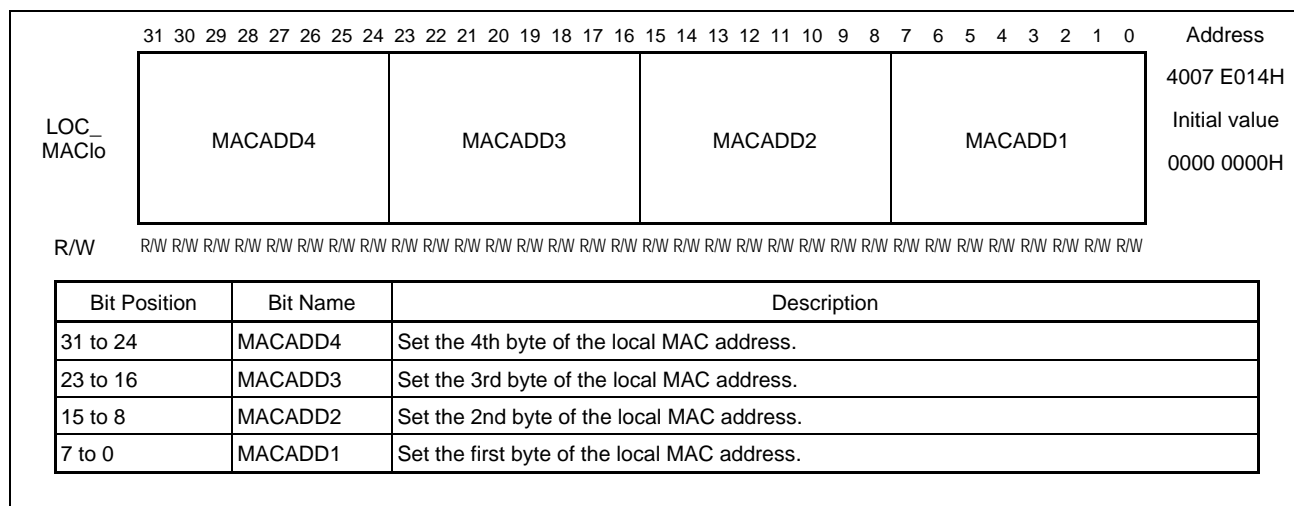
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
DLR_IRQ_STAT_ACK	(RESERVED)																FRMDS1	FRMDS0	BECEN1	BECEN0	INVTMR	IPCHANEVET	SUPIGNBEC	LINKSTAP1	LINKSTAP0	SUPRCHAG	BECTMRP1	BECTMRP0	STOPNBCHK1	STOPNBCHK0	FLUEVENT	STACHANGE	4007 E010H Initial value 0000 0180H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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.

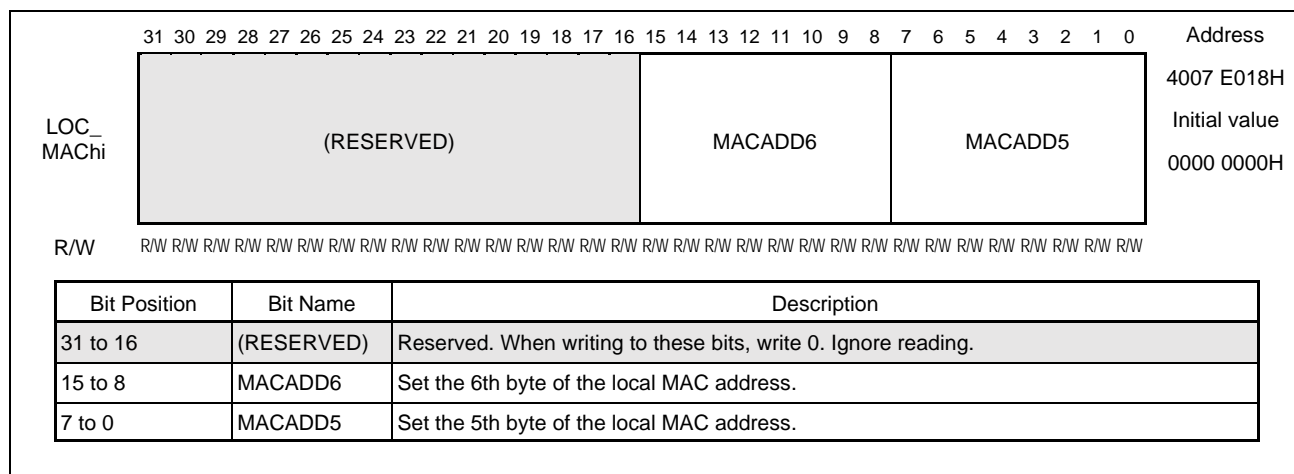
8.3.7.6 DLR Local MAC Address Low Register (LOC_MAClo)

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



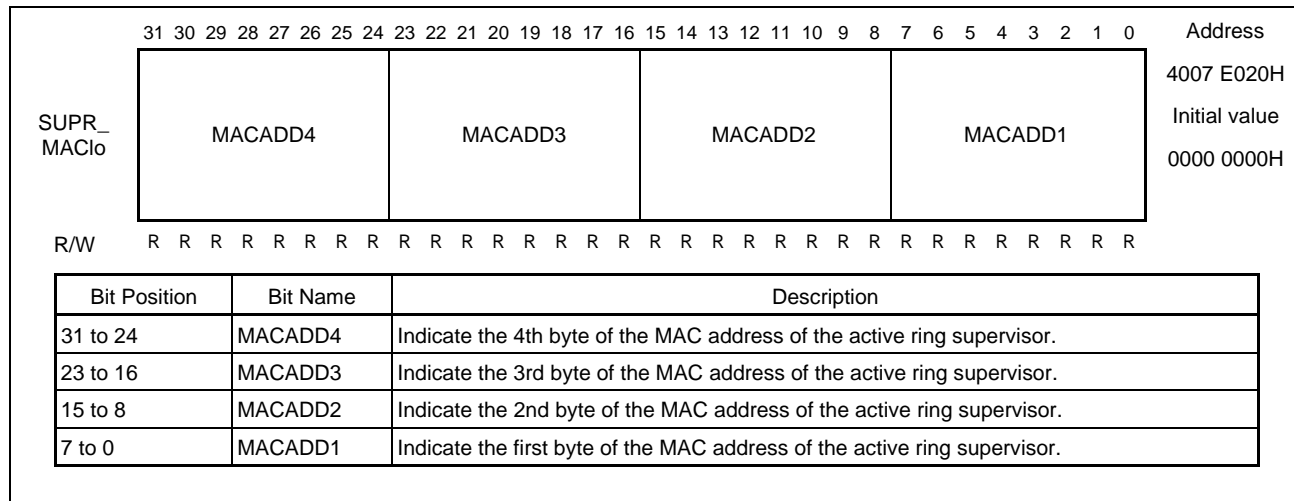
8.3.7.7 DLR Local MAC Address High Register (LOC_MACHi)

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



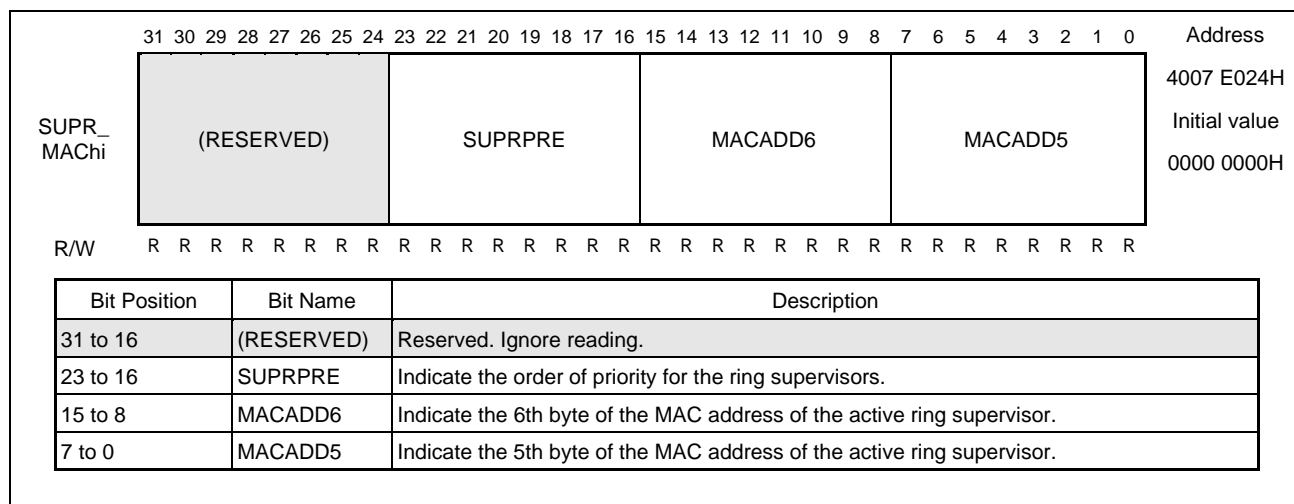
8.3.7.8 DLR Supervisor MAC Address Low Register (SUPR_MACLo)

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. This register is only readable in 32-bit units.



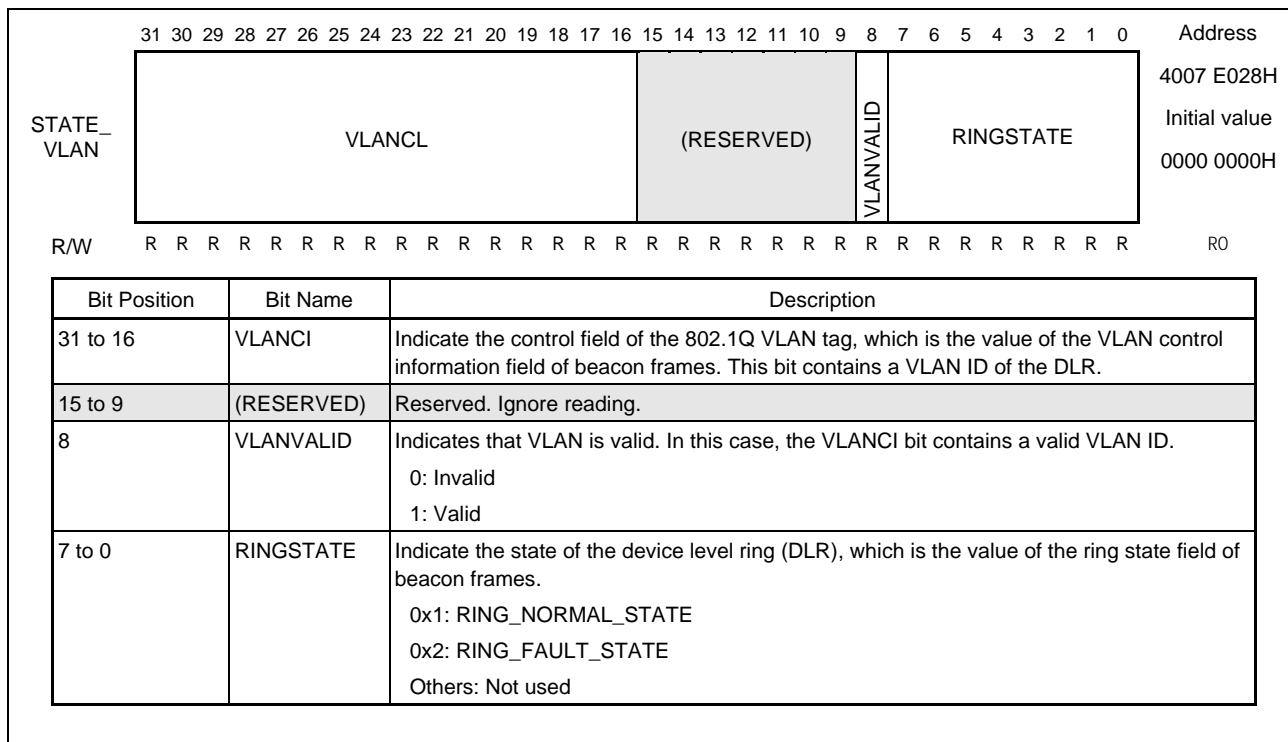
8.3.7.9 DLR Supervisor MAC Address High Register (SUPR_MACHi)

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. This register is only readable in 32-bit units.



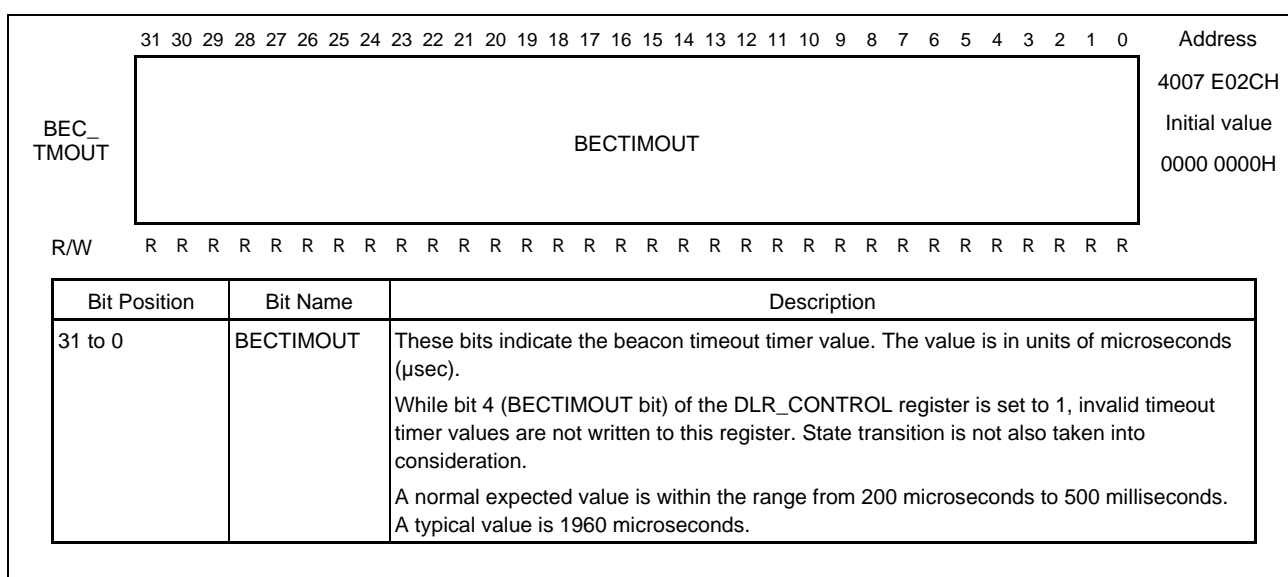
8.3.7.10 DLR Ring Status/VLAN Register (STATE_VLAN)

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. This register is only readable in 32-bit units.



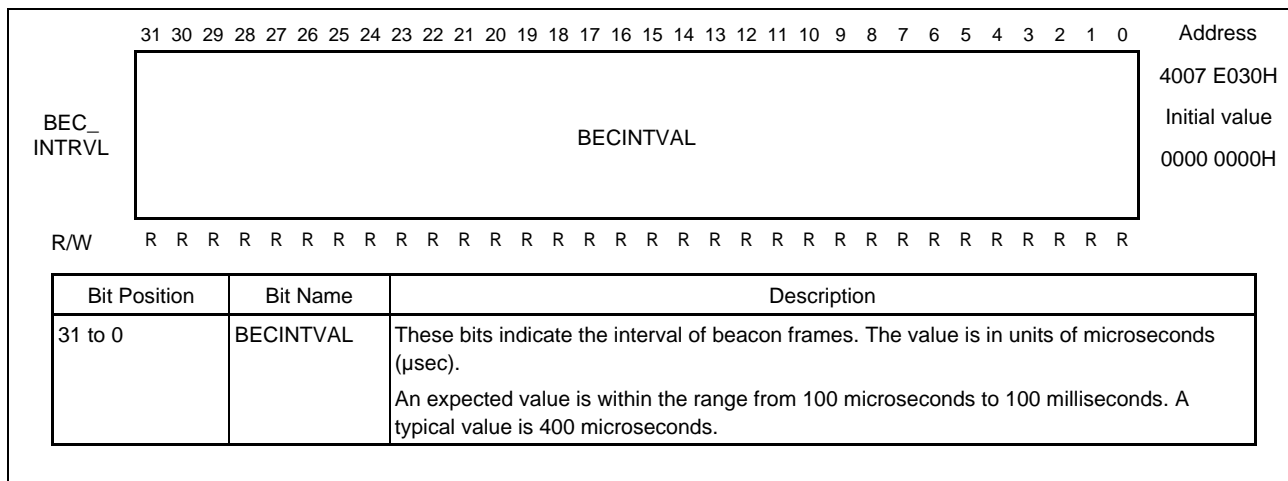
8.3.7.11 DLR Beacon Timeout Register (BEC_TMOUT)

This register indicates the timeout timer value of beacon frames. This is extracted from the beacon timeout field of beacon frames. This register is only readable in 32-bit units.



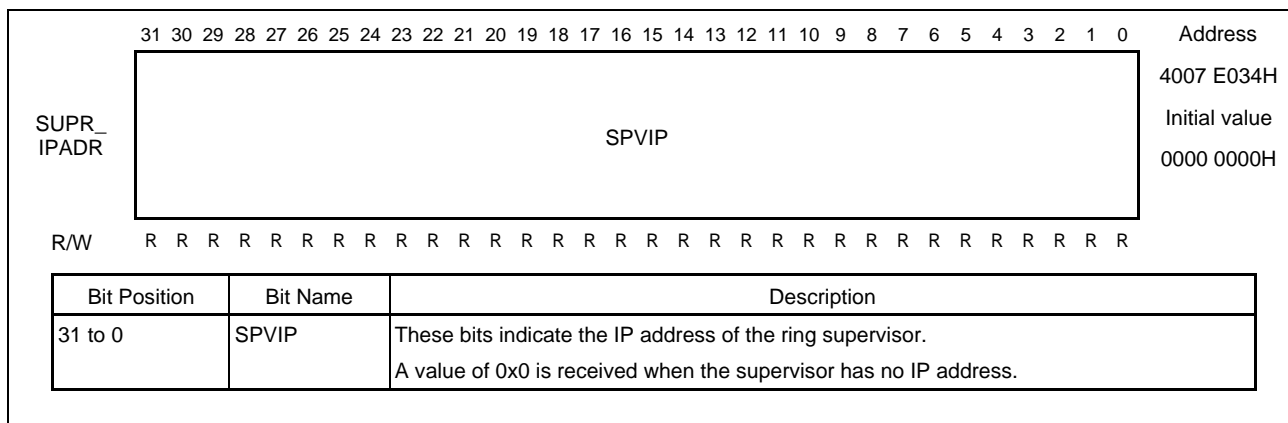
8.3.7.12 DLR Beacon Interval Register (BEC_INTRVL)

This register indicates the interval of beacon frames. This is extracted from the beacon interval field of beacon frames. This register is only readable in 32-bit units.



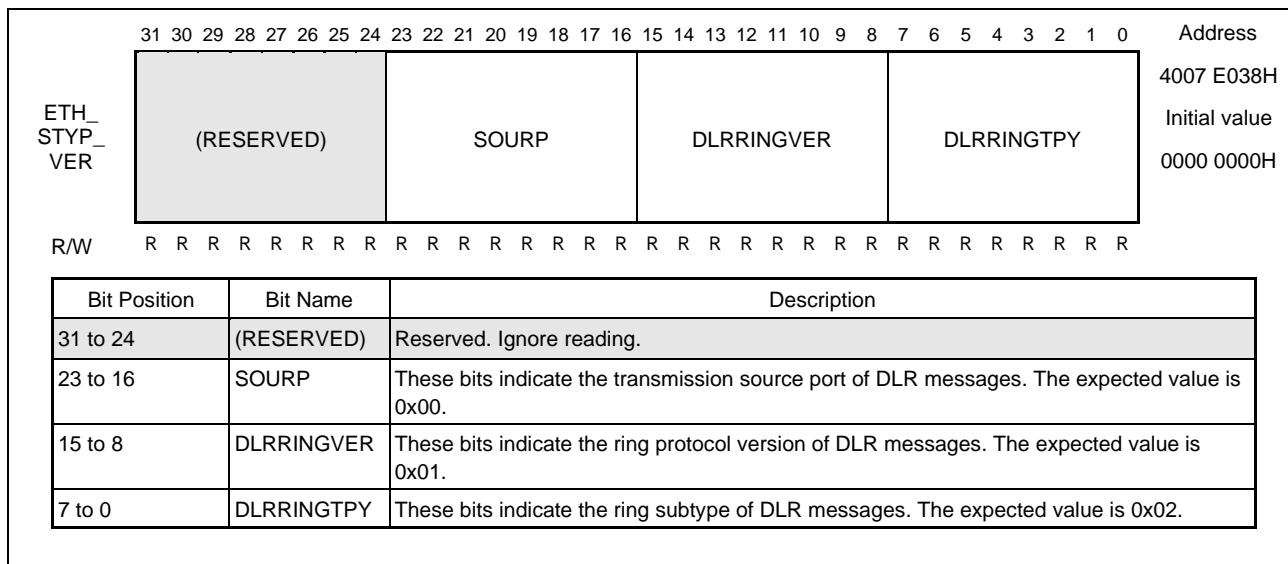
8.3.7.13 DLR Supervisor IP Address Register (SUPR_IPADR)

This register indicates the IP address of the ring supervisor. This is extracted from the source IP address field of beacon frames. This register is only readable in 32-bit units.



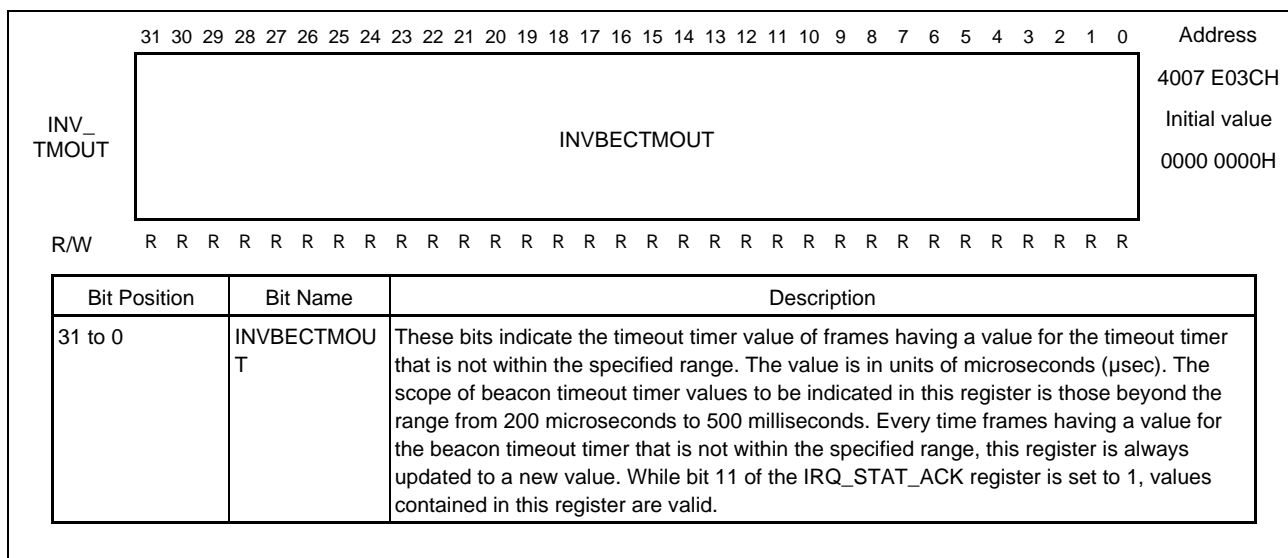
8.3.7.14 DLR Sub Type/Protocol Version Register (ETH_STYP_VER)

This register indicates information of DLR messages. This is extracted from the corresponding field in beacon frames. This register is only readable in 32-bit units.



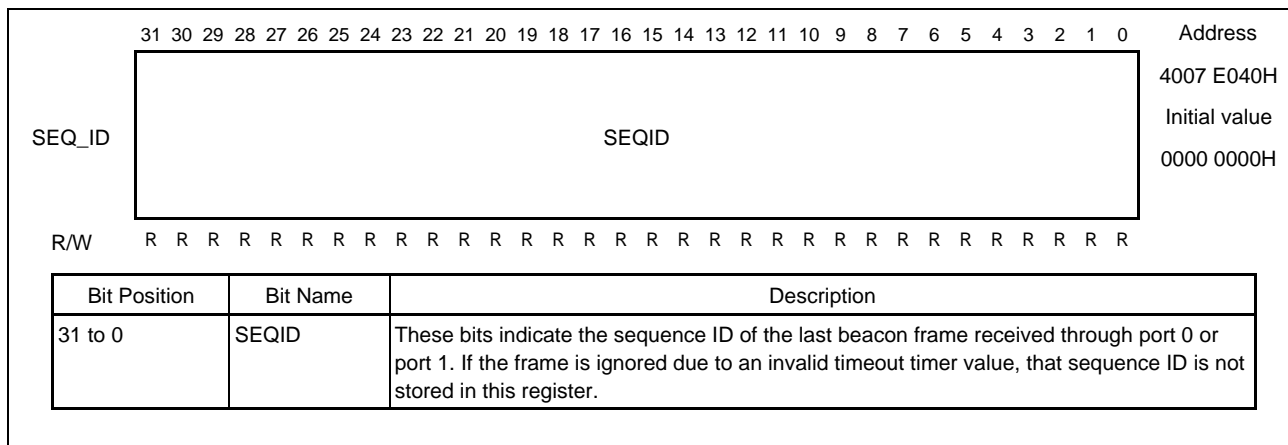
8.3.7.15 DLR Beacon Timeout Timer Register (INV_TMOUT)

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. This register is only readable in 32-bit units.



8.3.7.16 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. This register is only readable in 32-bit units.



8.3.7.17 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

8.4 Function Details

8.4.1 Switching Engine

8.4.1.1 Overview

The Ethernet switch implements the following functions:

- Input frame parsing and priority extraction
- Output port(s) resolution
- Frame queuing
- Output queue scheduling

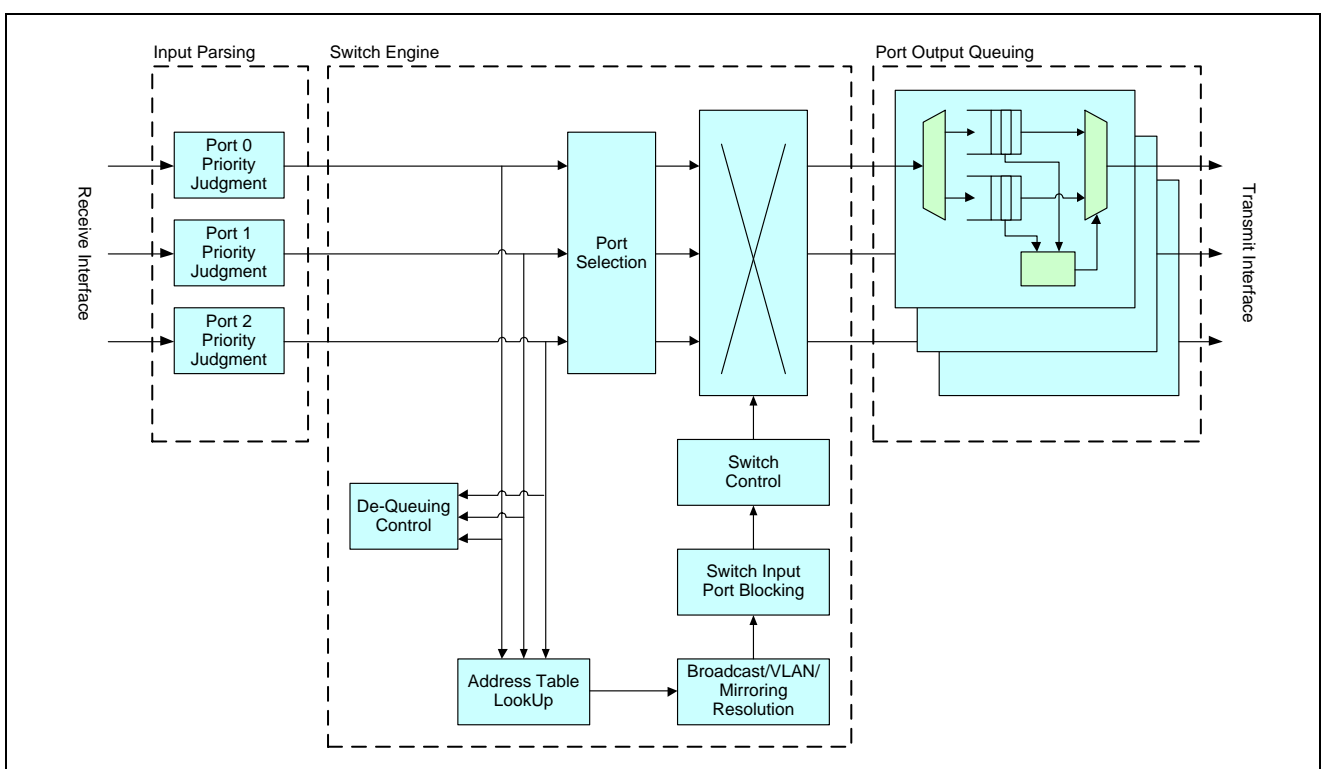


Figure 8.2 Switching Engine Overview

8.4.1.2 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.

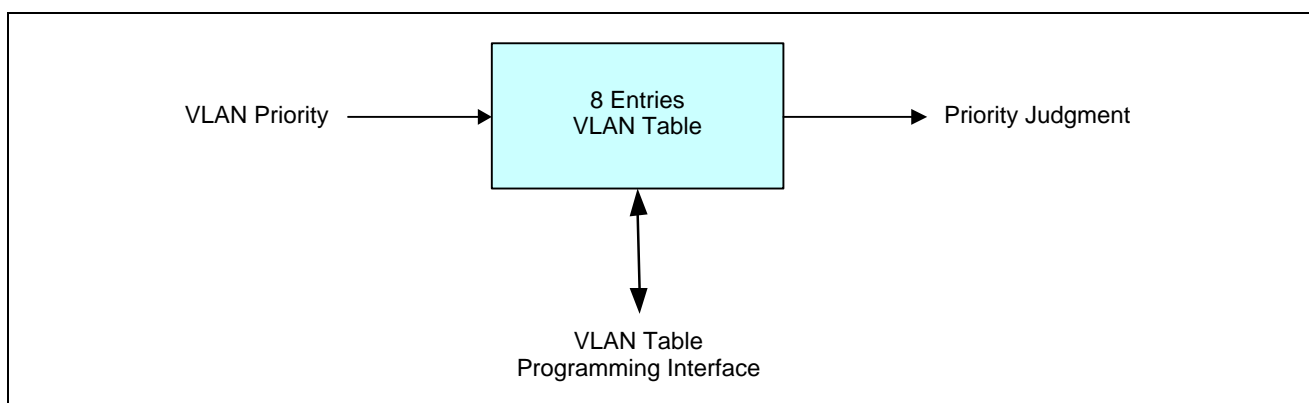


Figure 8.3 VLAN Priority Table Overview

(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.

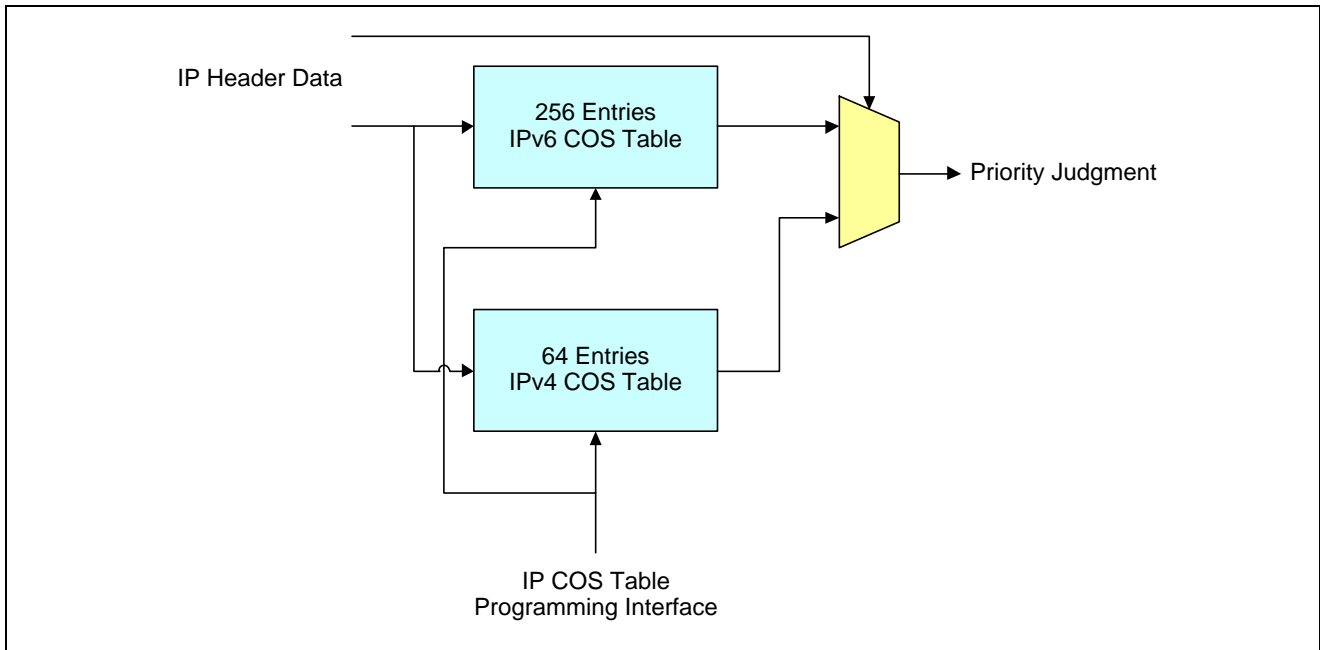


Figure 8.4 IP COS Tables Overview

(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.

8.4.1.3 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.

8.4.1.4 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.

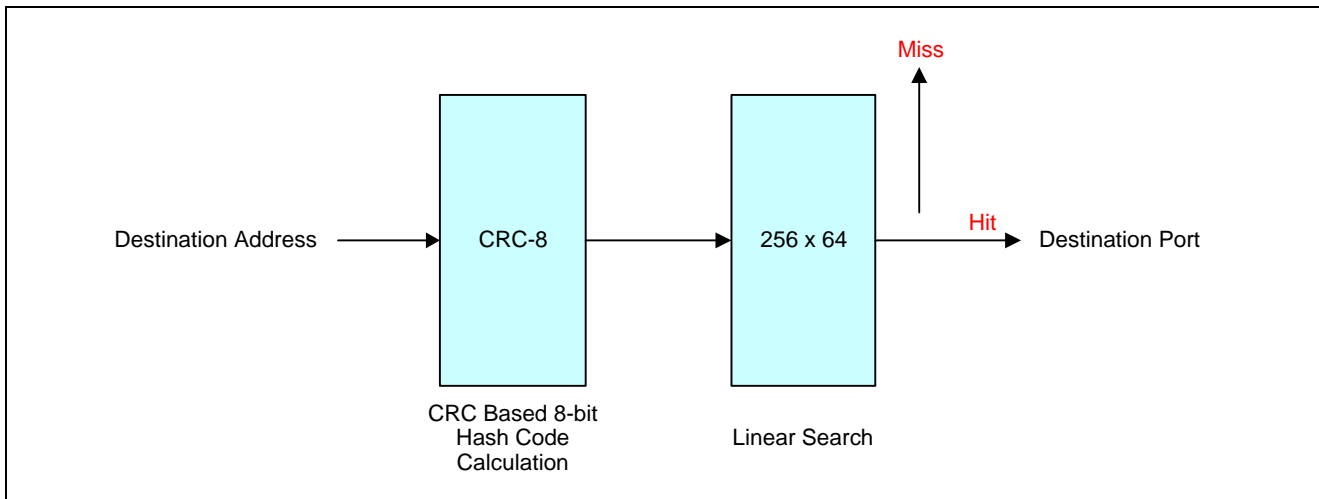


Figure 8.5 Port Look-Up Overview

(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.

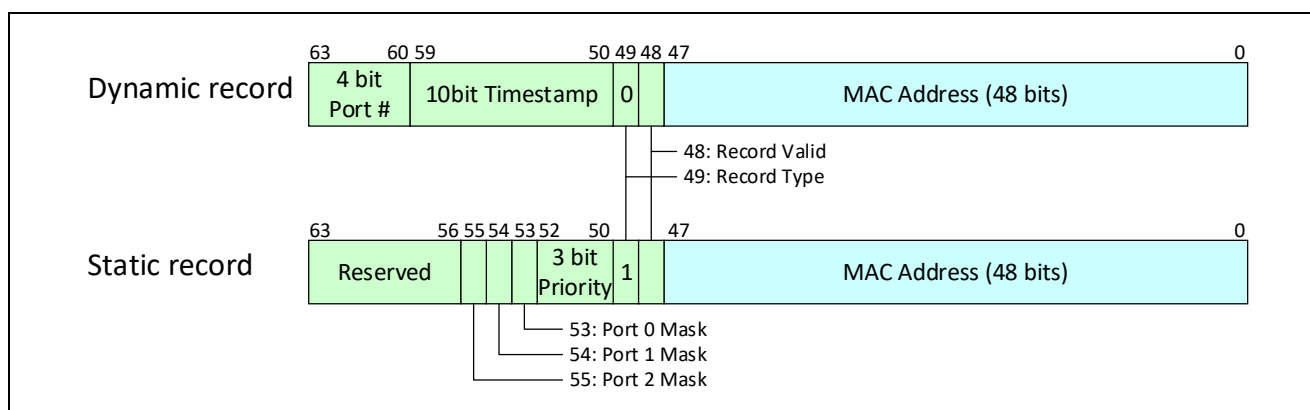


Figure 8.6 Record Types of Address Memory

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.

8.4.1.5 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.

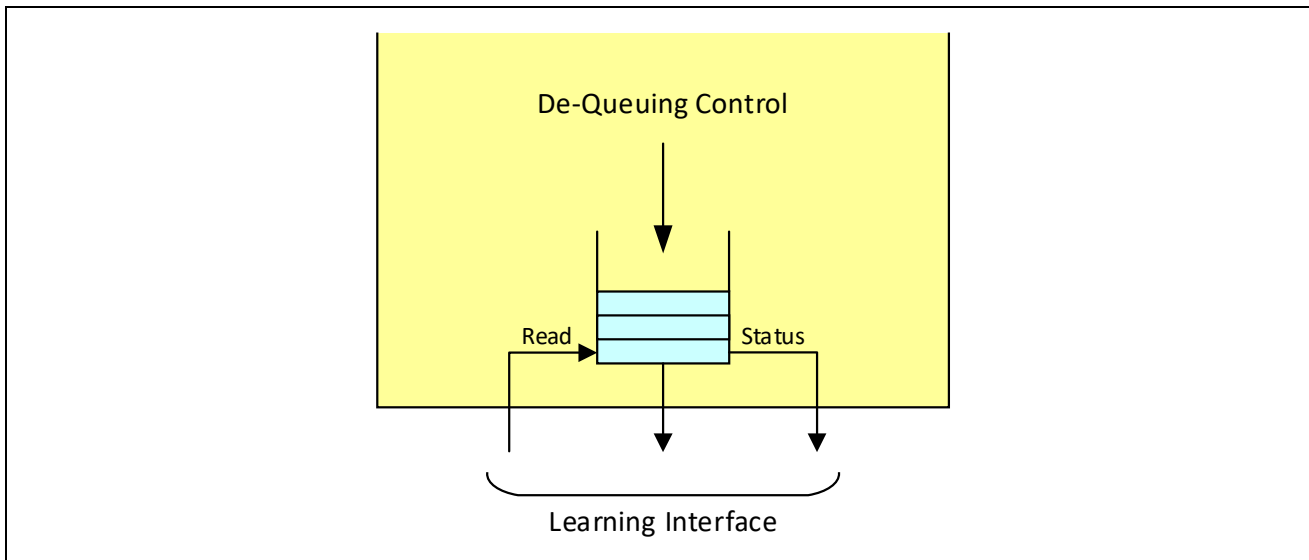


Figure 8.7 Learning Interface Overview

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.

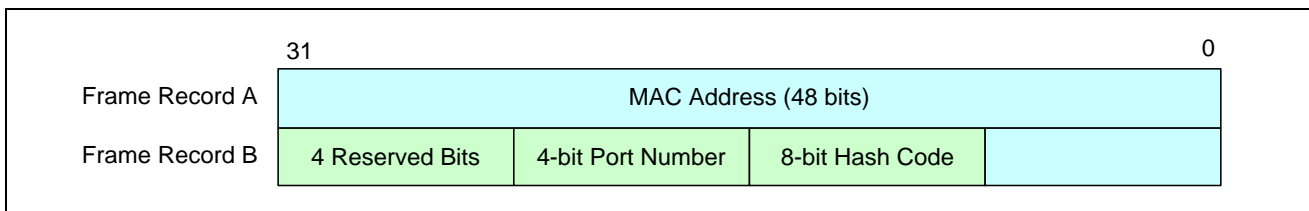


Figure 8.8 Record Formats

Software can read these records by using the LRN_REC_A and LRN_REC_B registers.

8.4.1.6 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.

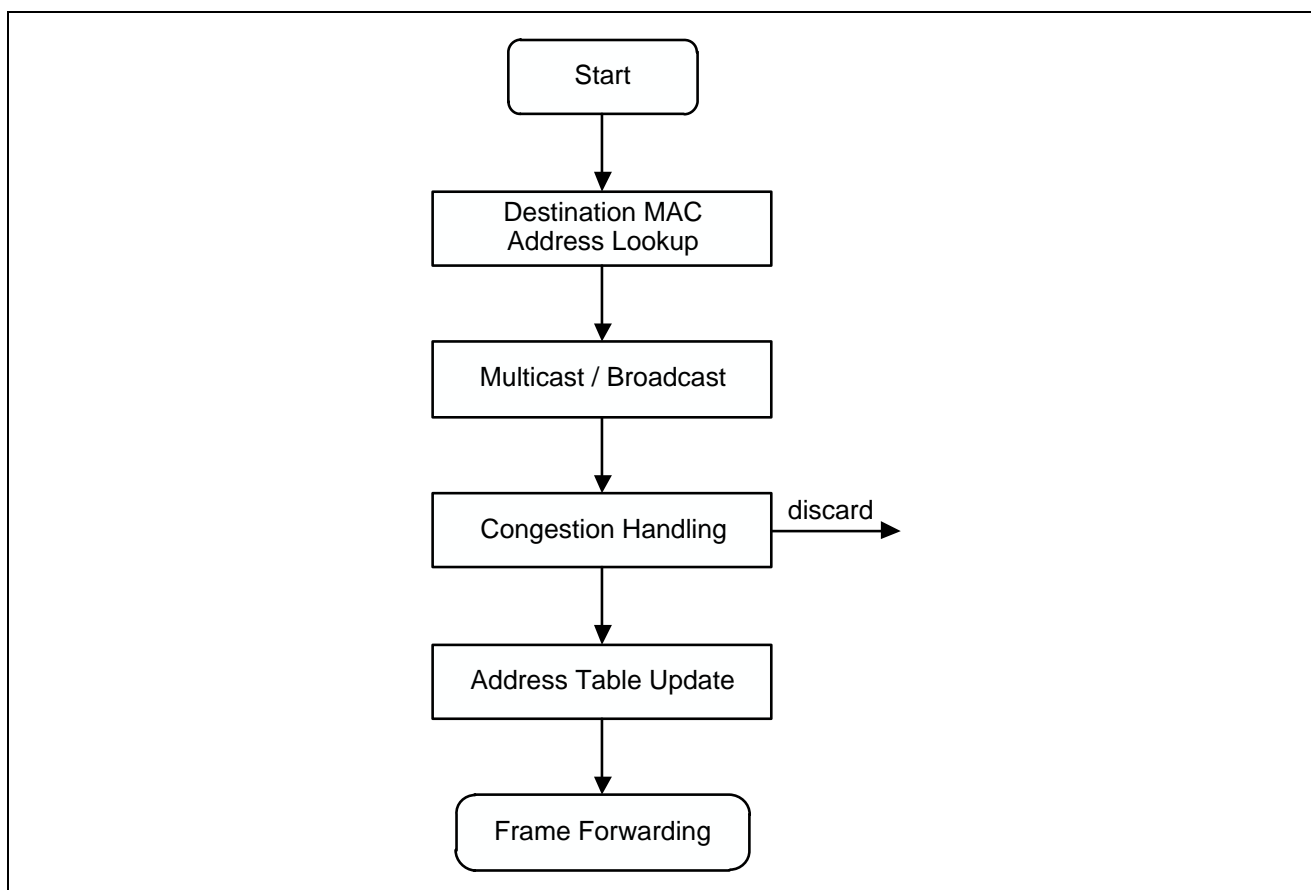


Figure 8.9 Overview of Processing for Frame Transfer

(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.

8.4.1.7 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.

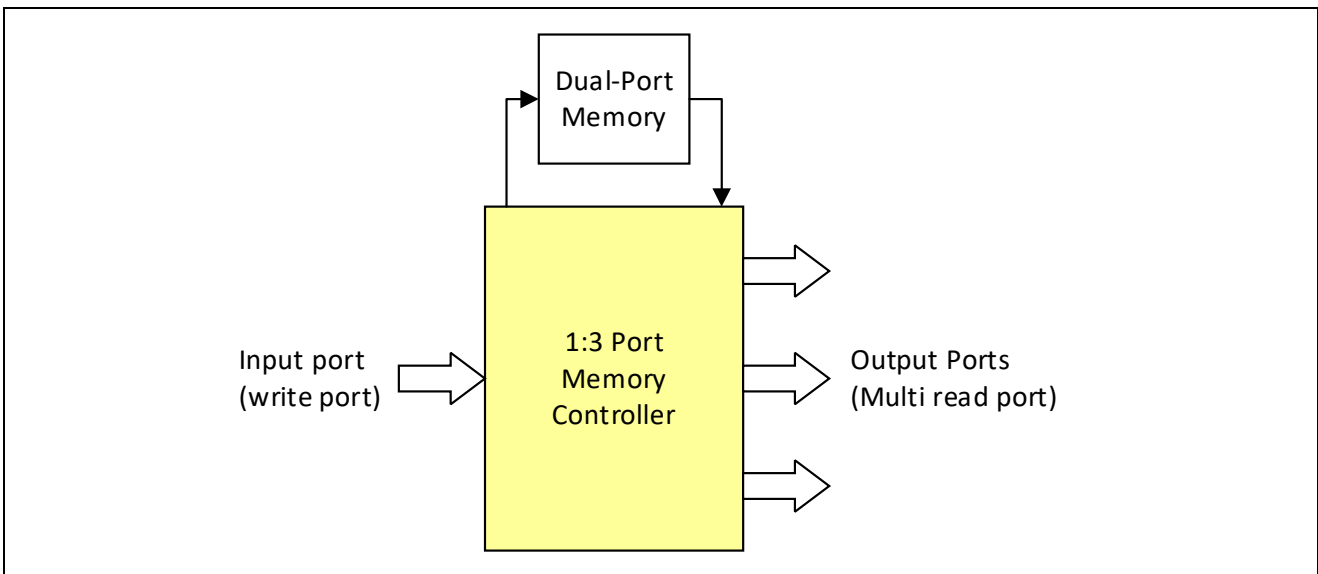


Figure 8.10 Overview of Output Port Memory Controller

(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.

8.4.2 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.

8.4.2.1 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.

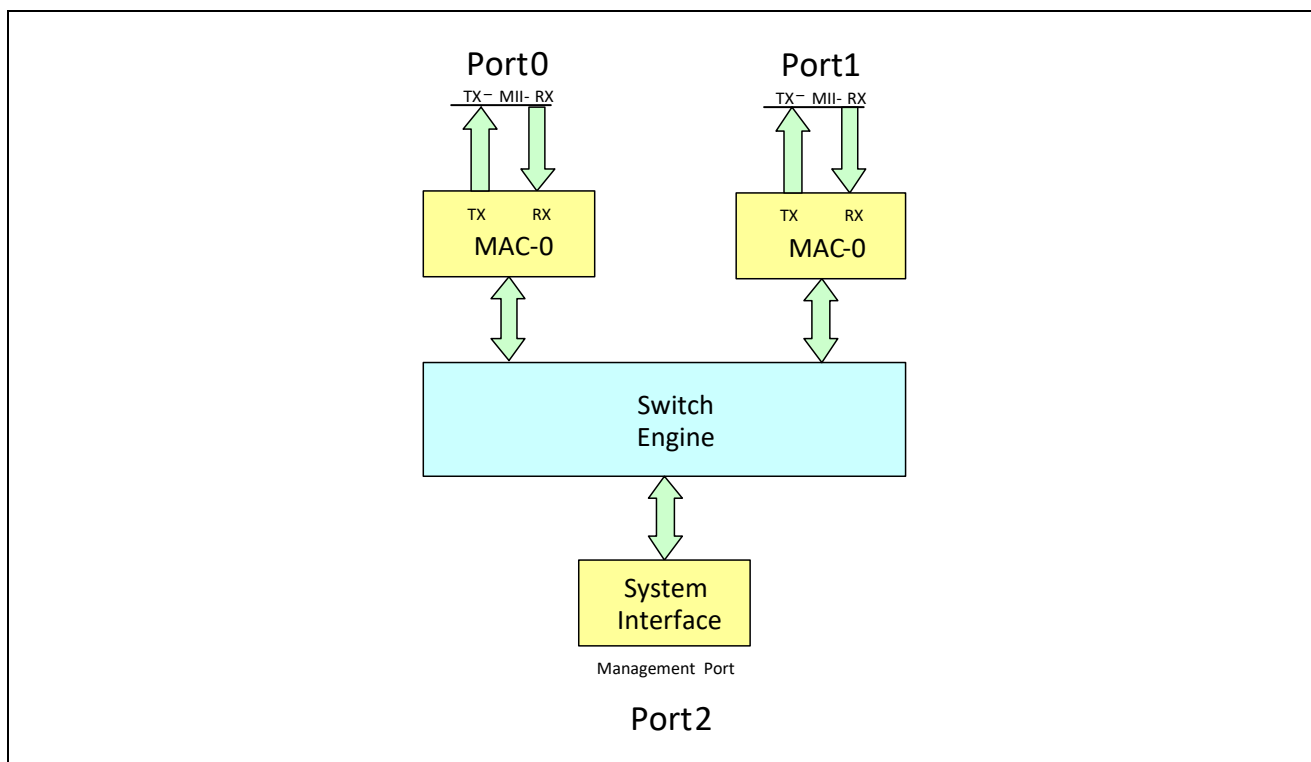


Figure 8.11 Normal Switch Mode Operation

8.4.2.2 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.

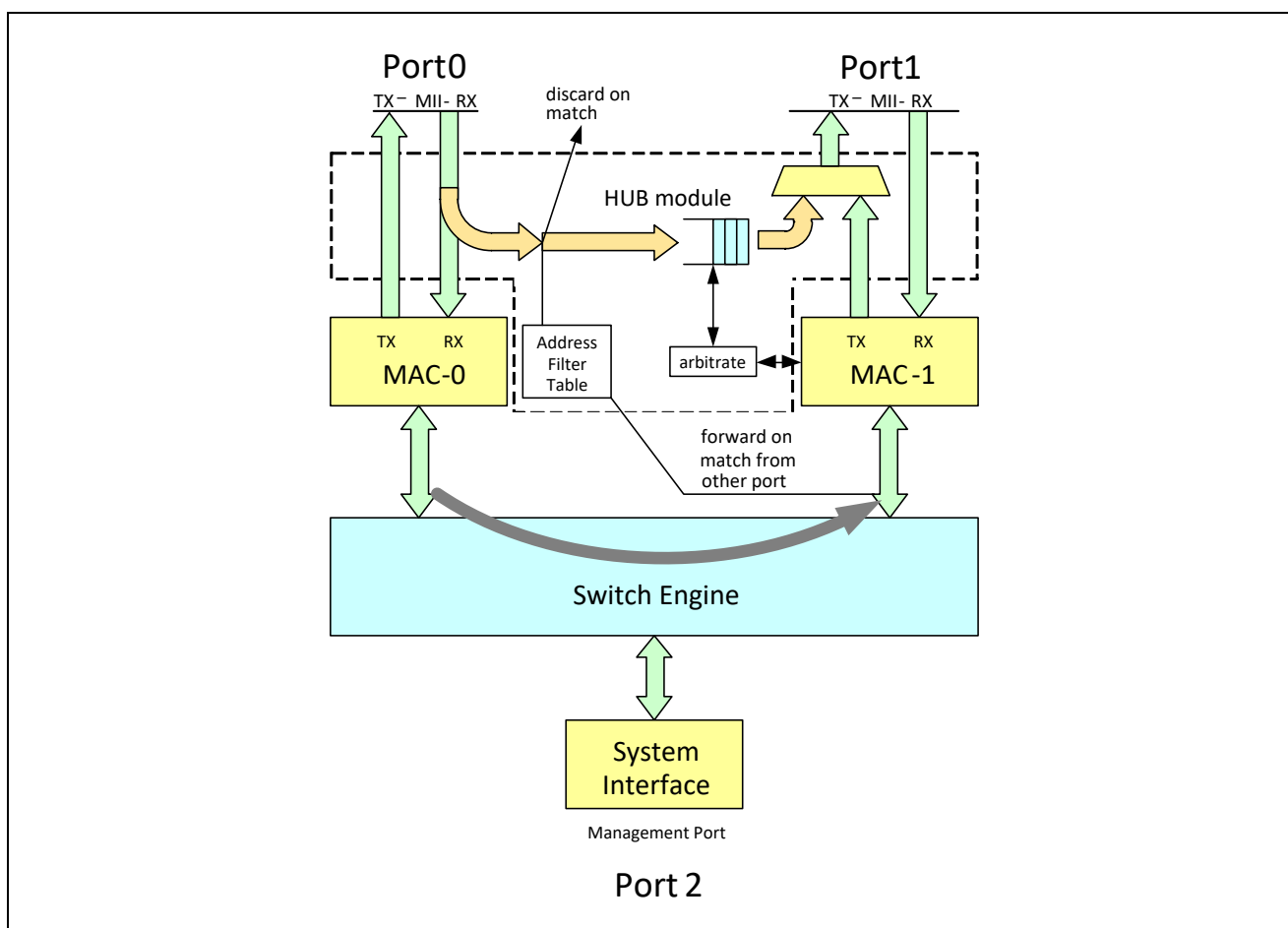


Figure 8.12 Operation of the Hub when Transfer from Port 0 to Port 1 is Enabled

8.4.2.3 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.

8.4.2.4 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 8.1 Operation of the Hub and Switch by Filter Setting

Forcible Transfer		Disable		Enable ^{Note1}	
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 ^{Note2}	Not transferred	— ^{Note3}	
Hub disabled	Hub	Not transferred	Not transferred	Cut-through	
	Switch	Store and forward ^{Note2}	Store and forward ^{Note2}	— ^{Note3}	

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 hub 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 8.2 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 8.3 PTP Multicast Domains: UDP/IP

Name	IP Address	MAC Address Mapping
DefaultPTPdomain	224.0.1.129	01-00-5e-00-01-81
AlternatePTPdomain1	224.0.1.130	01-00-5e-00-01-82
AlternatePTPdomain2	224.0.1.131	01-00-5e-00-01-83
AlternatePTPdomain3	224.0.1.132	01-00-5e-00-01-84

Table 8.4 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 8.5 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 8.6 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 8.7. 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 8.7 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

8.4.2.5 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.

8.4.2.6 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.

8.4.3 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.

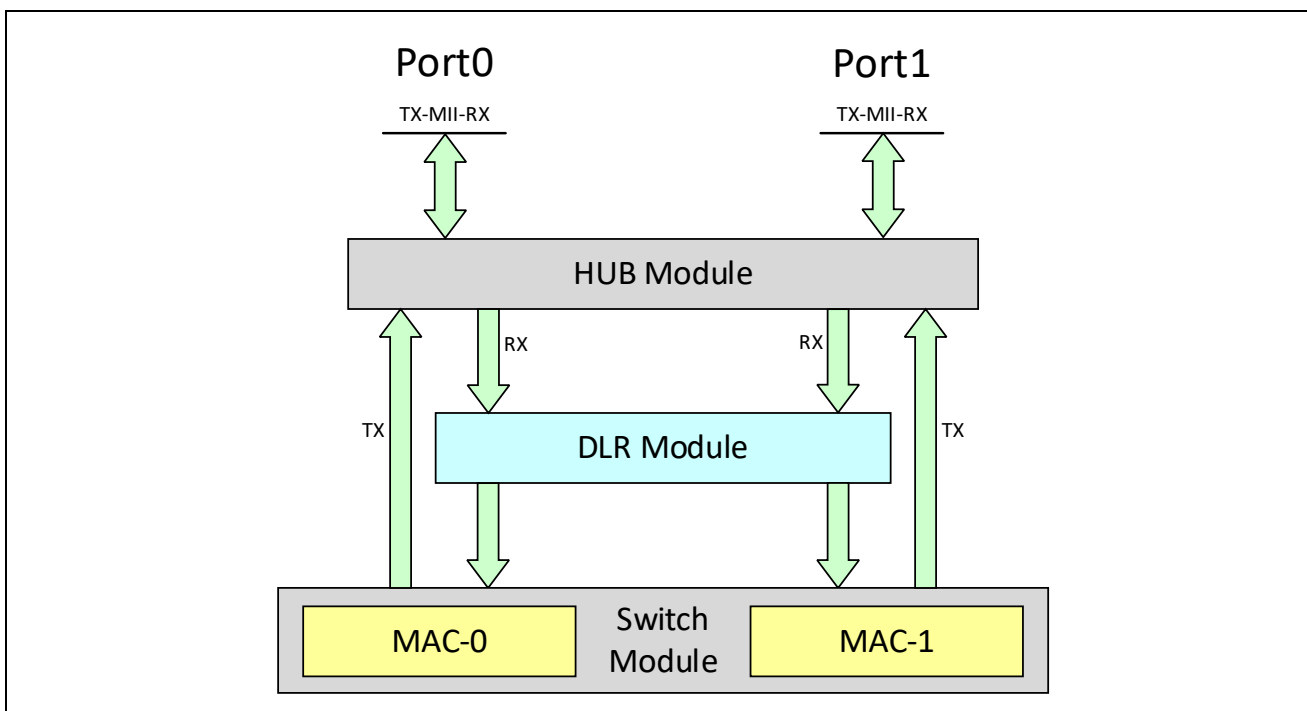


Figure 8.13 Connection between the Hub Module and the DLR Module

8.4.3.1 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	7 octets	PREAMBLE	Common DLR Protocol Header
	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 8.14 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 8.8 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	—

8.4.3.2 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.

8.4.4 IEEE 1588 Timer & Control Module

8.4.4.1 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.

8.4.4.2 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 8.9 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 8.10 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 8.11 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 8.12 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 8.13 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 8.14 Common PTPv2 Message Header

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
transportSpecific				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 8.15 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 8.16 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.

8.4.4.3 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 interrupt is connected to an external pin as a pulse per second (PPS) signal. The timer 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.

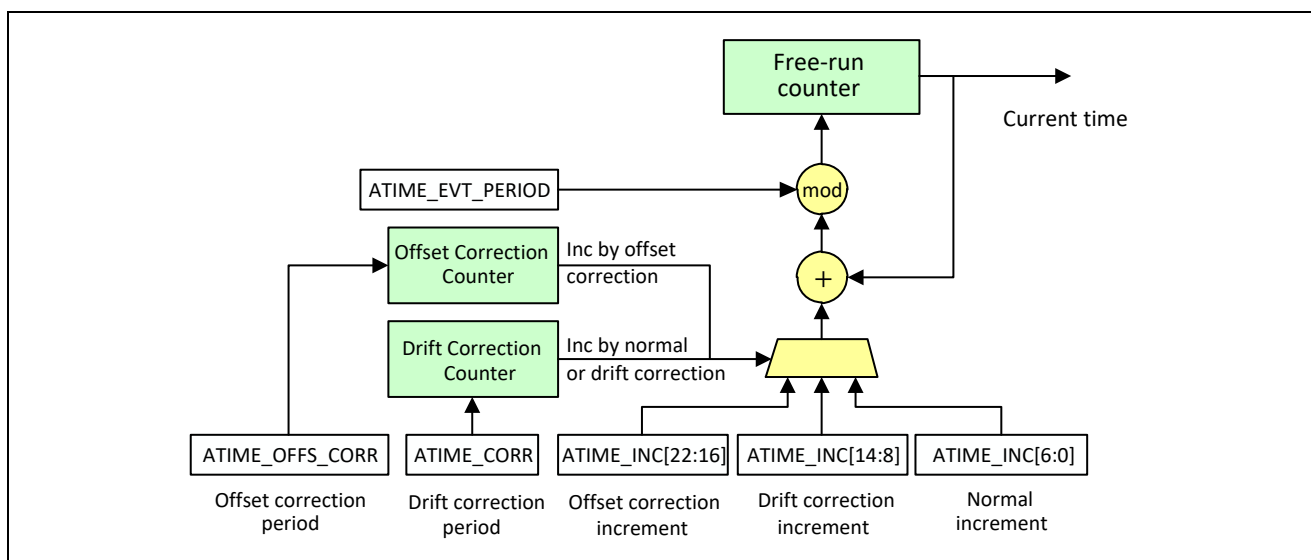


Figure 8.15 Configuration of Adjustable Timer

(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

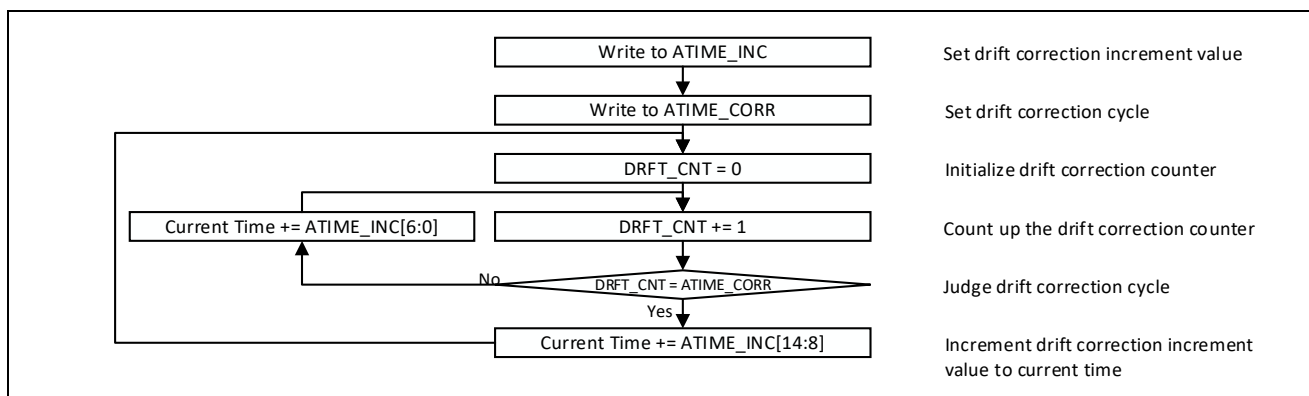


Figure 8.16 Drift Correction

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

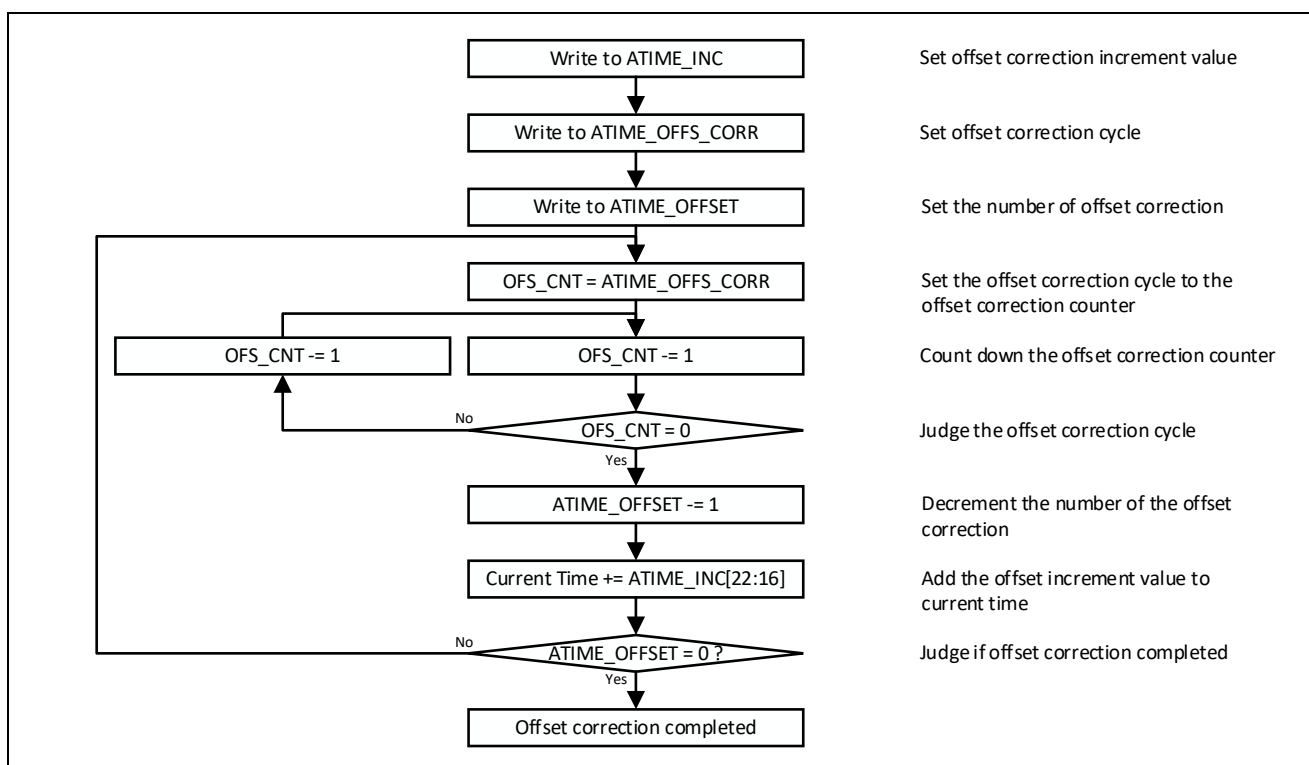


Figure 8.17 Offset Correction (when ATIME_OFFS_CORR is not zero)

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.

8.4.4.4 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 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.

8.4.4.5 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.

8.4.5 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.

8.4.5.1 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.

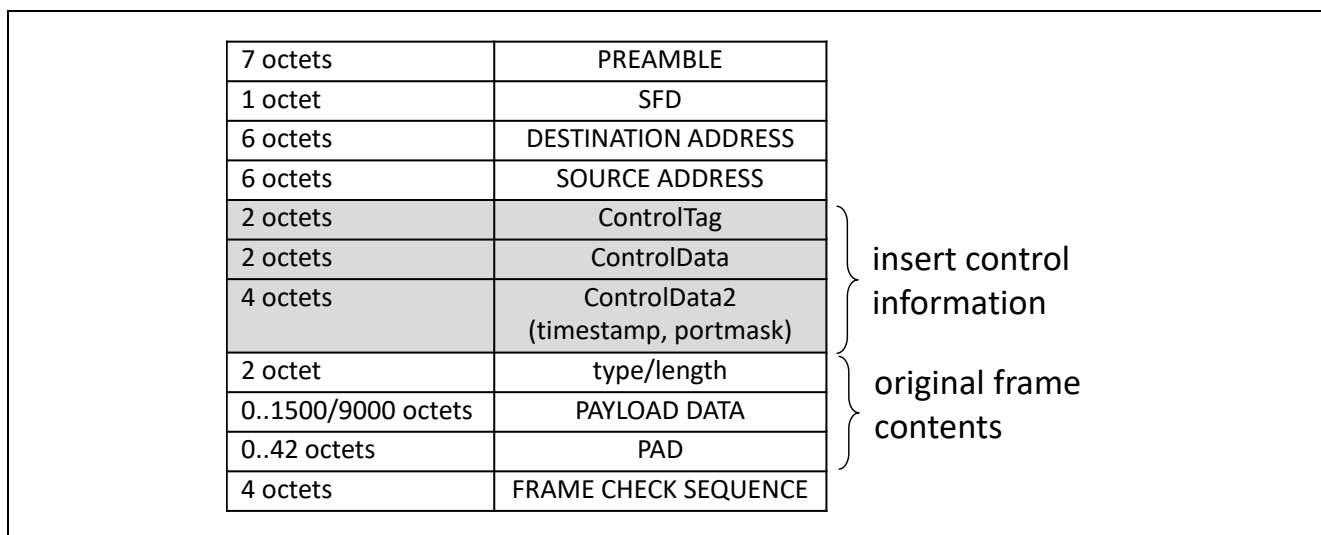


Figure 8.18 Format of Frame with Management Tag in Internal Port

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).

8.4.5.2 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 8.17 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.

8.4.5.3 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 8.18 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

8.4.5.4 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 7, Gigabit Ethernet MAC.**

8.5 Overview of Control Software

8.5.1 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.

8.5.2 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 8.19 Examples of Initial Settings of the Address Table

Address	Register	Example Setting	Description
4007 4000H to 4007 47FCH (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.
Address above + 4H	ADR_TABLE	0083 0605H	These settings are not required when set dynamically.

Table 8.20 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 8.21 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 8.22 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. For filtering the MAC address of 01-1b-19-00-00-00.
4007 01D4H	HUB_FLT_MAC1hi	00FF 0000H	
4007 01D8H	HUB_FLT_MAC2lo	005E 0001H	Example settings of UDP/IP messages of PTP. For filtering the MAC address of 01-00-5e-00-01-{80..87}.
4007 01DCH	HUB_FLT_MAC2hi	00F8 8001H	
4007 01E0H	HUB_FLT_MAC3lo	005E 0001H	Example settings of management frames. For filtering the MAC address of 01-00-5e-00-00-{00..03}.
4007 01E4H	HUB_FLT_MAC3hi	00FC 0000H	
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	
4007 01FCH	HUB_FLT_MAC6hi	01FF 0100H	Example settings of beacon frames of the DLR. 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 8.23 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	ATIME_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	PORT0_CTRL	0000 0000H	Clear the timestamp control and status registers.
4007 C028H	PORT1_CTRL		

Table 8.24 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_MACIo	0403 0201H	Set unicast addresses.
4007 E018H	LOC_MACHi	0000 0605H	The example settings are for 01-02-03-04-05-06.

8.5.3 Address Table Setting

8.5.3.1 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.

Figure 8.19 shows the principle of the address table layout. The software designer needs to understand this when writing the learning and aging functions.

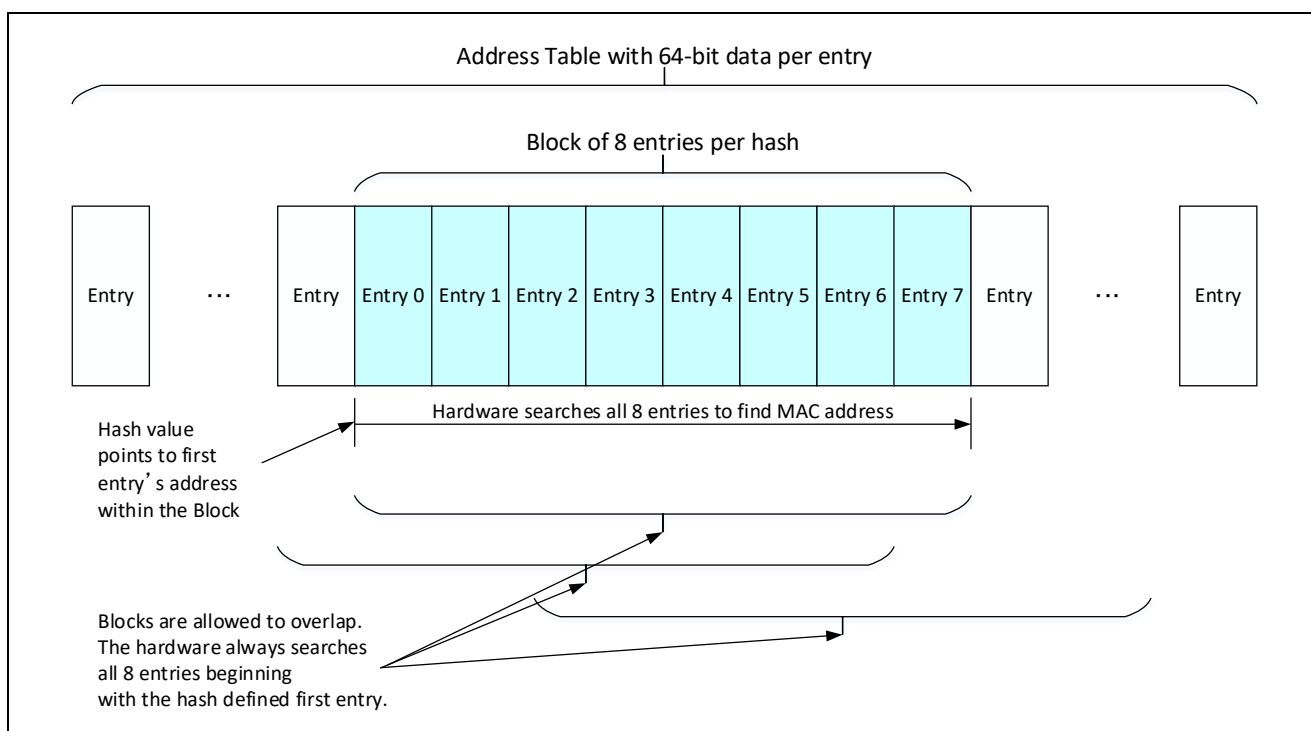


Figure 8.19 Entries of Address Table and Definition of Hash Block

8.5.3.2 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).

Figure 8.20 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.

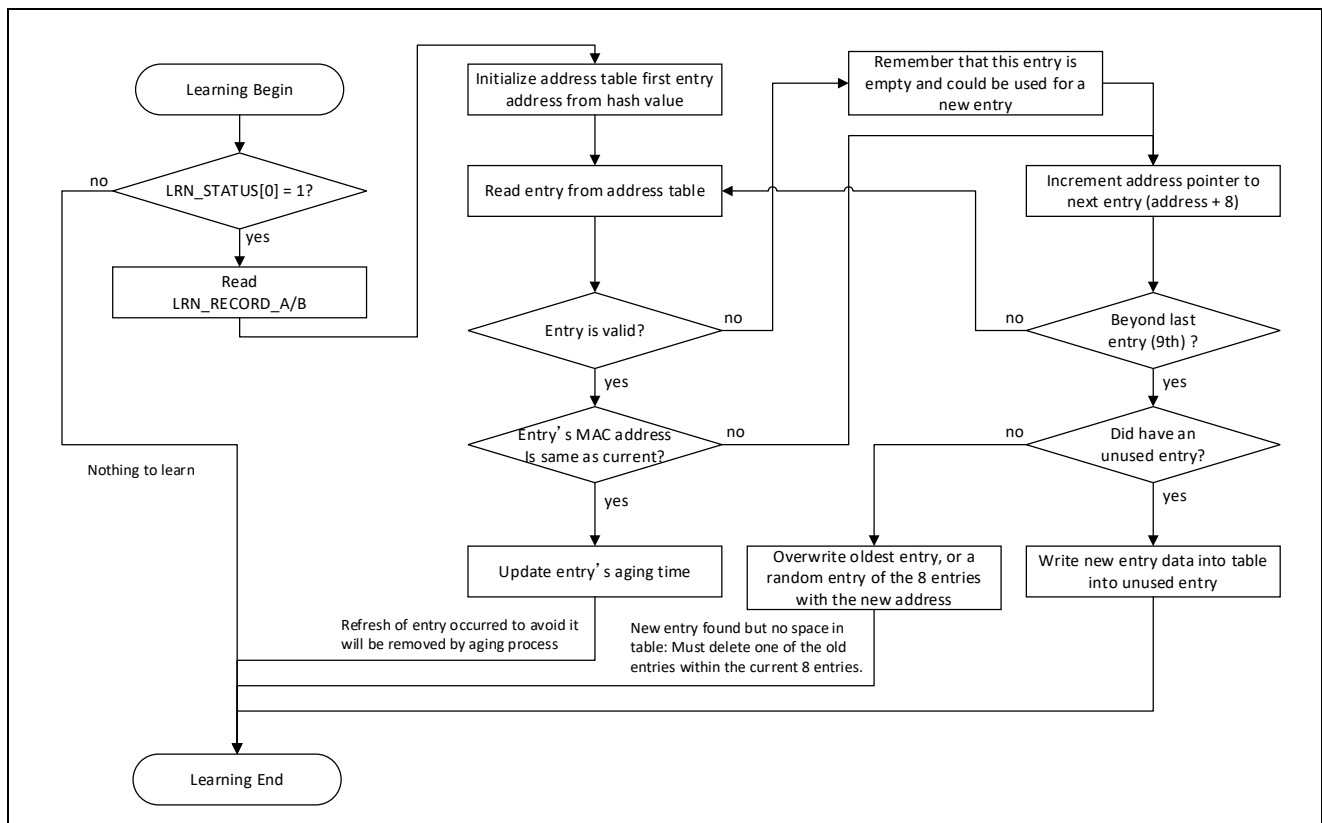


Figure 8.20 Address Learning Flow

9. Asynchronous SRAM Memory Controller (ROM/SRAM)

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.

9.1 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-IN32M3 do not support an 8-bit bus width.

9.2 Features

(1) Static memory control

The memory controllers of an R-IN32M3 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-IN32M3 always operate in little endian mode.

9.3 Bus Control

Operating an R-IN32M3 requires setting the bus control registers.

9.3.1 Overview of Registers

Table 9.1 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	4001 0100H

9.3.2 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.^{Note2}

	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
				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		RW	1
			RW	1
			RW	1
			RW	1
			RW	1
				1
		400A 2004H		
		Initial value		
		Note1		

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.**

9.3.3 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)

SMCn	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		
	0	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	400A 2008H+ n × 4H	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	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 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. <table border="1" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>IWn3</th> <th>IWn2</th> <th>IWn1</th> <th>IWn0</th> <th>Number of idle wait cycles for CSZn (IWn+1)</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>1 cycle of BUSCLK</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>2 cycles of BUSCLK</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>3 cycles of BUSCLK</td> </tr> <tr> <td colspan="4" style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> </tr> <tr> <td>1</td><td>1</td><td>0</td><td>1</td><td>14 cycles of BUSCLK</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>0</td><td>15 cycles of BUSCLK</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>16 cycles of BUSCLK (initial value)</td> </tr> </tbody> </table> <div style="background-color: #ffff00; padding: 5px; margin-top: 10px; border: 1px solid black;"> <p>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.</p> </div>	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)
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)																																						

Remarks 1. n = 0 to 3

2. BUSCLK = HCLK

Bit Position	Bit Name	Description																																																	
11 to 8	WWn3- WWn0	<p>Set a write recovery wait for each CSZn.</p> <p>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).</p> <p>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</p> <table border="1"> <thead> <tr> <th>WWn3</th> <th>WWn2</th> <th>WWn1</th> <th>WWn0</th> <th>Number of write recovery wait cycles for CSZn</th> </tr> </thead> <tbody> <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" style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</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> </tbody> </table> <p>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.</p>	WWn3	WWn2	WWn1	WWn0	Number of write recovery wait cycles for CSZn	0	0	0	0	1 cycle of BUSCLK	0	0	0	1	0	0	1	0	2 cycles of BUSCLK	0	0	1	1	3 cycles of BUSCLK	0	1	0	0	4 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)
WWn3	WWn2	WWn1	WWn0	Number of write recovery wait cycles for CSZn																																															
0	0	0	0	1 cycle of BUSCLK																																															
0	0	0	1																																																
0	0	1	0	2 cycles of BUSCLK																																															
0	0	1	1	3 cycles of BUSCLK																																															
0	1	0	0	4 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)																																															
7 to 4	DWn3-DWn0	<p>Set a data wait for each CSZn.</p> <p>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.</p> <table border="1"> <thead> <tr> <th>DWn3</th> <th>DWn2</th> <th>DWn1</th> <th>DWn0</th> <th>Number of data wait cycles for CSZn</th> </tr> </thead> <tbody> <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" style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</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> </tbody> </table> <p>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.</p>	DWn3	DWn2	DWn1	DWn0	Number of data wait cycles for CSZn	0	0	0	0	0 (Setting prohibited in case of Page ROM)	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)									
DWn3	DWn2	DWn1	DWn0	Number of data wait cycles for CSZn																																															
0	0	0	0	0 (Setting prohibited in case of Page ROM)																																															
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)																																															

Remarks 1. n = 0 to 3
2. BUSCLK = HCLK

Bit Position	Bit Name	Description																																								
3 to 0	ACn3-ACn0	<p>Sets an address setup wait for each CSZn.</p> <p>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).</p> <p>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.</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>ACn3</th> <th>ACn2</th> <th>ACn1</th> <th>ACn0</th> <th>Number of address setup waits for CSZn</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0 (for reading) or 1 cycle of BUSCLK (for writing) ^{Note}</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" style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</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> </tbody> </table> <p>Note: An address setup wait of one cycle of BUSCLK is always inserted for a write operation.</p> <p>Caution: The address setup wait set by this register is also inserted in on-page access to the page ROM.</p>	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)
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)																																						

Remarks 1. n = 0 to 3
 2. BUSCLK = HCLK

9.3.4 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.

		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	
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	0	ST	400A 2018H Initial value F000 0000H
R/W		RW	RW	RW	RW	0	0	0	0	0	0	0	0	RW	RW	RW	RW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RW	
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.																																	
		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)																													
19 to 16	MA6-MA3	Set the mask bits for comparison of addresses.																																	
		MA6	MA5	MA4	MA3	Page size of page ROM to be connected																													
		0	0	0	0	32 bits x 2, 16 bits x 4 (initial value)																													
		0	0	0	1	32 bits x 4, 16 bits x 8																													
		0	0	1	1	32 bits x 8, 16 bits x 16																													
		0	1	1	1	32 bits x 16, 16 bits x 32																													
		1	1	1	1	32 bits x 32, 16 bits x 64																													
		Other than the above				Setting prohibited (If set, correct operation is not guaranteed).																													
0	ST	Specifies the type of memory connected to chip select output 0 (CSZ0). 0 : SRAM, I/O device (initial value) 1 : Page ROM																																	

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.

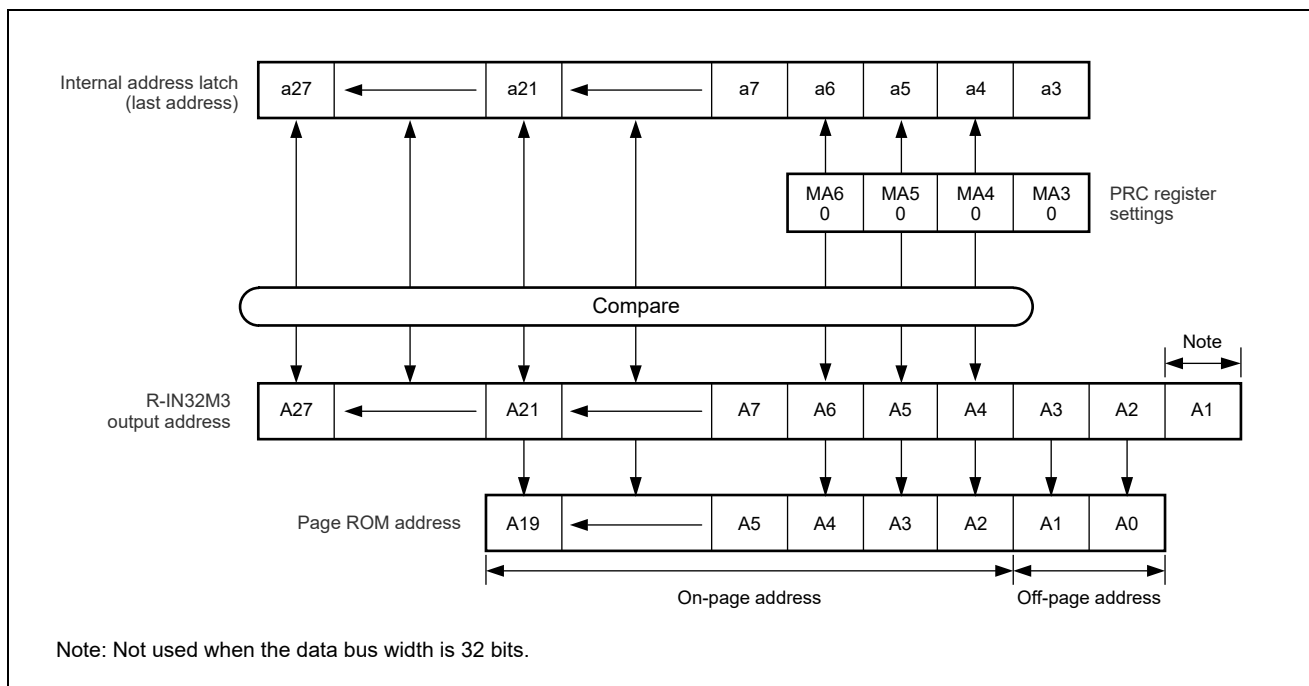


Figure 9.1 Example of Control Using the MA6 to MA3 Bits of the PRC Register

(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.

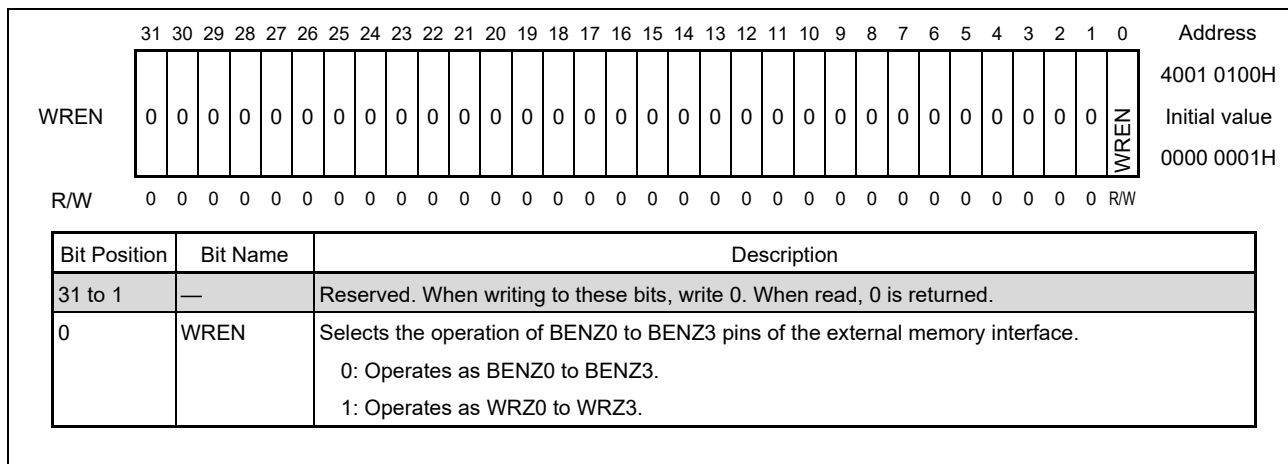
Caution: If an access request is made that requires a greater width for data transfer than the width of the paged ROM data bus, the access to the page ROM is divided and on-page access is judged separately for access to the low- and high-order parts. When access is divided, on-page access is judged per 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.

9.3.5 Write Enable Switching Register (WREN)

This register selects WRZ3-WRZ0 or BENZ3-BENZ0 for the BENZ3-BENZ0 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 BENZ3-BENZ0 pins operate as WRZ3-WRZ0.

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



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 how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.

9.4 Memory Connection Examples

9.4.1 SRAM Connection Example

An example of connection with SRAM is shown below.

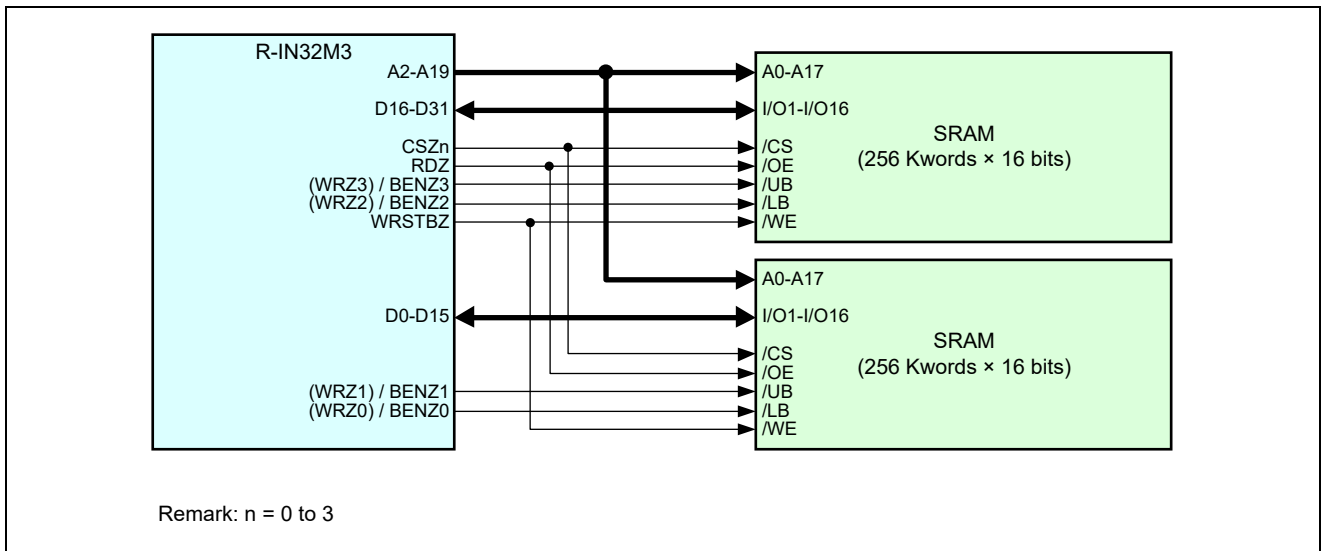


Figure 9.2 Example of Connection with 32-Bit SRAM

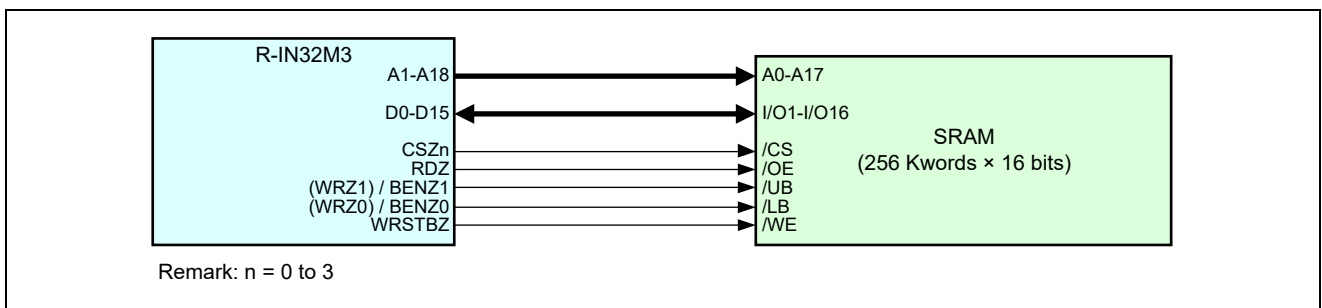


Figure 9.3 Example of Connection with 16-Bit SRAM

9.4.2 Page ROM Connection Example

An example of connection with page ROM is shown below.

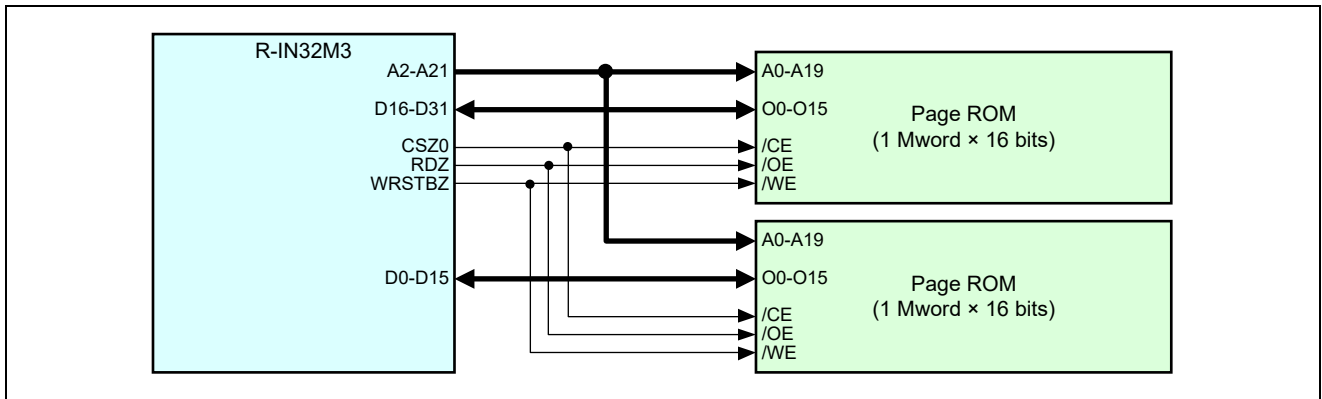


Figure 9.4 Example of Connection with 32-Bit Page ROM

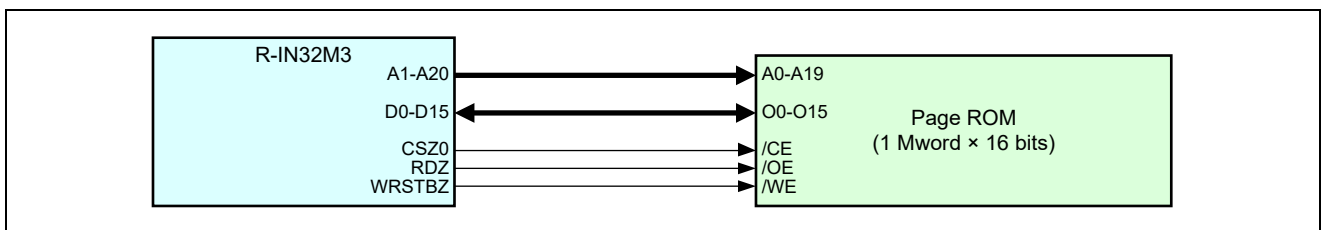


Figure 9.5 Example of Connection with 16-Bit Page ROM

Caution: On-page mode of the page ROM is only available when it is connected to CSZ0.

9.5 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.

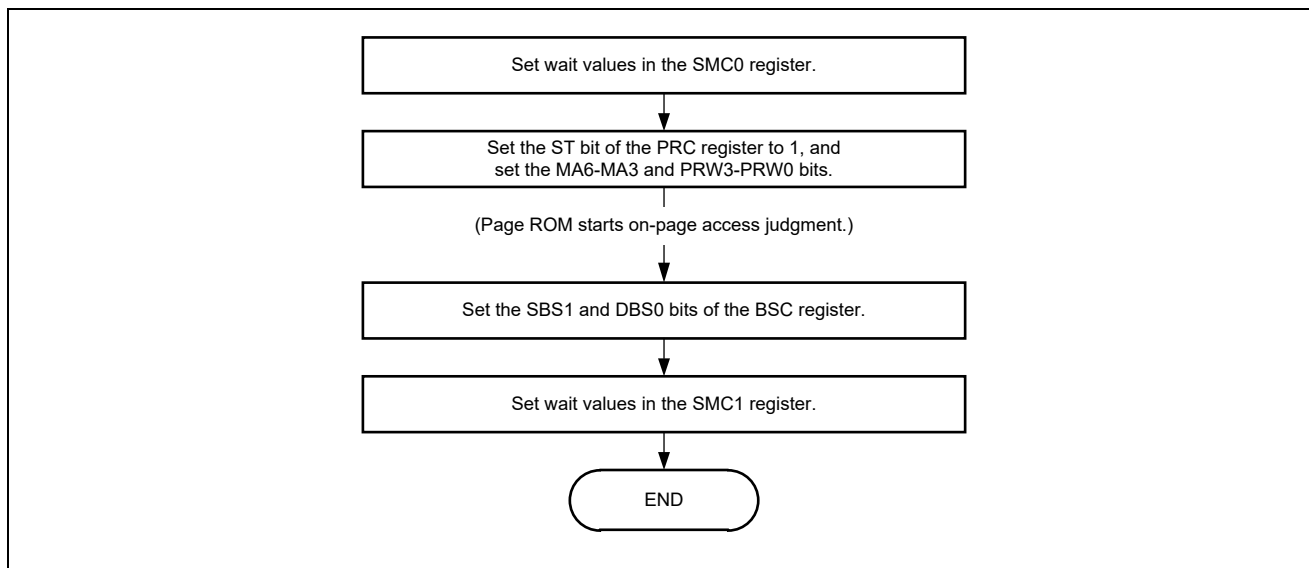


Figure 9.6 Example Procedure for Setting the Control Registers of the Memory Controller

9.6 External Wait Function

When a low-speed device or system is connected to the memory controller of an R-IN32M3, 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 for 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).

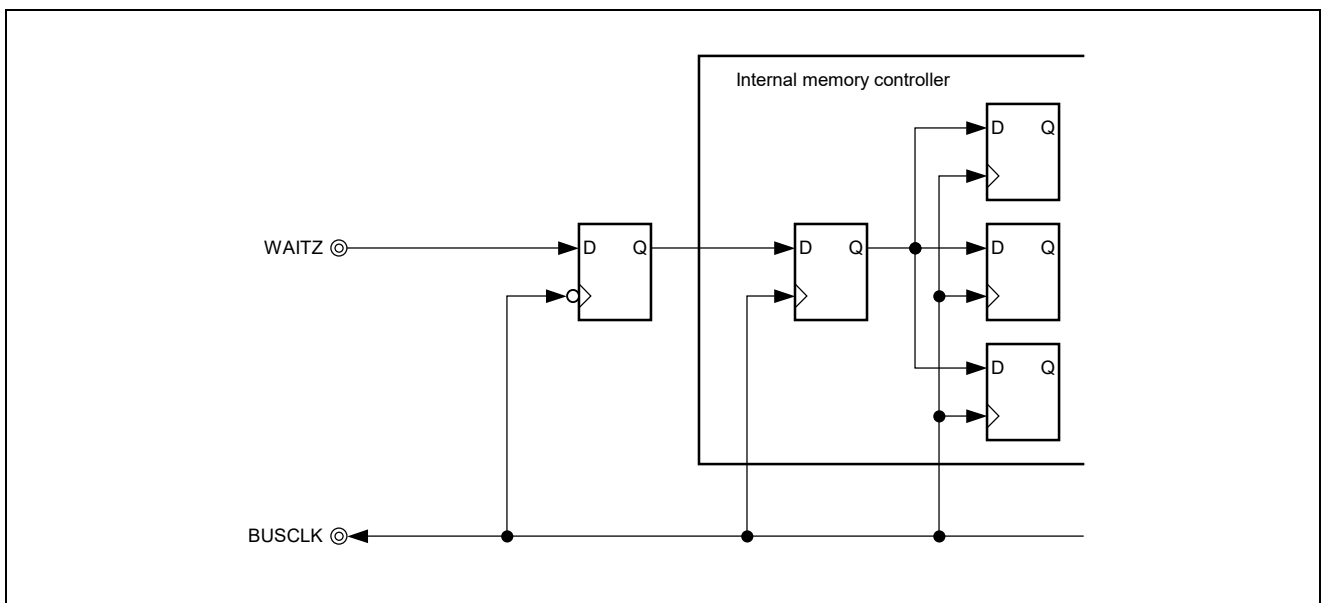


Figure 9.7 Configuration of the WAITZ Signal Sampling Circuit

9.7 Memory Access Timing Examples

The memory access timing examples are listed below.

Table 9.2 Memory Access Timing Examples

Figure Number	Memory Type	Access Condition	Page
Figure 9.8	SRAM	Reading without wait states	9-17
Figure 9.9	SRAM	Reading with wait states	9-18
Figure 9.10	SRAM	Reading with external wait insertion	9-19
Figure 9.11	SRAM	Writing without wait states	9-20
Figure 9.12	SRAM	Writing with wait states	9-21
Figure 9.13	SRAM	Writing with external wait insertion	9-22
Figure 9.14	Page ROM	Reading for a single transfer	9-23
Figure 9.15	Page ROM	Reading for a four-word burst transfer	9-24

BSC: SBS3-SBS0 = 1111B (32 bits), SMCn: IWn3-IWn0 = 0000B (1 wait cycle),
 DWn3-DWn0 = 0000B (no wait), ACn3-ACn0 = 0000B (no wait)

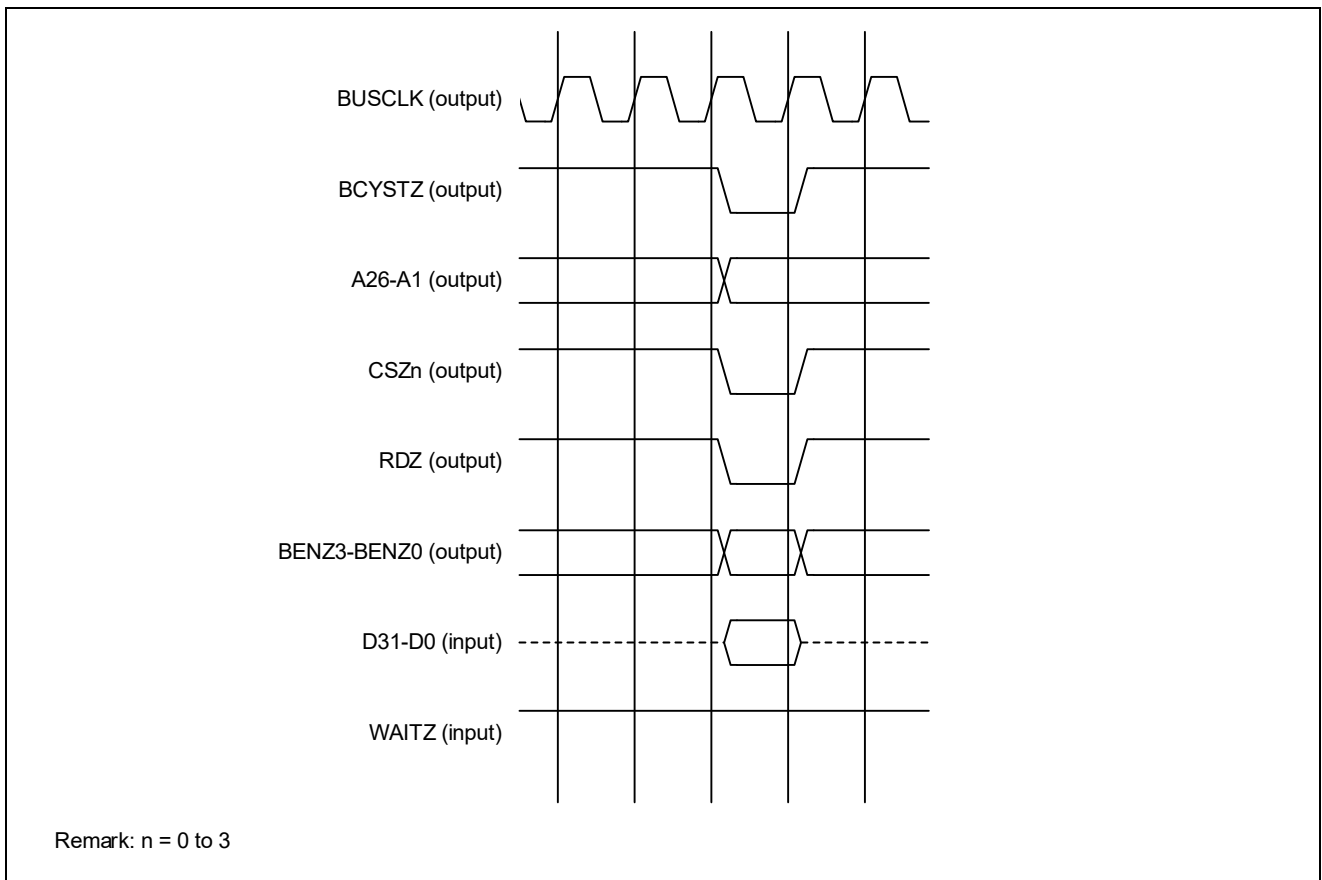


Figure 9.8 SRAM Read Cycles

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)

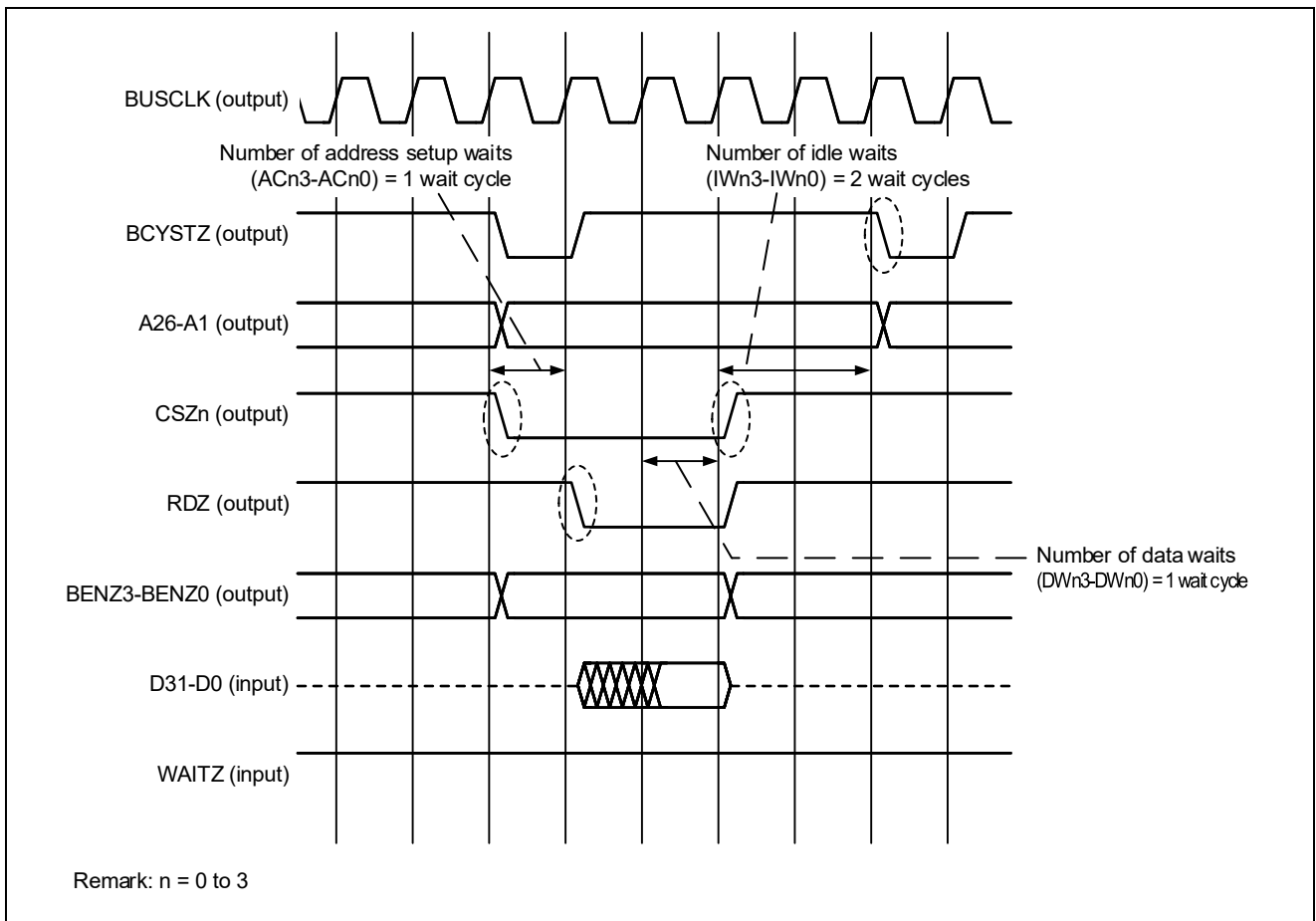


Figure 9.9 SRAM Read Cycles (with Wait Settings)

BSC: SBS3-SBS0 = 1111B (32 bits), SMCn: IWn3-IWn0 = 0000B (1 wait cycle),
 DWn3-DWn0 = 0011B (3 wait cycles), ACn3-ACn0 = 0000B (no wait)

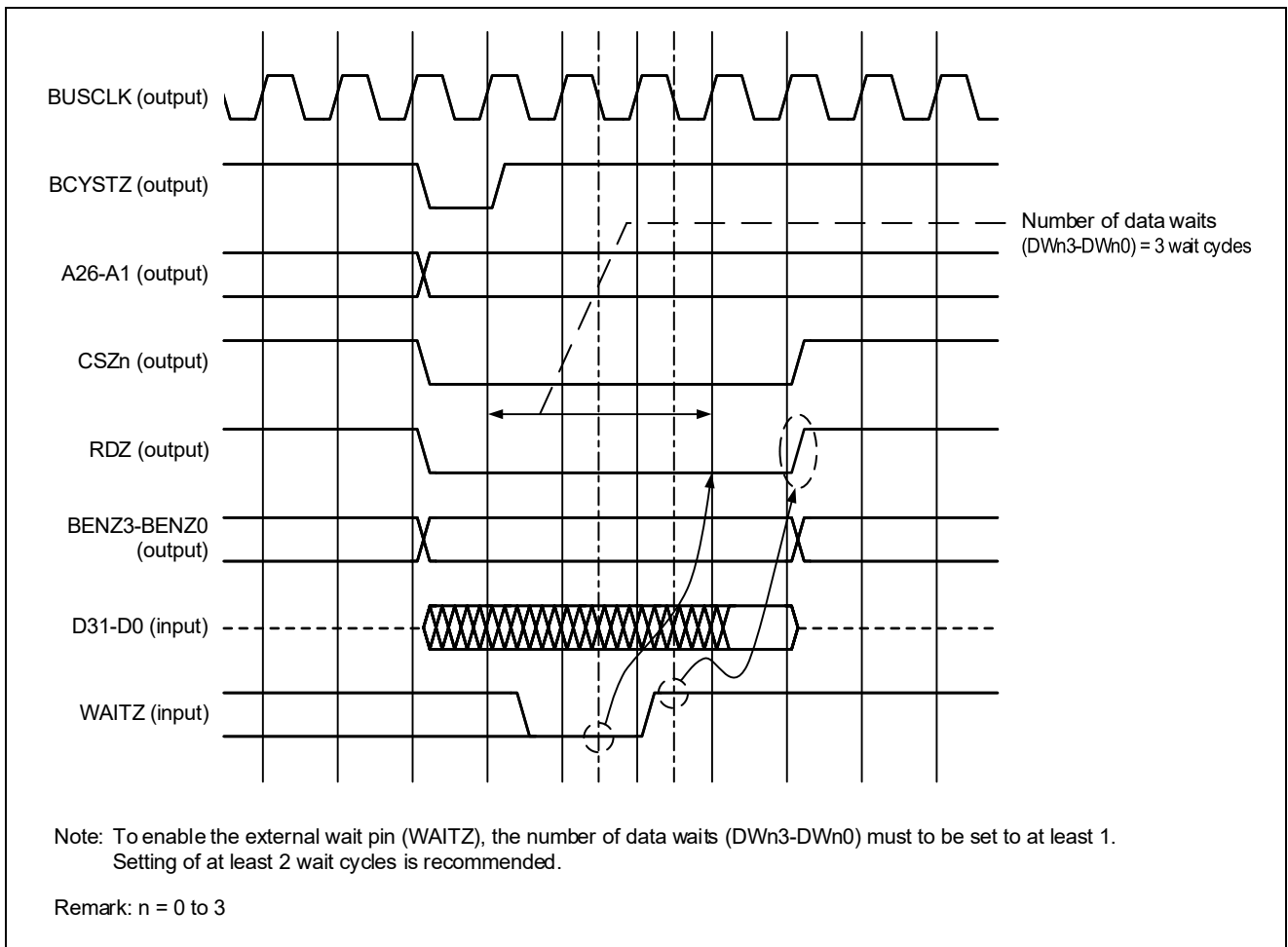


Figure 9.10 SRAM Read Cycles (External Wait Insertion)

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)

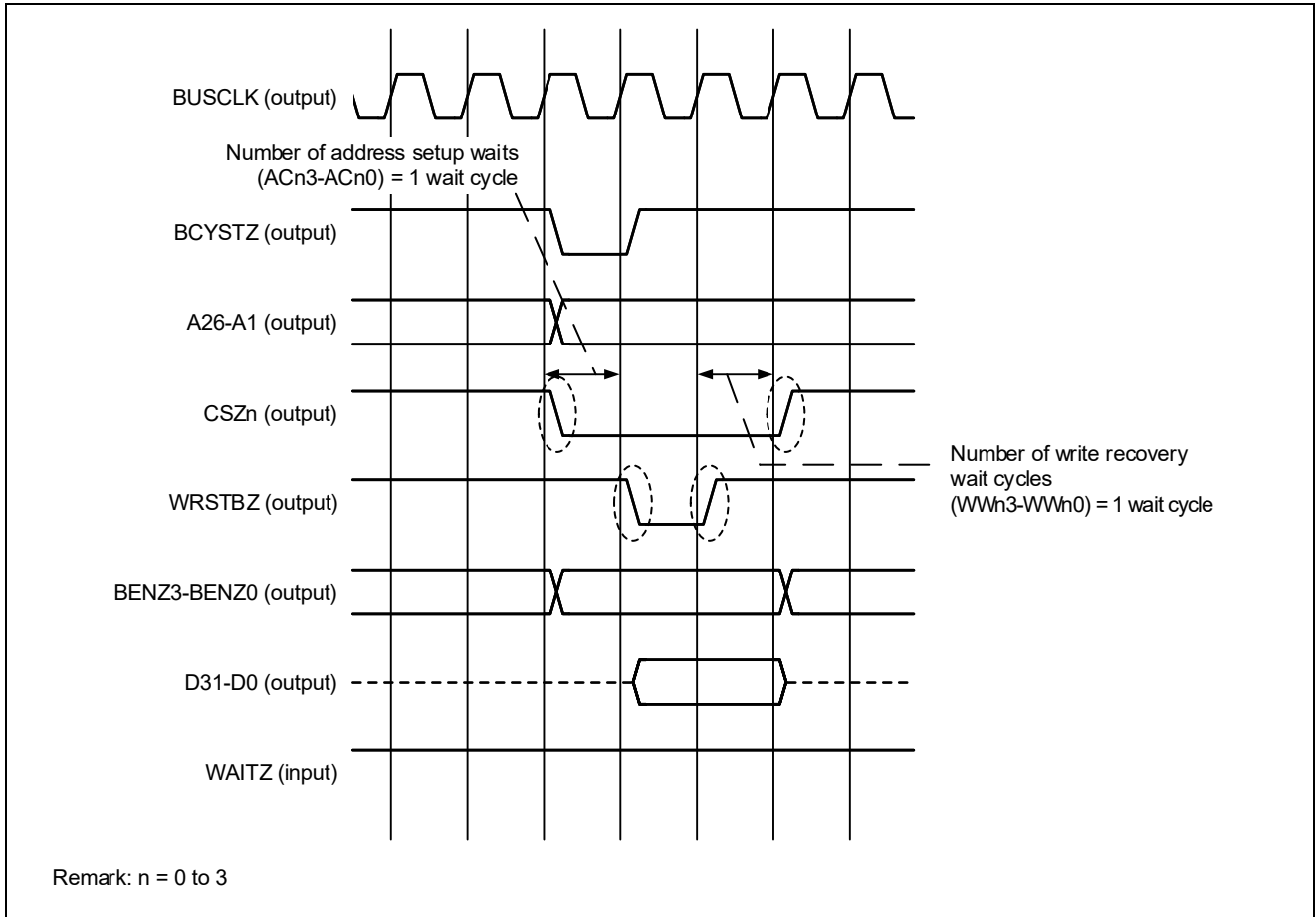


Figure 9.11 SRAM Write Cycles (with No Wait)

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)

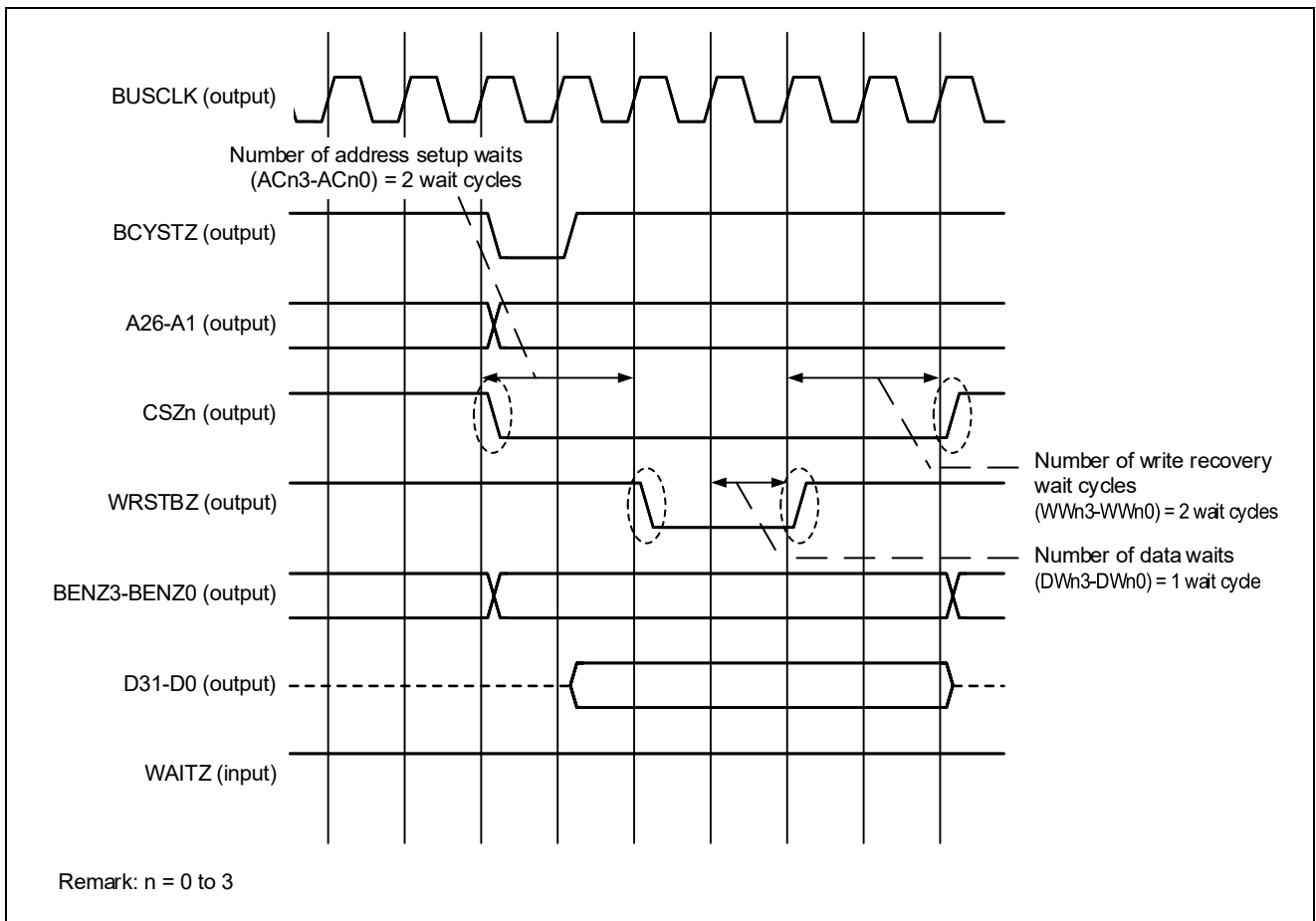


Figure 9.12 SRAM Write Cycles (with Wait States)

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)

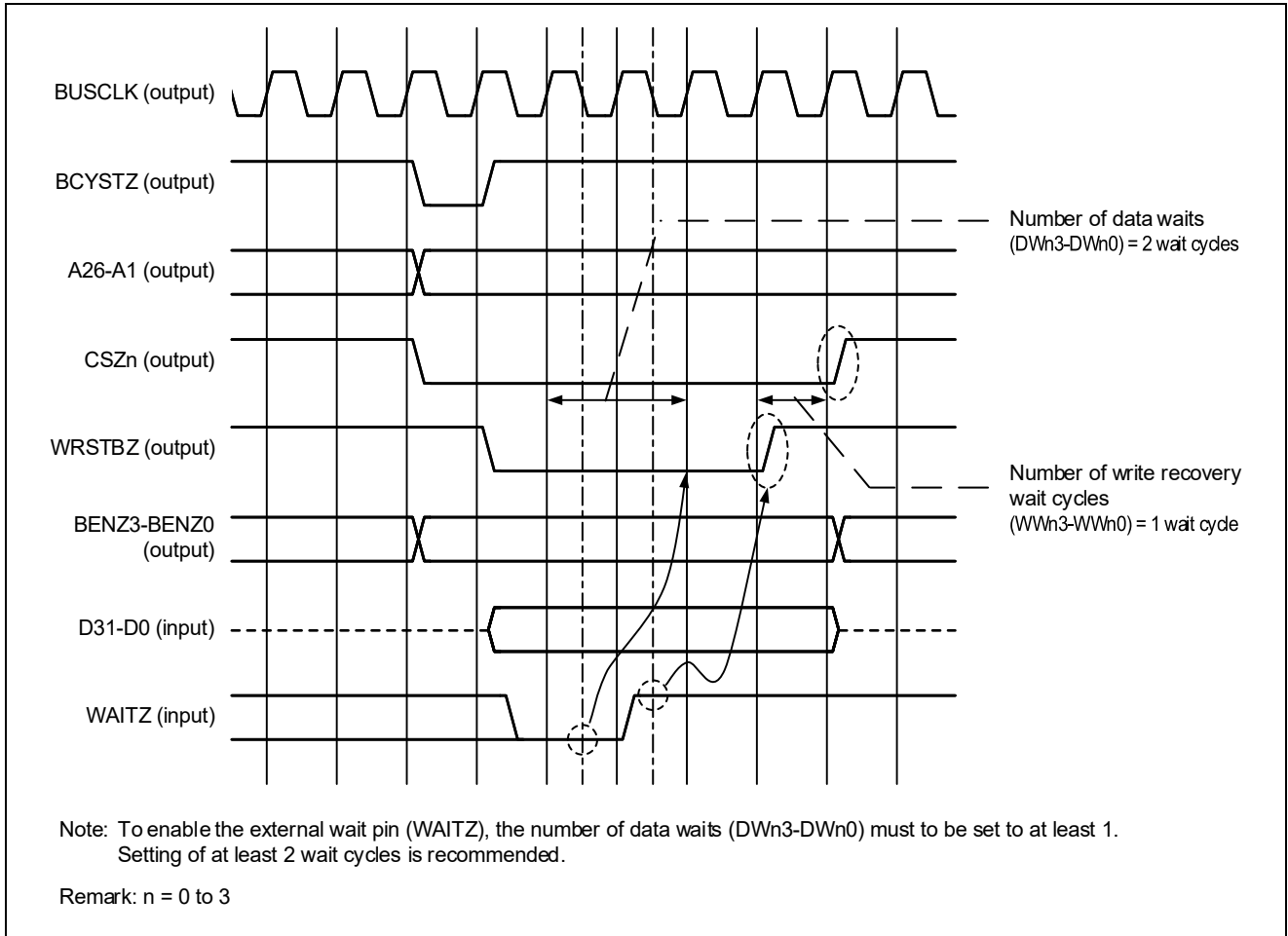


Figure 9.13 SRAM Write Cycles (External Wait Insertion)

BSC: SBS3-SBS0 = 1111B (32 bits), SMC0: DW3-DW0 = 0001B (1 wait cycle)

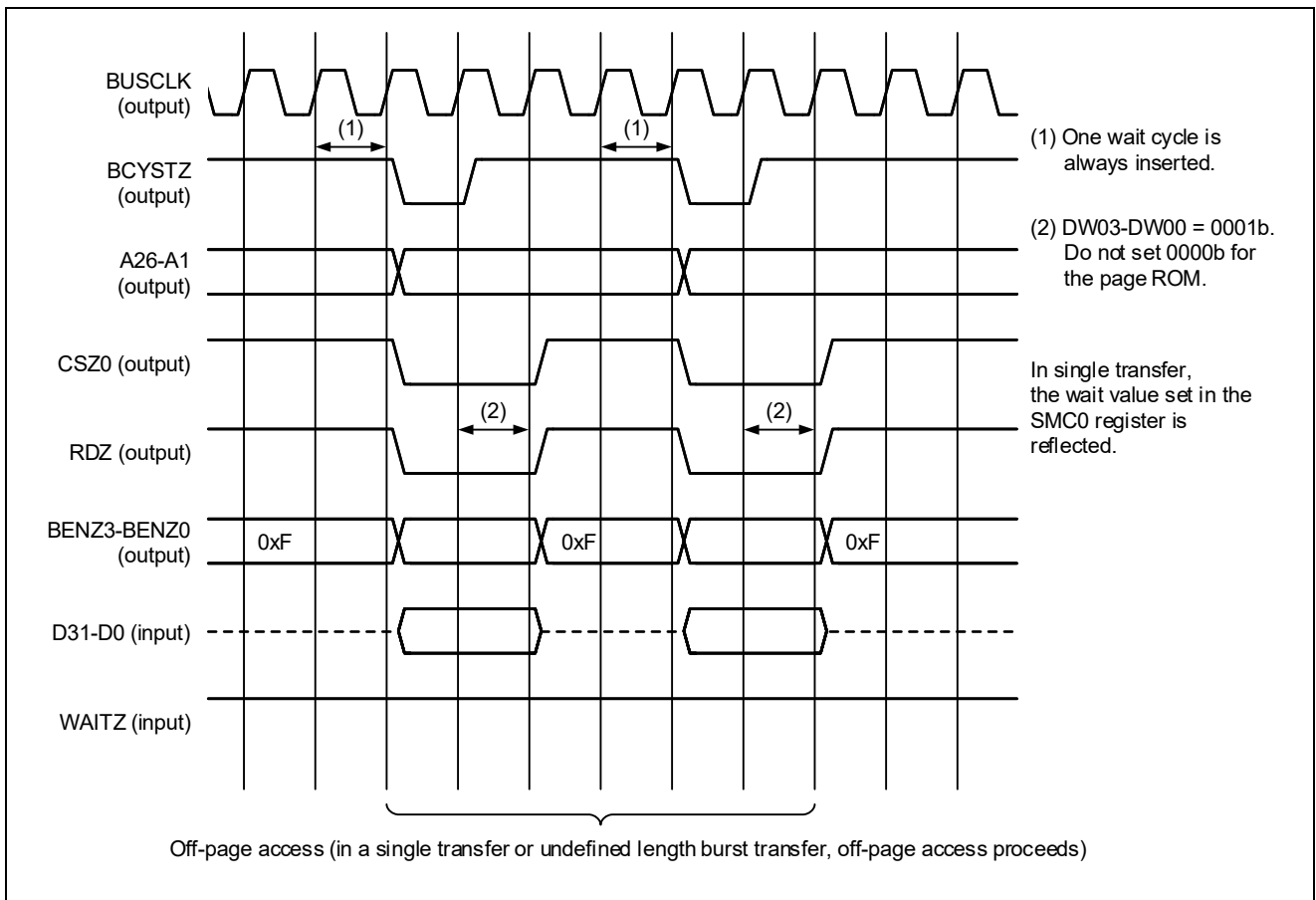


Figure 9.14 Page ROM Read Cycles (Single Transfer)

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)

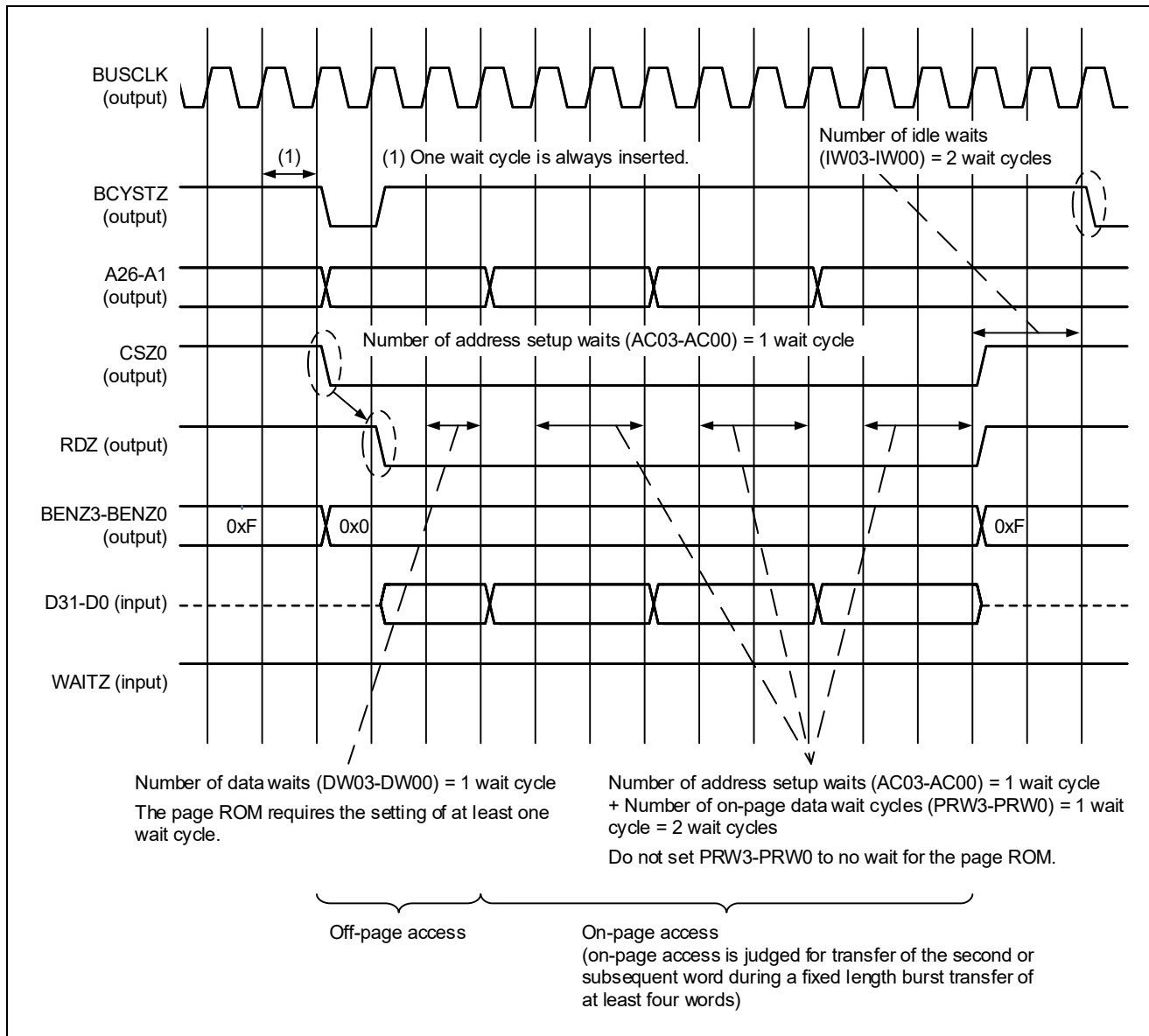


Figure 9.15 Page ROM Read Cycles (Four Burst Transfer)

10. 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 MEMCSEL and ADMUXMODE pins during operation. Fix the setting before the reset period ends.

10.1 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 WAITZ3) can be input.
 - The active level of the wait signals 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 taken in at the rising edge of BUSCLK.
 - Read data and the wait signals (WAITZ, WAITZ1 to WAITZ3) are taken in at the falling edge of BUSCLK.

10.2 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 10.1 Synchronous Burst Access Memory Controller Control Registers

Register Name	Symbol	Address
Wait signals selection register	WAITZSEL	4001 0108H
Synchronous burst access memory controller area select register 0	SMADSEL0	4001 0110H
Synchronous burst access memory controller area select register 1	SMADSEL1	4001 0114H
Synchronous burst access memory controller area select register 2	SMADSEL2	4001 0118H
Synchronous burst access memory controller area select register 3	SMADSEL3	4001 011CH
Bus clock division setting register	BCLKSEL	4001 0120H
Synchronous burst access memory controller operation mode setting register	SMCMD	4001 0124H
Synchronous burst access memory controller direct command register	DIRECTCMD	400A 8010H
Synchronous burst access memory controller cycle setting register	SETCYCLES	400A 8014H
Synchronous burst access memory controller mode setting register	SETOPMODE	400A 8018H
Synchronous burst access memory controller refresh setting register	REFRESH0	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

10.2.1 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 how to release protection, see the description of the system protection 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".

	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 4001 0108H Initial value 0000 000FH			
WAITZSEL	ESWT3	ESWT2	ESWT1	ESWT0																	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				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																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: The WAITZ3 pin is not used 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: The WAITZ2 pin is not used 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: The WAITZ1 pin is not used 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

Bit Position	Bit Name	Function
3 to 0	WSEL0n	Specify whether to enable the WAITZ input signal for each chip select area. 0000: The WAITZ pin is not used 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

10.2.2 Synchronous Burst Access Memory Controller Area Select Registers (SMADSEL0 to SMADSEL3)

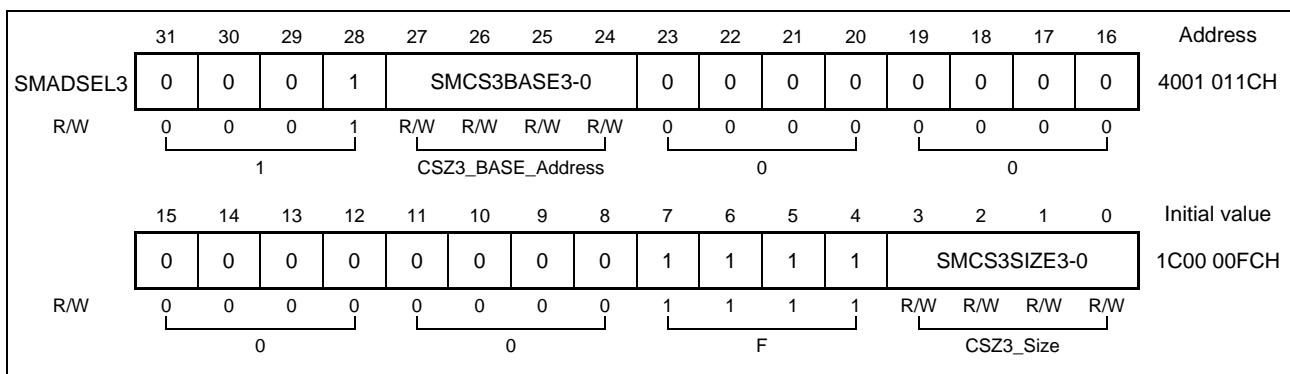
These registers are used to specify the allocation of the CSZ0 to CSZ3 areas. Before changing the initial value, be sure to copy the program to an area other than the external memory area.

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 how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.

- Access These registers can be read or written in 32-bit units.

SMADSEL0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td><td style="text-align: right;">Address</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td colspan="4" style="text-align: center;">SMCS0BASE3-0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: right;">4001 0110H</td> </tr> </table>	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address	0	0	0	1	SMCS0BASE3-0				0	0	0	0	0	0	0	0	0	4001 0110H	R/W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> <tr> <td colspan="4" style="text-align: center;">1</td><td colspan="4" style="text-align: center;">CSZ0_BASE_Address</td><td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">0</td> </tr> </table>	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			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address																																																					
0	0	0	1	SMCS0BASE3-0				0	0	0	0	0	0	0	0	0	4001 0110H																																																					
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																																																										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: right;">Initial value</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td colspan="4" style="text-align: center;">SMCS0SIZE3-0</td> </tr> </table>	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				R/W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td> </tr> <tr> <td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">F</td><td colspan="4" style="text-align: center;">CSZ0_Size</td> </tr> </table>	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						
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																																																										
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																																																										
SMADSEL1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td><td style="text-align: right;">Address</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td colspan="4" style="text-align: center;">SMCS1BASE3-0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: right;">4001 0114H</td> </tr> </table>	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address	0	0	0	1	SMCS1BASE3-0				0	0	0	0	0	0	0	0	0	4001 0114H	R/W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> <tr> <td colspan="4" style="text-align: center;">1</td><td colspan="4" style="text-align: center;">CSZ1_BASE_Address</td><td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">0</td> </tr> </table>	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			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address																																																					
0	0	0	1	SMCS1BASE3-0				0	0	0	0	0	0	0	0	0	4001 0114H																																																					
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																																																										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: right;">Initial value</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td colspan="4" style="text-align: center;">SMCS1SIZE3-0</td> </tr> </table>	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				R/W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td> </tr> <tr> <td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">F</td><td colspan="4" style="text-align: center;">CSZ1_Size</td> </tr> </table>	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						
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																																																										
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																																																										
SMADSEL2	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td><td style="text-align: right;">Address</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td colspan="4" style="text-align: center;">SMCS2BASE3-0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: right;">4001 0118H</td> </tr> </table>	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address	0	0	0	1	SMCS2BASE3-0				0	0	0	0	0	0	0	0	0	4001 0118H	R/W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> <tr> <td colspan="4" style="text-align: center;">1</td><td colspan="4" style="text-align: center;">CSZ2_BASE_Address</td><td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">0</td> </tr> </table>	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			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address																																																					
0	0	0	1	SMCS2BASE3-0				0	0	0	0	0	0	0	0	0	4001 0118H																																																					
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																																																										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: right;">Initial value</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td colspan="4" style="text-align: center;">SMCS2SIZE3-0</td> </tr> </table>	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				R/W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td><td style="text-align: center;">R/W</td> </tr> <tr> <td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">0</td><td colspan="4" style="text-align: center;">F</td><td colspan="4" style="text-align: center;">CSZ2_Size</td> </tr> </table>	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						
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																																																										
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.



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**
- The total size of all CSZn areas is 256 MB.
 - The specifiable address space is from 1000 0000H to 1FFF FFFFH.
 - The CSZn areas must not overlap. Specify base addresses and sizes such that the CSZ areas do not overlap.
 - 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**
- 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
 - n = 0 to 3

10.2.3 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 how to release protection, see the description of the system protection 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				Address
BCLKSEL	0 0				4001 0120H
					Initial value
					0000 0004H
R/W	0 0			R/W R/W R/W	

Bit Position	Bit Name	Function
31 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 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

10.2.4 Synchronous Burst Access Memory Controller Operation Mode Setting Register (SMCMD)

This register is used to specify the operating mode of the synchronous burst access memory controller.

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 how to release protection, see the description of the system protection 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.

SMCMD	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 4001 0124H Initial value 0000 0000H				
	0	0	0	0	0	0	0	0	0	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	0	0	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

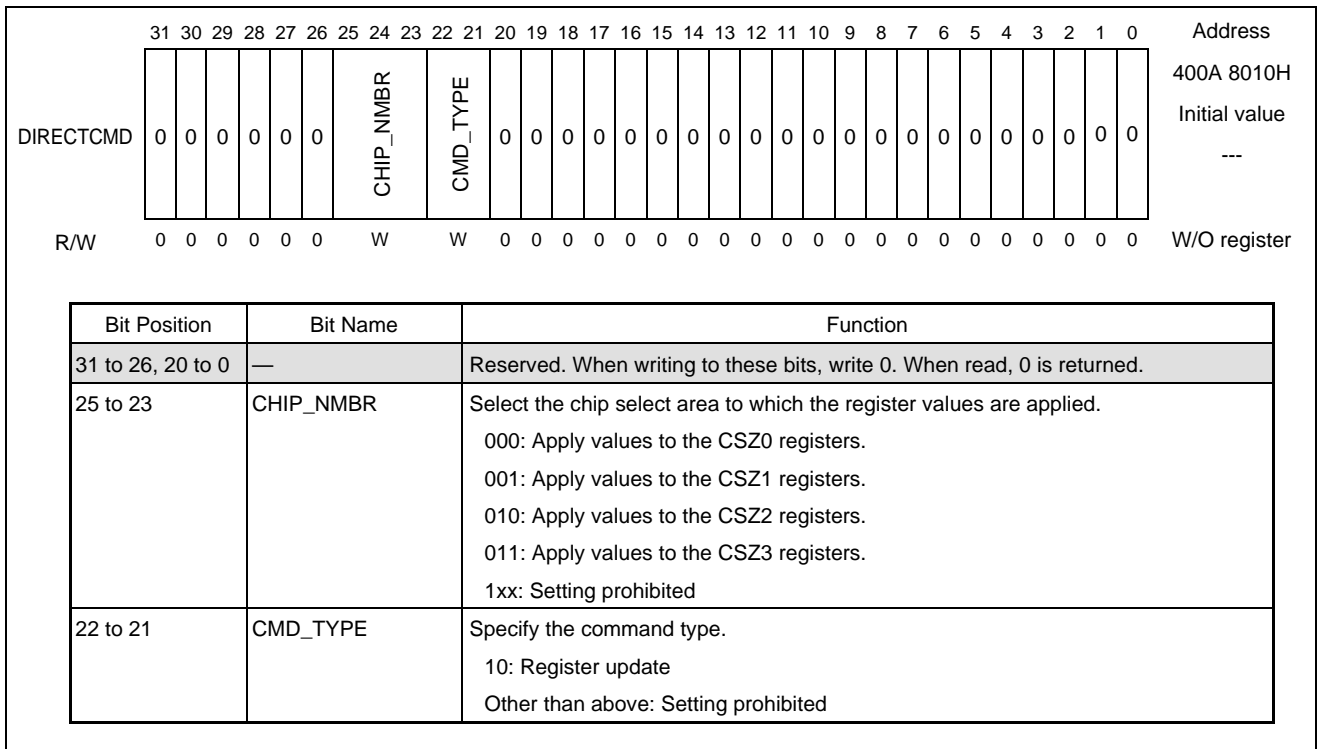
Bit Position	Bit Name	Function
31 to 5	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
4	MAGTD1	Fix the output from the MA16 to MA26 pins to low level. (Pins that function alternately as port pins output a low level only when used as port pins.) ^{Note1} 0: Regular usage 1: 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.) ^{Note1} 0: Regular usage 1: Fix the output from the MA0 to MA15 pins to low level.
2	SMCRDLTH	Select the SRAM read timing ^{Note2} 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 SETCYCLES 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. Timing examples in each mode are shown in 10.3.1(2) Bus Clock 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.

10.2.5 Synchronous Burst Access Memory Controller Direct Command Register (DIRECTCMD)

This register is used to apply the values set in the synchronous burst access memory controller cycle setting register (SETCYCLES) and synchronous burst access memory controller mode setting register (SETOPMODE) 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.



Remark: n = 0 to 3

10.2.6 Synchronous Burst Access Memory Controller Cycle Setting Register (SETCYCLES)

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 (SETOPMODE), and then apply the values to each chip select area by using the synchronous burst access memory controller direct command register (DIRECTCMD).

	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 CYCLES	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">WE_TIME</td><td style="width: 2.5%;">T_TR</td><td style="width: 2.5%;">T_PC</td><td style="width: 2.5%;">T_WP</td><td style="width: 2.5%;">T_CEOE</td><td style="width: 2.5%;">T_WC</td><td style="width: 2.5%;">T_RC</td> </tr> </table>	0	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	
0	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				
		Initial value	—																			
R/W	0 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 SMCMD 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) ^{Note 1} 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 signal 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 10.23, 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 10.22, Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled.

10.2.7 Synchronous Burst Access Memory Controller Mode Setting Register (SETOPMODE)

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 (SETCYCLES), and then apply the values to each chip select area by using the synchronous burst access memory controller direct command register (DIRECTCMD).

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 BENZO 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 is valid when the ADVZ signal is at the 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

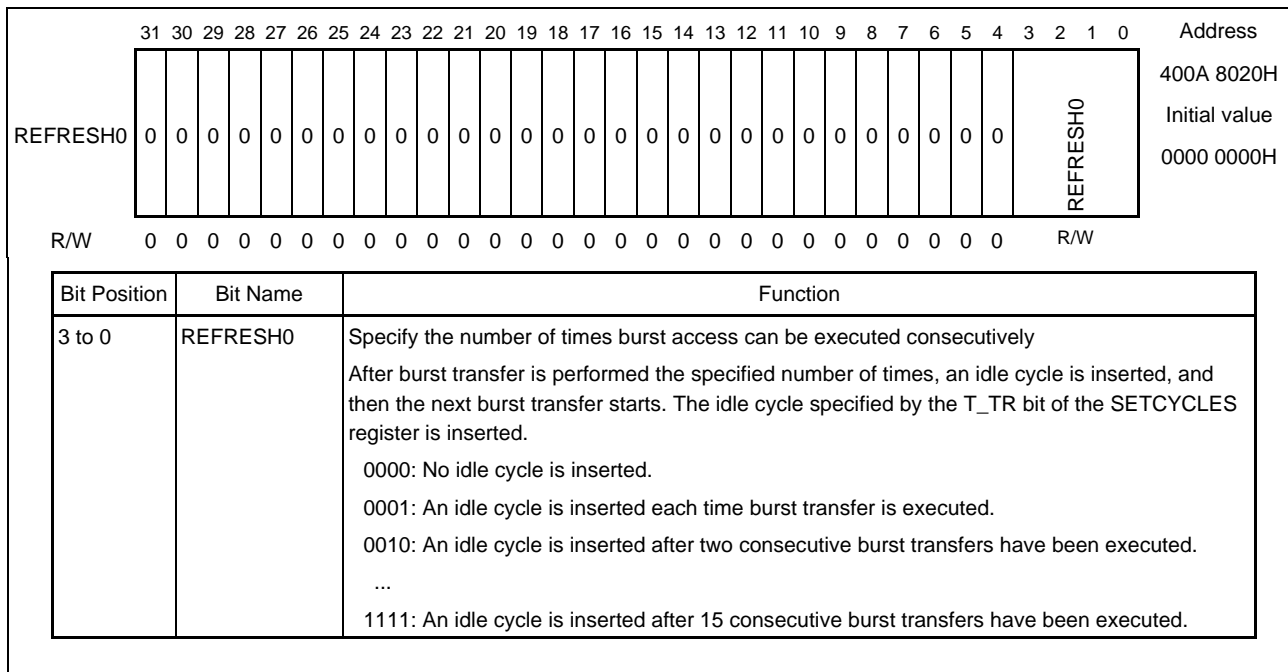
Note: Only single access can be specified for asynchronous access. Otherwise, setting is prohibited.

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 for asynchronous access other than page read access. Otherwise, setting is prohibited.

10.2.8 Synchronous Burst Access Memory Controller Refresh Setting Register (REFRESH0)

This register is used to specify the number of times burst access can be executed consecutively.

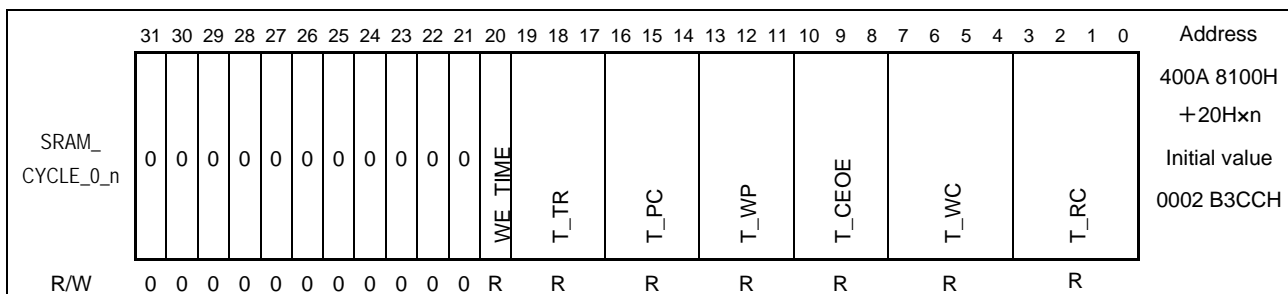


Caution: Set 0x000_0001 in this register if the SMCWETH bit of the SMCMD register is set to 1 enabling use of the address/data signal in separate bus mode.

10.2.9 Synchronous Burst Access Memory Controller CSZn Cycle Setting Registers (SRAM_CYCLE0_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 (SETCYCLES) can be read from each bit.

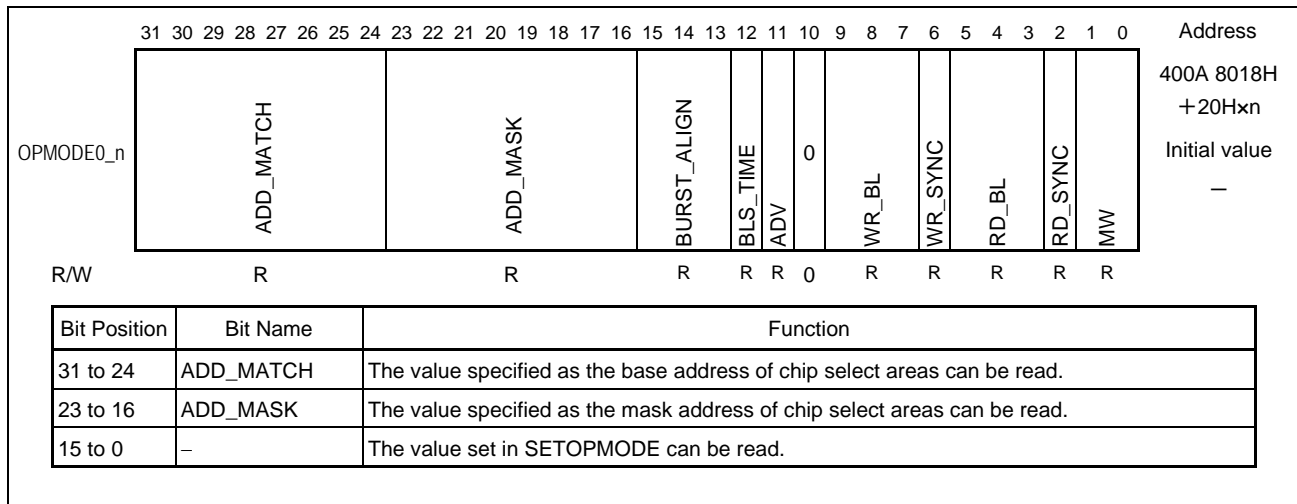


Remark: n = 0 to 3

10.2.10 Synchronous Burst Access Memory Controller CSZn Mode Registers (OPMODE0_n)

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 (SETOPMODE) can be referenced by using the lower-order 16 bits of each register.



Remark: n = 0 to 3

10.2.11 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.

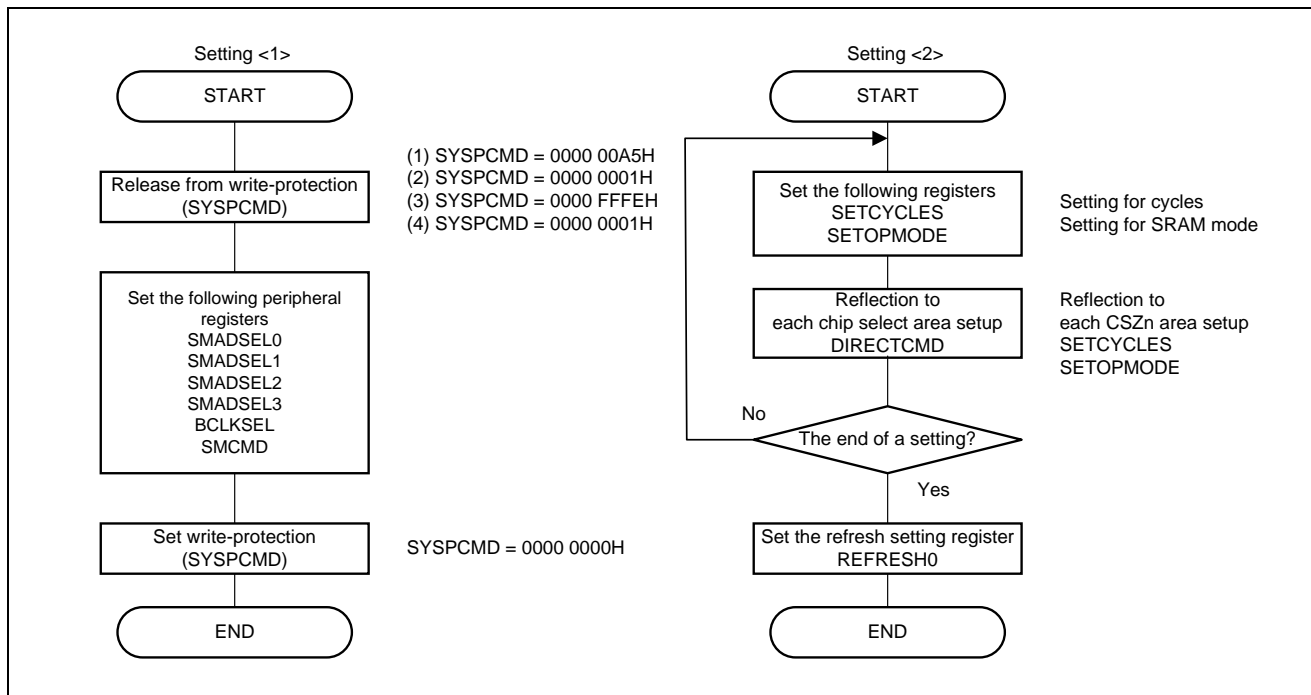


Figure 10.1 Register Setup Procedure

10.3 Function Details

10.3.1 Bus Clock Control

(1) Bus Clock 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) Bus Clock Masking

The bus clock (BUSCLK) can be output for the period in which the CSZn signal is active, which is specified by the SMCMD register.

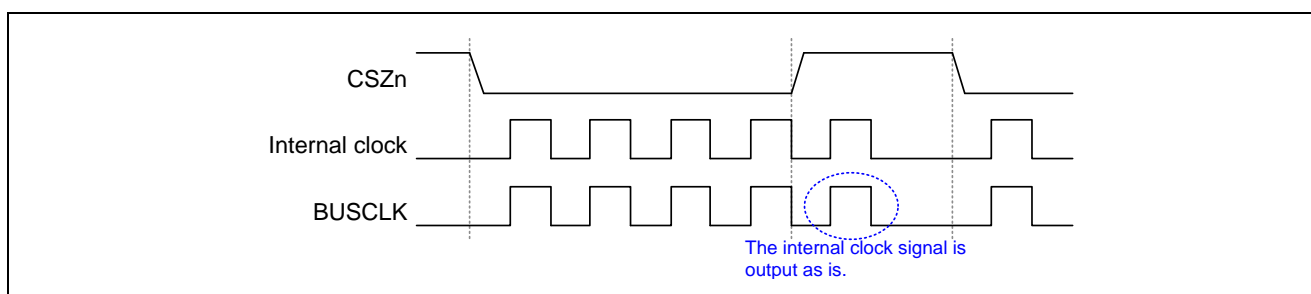


Figure 10.2 Clock Output Timing Example (SMCMD.SMCCLKTH = 0)

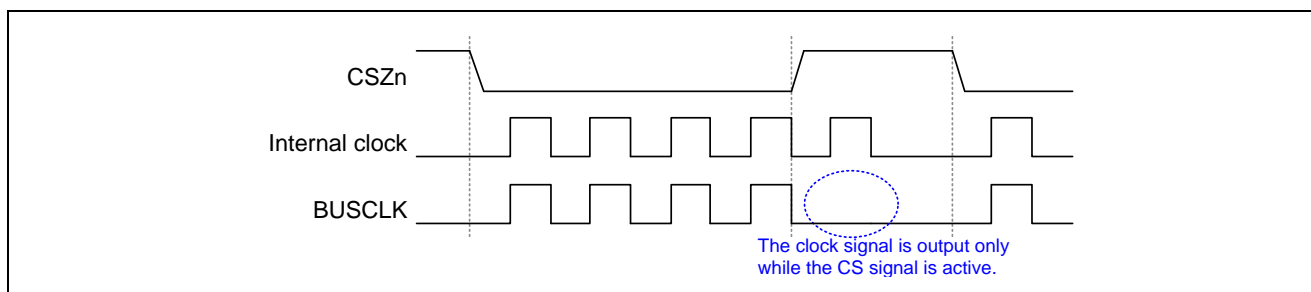


Figure 10.3 Clock Output Timing Example (SMCMD.SMCCLKTH = 1)

Remark: n = 0 to 3

10.3.2 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

10.3.3 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 "10.4 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 10.10 Asynchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Write: Figure 10.13 Asynchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled, WE_TIME = 0

Synchronous access

Read: Figure 10.16 Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Write: Figure 10.20 Synchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled

10.3.4 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 CSZn signal 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 SMCMD register to 1.

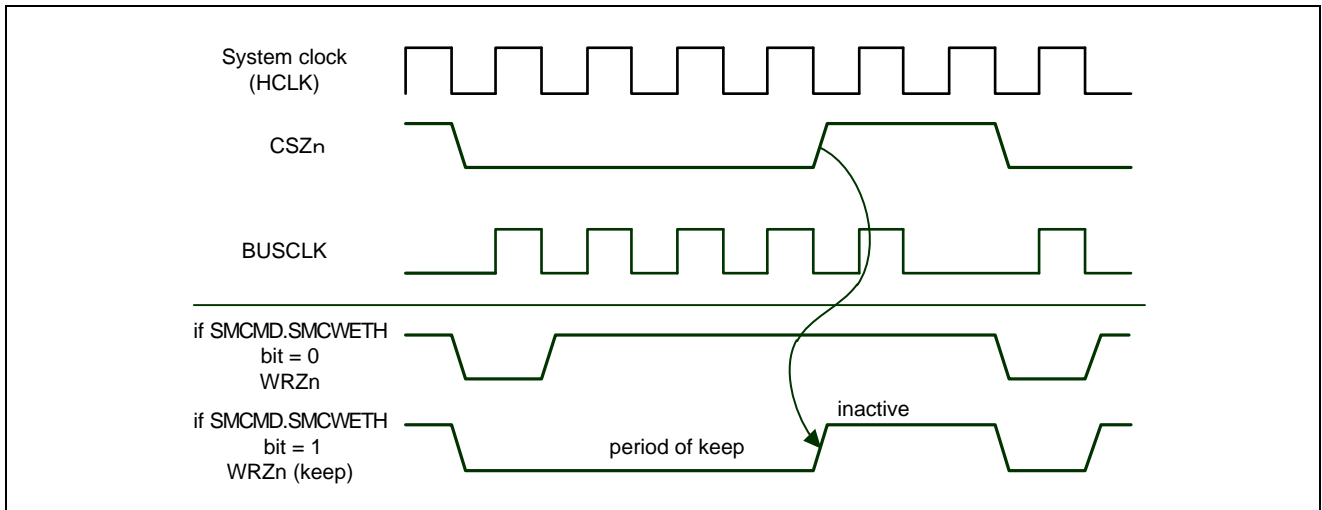


Figure 10.4 Write Enable Signal Operation

Remark: n = 0 to 3

10.3.5 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-IN32M3 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.

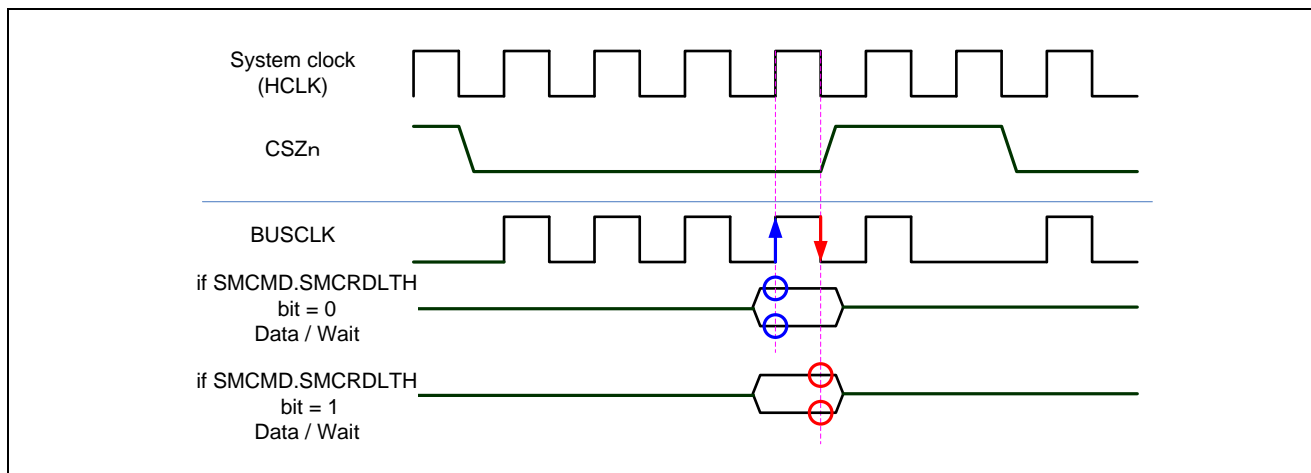


Figure 10.5 Read Data Timing Control

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.

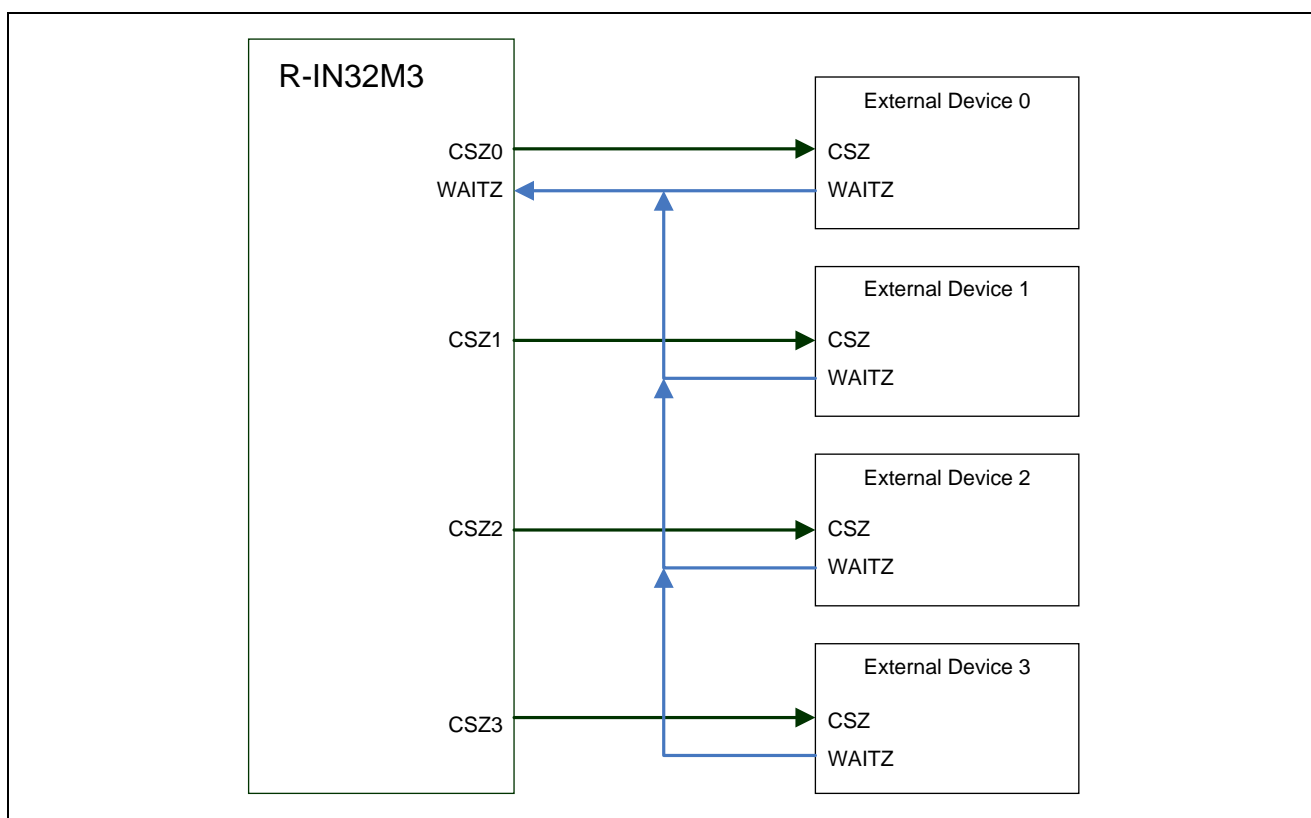
10.3.6 Wait Signals Control

The synchronous burst access memory controller can use up to four external wait input pins (WAITZ, WAITZ1 to WAITZ3) for 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-IN32M3, the external devices, and external memory interface pins, refer to the R-IN32M3 Series 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.



Remark: The settings of the wait signals selection register are as follows.

WAITZSEL.WSEL0[3:0] = 1111B

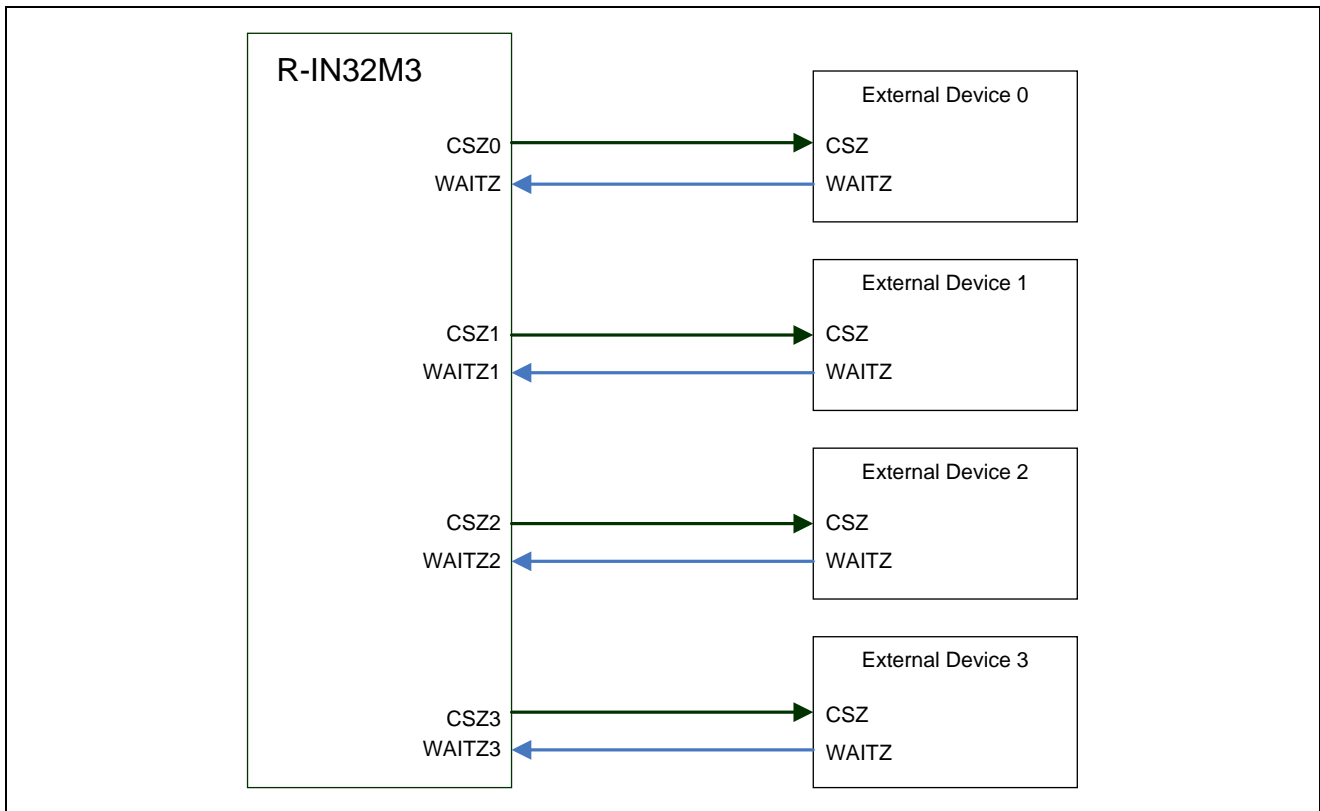
WAITZSEL.WSEL1[3:0] = 0000B

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.



Remark: The settings of the wait signals selection register are as follows.

WAITZSEL.WSEL0[3:0] = 0001B

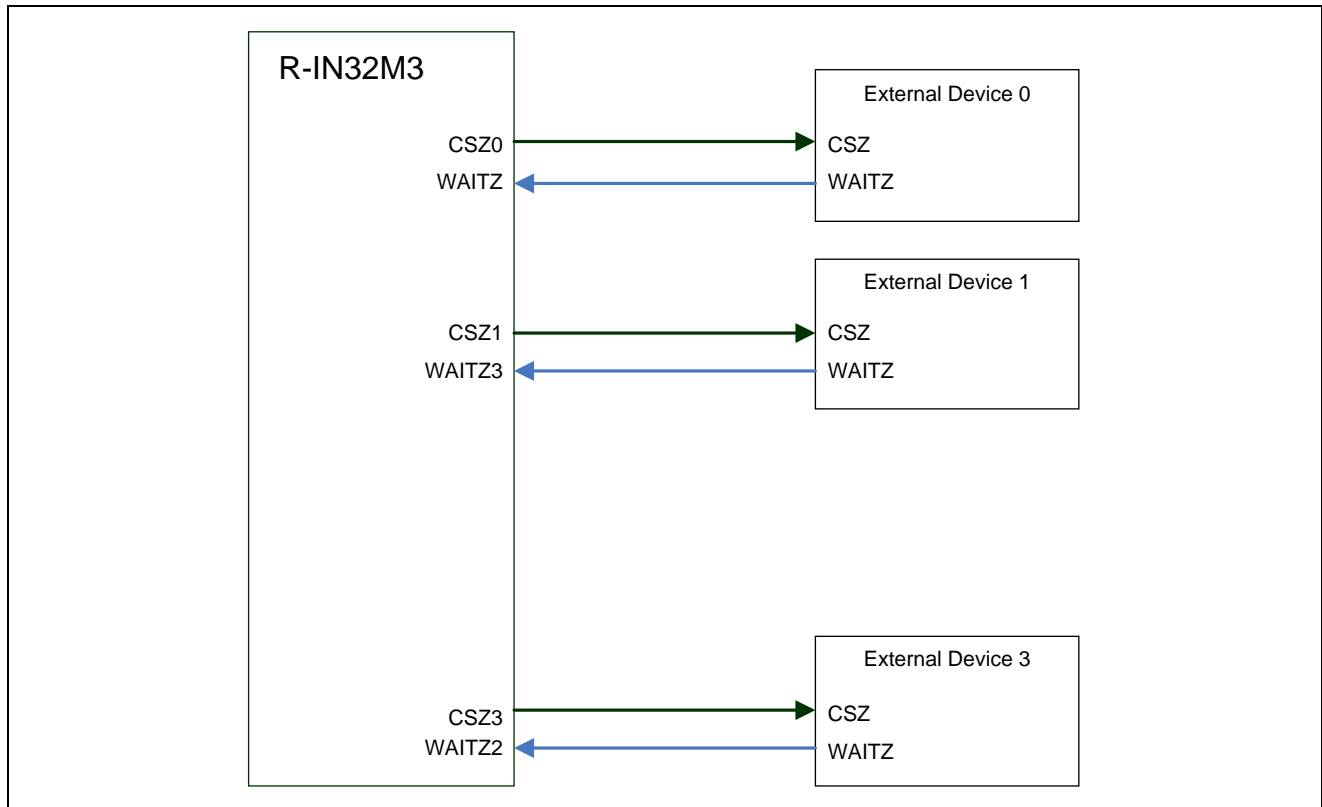
WAITZSEL.WSEL1[3:0] = 0010B

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.



Remarks 1. The wait signals selection register (WAITZSEL) 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

10.3.7 Specify the Operating Mode of the Synchronous Burst Access Memory Controller

Specify the operating mode for R-IN32M3 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 CS area bus width. 0: 16-bit bus 1: 32-bit bus

10.3.8 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 the 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.**

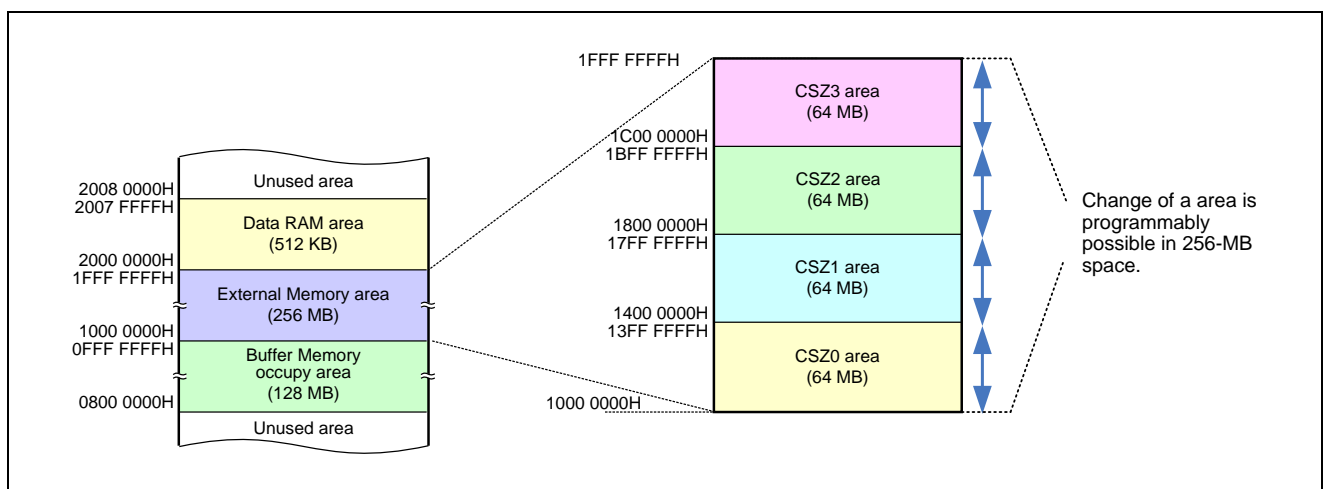


Figure 10.6 External Memory Space

10.4 Memory Access Timing Example

Memory access timing examples are shown below.

Table 10.2 Memory Access Timing Examples

Figure	Memory Type	Access Conditions	Page
Figure 10.7	Asynchronous SRAM	Read access, separate bus mode, ADVZ enabled	10-28
Figure 10.8	Asynchronous SRAM	Read access, separate bus mode, ADVZ disabled	10-29
Figure 10.9	Page ROM	Read access, separate bus mode, ADVZ enabled	10-30
Figure 10.10	Asynchronous SRAM	Read access, multiplexed bus mode, ADVZ enabled	10-31
Figure 10.11	Asynchronous SRAM	Write access, separate bus mode, ADVZ disabled	10-32
Figure 10.12	Asynchronous SRAM	Write access, separate bus mode, ADVZ enabled	10-33
Figure 10.13	Asynchronous SRAM	Write access, multiplexed bus mode, ADVZ enabled, WE_TIME = 0	10-34
Figure 10.14	Asynchronous SRAM	Write access, multiplexed bus mode, ADVZ enabled, WE_TIME = 1	10-35
Figure 10.15	Synchronous SRAM	Read access, separate bus mode, ADVZ enabled	10-36
Figure 10.16	Synchronous SRAM	Read access, multiplexed bus mode, ADVZ enabled	10-37
Figure 10.17	Synchronous SRAM	4-data burst read access, multiplexed bus mode, ADVZ enabled	10-38
Figure 10.18	Synchronous SRAM	Write access, separate bus mode, ADVZ enabled	10-39
Figure 10.19	Synchronous SRAM	8-data burst write access, separate bus mode, ADVZ enabled	10-40
Figure 10.20	Synchronous SRAM	Write access, multiplexed bus mode, ADVZ enabled	10-41
Figure 10.21	Synchronous SRAM	4-data burst write access, multiplexed bus mode, ADVZ enabled	10-42
Figure 10.22	Synchronous SRAM	Read, external wait timing	10-43
Figure 10.23	Synchronous SRAM	Write, external wait timing	10-44

10.4.1 Asynchronous Access Timing

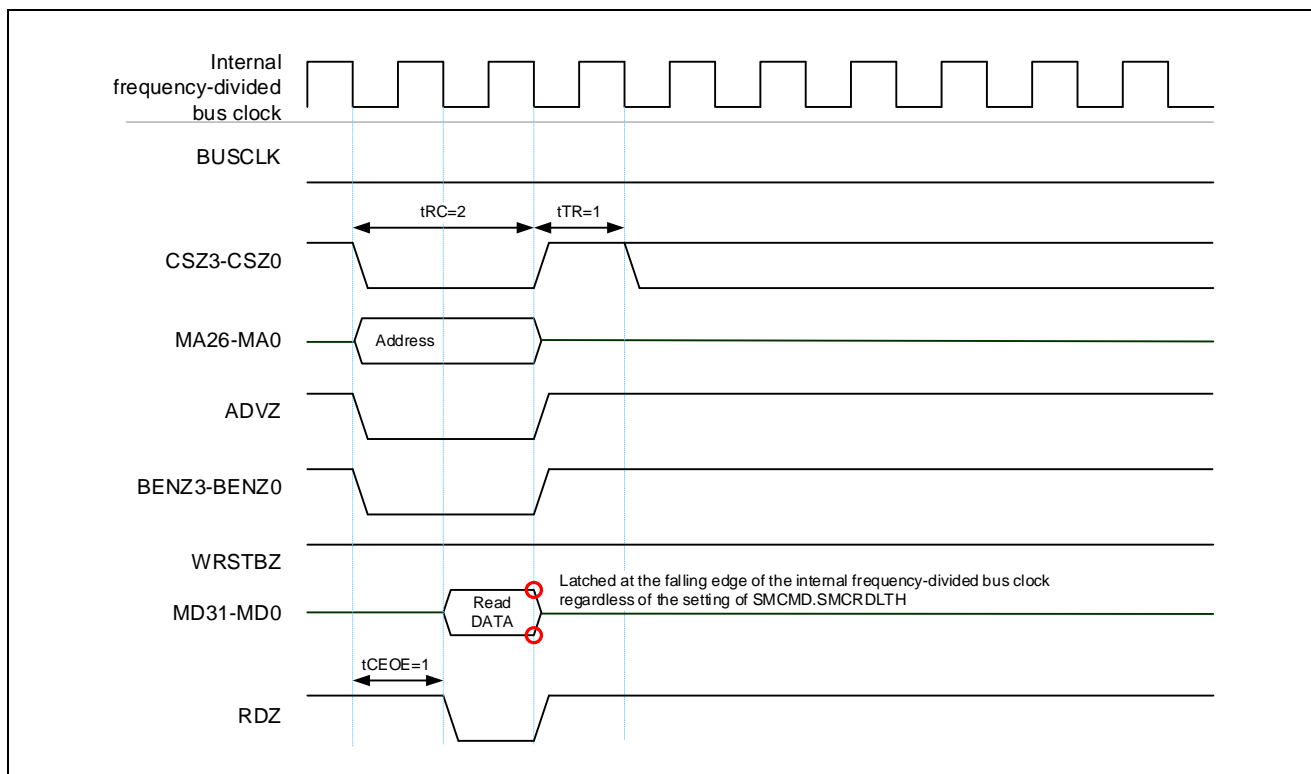


Figure 10.7 Asynchronous SRAM, Separate Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate bus mode)

SETCYCLES.T_TR[2:0] = 001B (1 cycle)

T_CEOE[2:0] = 001B (1 cycle)

T_RC[3:0] = 0010B (2 cycles)

SETOPMODE.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)

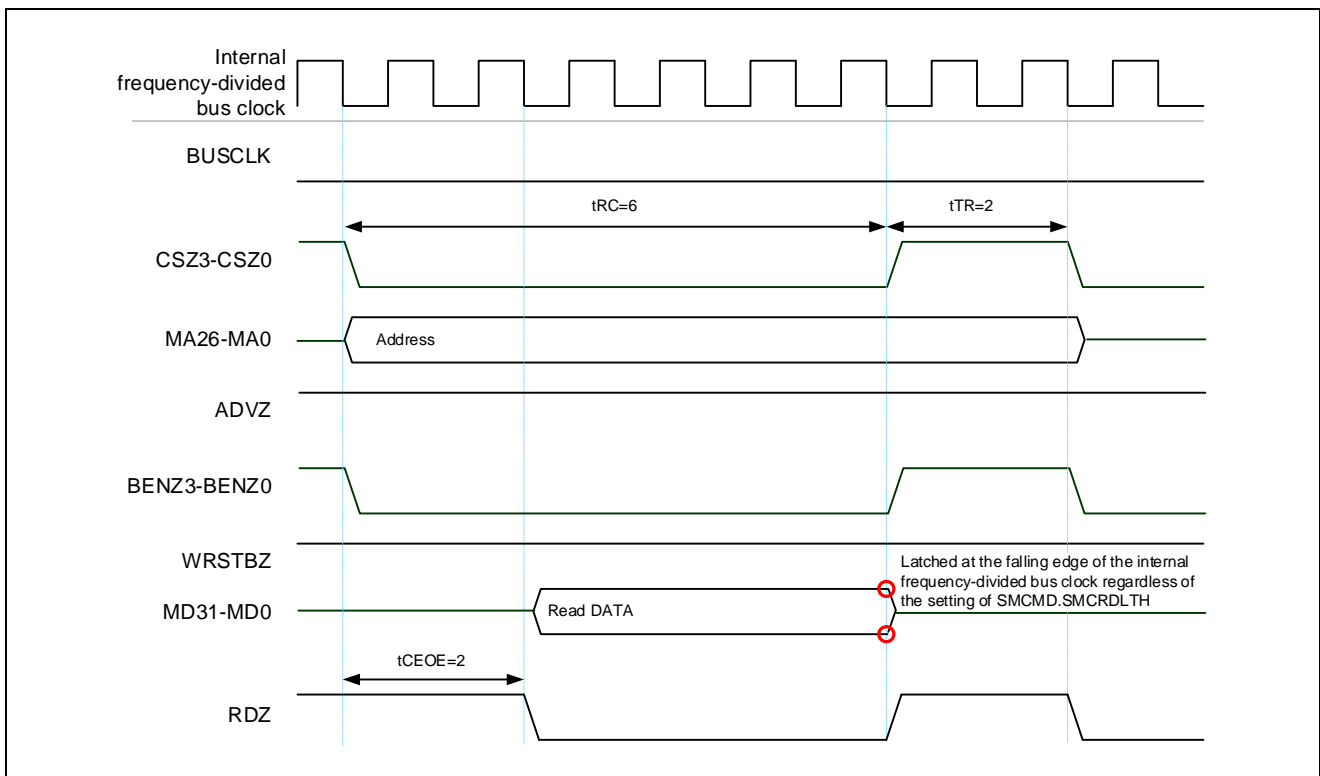


Figure 10.8 Asynchronous SRAM, Separate Bus Mode, Read Access, ADVZ Disabled

Remark: ADMUXMODE pin = Low level (separate bus mode)

SETCYCLES.T_TR[2:0] = 010B (2 cycles)

T_CEOE[2:0] = 010B (2 cycles)

T_RC[3:0] = 0110B (6 cycles)

SETOPMODE.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)

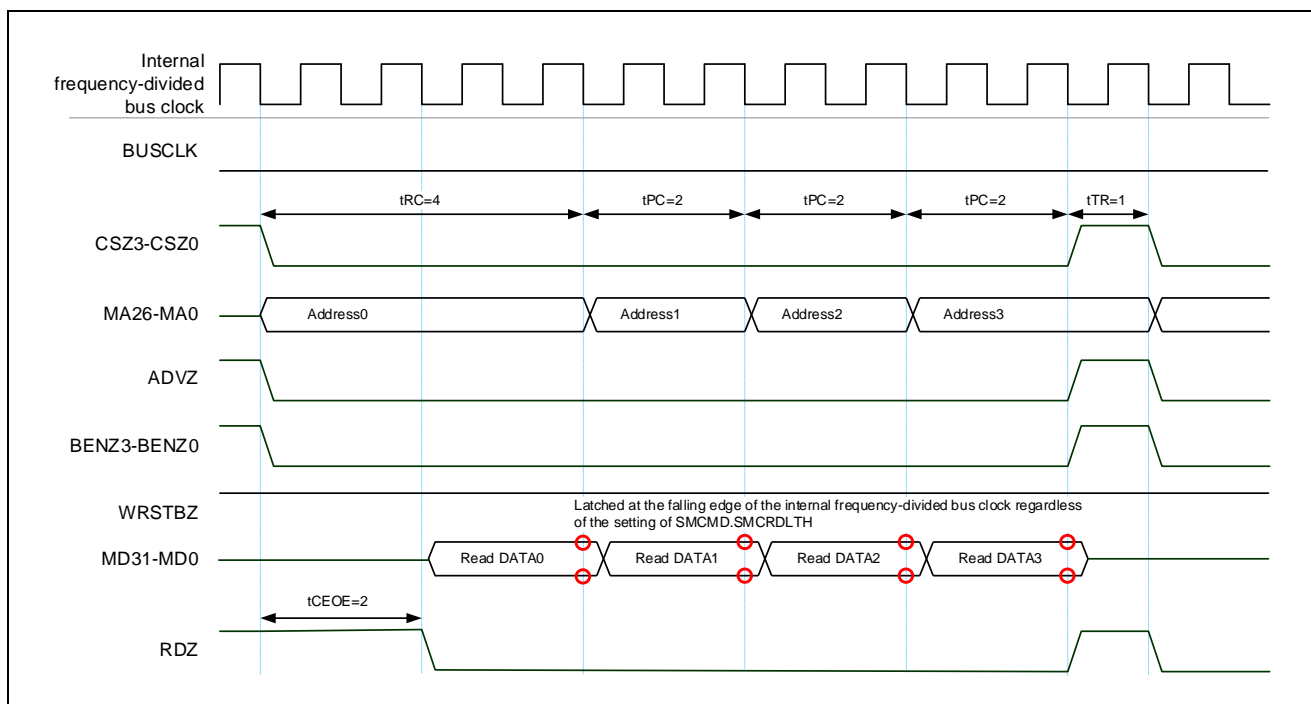


Figure 10.9 Asynchronous Page ROM, Separate Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate bus mode)
SETCYCLES.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)
SETOPMODE.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)

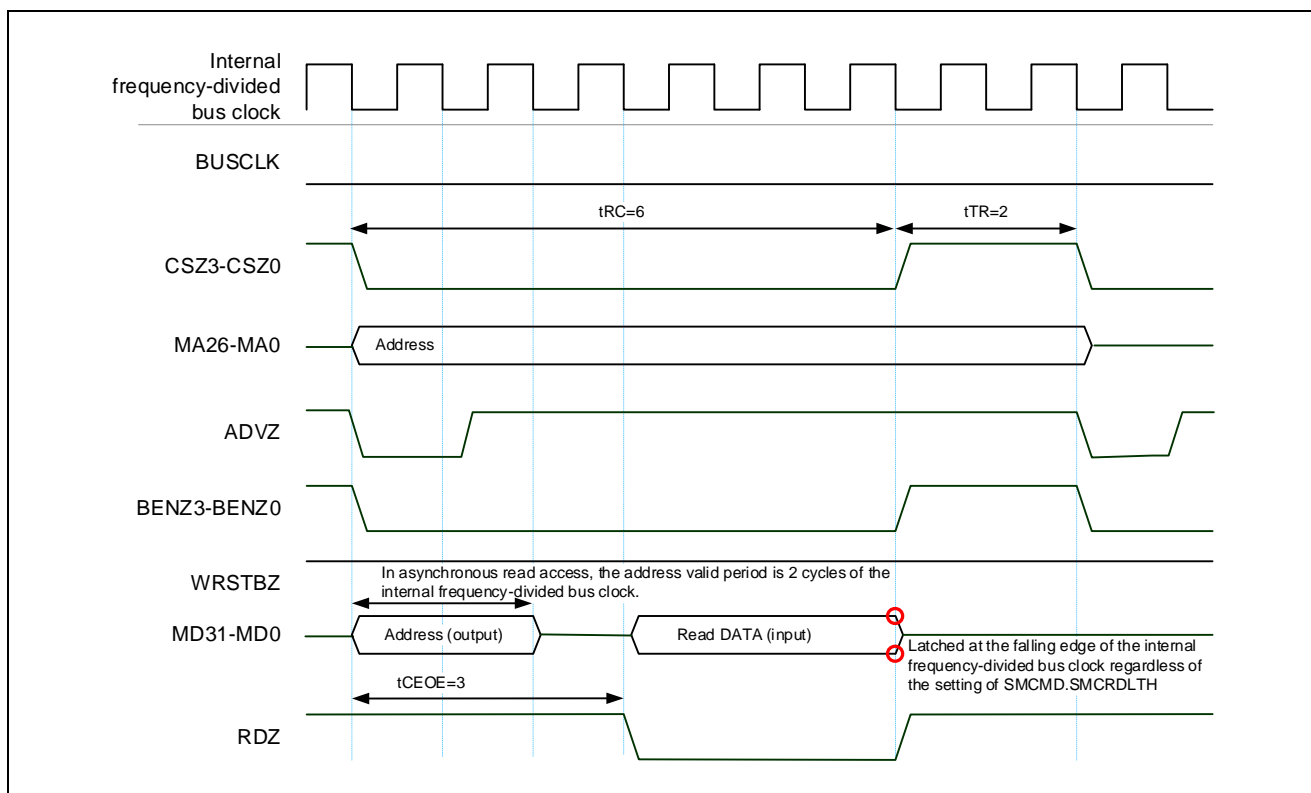


Figure 10.10 Asynchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed bus mode)
SETCYCLES.T_TR[2:0] = 010B (2 cycles)
T_CEOE[2:0] = 011B (3 cycles)
T_RC[3:0] = 0110B (6 cycles)
SETOPMODE.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)

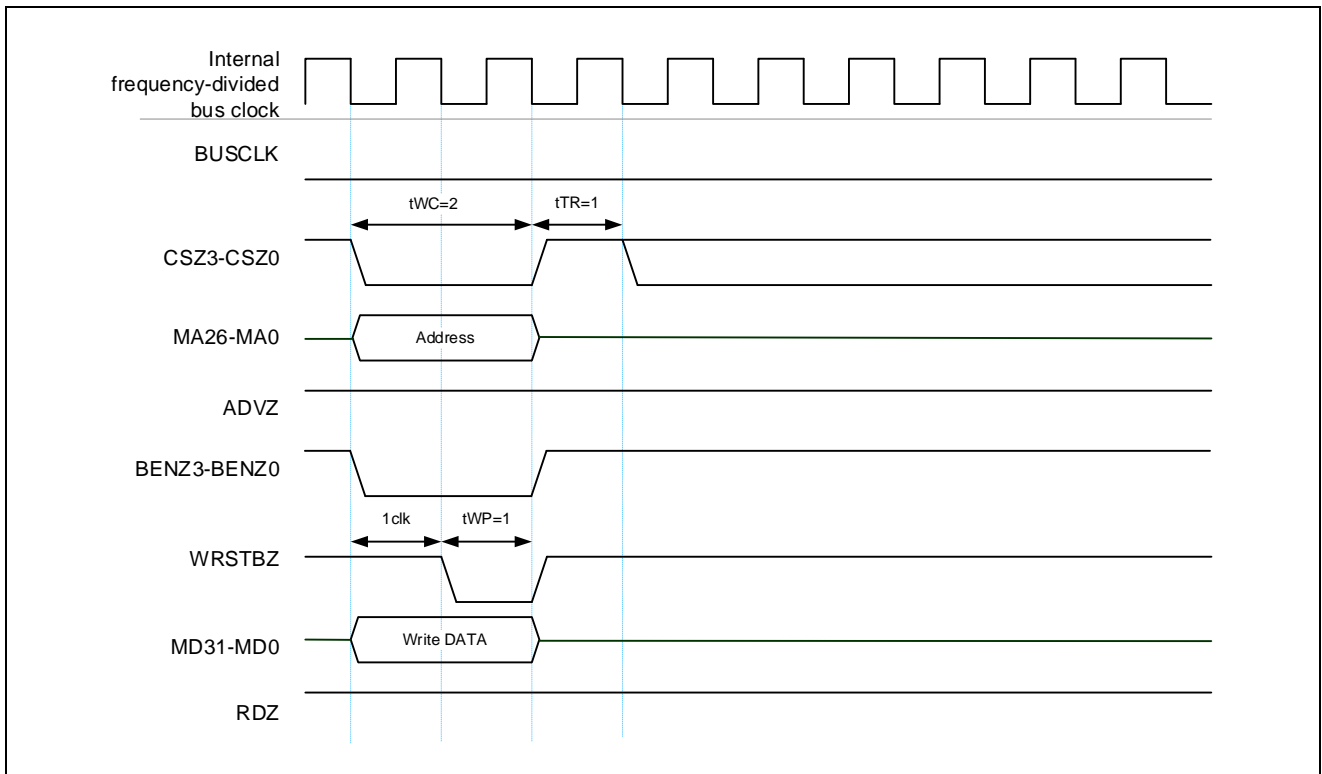


Figure 10.11 Asynchronous SRAM, Separate Bus Mode, Write Access, ADVZ Disabled

Remark: ADMUXMODE pin = Low level (separate bus mode)
 SETCYCLES.T_TR[2:0] = 001B (1 cycle)
 T_WP[2:0] = 001B (1 cycle)
 T_WC[3:0] = 0010B (2 cycles)
 SETOPMODE.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)

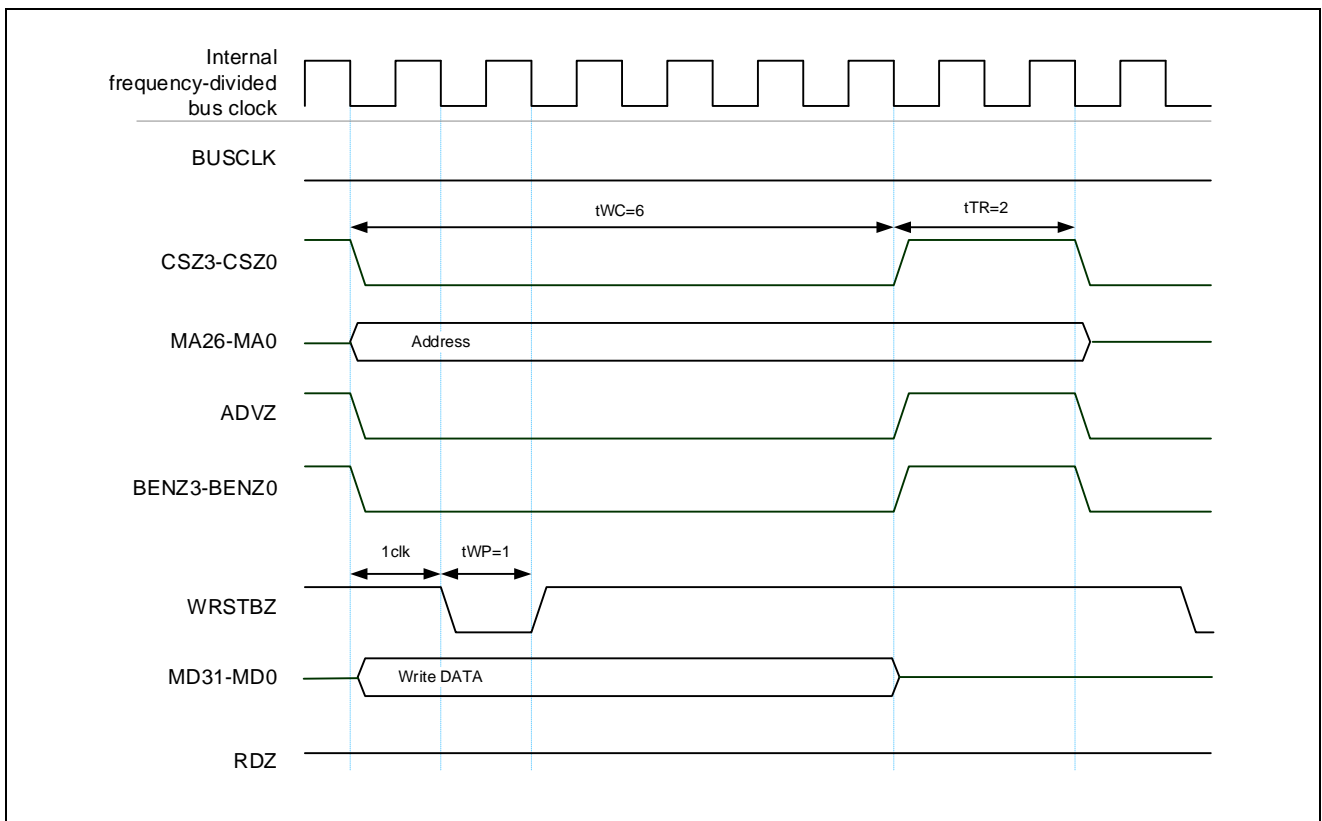


Figure 10.12 Asynchronous SRAM, Separate Bus Mode, Write Access, ADVZ Enabled

Remark: **ADMUXMODE pin = Low level (separate bus mode)**
SETCYCLES.T_TR[2:0] = 010B (2 cycles)
T_WP[2:0] = 001B (1 cycle)
T_WC[3:0] = 0110B (6 cycles)
SETOPMODE.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)

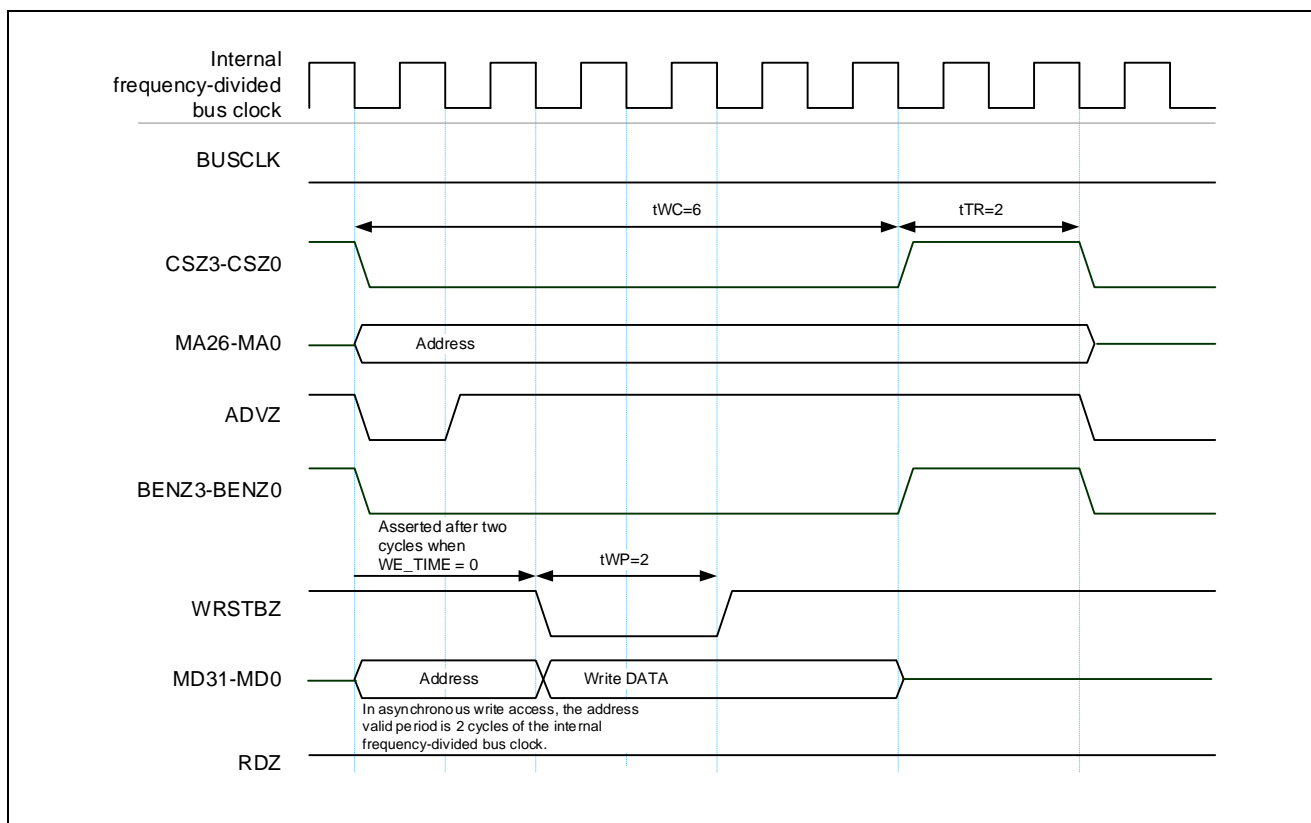


Figure 10.13 Asynchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled, WE_TIME = 0

Remark: ADMUXMODE pin = High level (multiplexed bus mode)

SETCYCLES.WE_TIME = 0B (2 cycles after the CSZ0 to CSZ3 signal is asserted)

T_TR[2:0] = 010B (2 cycles)

T_WP[2:0] = 010B (2 cycles)

T_WC[3:0] = 0110B (6 cycles)

SETOPMODE.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)

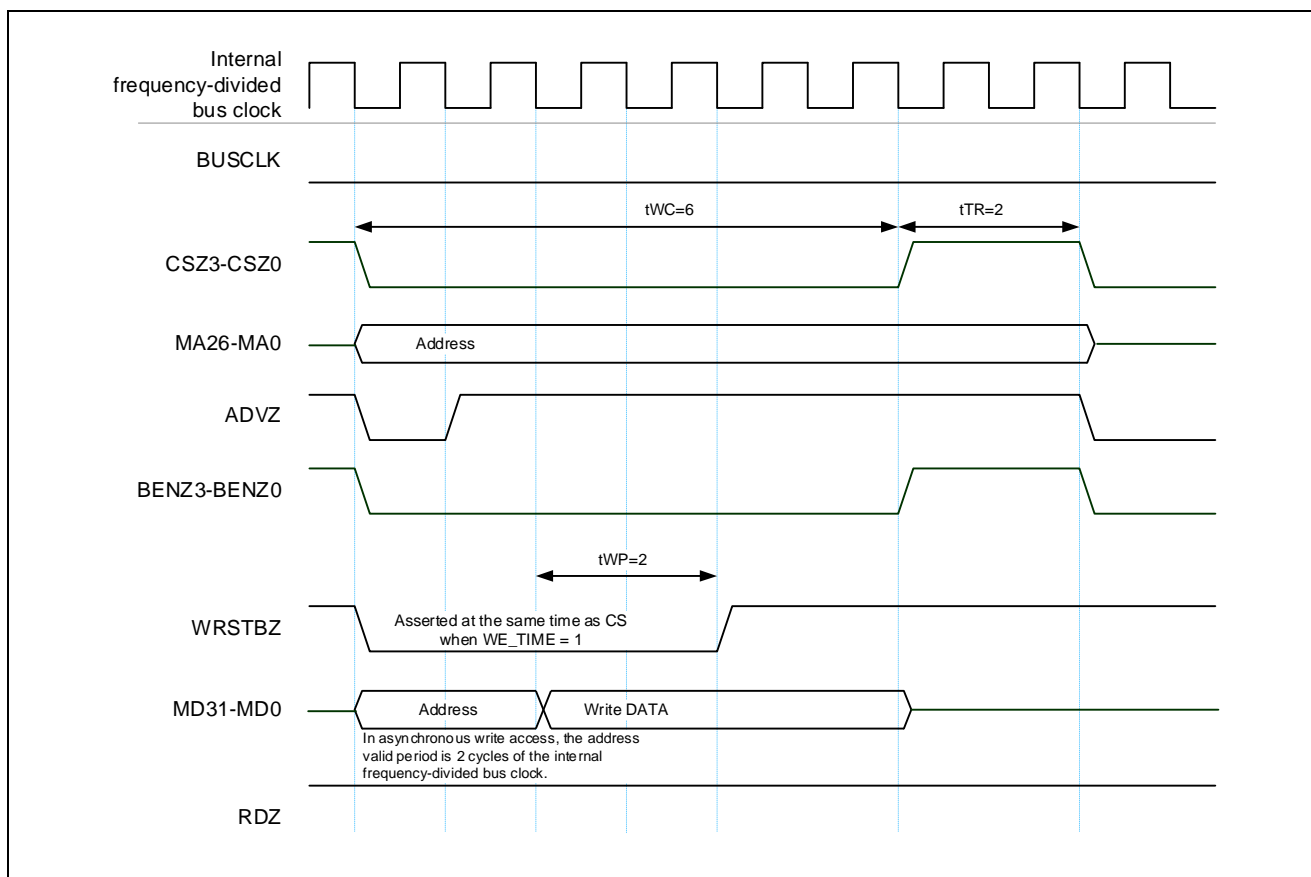


Figure 10.14 Asynchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled, WE_TIME = 1

Remark: ADMUXMODE pin = High level (multiplexed bus mode)
SETCYCLES.WE_TIME = 1B (WRSTBZ is asserted at the same time as the CSZ0 to CSZ3 signal)

T_TR[2:0] = 010B (2 cycles)
T_WP[2:0] = 010B (2 cycles)
T_WC[3:0] = 0110B (6 cycles)

SETOPMODE.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)

10.4.2 Synchronous Access Timing

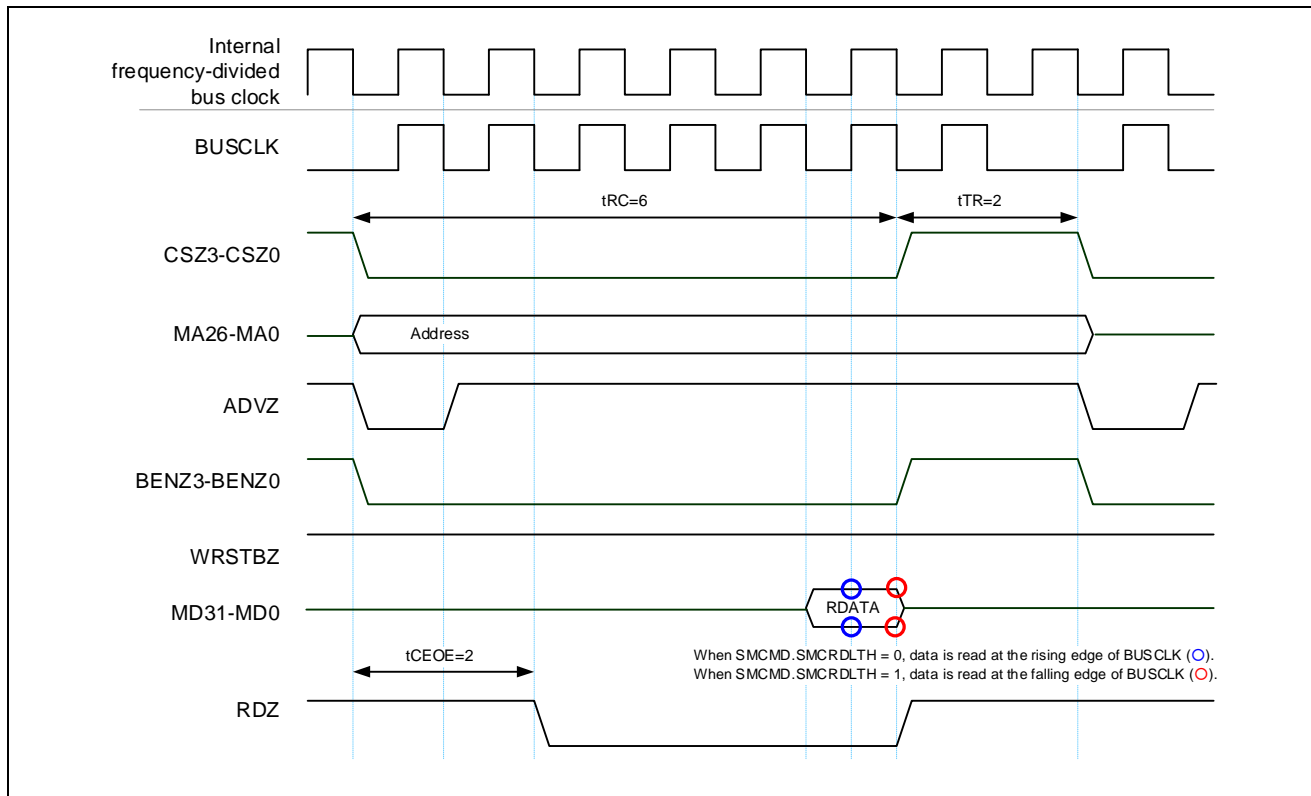


Figure 10.15 Synchronous SRAM, Separate Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate bus mode)
 SETCYCLES.T_TR[2:0] = 010B (2 cycles)
 T_CEOE[2:0] = 010B (2 cycles)
 T_RC[3:0] = 0110B (6 cycles)
 SETOPMODE.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)

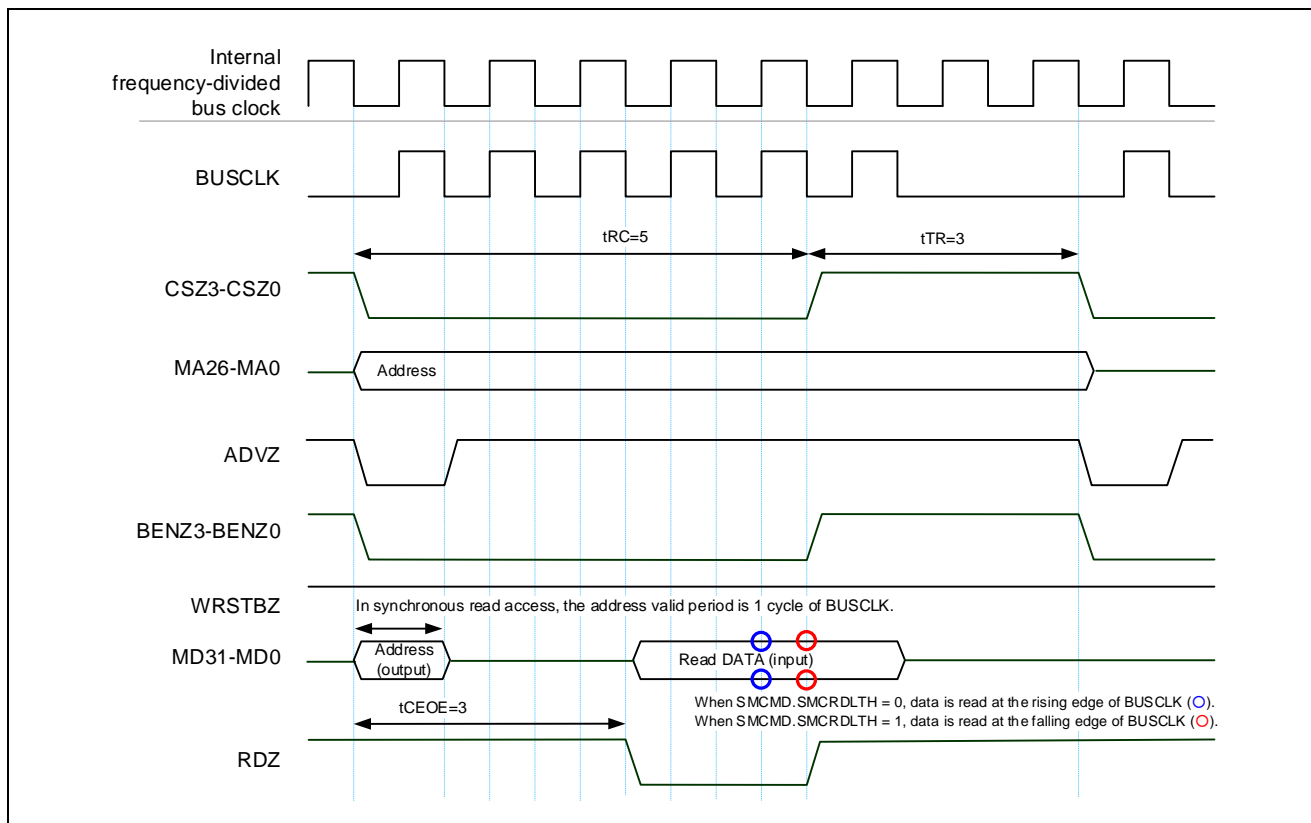


Figure 10.16 Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed bus mode)
 SETCYCLES.T_TR[2:0] = 011B (3 cycles)
 T_CEOE[2:0] = 011B (3 cycles)
 T_RC[3:0] = 0101B (5 cycles)
 SETOPMODE.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)

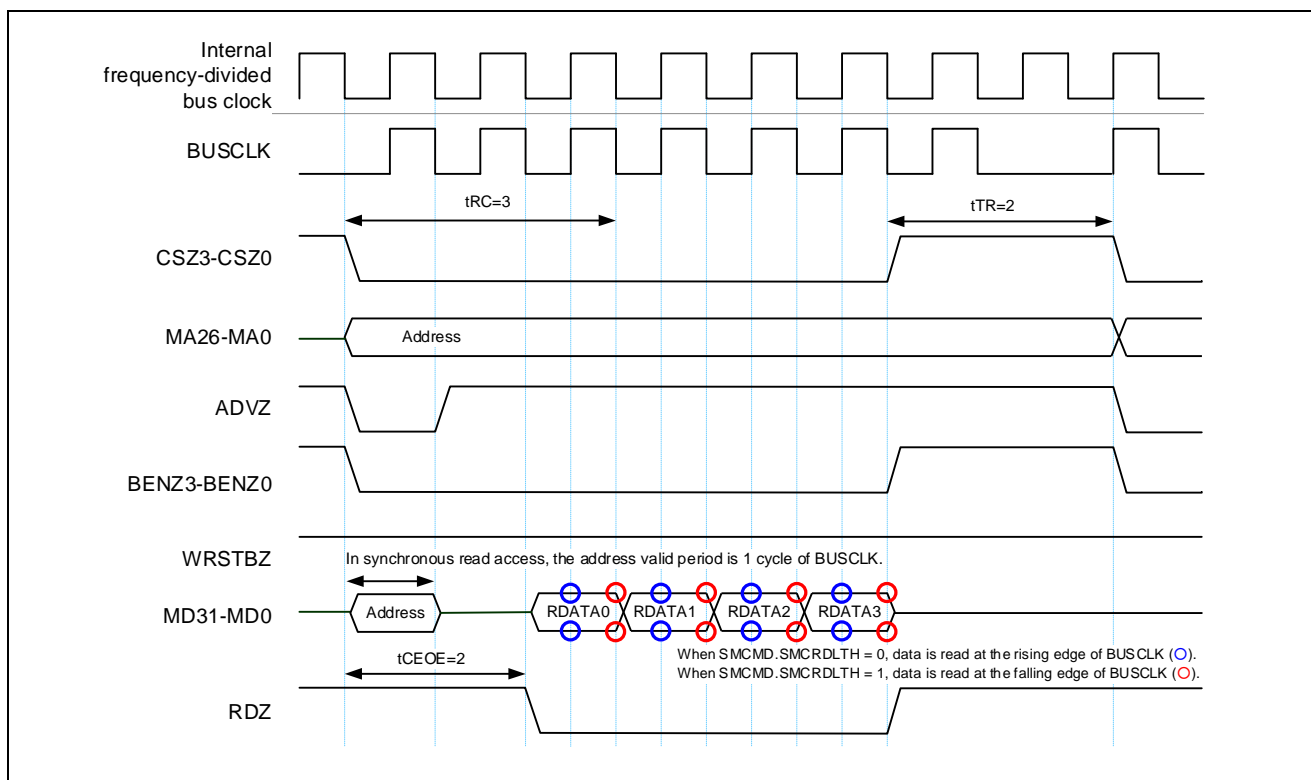


Figure 10.17 Synchronous SRAM, Multiplexed Bus Mode, Burst Read Access (4-Beat), ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed bus mode)

SETCYCLES.T_TR[2:0] = 010B (2 cycles)

T_CEOE[2:0] = 010B (2 cycles)

T_RC[3:0] = 0011B (3 cycles)

SETOPMODE.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)

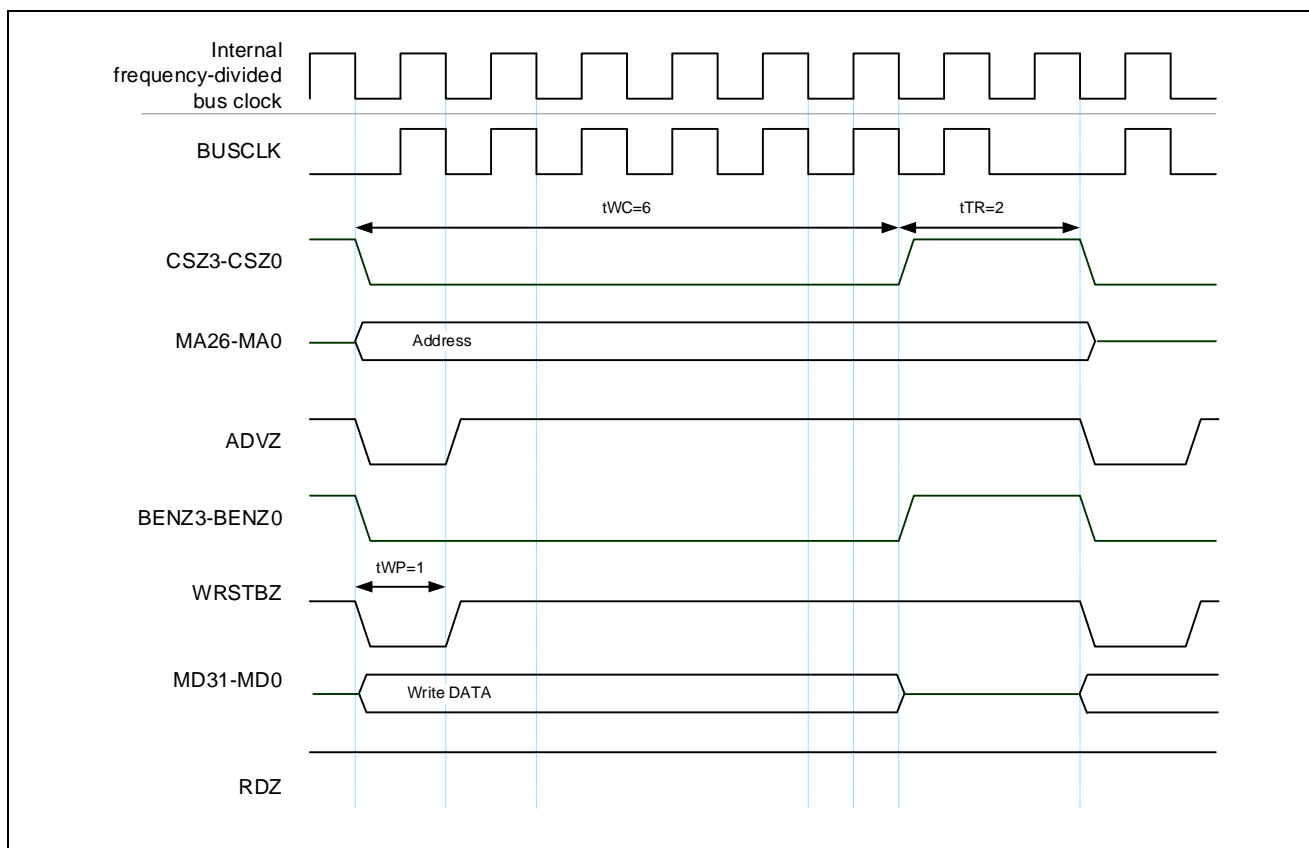


Figure 10.18 Synchronous SRAM, Separate Bus Mode, Write Access, ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate bus mode)
SETCYCLES.T_TR[2:0] = 010B (2 cycles)
T_WP[2:0] = 001B (1 cycle)
T_WC[3:0] = 0110B (6 cycles)
SETOPMODE.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)

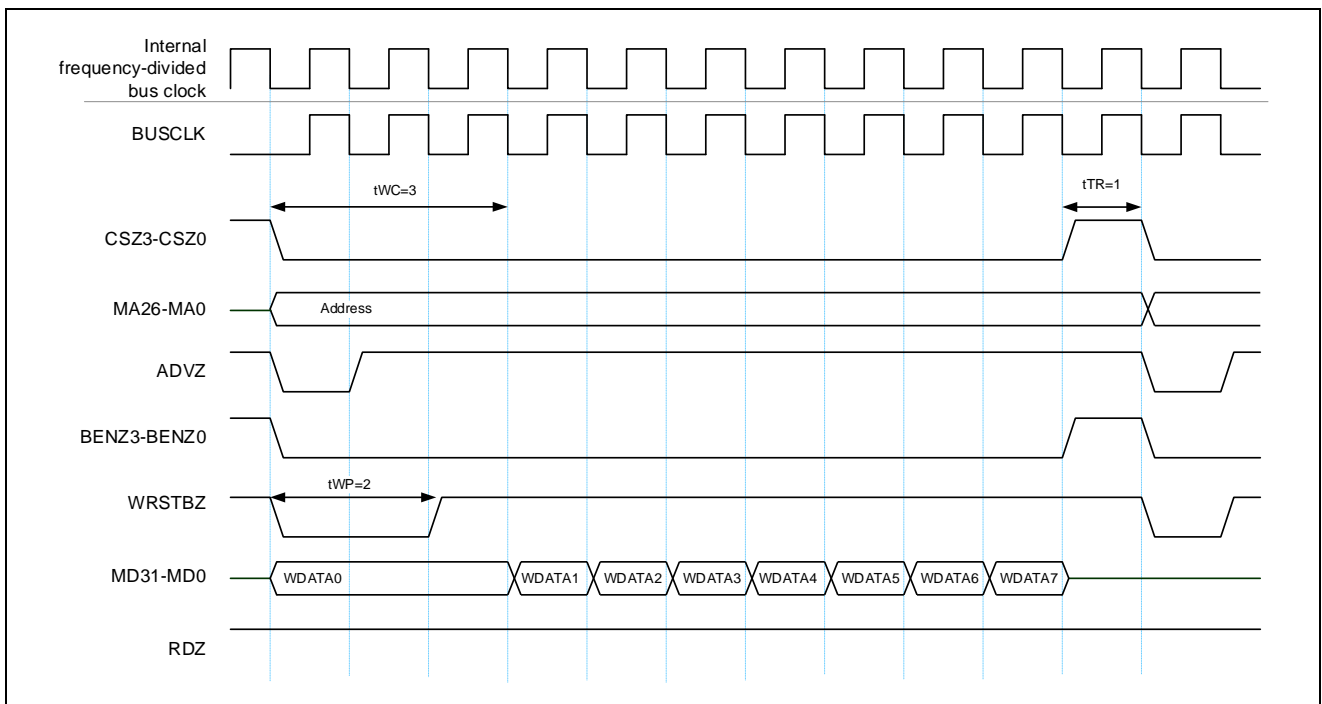


Figure 10.19 Synchronous SRAM, Separate Bus Mode, Burst Write Access (8-Beat), ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate bus mode)

SETCYCLES.T_TR[2:0] = 001B (1 cycle)

T_WP[2:0] = 010B (2 cycles)

T_WC[3:0] = 0011B (3 cycles)

SETOPMODE.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)

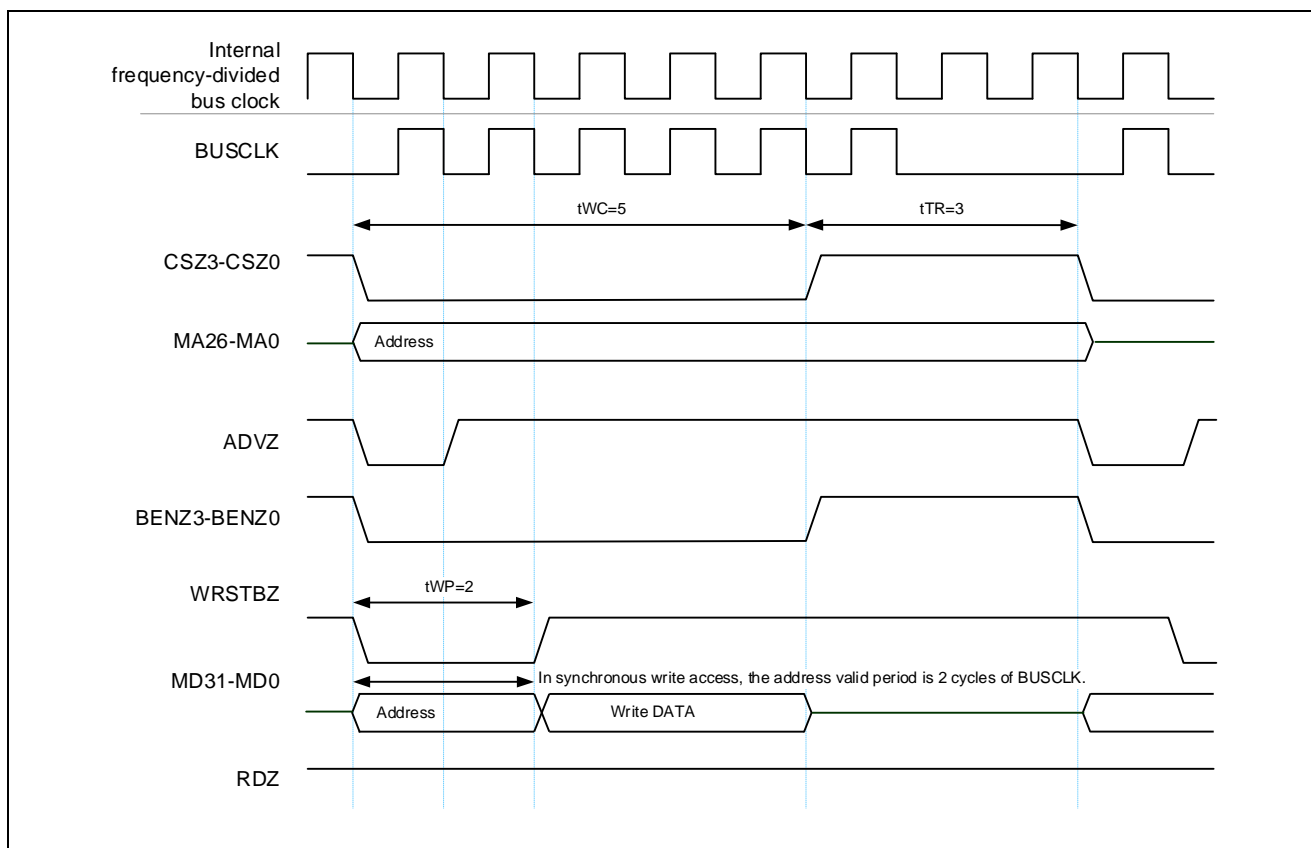


Figure 10.20 Synchronous SRAM, Multiplexed Bus Mode, Write Access, ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed bus mode)
SETCYCLES.T_TR[2:0] = 011B (3 cycles)
T_WP[2:0] = 010B (2 cycles)
T_WC[3:0] = 0101B (5 cycles)
SETOPMODE.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)

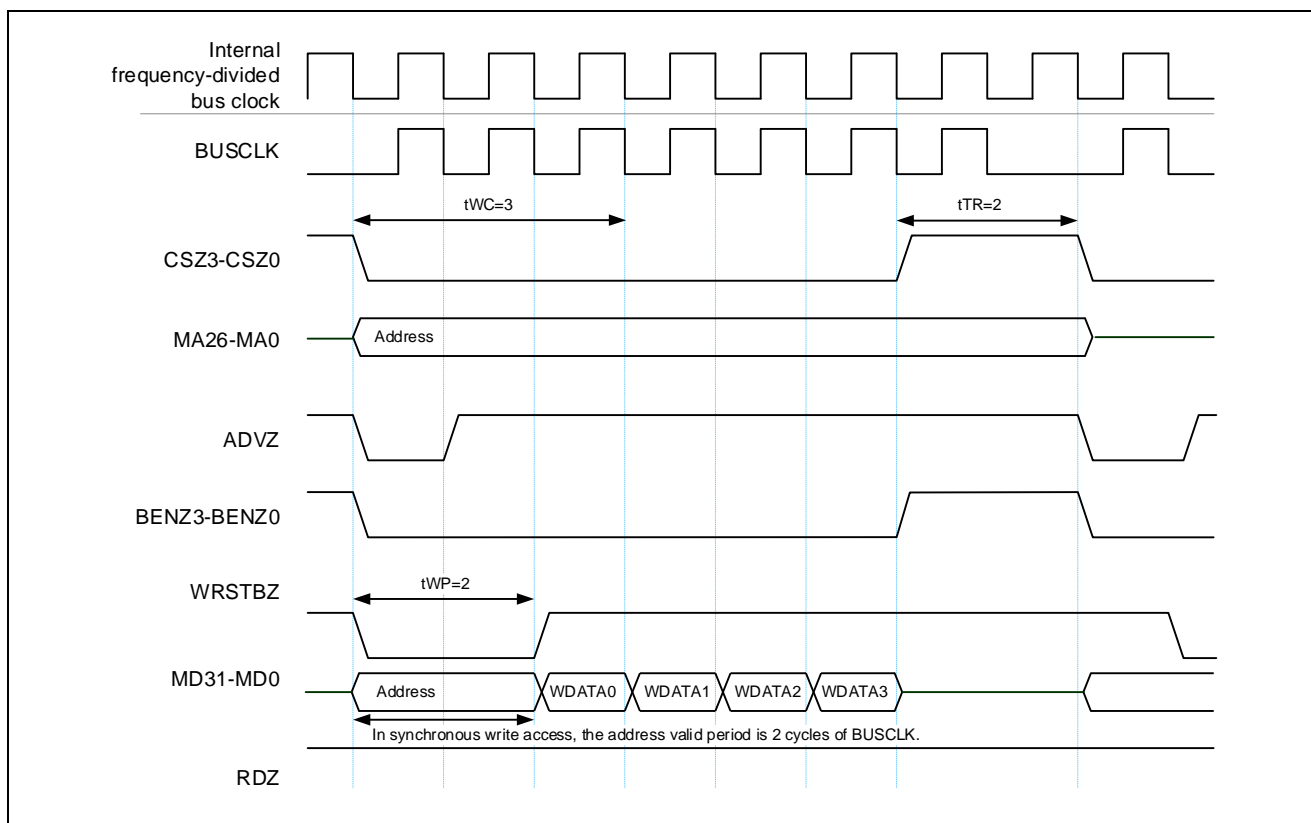


Figure 10.21 Synchronous SRAM, Multiplexed Bus Mode, Burst Write Access (4-Beat), ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed bus mode)
SETCYCLES.T_TR[2:0] = 010B (2 cycles)
T_WP[2:0] = 010B (2 cycles)
T_WC[3:0] = 0011B (3 cycles)
SETOPMODE.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)

10.4.3 Wait Timing

Wait signals (WAITZ, WAITZ1 to WAITZ3) 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.

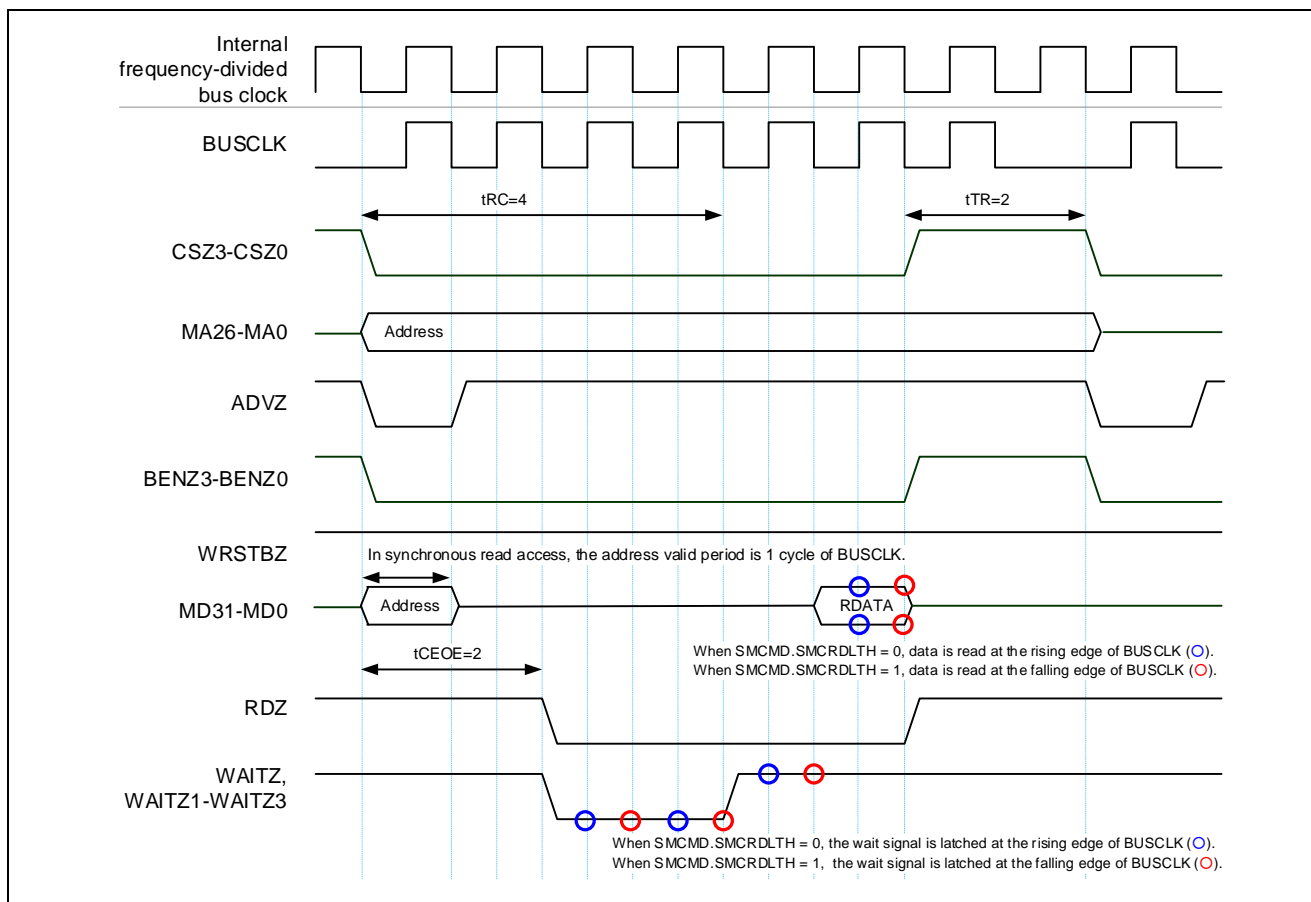


Figure 10.22 Synchronous SRAM, Multiplexed Bus Mode, Read Access, ADVZ Enabled

Remark: ADMUXMODE pin = High level (multiplexed bus mode)
SETCYCLES.T_TR[2:0] = 010B (2 cycles)
T_CEOE[2:0] = 010B (2 cycles)
T_RC[3:0] = 0100B (4 cycles)
SETOPMODE.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)

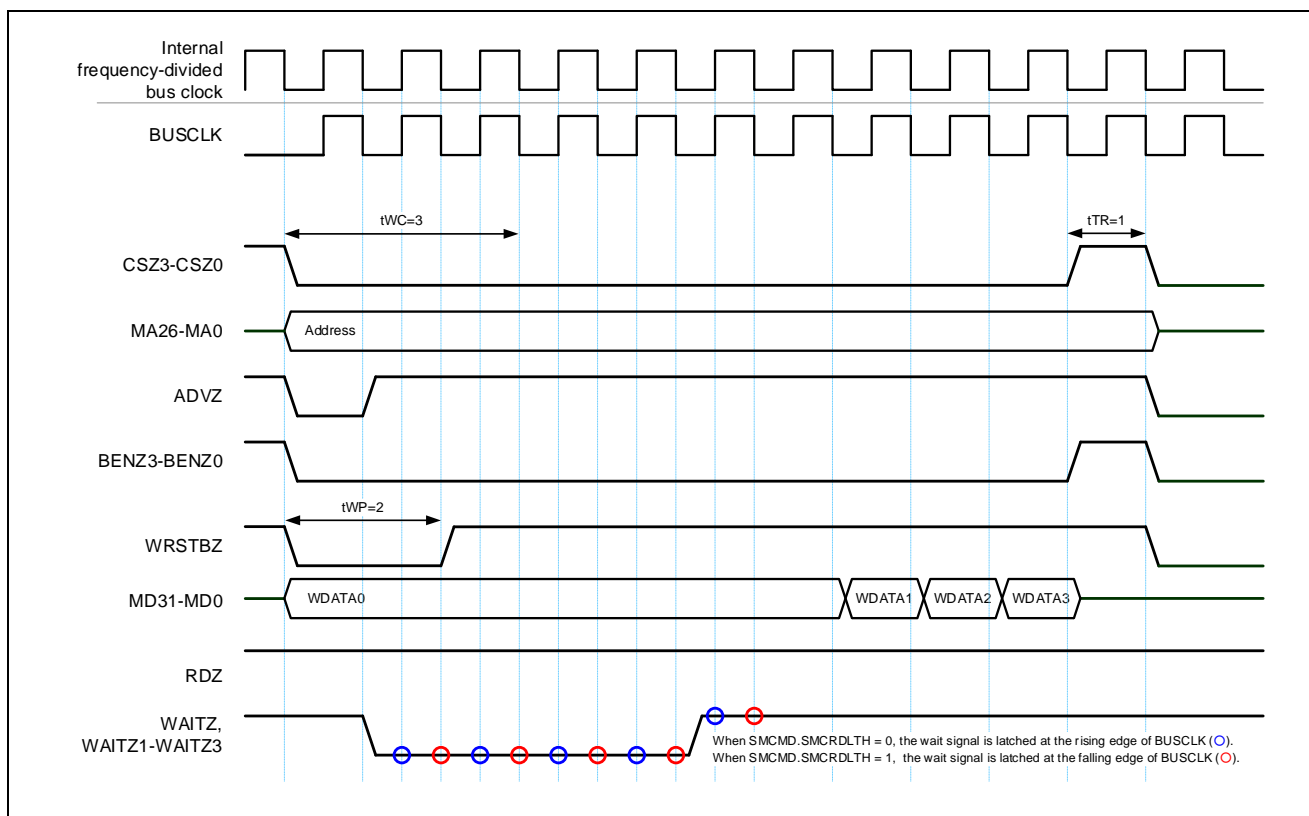


Figure 10.23 Synchronous SRAM, Separate Bus Mode, Burst Write Access (4-Beat), ADVZ Enabled

Remark: ADMUXMODE pin = Low level (separate bus mode)
 SETCYCLES.T_TR[2:0] = 001B (1 cycle)
 T_WP[2:0] = 010B (2 cycles)
 T_WC[3:0] = 0011B (3 cycles)
 SETOPMODE.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)

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.

11. External MCU Interface

The external MCU interface is for the connection of an external MPU. Specifically, the external MCU interface is provided to allow use of the internal resources of an R-IN32M3 by an external host MPU. The signals of the interface are multiplexed with the same pins as are used for the external memory interface, and usage as an external MCU interface is selected by placing the high level on the MEMIFSEL pin. Please set the MEMIFSEL pin level by the time of de-assertion of whichever is the later of the PONRZ and RESETZ reset signals after power is initially supplied. Dynamic switching is not supported.

When the external MCU interface is in use, booting of the R-IN32M3 from the external MCU or serial flash memory is possible. However, external memory access (to external ROM and SRAM) is not available.

The external MCU interface supports the asynchronous SRAM interface and the synchronous SRAM interface. When the HIFSYNC pin is at the high level, it becomes a synchronous SRAM MCU interface, and when HIFSYNC is at the low level, it becomes an asynchronous SRAM MCU interface.

Moreover, the external MCU interface supports clock-synchronous SRAM type transfer so that large volumes of data can be accessed at high speed. Placing the high level on both the MEMIFSEL pin and MEMCSEL pins selects this mode.

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.

Table 11.1 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 an external MCU (operation is as an external memory interface)
High	Low	Low	Low	Asynchronous SRAM interface mode is entered. Connection of the bus clock signal to HBUSCLK is not required.
			High	Setting prohibited
		High	Low	Synchronous SRAM interface mode is entered. Connection of the bus clock signal to HBUSCLK is required. ^{Note}
			High	Setting prohibited
	High	Low	Low	Setting prohibited
			High	Setting prohibited
		High	Low	Setting prohibited
			High	Synchronous SRAM type transfer mode is entered.

**Note: For access to the CC-Link IE field, synchronous SRAM interface mode must be set (MEMIFSEL = high, MEMCSEL = low, HIFSYNC = high).
The CC-Link IE field is only included in the R-IN32M3-CL.**

- Cautions**
1. To prevent a malfunction, input the high level to the MEMIFSEL pin. To stop an external MCU interface, fix the HRDZ and HWRSTBZ pins to the high level.
 2. The BUS32EN, HWRZSEL, MEMCSEL, MEMIFSEL, and HIFSYNC pins do not support the dynamic switching. Determine the input values during a reset.
 3. For access to the CC-Link IE Field by the external MCU interface, place the following levels on the given pins.

	HWRZSEL pin	
	Low-level	High-level
CCI_WRLLENH pin	Open* or high-level	Low-level
HWRSTBZ pin	Write strobe signal of external MCU	Open* or high-level

*: High-level with the internal pull-up resistor

11.1 Memory Map

A 2-Mbyte space is provided as the external MCU interface.

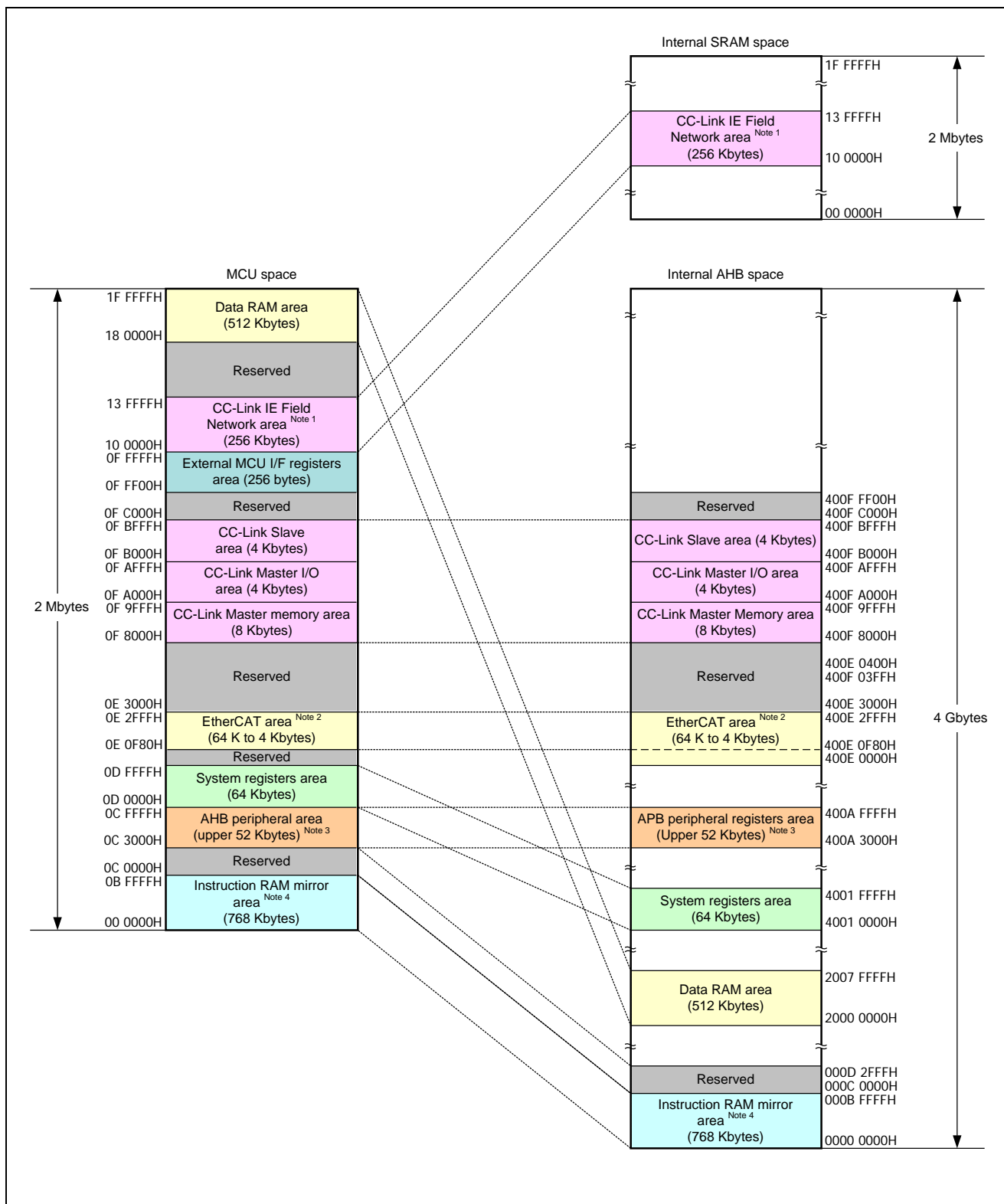


Figure 11.1 External MCU Interface Space

- Notes 1.** This is only provided in the R-IN32M3-CL.
- 2.** This is only provided in the R-IN32M3-EC.
- 3.** 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, see the memory map of the R-IN32M3 Series User's Manual.
- 4.** The addresses of the instruction RAM mirror area (768 Kbytes) where access actually occurs will change according to the selected boot mode, as shown in the table below. For details, see section 5.3, Memory Map in Each Boot Mode, and section 4, Bus Architecture.

BOOT1	BOOT0	Boot Mode	Access Destination Area	Remarks
0	0	External memory boot	—	External MCU interface is disabled
0	1	External serial flash ROM boot	Reserved	Access disabled
1	0	External MCU boot	Instruction RAM area	—
1	1	Instruction RAM boot	Instruction RAM area	Enabled only for debugging

11.2 Synchronous/Asynchronous SRAM Interface Mode

The external MCU interface supports the asynchronous and synchronous SRAM interfaces. When the HIFSYNC pin is at the high level, it becomes a synchronous SRAM MCU interface, and when HIFSYNC is at the low level, it becomes an asynchronous SRAM MCU interface.

It is used when the MEMIFSEL pin is at the high level and the MEMCSEL pin is at the low level. Please set the MEMIFSEL and MEMCSEL pin level by the time of de-assertion of the reset signal after power is initially supplied. Dynamic switching is not supported.

11.2.1 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 (up to 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 (synchronous mode is selected) or one stage (asynchronous mode is selected)
- Read buffer: Advance reading of up to 32 bytes is possible.
- Multiplexing of addresses and data
 - No multiplexing of addresses and data
- Checking of various states
 - Internal reset state (can only be checked while asynchronous mode is set)
 - States of the HIFSYNC and BUS32EN pins

11.2.2 Operation

(1) Method of transfer

Table 11.2 Method of Transfer

Mode Settings			Allowable Transfer Methods	
MEMIFSEL	HIFSYNC	BUS32EN	Page access permitted area	Page access prohibited area
Low	—	—	Not accessible	Not accessible
High	Low (Asynchronous)	Low (16 bits)	SRAM reading	SRAM reading
		High (32 bits)	SRAM writing	SRAM writing
	High (Synchronous)	Low (16 bits)	Page ROM reading	
		High (32 bits)		

Remark: Page ROM reading can only be used for the area permitted by the HIFBCC or HIFPRC register.

(2) Bus sizing

The bus size for internal access is in accord with the external bus width.

Table 11.3 Bus Sizing

BUS32EN (External MCU Interface Bus width)	Access area in R-IN32M3		Operation for Transfer to Internal Bus from External MCU Interface
	Area	Internal Bus Width	
0 (16 bits)	External interface registers area	32 bits	This area is always read in 32 bits. Writing is only to the specified byte lanes.
	CC-Link IE Field Network area ^{Note}	16 bits	Same access as an external MCU
	Other area	32 bits	A buffer enabled area is read in 32 bits and a buffer disabled area is read in 16 bits (MCU bus width). Writing is only to the specified byte lanes.
1 (32 bits)	External interface registers area	32 bits	This area is always read in 32 bits. Writing is only to the specified byte lanes.
	CC-Link IE Field Network area ^{Note}	16 bits	The 16 higher-order bits or the 16 lower-order bits are accessible. Simultaneous access to both the 16 higher-order bits and 16 lower-order bits is prohibited.
	Other areas	32 bits	This area is always read in 32 bits. Writing is only to the specified byte lanes.

Caution: Each access to the CC-Link IE Field Network by the external MCU is converted by the bus sizing facility of the external MCU interface. Since the bus cycle is not separated into multiple cycles, transfer across two-byte boundaries to the CC-Link IE Field Network area is prohibited if the bus width of the external MCU interface is 32 bits.

Note: This is only provided in the R-IN32M3-CL.

(3) Synchronous mode and asynchronous mode

Synchronous mode or asynchronous mode is selectable by the setting of the HIFSYNC pin for areas other than the CC-Link IE Field Network area. When the CC-Link IE Field Network is to be used, select synchronous mode.

The synchronous relationship of the interface signals is shown below.

Table 11.4 Synchronous Relationship of External MCU interface Signals

Signal Name	I/O	HIFSYNC (Selection of Synchronous Relation)		
		H (Synchronous Mode)		L (Asynchronous Mode)
		WRITE	READ	
HCSZ	Input	HBUSCLK synchronous	Asynchronous	Asynchronous
HPGCSZ	Input	HBUSCLK synchronous	Asynchronous	Asynchronous
HA20-HA1	Input	HBUSCLK synchronous	Asynchronous	Asynchronous
HRDZ	Input	—	Asynchronous	Asynchronous
HWRSTBZ	Input	HBUSCLK synchronous	—	Asynchronous
HWRZ3-HWRZ0, HBENZ3-HBENZ0	Input	HBUSCLK synchronous	—	Asynchronous
HD31-HD0 (input)	Input	HBUSCLK synchronous	—	Asynchronous
HD31-HD0 (output)	Output	—	Asynchronous	Asynchronous
HWAITZ	Output	HBUSCLK synchronous		Asynchronous
HERROUTZ	Output	Asynchronous		

(4) Buffer reading

Buffer reading can be used for the data RAM area, etc.

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 are read from the buffer. This improves throughput, since data can be read from the advance-read buffer at high speed.

Remark: Data at addresses for advance reading are always read in ascending order of addresses.

(5) Page ROM reading

In addition to buffer reading, page ROM reading can be used for the data RAM area, etc.

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 paged ROM, the wait signal is de-asserted (the HWAITZ signal output is at the high level) once all data in a page have been prepared for off-page reading. This improves throughout, since on-page reading following off-page reading does not require a wait.

- 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 may 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, xx40H.

Remark: The page size is not affected by the bus width.

A large page size will increase the on-page ratio and transfer rate. We recommend selecting 16 bytes as the page size unless this creates a problem for the system.

11.2.3 Basic Operation Timing of the External MCU Interface

(1) Access to CC-Link IE Field Network Area

Remark: Only the R-IN32M3-CL supports CC-Link IE Field Network.

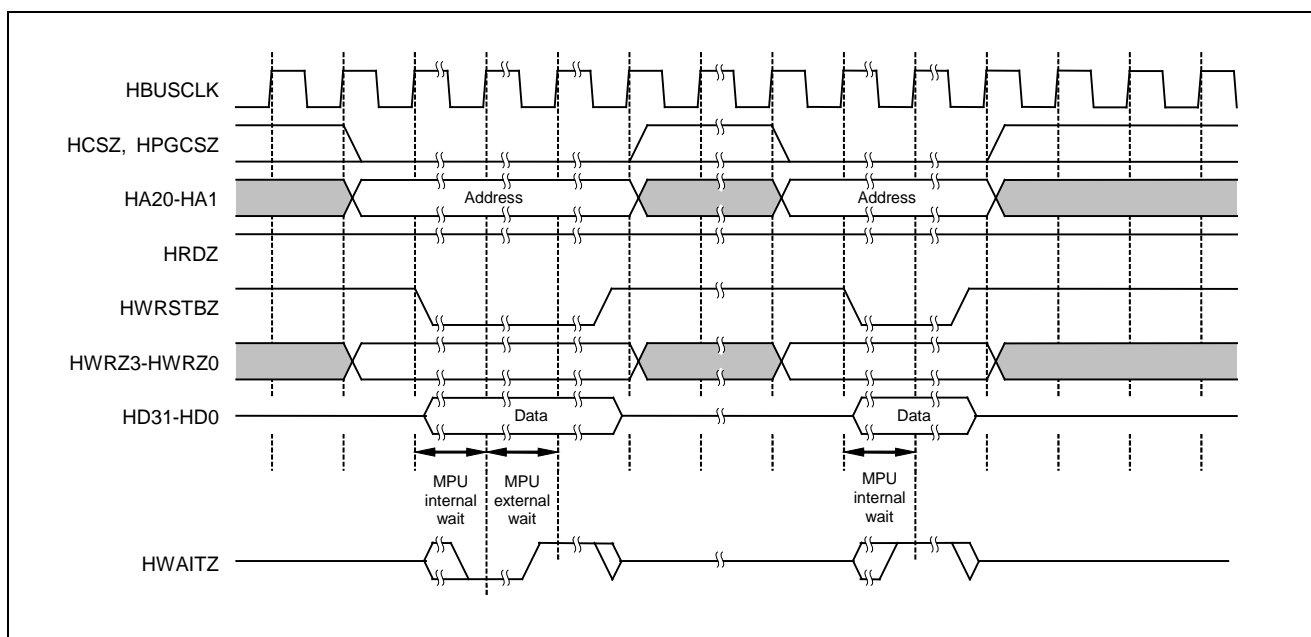


Figure 11.2 Writing to the CC-Link IE Field Network Area (SRAM writing)

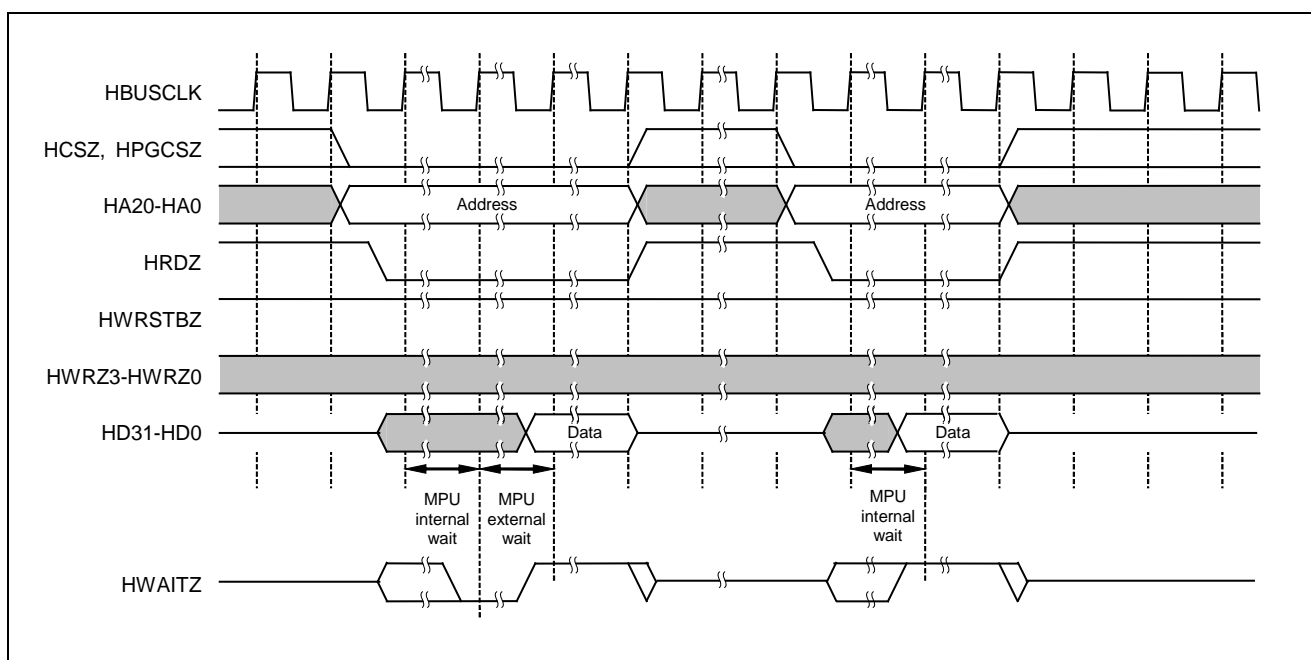


Figure 11.3 Reading from CC-Link IE Field Network Area (SRAM reading)

(2) Access to the External MCU Interface Register Area

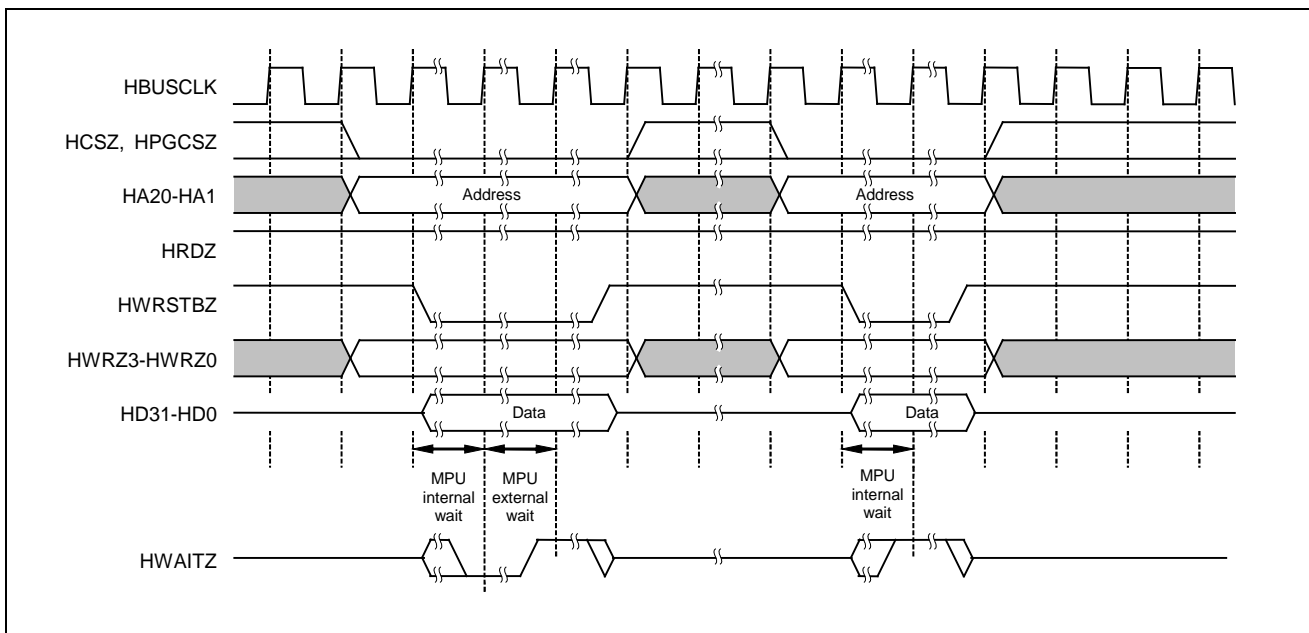


Figure 11.4 Writing to the External MCU Interface Register Area (SRAM writing)

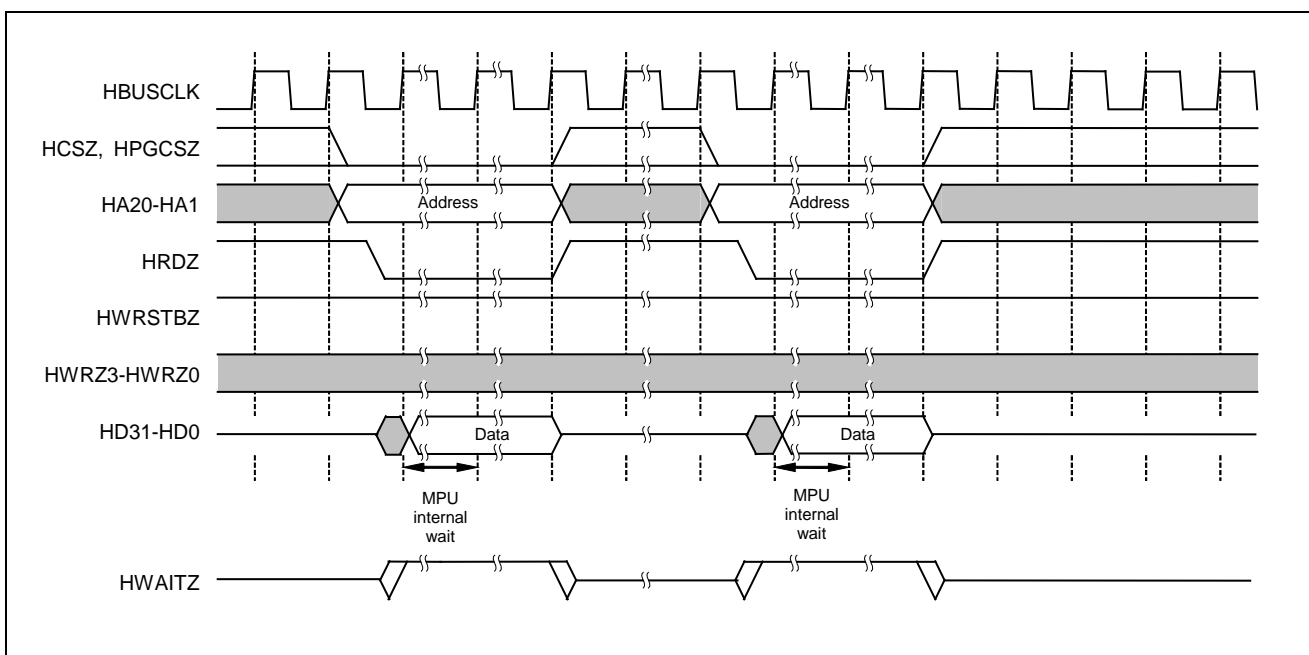


Figure 11.5 Reading from External MCU Interface Register Area (SRAM reading)

(3) Access to Other Areas

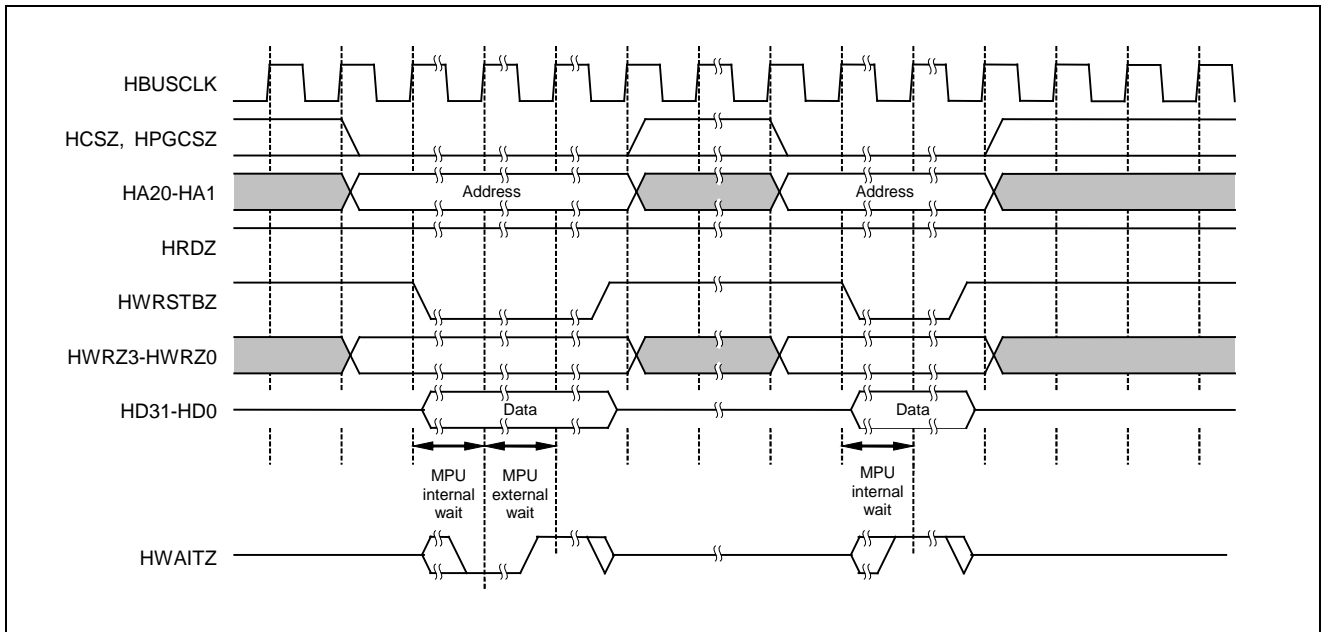


Figure 11.6 Writing to Other Areas (SRAM writing)

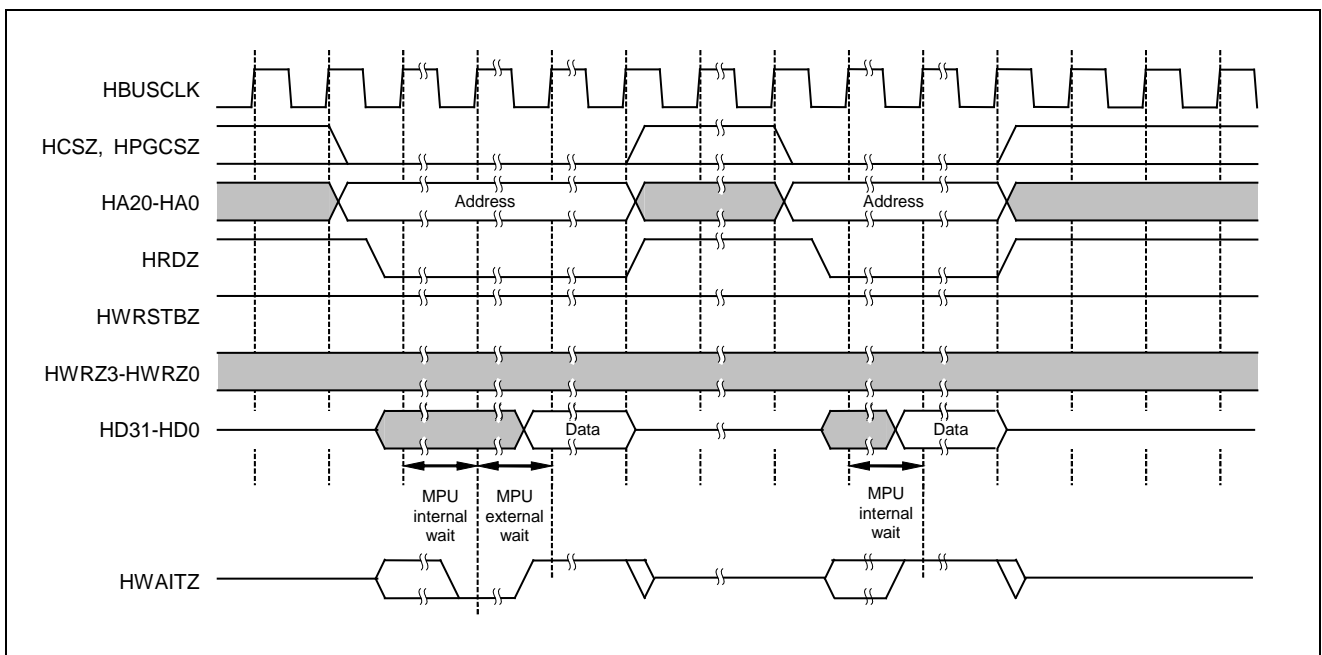


Figure 11.7 Reading from Other Areas (SRAM reading)

11.2.4 Timing Adjustment of the External MCU Interface

(1) Outline of timing adjustment

Adjustment of timing is provided to adjust for variations in the relative timing between the external MCU interfaces.

Table 11.5 Timing Adjustment of the External MCU Interface

Method of Transfer	Timing Adjustment				Target Areas for Adjustment		
	Adjustment Timing		Setting Register		External MCU Interface Registers Area	CC-Link IE Field Network	Other Areas
	Target Signal	Relativity Signal	Register Name	Bit Name			
SRAM writing	HCSZ	HWRSTBZ	HIFBTC	WRSTD1-0	✓	—	✓
	HPGCSZ						
	HA20-HA1						
	HWRZ						
	HD31-HD0						
SRAM reading	HCSZ	HRDZ	HIFBTC	RDSTD1-0	—	—	✓
	HPGCSZ						
	HA20-HA1	HWAITZ	HIFBTC	RDDTS1-0	—	—	✓
	HD31-HD0						
Page ROM reading	HCSZ	HA20-HA1	HIFBTC	RDSTD1-0	—	—	✓
	HPGCSZ						
	HA20-HA1	HIFBTC	PASTD2-0	—	—	—	✓
	HD31-HD0						

Cautions 1. Timing adjustment does not apply to the following accesses.

- Access to the CC-Link IE Field Network area
- Read from the external MCU interface registers area

2. Timing adjustment affects an access latency.

(2) Timing adjustment (SRAM writing)

Writing to internal resources starts in response to 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 HWRZSEL pin or the BUS32EN pin.

When externally writing to internal resources of an R-IN32M3, stable addresses and data are required.

An R-IN32M3 has a function to adjust timing for sampling of addresses and data.

Sampling timing adjustment can be handled by the HIFBTC register.

Table 11.6 Write Strobe Signal

Condition		Write Strobe Signal (Active Low)	Remark
HWRZSEL	BUS32EN		
Low	—	HWRSTBZ	
High	Low	HWRZ1 & HWRZ0	
	High	HWRZ3 & HWRZ2 & HWRZ1 & HWRZ0	

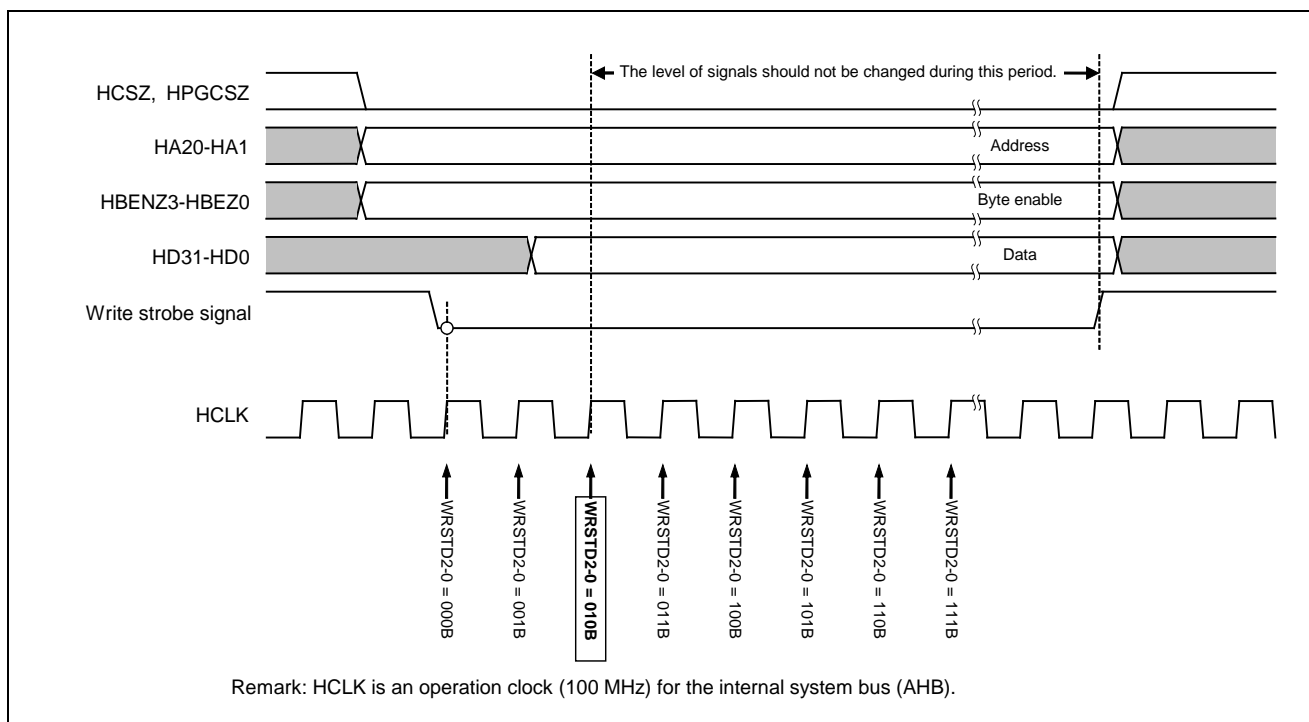


Figure 11.8 Timing Adjustment (SRAM writing)

(3) 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.

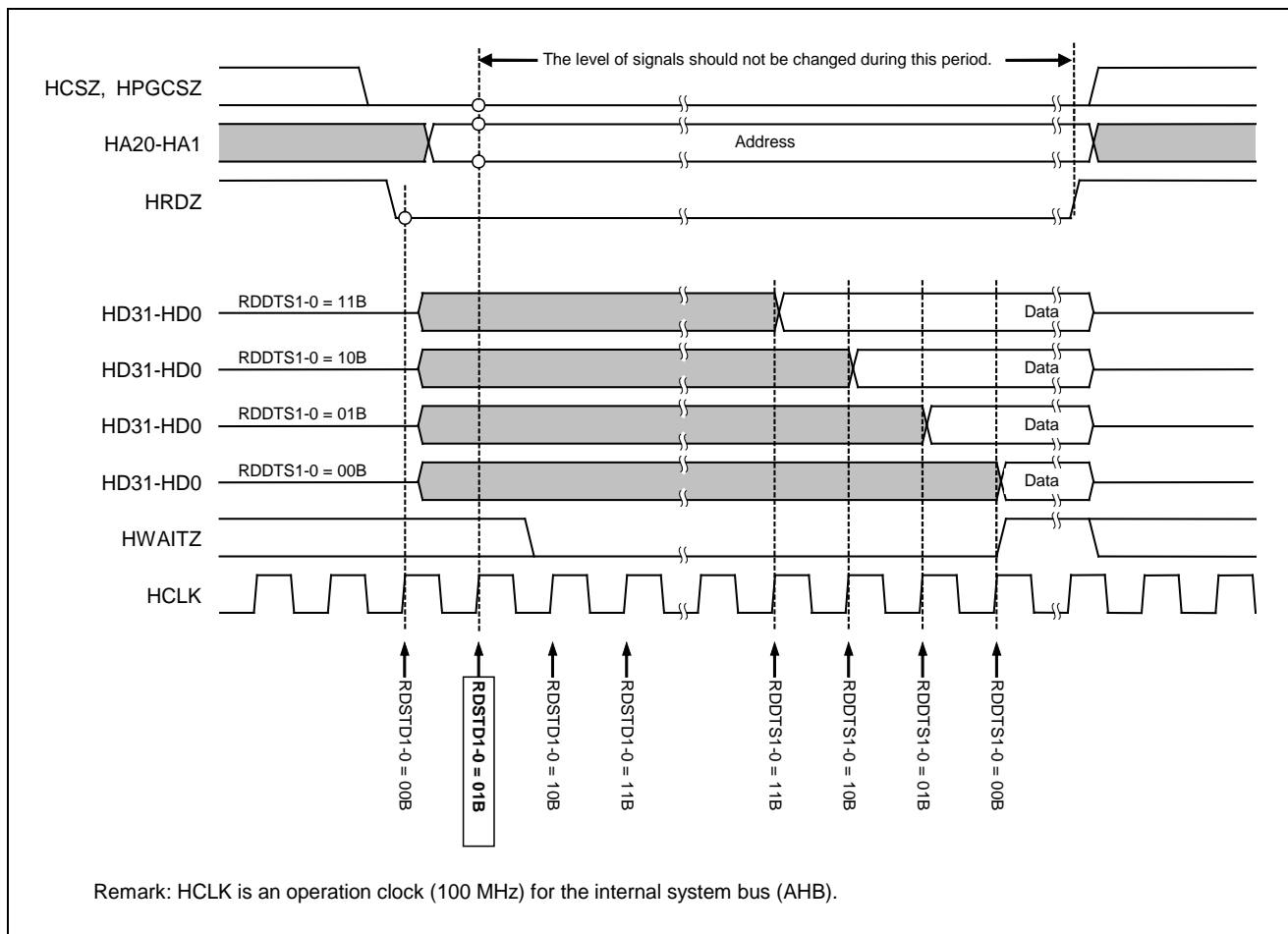


Figure 11.9 Timing Adjustment (SRAM reading, page ROM reading)

Caution: In reading of 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) 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 reading, 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.

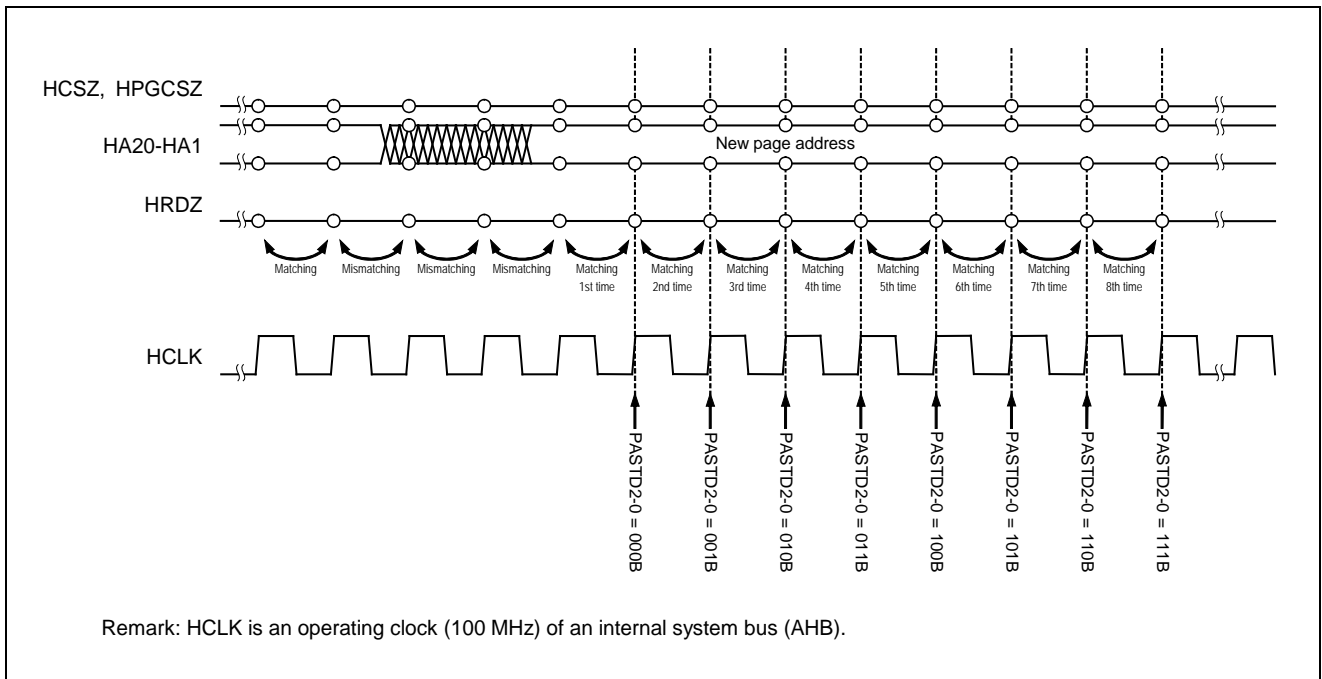


Figure 11.10 Timing Adjustment (page ROM reading)

11.2.5 Control Registers

(1) List of registers

The external MCU interface control registers are accessible by the MCU externally connected to an R-IN32M3.

Address	Register Name	Symbol	R/W	Operation Unit [bit]			Time of Reset
				8	16	32	
0F FF00H	External MCU I/F bus control register	HIFBCC	R/W	✓	✓	—	0001H
0F FF04H	External MCU I/F timing control register	HIFBTC	R/W	✓	✓	—	3733H
0F FF08H	External MCU I/F page ROM control register	HIFPRC	R/W	✓	✓	—	0000H
0F FF0CH	External MCU I/F interrupt request control register	HIFIRC	R/W	✓	✓	—	0000H
0F FF10H	External MCU I/F error source register 0	HIFECR0	R	✓	✓	✓	0000 0000H
0F FF14H	External MCU I/F error source register 1	HIFECR1	R	✓	✓	—	0000H
0F FF20H	External MCU I/F monitor register	HIFMON	R	✓	✓	—	0000H / 0004H 0008H / 000CH

Remark: The initial value of the HIFMON register is determined according to the state of the input pins (HIFSYNC, BUS32EN).

(2) HOSTIF 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	WRP ON	BST ON	0	0	0	0	0	0	0	RBU FON 4	RBU FON 3	0	RBU FON 1	RBU FON 0	0F FF00H	0001H
R/W	0	0	R/W	R/W	0	0	0	0	0	0	0	R/W	R/W	0	R/W	R/W		

Bit Position	Bit Name	Description
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.
4	RBUFON4	Enables or disables advance reading of a part of EtherCAT. 0: Advance reading is disabled. 1: Advance reading is enabled.
3	RBUFON3	Enables or disables advance reading of a part of the CC-Link area. 0: Advance reading is disabled. 1: Advance reading is enabled.
2	RBUFON2	Unused
1	RBUFON1	Enables or disables advance reading of the instruction RAM 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. Only accessible by the external MCU.

2. Clearing of read buffers requires write access to any of the internal 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 11.7 Address Range for which Advance Reading and Page ROM Reading are Selectable

Target Macro	Address Range		Related Enable Bits	
	MPU Space	Internal AHB Space	Advance Reading	Page ROM
Ether CAT	0E FFFFH to 0E 1000H	400E FFFFH to 400E 1000H	HIFBCC. RBUFON4	HIFPRC. PAGEON4
CC-Link	0F BFFFH to 0F A000H	400F BFFFH to 400F A000H	HIFBCC. RBUFON3	HIFPRC. PAGEON3
Instruction RAM mirror area	0B FFFFH to 00 0000H	000B FFFFH to 0000 0000H	HIFBCC. RBUFON1	HIFPRC. PAGEON1
Data RAM	1F FFFFH to 18 0000H	2007 FFFFH to 2000 0000H	HIFBCC. RBUFON0	HIFPRC. PAGEON0

- Cautions**
1. Some areas cannot be read in advance depending on the target macro even if advance reading is enabled.
 2. If the last 16-byte area of the instruction RAM mirror area is read while advance reading is enabled, this will lead to assertion of the HERROUTZ pin.

(3) HOSTIF 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	RDD TS1	RDD TS0	0	PAS TD2	PAS TD1	PAS TD0	0	0	RDS TD1	RDS TD0	0	WRS TD2	WRS TD1	WRS TD0	0F FF04H	3737H
R/W	0	0	0	0	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
13, 12	RDDTS1, RDDTS0	These bits set the waiting time from fixing of the HD 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
10 to 8	PASTD2, PASTD1, PASTD0	The stable waiting time of off-page detection is set up. 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
5, 4	RDSTD1, RDSTD0	These bits set up the timing for detecting the start of read operation by the HRDZ signal. The setup time of address input signals for falling edges of the HRDZ signal is adjusted. 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
2 to 0	WRSTD2, WRSTD1, WRSTD0	These bits set up the timing for detecting the start of write operation by the HWRSTBZ signal. The setup time of address input signals and write data input signals for falling edges of the HWRSTBZ signal is adjusted. 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: Only accessible by the external MCU.

(4) HOSTIF page ROM control register (HIFPRC)

This register sets up 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	0	0	0	0	PAGEON4	PAGEON3	0	PAGEON1	PAGEON0	0F FF08H	0000H
R/W	0	0	0	R/W	0	0	0	0	0	0	0	R/W	R/W	0	R/W	R/W		

Bit Position	Bit Name	Description
12	PAGESZ	The page size for page ROM reading is set up. 0: 8 bytes 1: 16 bytes However, access across 16-byte boundaries is prohibited.
4	PAGEON4	Page ROM reading of a part of EtherCAT is set up. 0: SRAM reading 1: Page ROM reading
3	PAGEON3	Page ROM reading of a part of the CC-Link area is set up. 0: SRAM reading 1: Page ROM reading
2	PAGEON2	Unused
1	PAGEON1	Page ROM reading of the instruction RAM mirror area is set up. 0: SRAM reading 1: Page ROM reading
0	PAGEON0	Page ROM reading of the data RAM area is set up. 0: SRAM reading 1: Page ROM reading

Caution: The page size to be set in the PAGESZ bit must match the page size setting of the external MCU.

Remark: Only accessible by the external MCU.

(5) HOSTIF 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	ERRRSP	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
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. Only accessible by the external MCU.**
- 2. While the setting of the ERRRSP bit is 1, the HOSTIF error source registers (HIFECR0, HIFECR1) are not updated even if a new error response was generated. The first error information is held in the HOSTIF error source register.**

(6) HOSTIF 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 HOSTIF interrupt request control register (HIFIRC) is 1, that address formation is not stored.

	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	
HIFECR0	ERRAD31	ERRAD30	ERRAD29	ERRAD28	ERRAD27	ERRAD26	ERRAD25	ERRAD24	ERRAD23	ERRAD22	ERRAD21	ERRAD20	ERRAD19	ERRAD18	ERRAD17	ERRAD16	ERRAD15	ERRAD14	ERRAD13	ERRAD12	ERRAD11	ERRAD10	ERRAD9	ERRAD8	ERRAD7	ERRAD6	ERRAD5	ERRAD4	ERRAD3	ERRAD2	ERRAD1	ERRAD0	0F FF10H Initial value 0000 0000H	
_W																																		
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	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: 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 to 0	ERRAD15-ERRAD0	These bits hold the address (lower-order 16-bit address) at which an error occurred.

	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	0	0	0	0	0	0	0	0	R	R	R	R	R		

Bit Position	Bit Name	Description
15 to 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: Only accessible by the external MCU.

(7) HOSTIF 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 HOSTIF 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	ERR WR	ERR SZ2	ERR SZ1	ERR SZ0	0F FF14H	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 to 4	0	Unused (0 Fixed)
3	ERRWR	The information on whether an error was caused by reading or writing is stored. 0: Reading 1: Writing
2 to 0	ERRSZ2, ERRSZ1, ERRSZ0	The transfer size (access width) when the error occurred is stored. 000: 8 bits 001: 16 bits 010: 32 bits 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: Only accessible by the external MCU.

(8) HOSTIF monitor register (HIFMON)

This register monitors input pins for HOSTIF and their internal states.

This register can also be read during the reset period.

HIFMON	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
	0	0	0	0	0	0	0	0	0	0	0	0	HIFS YNC	BUS 32EN	0	HIF RDY	0F FF20H	0000H 0004H 0008H 000CH
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		

Bit Position	Bit Name	Description
3	HIFS YNC	Indicates the state of the HIFS YNC pin. 0: Low level (asynchronous mode) 1: High level (synchronous mode)
2	BUS 32EN	Indicates the state of the BUS 32EN pin. 0: Low level (16-bit bus width) 1: High level (32-bit bus width)
0	HIF RDY	Indicates the internal initialization state of HOSTIF. 0: Under internal initialization 1: Completion of internal initialization

Caution: 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 an R-IN32M3.

Remark: Only accessible by the external MCU.

11.2.6 Precautions

Precautions on usage of the external MCU interface in an R-IN32M3 are described below.

Table 11.8 Register Settings for Each Area and Method of Access

Area	Register Settings		Method of Access		
	HIFPRC. PAGEONn	HIFBCC. RBUFONn	Page ROM	SRAM	
			Read	Read	Write
Area not to be buffered	—	—	Setting prohibited <small>Note</small>	OK	OK
Area to be buffered	0	0	Setting prohibited <small>Note</small>	OK	OK
	0	1	Setting prohibited <small>Note</small>	OK	OK
	1	0	Setting prohibited <small>Note</small>	OK	OK
	1	1	OK	OK	OK

Note: Attempted access via the external MCU interface may lead to a deadlock.

11.3 Synchronous SRAM Type Transfer Mode

The external MCU interface supports synchronous SRAM type transfer for faster transfer of large volumes of data.

11.3.1 Functional Overview

- Interface system
 - Synchronous SRAM single transfer (for reading and writing)
- Synchronous relations
 - HBUSCLK Sync. (up to 50 MHz)
- Bus width
 - 16 bits / 32 bits (set up with the BUS32EN pin)
- Transmission data size
 - 32 bits / 16 bits / 8 bits
- Write buffer: 8 stages
- Read buffer: Advance reading of up to 32 bytes is possible.
- Multiplexing of addresses and data
 - Multiplexing of addresses and data
 - Separation is not supported
- Checking of each state
 - State of the HIFSYNC and BUS32EN pins

11.3.2 Selection of Operational Mode

In synchronous SRAM type transfer 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 11.9 Operating Mode Settings

Mode Setting Pins				Operating Mode	
MEMCSEL	BUS32EN	HIFSYNC	ADMUXMODE		
H	L	L	—	Setting prohibited	
		H	L	Multiplexing of 16-bit synchronous SRAM word addresses and data	
	H	L	—	Setting prohibited	
		H	L	Multiplexing of 32-bit synchronous SRAM word addresses and data	
	H	H	L	—	Setting prohibited
			H	L	Multiplexing of 32-bit synchronous SRAM word addresses and data

Caution: With a synchronous SRAM type transfer mode, the asynchronous interface cannot be selected.

11.3.3 Write Status Mode and Write Strobe Mode

There are two types of write operation 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.

If 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 pin) or the write bus cycle continues until the next bus cycle starts (the low level is sampled from the HBCYSTZ pin).

If the HWRSTBZ pin is at the high level while the low level is input on the HBCYSTZ pin, 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).

11.3.4 Control Registers

(1) Register overview

The synchronous SRAM control registers of the external MCU interface are accessible by the MCU externally connected to an R-IN32M3.

Table 11.10 Synchronous SRAM Control Registers of the External MCU Interface

Address	Register name	Symbol	R/W	Access size [bit]			Initial Value
				8	16	32	
0F FF80H	HOSTIF synchronous SRAM control register 0	HIFEXT0	R/W	✓	✓	—	0000H
0F FF84H	HOSTIF synchronous SRAM control register 1	HIFEXT1	R/W	✓	✓	—	0202H

Remark: The synchronous SRAM control registers can only be accessed when the high level is input on the MEMCSEL pin.

(2) HOSTIF synchronous SRAM control register 0 (HIFEXT0)

This register sets up operation for synchronous SRAM type 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	R/W	R/W	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	MODTRN	Transfer mode selection 0: Synchronous SRAM single transfer 1: Setting prohibited
9	CNDWEO	WAIT release timing selection 0: Simultaneous with data 1: Precedes the data by one clock cycle
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
0	KESSBI	Effective edge selection for strobe signal input (HRDZ, HWRSTBZ) 0: Rising edge 1: Falling edge

Remarks 1. Only accessible by the external MCU.
2. This register can only be accessed when the high level is input on the MEMCSEL pin.

Caution: Do not write a value other than 0 to the bits fixed to 0. Writing any other value to these bits may lead to a malfunction.

(3) HOSTIF synchronous SRAM control register 1 (HIFEXT1)

This register sets up operation for synchronous SRAM type 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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

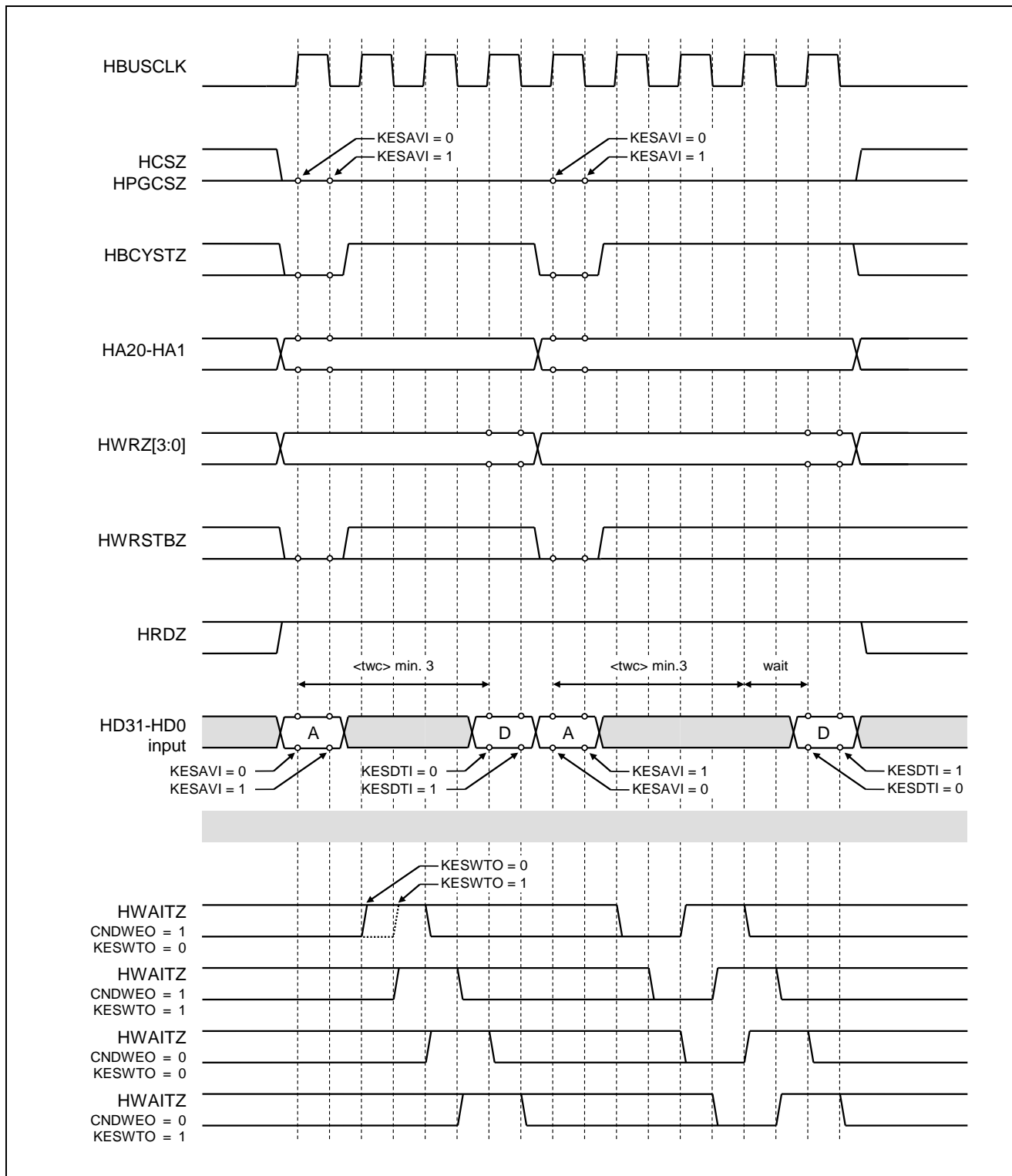
Bit Position	Bit Name	Function
11 to 8	DLYRA3- DLYRA0	Minimal time from the last input of the low level on the BCYSTZ pin to the point where read data can be acquired. (trc) 0000: 3 1000: 11 0001: 4 1001: 12 0010: 5 1010: 13 0011: 6 1011: 14 0100: 7 1100: 15 0101: 8 1101: 16 0110: 9 1110: 17 0111: 10 1111: 18
3 to 0	DLYWA3- DLYWA0	Minimal 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. Only accessible by the external MCU.
- 2. This register can only be accessed when the high level is input on the MEMCSEL pin.

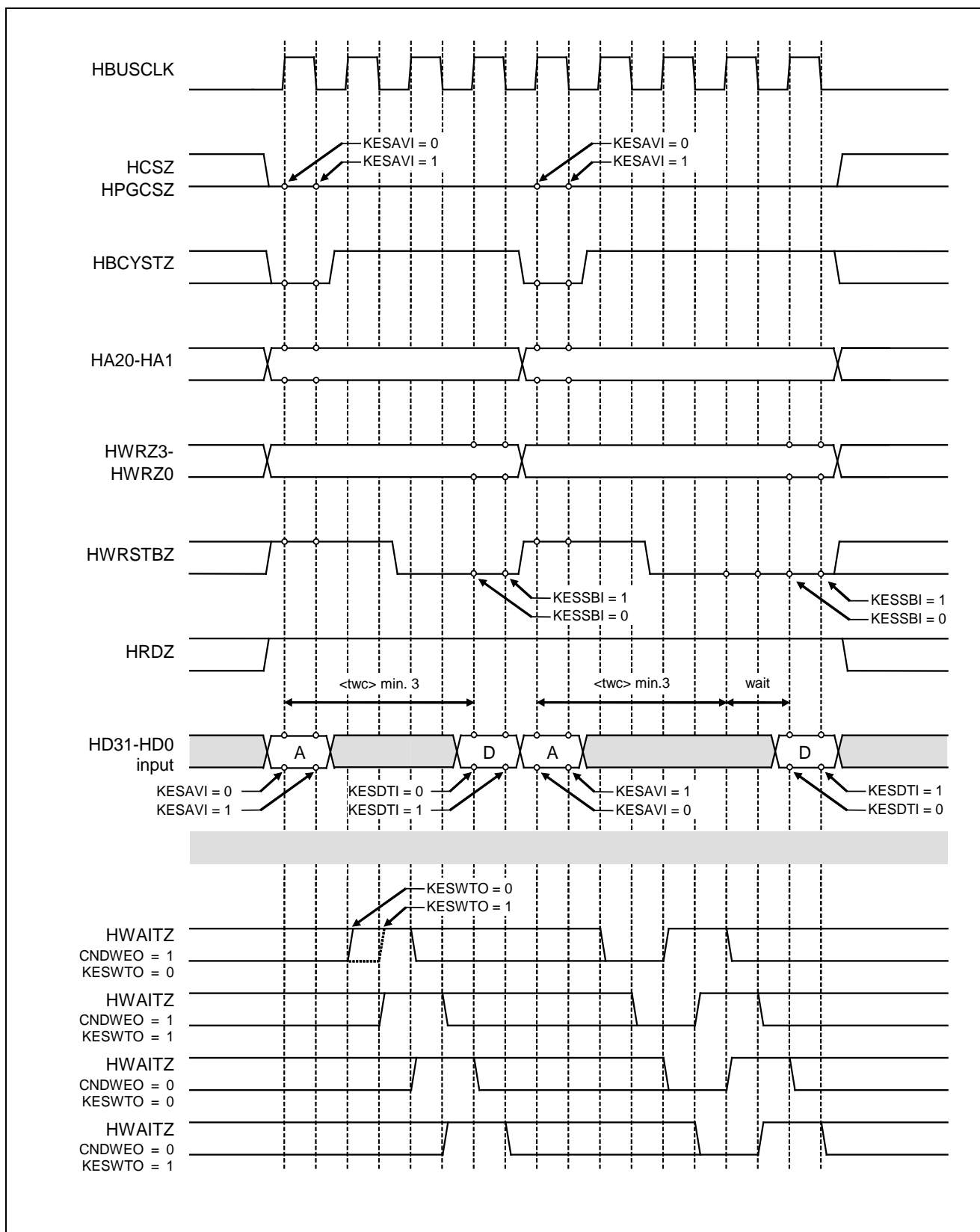
Caution: Do not write a value other than 0 to the bits fixed to 0. Writing any other value to these bits may lead to a malfunction.

11.3.5 Basic Operation Timing in Synchronous SRAM Type Transfer Mode

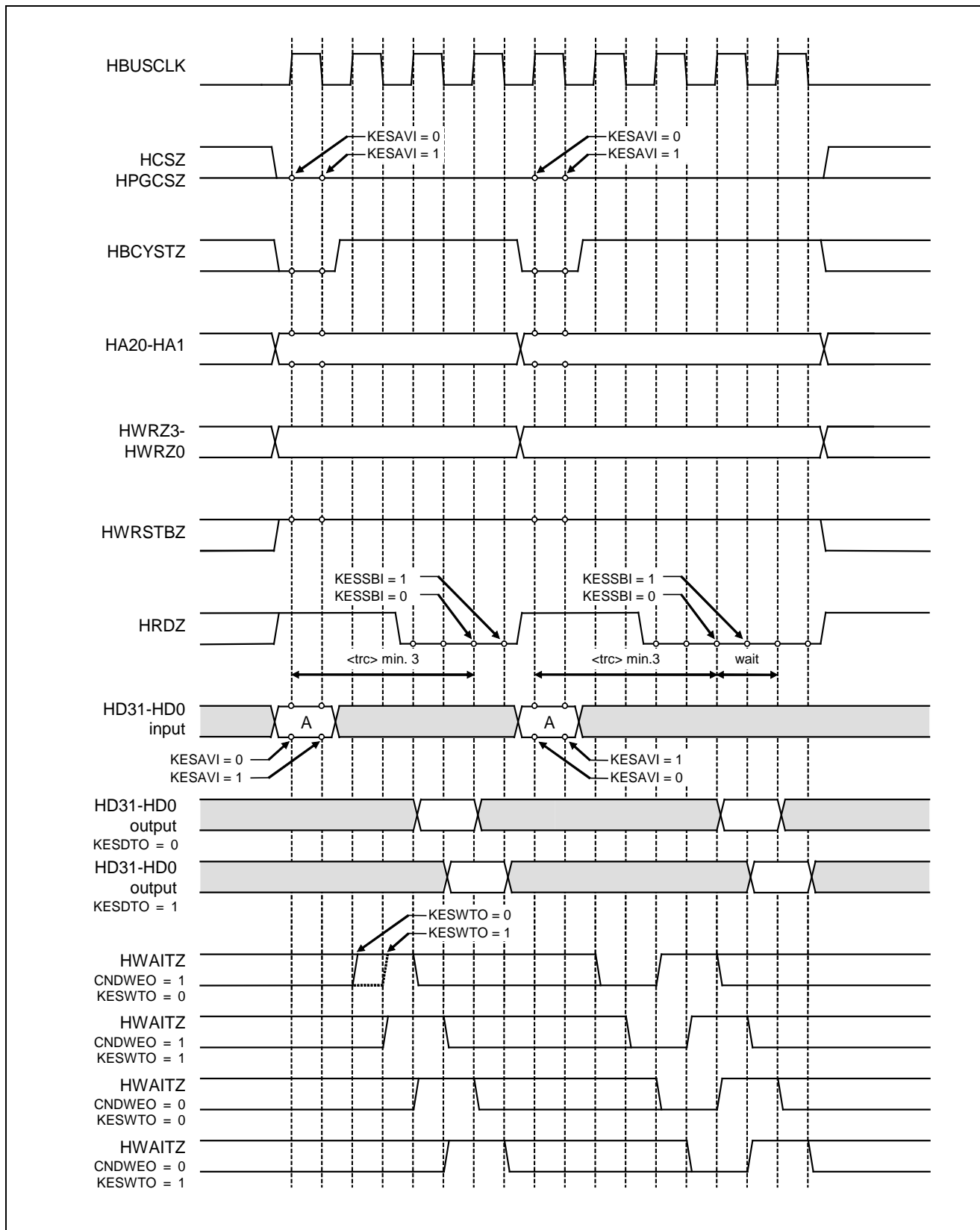
(1) Write access by the external MCU (synchronous SRAM single transfer, multiplexing of addresses and data, write status)



(2) Write access by the external MCU (synchronous SRAM single transfer, multiplexing of addresses and data, write strobe)



(3) Read access by the external MCU (synchronous SRAM single transfer, multiplexing of addresses and data)



11.3.6 Precautions

Precautions on usage of synchronous SRAM type transfer mode are described below.

(1) Method of access for HCSZ and HPGCSZ

Table 11.11 Register Settings and Method of Access Using the HCSZ Pin

Area	Register Settings			Method of Access for HCSZ	
	HIFEXT0. MODTRN	HIFPRC. PAGEONn	HIFBCC. RBUFONn	Synchronous SRAM Single Transfer	
				Write	Read
Area not to be buffered	—	—	—	OK	OK
Area to be buffered	0	—	—	OK	OK
	1	0	0	Prohibited	Prohibited ^{Note}
	1	0	1	Prohibited	Prohibited ^{Note}
	1	1	0	Prohibited	Prohibited ^{Note}
	1	1	1	Prohibited	Prohibited

Note: Attempted access via the external MCU interface may lead to a deadlock.

Table 11.12 Register Settings and Method of Access Using the HPGCSZ Pin

Area	Register Settings			Method of Access for HPGCSZ	
	HIFEXT0. MODTRN	HIFPRC. PAGEONn	HIFBCC. RBUFONx	Synchronous SRAM Single Transfer	
				Write	Read
All areas	—	—	—	OK	OK

(2) Timing for starting the internal bus cycle for writing

In bus cycles for writing in synchronous SRAM mode, data for writing are sampled in synchronization with HBUSCLK and an access request is issued to an internal block on the next falling edge of HBUSCLK.

Accordingly, if the connected external MCU only supplies the clock signal to the HBUSCLK pin during the bus cycle periods, writing to the target internal resource may not be completed.

If completion of writing to the target internal resource is essential, add some access for supplying HBUSCLK to supply the clock signal the HBUSCLK pin after the number of bus cycles that would otherwise be required for writing.

(3) Access during the internal reset period

In synchronous SRAM type transfer mode, register values cannot be read during the internal reset period.

(4) De-asserting the reset signal

The internal reset signal is de-asserted in synchronization with HBUSCLK supplied by the external MCU. Accordingly, if the external MCU only supplies HBUSCLK during bus cycles and the internal reset signal is not de-asserted at the point at which the first bus cycle starts, so that bus cycle is not recognizable. If you are using an applicable MCU, perform dummy access to supply HBUSCLK.

12. Serial Flash ROM Memory Controller

An R-IN32M3 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.

12.1 Features

- SPI interface:
 - 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, and fast read dual 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

12.2 Control Registers

To use the serial flash ROM memory controller, set the operating mode by using the control registers.

Table 12.1 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

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.

12.2.1 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 12, 3, and 2 to 0.

(1/2)

SFMSMD		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	
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SFMOSW	SFMOHW	SFMOEX	SFMMD3	SFMPAE	SFMPE	SFMSE1	SFMSE0	0	0	SFMRM1	SFMRM0	400A 2400H
		Initial value																											0000 0110H						
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	0	0	R/W	R/W	

Bit Position	Bit Name	Description
31 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	SFMPE	Selects prefetching. 0: Disables prefetching (initial value). 1: Enables prefetching.

(2/2)

Bit Position	Bit Name	Description															
5, 4	SFMSE1, SFMSE0	<p>Selects extension of the SMCSZ (chip select) signal after access to the SPI bus.</p> <table border="1"> <thead> <tr> <th>SFMSE1</th> <th>SFMSE0</th> <th>SMCSZ (chip select) signal extension mode</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>While the SMCSZ signal is at the high level, power consumption of the serial flash ROM is reduced.</p>	SFMSE1	SFMSE0	SMCSZ (chip select) signal extension mode	0	0	Does not extend the SMCSZ signal.	0	1	Extends the SMCSZ signal by up to 33 serial clock cycles (initial value).	1	0	Extends the SMCSZ signal by up to 129 serial clock cycles.	1	1	Extends the SMCSZ signal infinitely.
SFMSE1	SFMSE0	SMCSZ (chip select) signal extension mode															
0	0	Does not extend the SMCSZ signal.															
0	1	Extends the SMCSZ signal by up to 33 serial clock cycles (initial value).															
1	0	Extends the SMCSZ signal by up to 129 serial clock cycles.															
1	1	Extends the SMCSZ signal infinitely.															
3, 2	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.															
1, 0	SFMRM1, SFMRM0	<p>Selects the read mode of the serial flash ROM.</p> <table border="1"> <thead> <tr> <th>SFMRM1</th> <th>SFMRM0</th> <th>Serial flash ROM read mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Reading (initial value)</td> </tr> <tr> <td>0</td> <td>1</td> <td>Fast Read</td> </tr> <tr> <td>1</td> <td>0</td> <td>Fast Read Dual Output</td> </tr> <tr> <td>1</td> <td>1</td> <td>Fast Read Dual I/O</td> </tr> </tbody> </table>	SFMRM1	SFMRM0	Serial flash ROM read mode	0	0	Reading (initial value)	0	1	Fast Read	1	0	Fast Read Dual Output	1	1	Fast Read Dual I/O
SFMRM1	SFMRM0	Serial flash ROM read mode															
0	0	Reading (initial value)															
0	1	Fast Read															
1	0	Fast Read Dual Output															
1	1	Fast Read Dual I/O															

12.2.2 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	Address	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	400A 2404H
																												Initial value					
																												0000 0037H					
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	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	0	0	0	0	0	0	R/W	
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 SMCSZ 0.5 clock cycles before the first rising edge of SMSCK. 1: Outputs SMCSZ 1.5 clock cycles before the first rising edge of SMSCK (initial value).																															
4	SFMSHD	Selects the timing for de-asserting the SMCSZ (chip select) signal. 0: De-asserts SMCSZ 0.5 clock cycles after the last rising edge of SMSCK. 1: De-asserts SMCSZ 1.5 clock cycles after the last rising edge of SMSCK (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																											

12.2.3 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	Address
SFMSKC																												400A 2408H				
0 0																												Initial value				
																												0000 0008H				
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	R/W	R/W	R/W	R/W	

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). <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>SFMDV4</th><th>SFMDV3</th><th>SFMDV2</th><th>SFMDV1</th><th>SFMDV0</th><th>Serial clock selection</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>HCLK/2</td></tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>HCLK/3 (divided by an odd number) <small>Note</small></td></tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>HCLK/4</td></tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>HCLK/5 (divided by an odd number) <small>Note</small></td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>HCLK/6</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>HCLK/7 (divided by an odd number) <small>Note</small></td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>HCLK/8</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>HCLK/9 (divided by an odd number) <small>Note</small></td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>HCLK/10 (initial value)</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>HCLK/11 (divided by an odd number) <small>Note</small></td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>HCLK/12</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>HCLK/13 (divided by an odd number) <small>Note</small></td></tr> </tbody> </table>	SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection	0	0	0	0	0	HCLK/2	0	0	0	0	1	HCLK/3 (divided by an odd number) <small>Note</small>	0	0	0	1	0	HCLK/4	0	0	0	1	1	HCLK/5 (divided by an odd number) <small>Note</small>	0	0	1	0	0	HCLK/6	0	0	1	0	1	HCLK/7 (divided by an odd number) <small>Note</small>	0	0	1	1	0	HCLK/8	0	0	1	1	1	HCLK/9 (divided by an odd number) <small>Note</small>	0	1	0	0	0	HCLK/10 (initial value)	0	1	0	0	1	HCLK/11 (divided by an odd number) <small>Note</small>	0	1	0	1	0	HCLK/12	0	1	0	1	1	HCLK/13 (divided by an odd number) <small>Note</small>
SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection																																																																											
0	0	0	0	0	HCLK/2																																																																											
0	0	0	0	1	HCLK/3 (divided by an odd number) <small>Note</small>																																																																											
0	0	0	1	0	HCLK/4																																																																											
0	0	0	1	1	HCLK/5 (divided by an odd number) <small>Note</small>																																																																											
0	0	1	0	0	HCLK/6																																																																											
0	0	1	0	1	HCLK/7 (divided by an odd number) <small>Note</small>																																																																											
0	0	1	1	0	HCLK/8																																																																											
0	0	1	1	1	HCLK/9 (divided by an odd number) <small>Note</small>																																																																											
0	1	0	0	0	HCLK/10 (initial value)																																																																											
0	1	0	0	1	HCLK/11 (divided by an odd number) <small>Note</small>																																																																											
0	1	0	1	0	HCLK/12																																																																											
0	1	0	1	1	HCLK/13 (divided by an odd number) <small>Note</small>																																																																											

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.

Remark: HCLK: Internal system bus clock

(2/2)

Bit Position	Bit Name	Description																																																																																																																														
4 to 0	SFMDV4-SFMDV0	<p>Selects the serial clock (SMSCK) based on the internal system bus clock (HCLK).</p> <table border="1"> <thead> <tr> <th>SFMDV4</th> <th>SFMDV3</th> <th>SFMDV2</th> <th>SFMDV1</th> <th>SFMDV0</th> <th>Serial clock selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>HCLK/14</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>HCLK/15 (divided by an odd number) ^{Note}</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>HCLK/16</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>HCLK/17 (divided by an odd number) ^{Note}</td> </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> </tbody> </table>	SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection	0	1	1	0	0	HCLK/14	0	1	1	0	1	HCLK/15 (divided by an odd number) ^{Note}	0	1	1	1	0	HCLK/16	0	1	1	1	1	HCLK/17 (divided by an odd number) ^{Note}	1	0	0	0	0	HCLK/18	1	0	0	0	1	HCLK/20	1	0	0	1	0	HCLK/22	1	0	0	1	1	HCLK/24	1	0	1	0	0	HCLK/26	1	0	1	0	1	HCLK/28	1	0	1	1	0	HCLK/30	1	0	1	1	1	HCLK/32	1	1	0	0	0	HCLK/34	1	1	0	0	1	HCLK/36	1	1	0	1	0	HCLK/38	1	1	0	1	1	HCLK/40	1	1	1	0	0	HCLK/42	1	1	1	0	1	HCLK/44	1	1	1	1	0	HCLK/46	1	1	1	1	1	HCLK/48
SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection																																																																																																																											
0	1	1	0	0	HCLK/14																																																																																																																											
0	1	1	0	1	HCLK/15 (divided by an odd number) ^{Note}																																																																																																																											
0	1	1	1	0	HCLK/16																																																																																																																											
0	1	1	1	1	HCLK/17 (divided by an odd number) ^{Note}																																																																																																																											
1	0	0	0	0	HCLK/18																																																																																																																											
1	0	0	0	1	HCLK/20																																																																																																																											
1	0	0	1	0	HCLK/22																																																																																																																											
1	0	0	1	1	HCLK/24																																																																																																																											
1	0	1	0	0	HCLK/26																																																																																																																											
1	0	1	0	1	HCLK/28																																																																																																																											
1	0	1	1	0	HCLK/30																																																																																																																											
1	0	1	1	1	HCLK/32																																																																																																																											
1	1	0	0	0	HCLK/34																																																																																																																											
1	1	0	0	1	HCLK/36																																																																																																																											
1	1	0	1	0	HCLK/38																																																																																																																											
1	1	0	1	1	HCLK/40																																																																																																																											
1	1	1	0	0	HCLK/42																																																																																																																											
1	1	1	0	1	HCLK/44																																																																																																																											
1	1	1	1	0	HCLK/46																																																																																																																											
1	1	1	1	1	HCLK/48																																																																																																																											

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.

Remark: HCLK: Internal system bus clock

12.2.4 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.

	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	
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	0	PFOFF	PFFUL	0	0	0	PFCNT2	PFCNT1	PFCNT0	400A 240CH
																												Initial value						
																												0000 0080H						
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	R	

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 to 3	—	Reserved. These bits are read as 0.																																				
2 to 0	PFCNT2- PFCNT0	Indicates the number of bytes of the prefetched data. <table border="1" style="margin: 5px auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>PFCNT2</th> <th>PFCNT1</th> <th>PFCNT0</th> <th>Number of bytes of prefetched data</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>No prefetched data (initial value)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1 byte of data prefetched</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>2 bytes of data prefetched</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>3 bytes of data prefetched</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>4 bytes of data prefetched</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>5 bytes of data prefetched</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>6 bytes of data prefetched</td> </tr> <tr> <td colspan="3">Other than the above</td> <td>No other combination is available.</td> </tr> </tbody> </table>	PFCNT2	PFCNT1	PFCNT0	Number of bytes of prefetched data	0	0	0	No prefetched data (initial value)	0	0	1	1 byte of data prefetched	0	1	0	2 bytes of data prefetched	0	1	1	3 bytes of data prefetched	1	0	0	4 bytes of data prefetched	1	0	1	5 bytes of data prefetched	1	1	0	6 bytes of data prefetched	Other than the above			No other combination is available.
PFCNT2	PFCNT1	PFCNT0	Number of bytes of prefetched data																																			
0	0	0	No prefetched data (initial value)																																			
0	0	1	1 byte of data prefetched																																			
0	1	0	2 bytes of data prefetched																																			
0	1	1	3 bytes of data prefetched																																			
1	0	0	4 bytes of data prefetched																																			
1	0	1	5 bytes of data prefetched																																			
1	1	0	6 bytes of data prefetched																																			
Other than the above			No other combination is available.																																			

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 PFCNT2 to PFCNT0 bits.

12.2.5 Communications Port Register (SFMCOM)

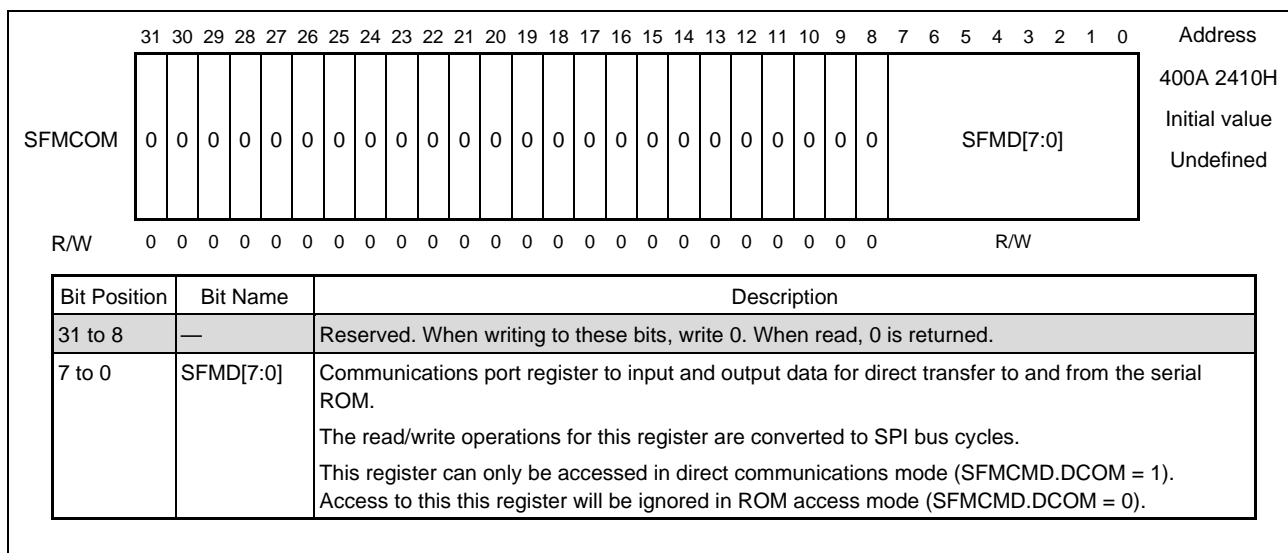
This is an I/O port used to output instruction codes, addresses, and write data to the serial flash ROM, as well as to retrieve read data and status information from the serial flash ROM.

When data is written to the SFMCOM port in direct communications mode (SFMCMDCOM = 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 SFMCMDCOM 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-IN32M3 products, software control via SFMCOM is necessary for programming and erasure.

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



12.2.6 Communications Mode Control Register (SFMCMMD)

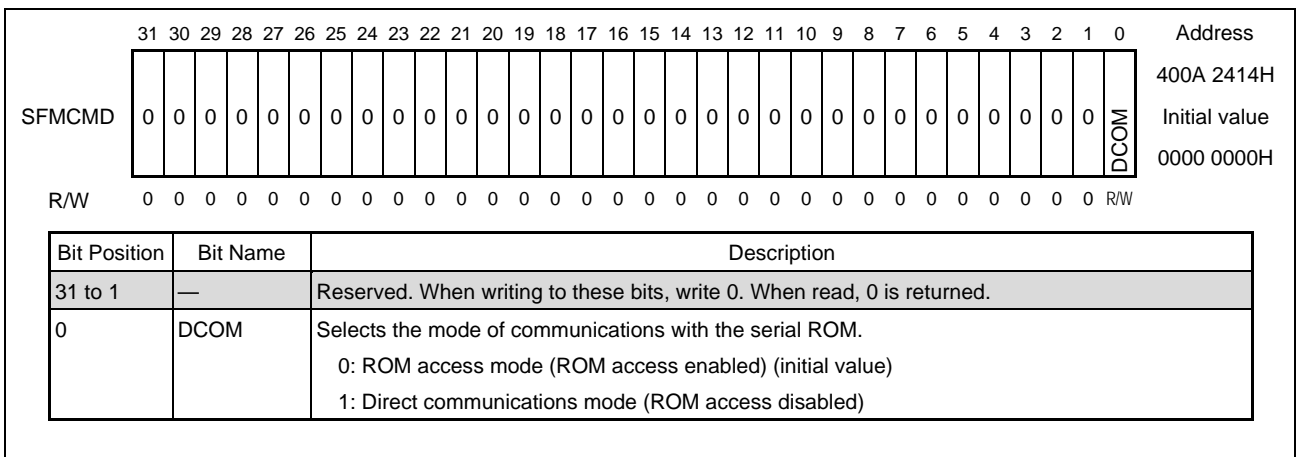
This register is used to select the mode of communications between the system bus and SPI bus. There are two modes of communications between the system bus and SPI bus: ROM access mode and direct communications mode.

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 SFMCMMD 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.



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.

12.2.7 Communications Status Register (SFCMST)

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.

SFCMST	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 2418H Initial value 0000 0000H
	0 R/W 0 0 0 0 0 0 0 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	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-IN32M3. The specifications and restrictions on the individual serial ROM products from various vendors require software control.

12.3 Connection with Serial Flash ROM

An R-IN32M3 device is connected with the serial flash ROM as shown below.

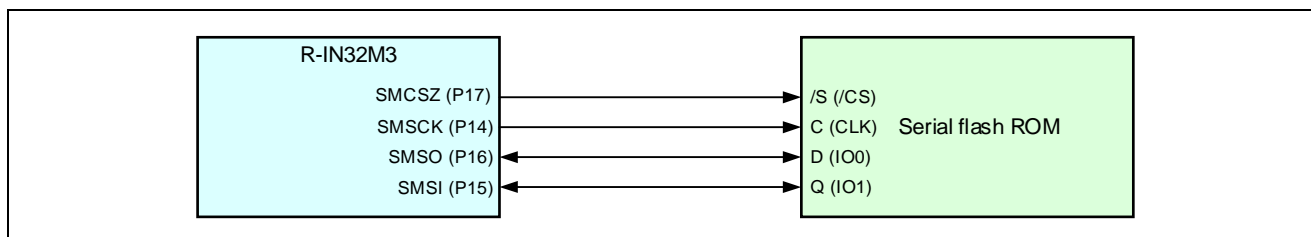


Figure 12.1 Connection with Serial Flash ROM

12.4 Operation

12.4.1 SPI Bus Operation

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 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 SMSCK signal on standby. In SPI mode 0, the standby level of the SMSCK signal is the low level. In SPI mode 3, the standby level of the SMSCK signal is the high level.

Serial data for output is output from the flash memory in synchronization with the falling edges of the serial clock signal (SMSCK) and acquired in synchronization with the rising edges of the serial clock signal (SMSCK).

Serial data for input are output from the flash memory in synchronization with falling edges of the serial clock signal (SMSCK) and acquired in synchronization with the following falling edges.

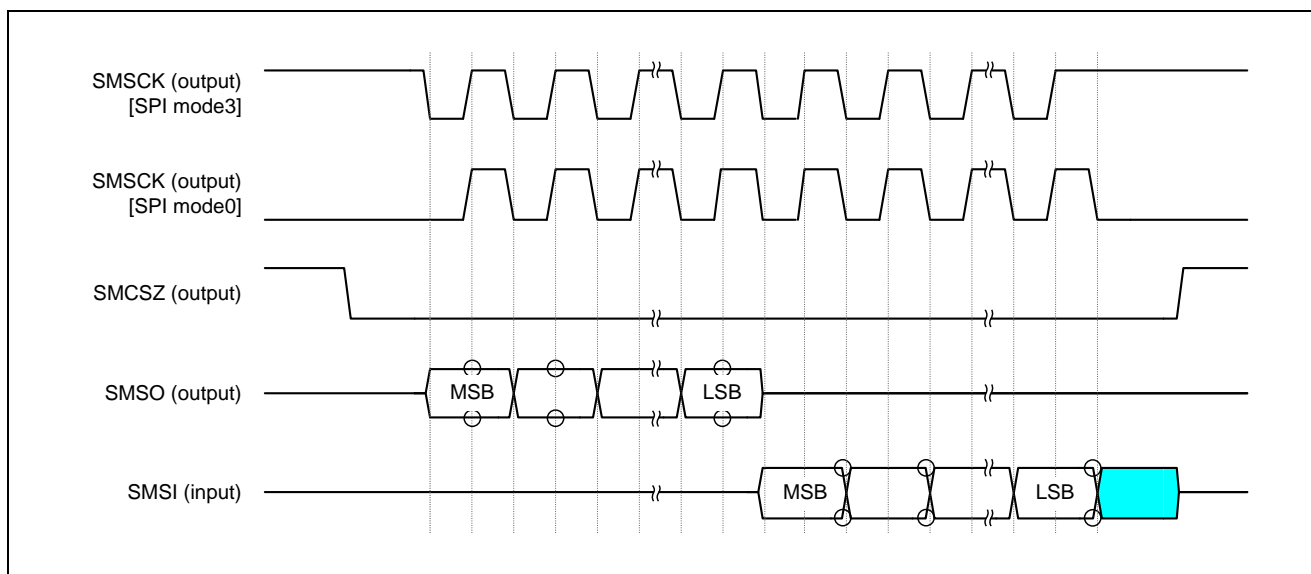


Figure 12.2 Basic Operation of SPI Bus

12.4.2 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.

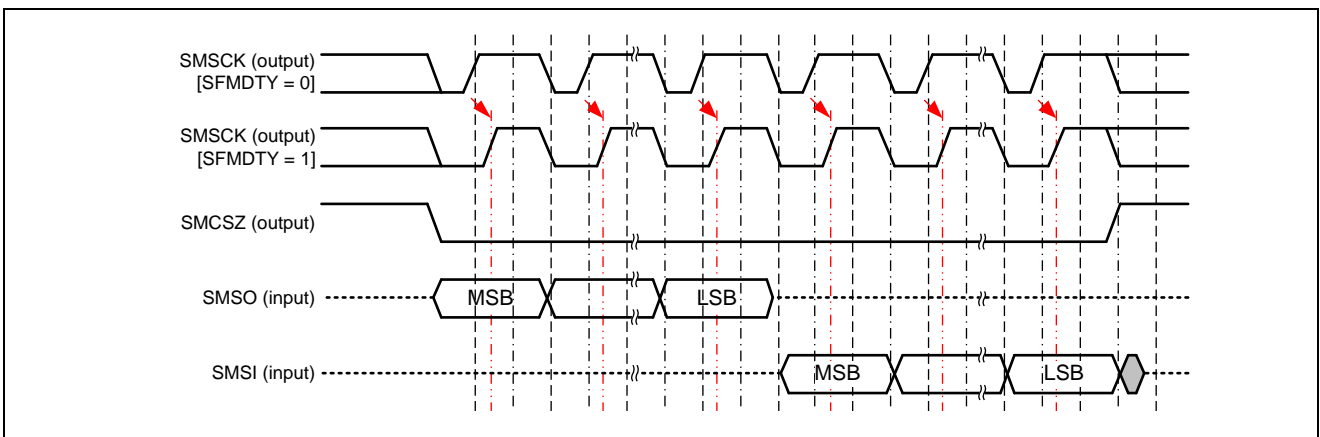


Figure 12.3 Correction of the SMSCK Signal Duty Factor by Using the SFMDTY Bit (Example of HCLK/3)

(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.

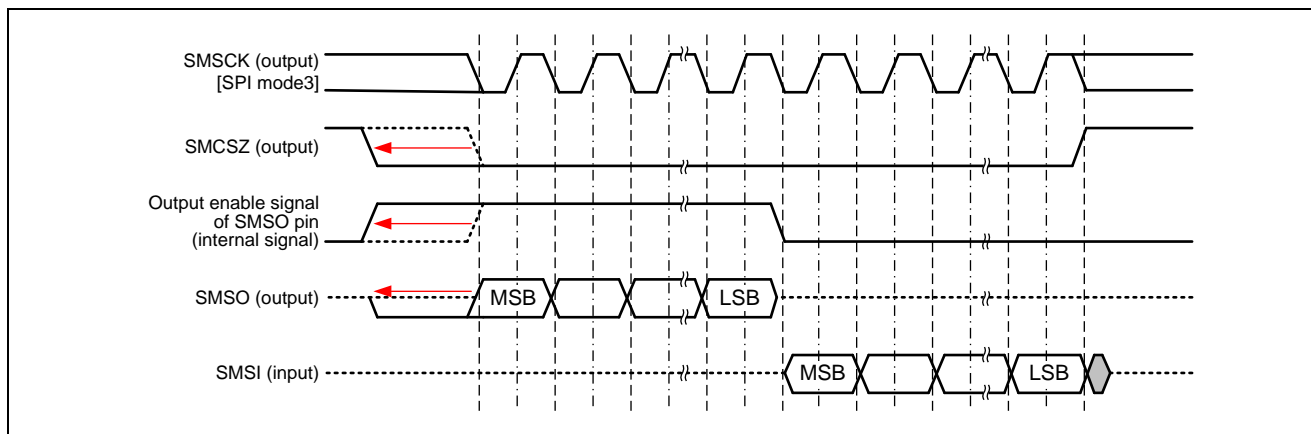


Figure 12.4 SMCSZ Signal Setup Time Adjustment by Using the SFMSLD Bit

(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.

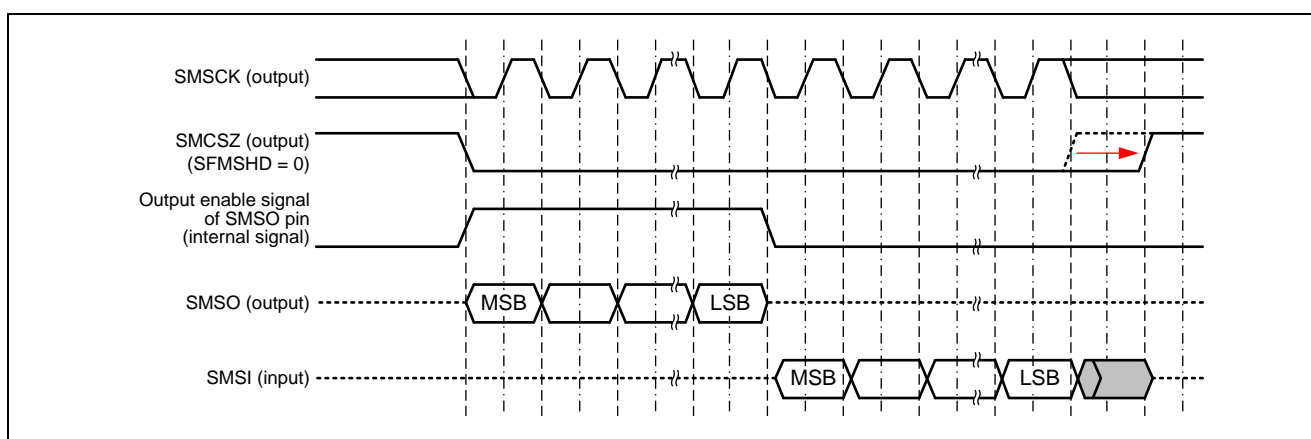


Figure 12.5 SMCSZ Signal Hold Time Adjustment by Using the SFMSHD Bit

(6) Output Enable Time of the Serial Data Output Buffer

The buffer output enable time of the SMSO and SMSI pins can be extended by one SMSCK clock cycle by using SFMSMD.SFMOEX.

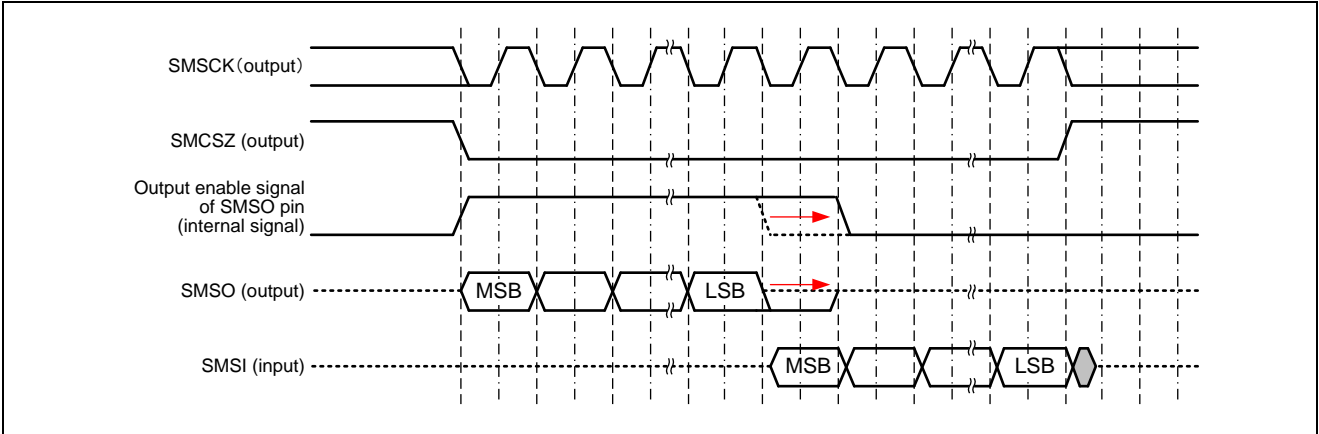


Figure 12.6 Output Enable Time Adjustment by Using the SFMOEX Bit

(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 SMSCK clock cycle.

If the SFMOSW bit is set to 1, the width at low level of SMSCK is extended by one SMSCK clock cycle when serial data is transmitted during a data output cycle. This function does not take effect for serial data reception.

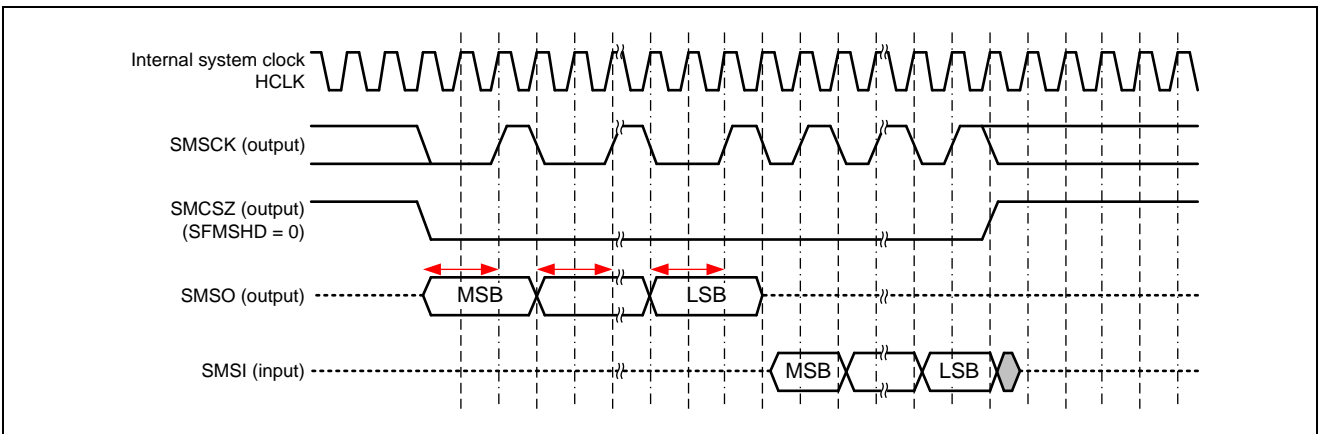


Figure 12.7 Serial Data Setup Time Adjustment by Using the SFMOSW Bit

(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 SMSCK clock cycle.

If the SFMOHW bit is set to 1, the width at high level of SMSCK is extended by one SMSCK clock cycle when serial data is transmitted during a data output cycle. This function does not take effect for serial data reception.

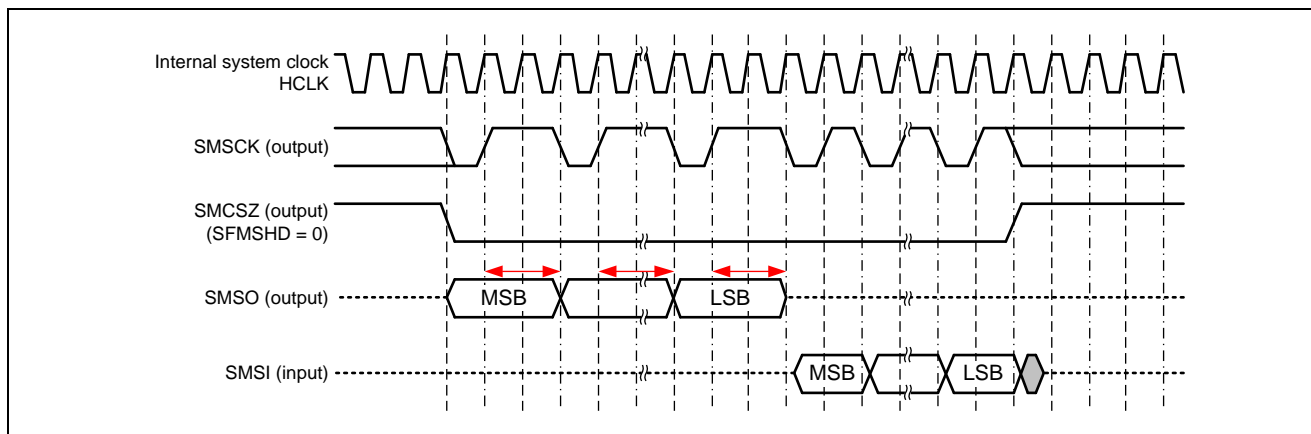


Figure 12.8 Serial Data Hold Time Adjustment by Using the SFMOHW Bit

12.4.3 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 12.2 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 read	03H	3	—	1 to infinite	SFMRM[1:0] = 00B
Fast Read	0BH	3	1	1 to infinite	SFMRM[1:0] = 01B
Fast Read Dual Output	3BH	3	1	1 to infinite	SFMRM[1:0] = 10B
Fast Read Dual I/O	BBH	3	1	1 to infinite	SFMRM[1:0] = 11B
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.

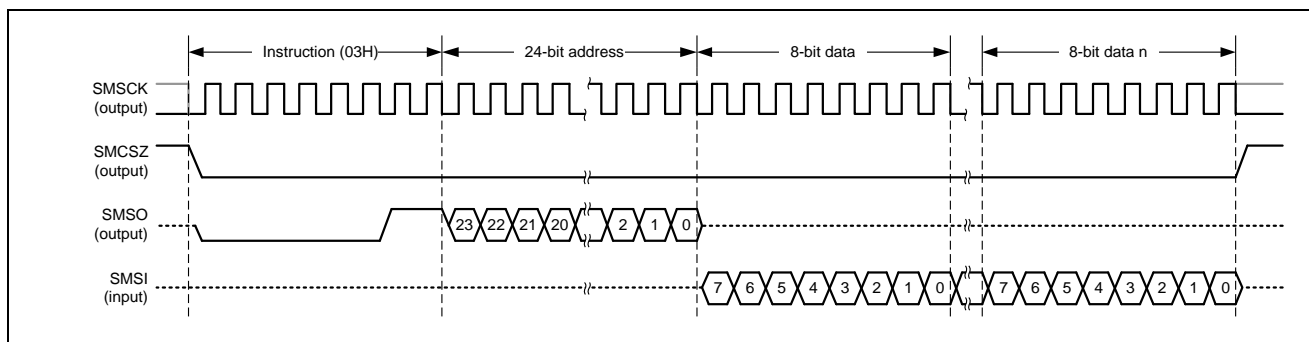


Figure 12.9 Bus Cycles for Standard Reading

(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.

For switching to fast read, use SFMSMD.SFMRM1 or SFMSMD.SFMRM0.

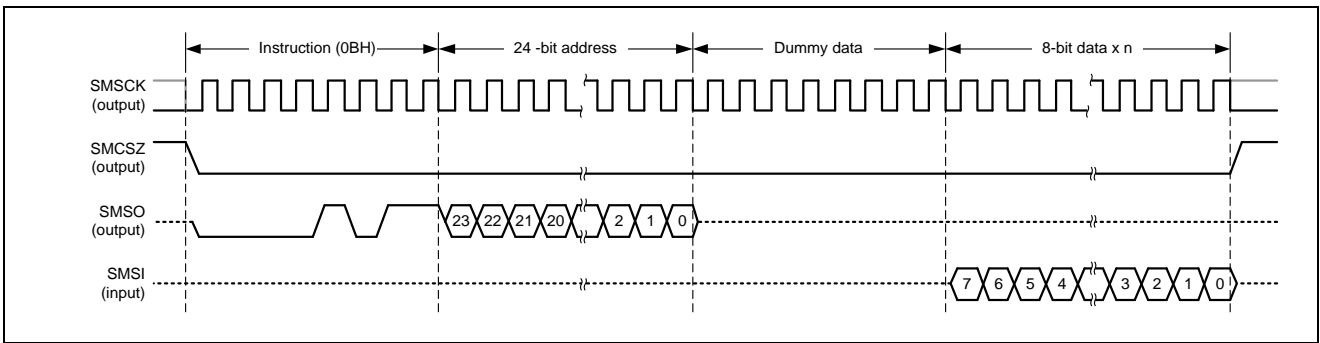


Figure 12.10 Bus Cycles for Fast Read

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 SMSO and SMSI pins.

Even-numbered bits of data are received via the SMSO pin, and odd-numbered bits are received via the SMSI pin.

For switching to fast read dual output, use SFMSMD.SFMRM1 or SFMSMD.SFMRM0.

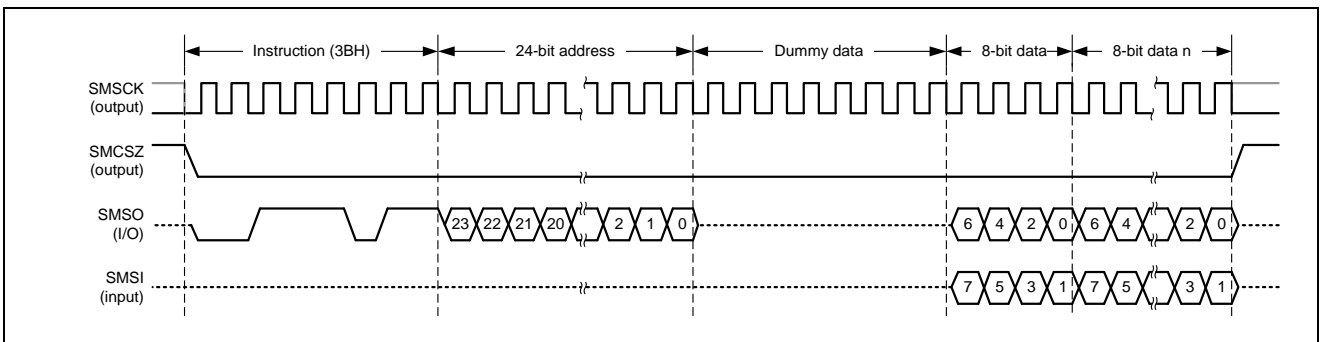


Figure 12.11 Bus Cycles for Fast Read Dual Output

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 SMSO and SMSI pins, followed by reception of the data by using the SMSO and SMSI pins.

When an address and dummy data are transmitted and data is received, the SMSO pin is used for even-numbered bits and the SMSI pin is used for odd-numbered bits.

For switching to fast read dual I/O, use SFMSMD.SFMRM1 or SFMSMD.SFMRM0.

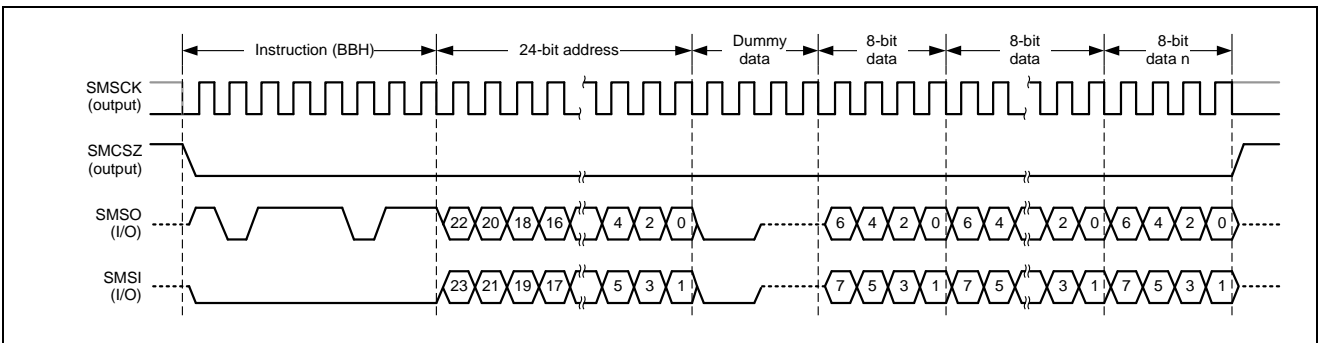


Figure 12.12 Bus Cycles for Fast Read Dual I/O

Caution: Use fast read dual I/O only for serial flash ROMs that support fast read dual I/O.

(6) 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.

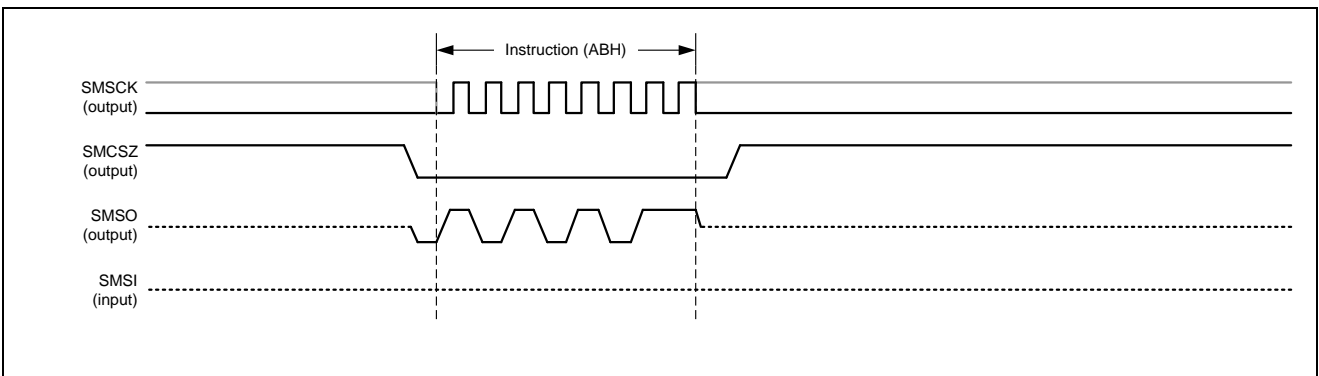


Figure 12.13 Bus Cycles for Release from Deep Power-Down

12.4.4 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.

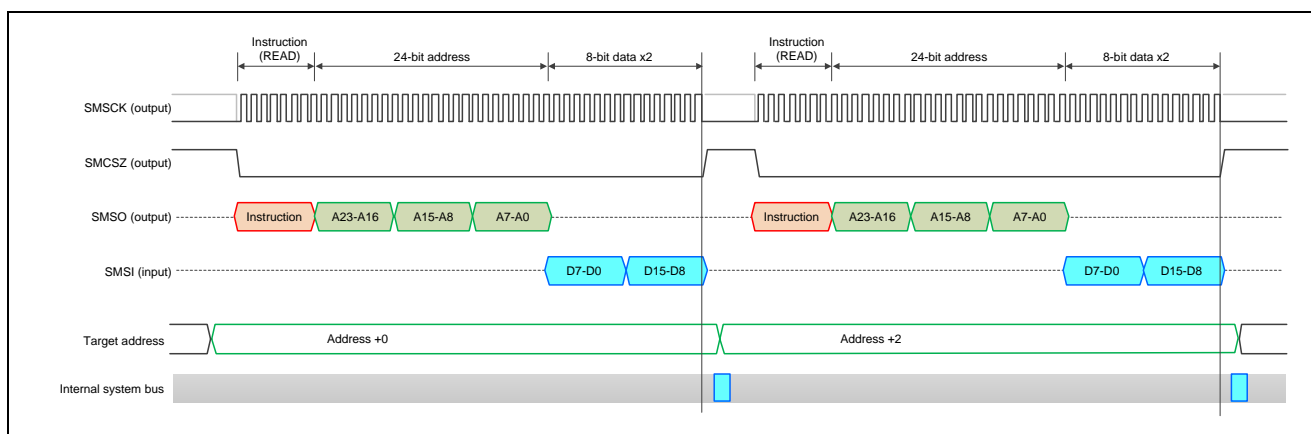


Figure 12.14 Continuous Data Reading by Individual Conversion

(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-IN32M3 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 6 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.

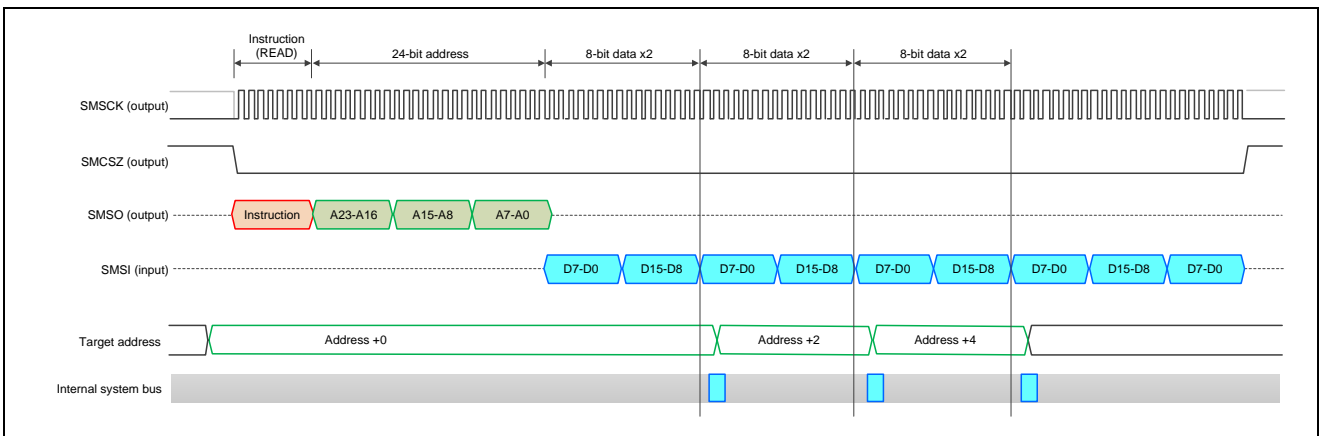


Figure 12.15 Continuous Data Reading by Using Prefetching

(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 SMCLK 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 SMCLK 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.

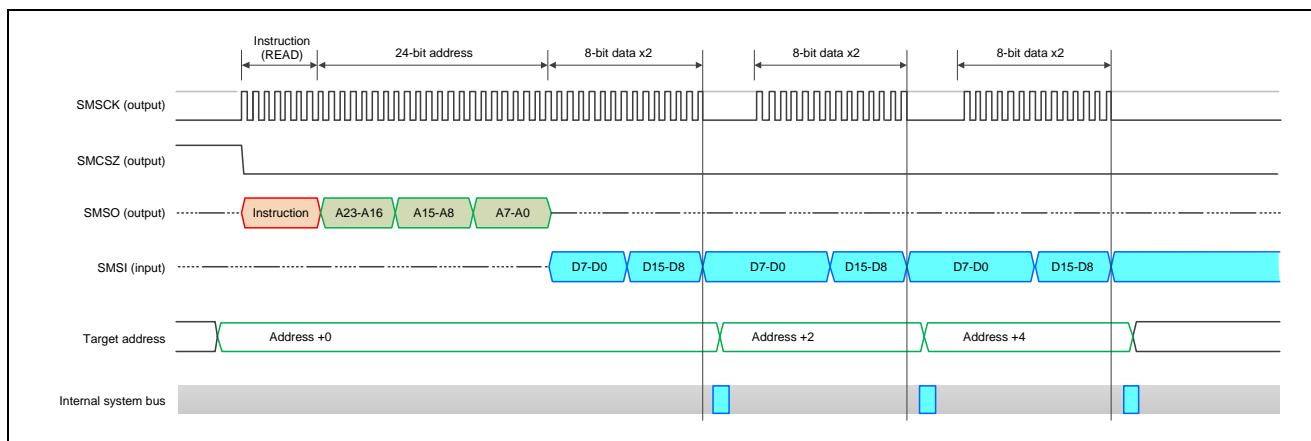


Figure 12.16 Continuous Data Reading by Using SPI Bus Cycle Extension

12.4.5 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 12336 cycles of the internal system bus clocks (HCLK). When the system is configured to boot from the serial flash ROM, booting-up starts after this wait.

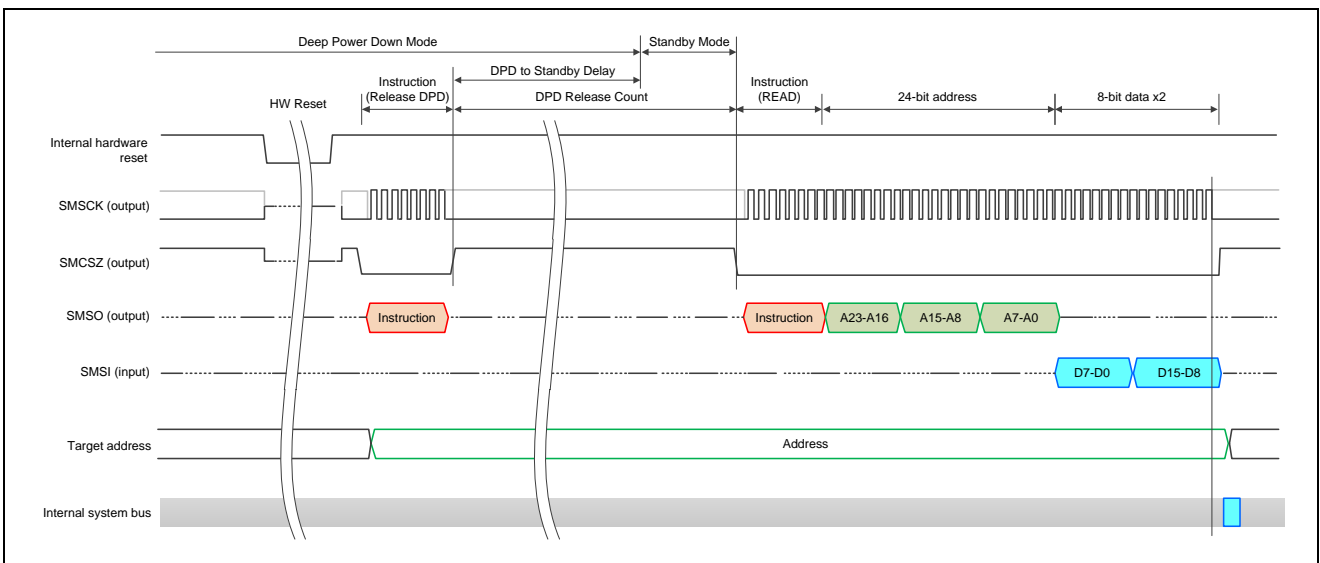


Figure 12.17 Operation for Automatic Release from the Deep Power-Down State

12.4.6 Direct Communications

(1) About Direct Communications

While the serial flash ROM memory controller of an R-IN32M3 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-IN32M3, 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 (SFMSMD, SFMSSC, SFMSKC, SFMSST, and SFMCST) is prohibited.**
- 2. The completion of an SPI bus cycle by writing to a register other than SFMCMD is not guaranteed as official functionality.**

12.5 Example of Configuration

This section describes the settings of the registers and serial flash ROM for standard reading and fast read dual I/O. The settings assume use of serial flash ROM “S25FL032P0XNFI010” and “S25FL064P0XNFI001” in “TS-R-RIN32M3-EC” and “TS-R-RIN32M3-CL” from TESSERA TECHNOLOGY. Please make appropriate settings according to your usage environment. For details about the registers of the serial flash ROM memory controller, see section 12.2, Control Registers.

12.5.1 Standard Reading

(1) Operation

Standard reading is a common method of reading supported by most serial flash ROMs. For details, see section 12.4.3, 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

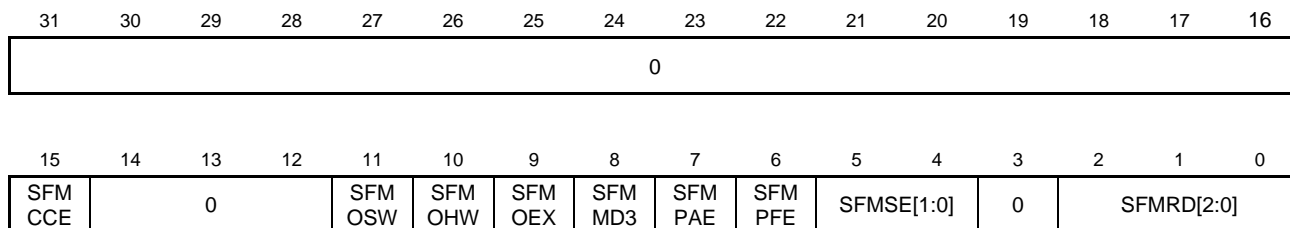


Table 12.3 SFMSMD Register Settings for Standard Reading

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

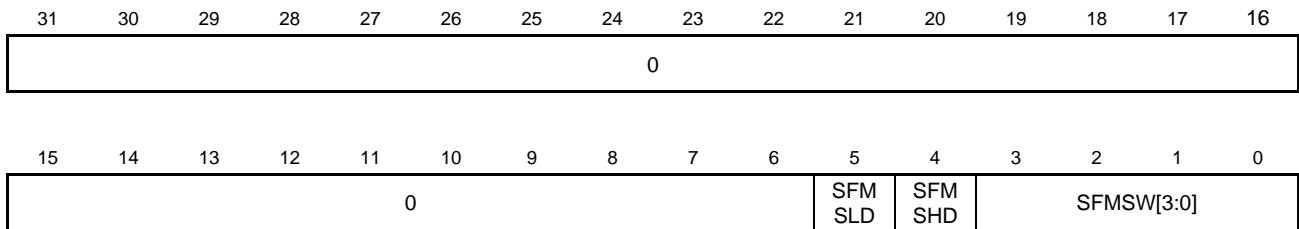


Table 12.4 SFMSSC Register Settings for Standard Reading

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 border="1" style="margin-left: 20px;"> <thead> <tr> <th>SFMSW3</th> <th>SFMSW2</th> <th>SFMSW1</th> <th>SFMSW0</th> <th>Minimum width at high level of SMCSZ signal</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>8 SMSCK cycles (initial value)</td> </tr> </tbody> </table>	SFMSW3	SFMSW2	SFMSW1	SFMSW0	Minimum width at high level of SMCSZ signal	0	1	1	1	8 SMSCK cycles (initial value)
SFMSW3	SFMSW2	SFMSW1	SFMSW0	Minimum width at high level of SMCSZ signal							
0	1	1	1	8 SMSCK cycles (initial value)							

(c) SFMSKC Register

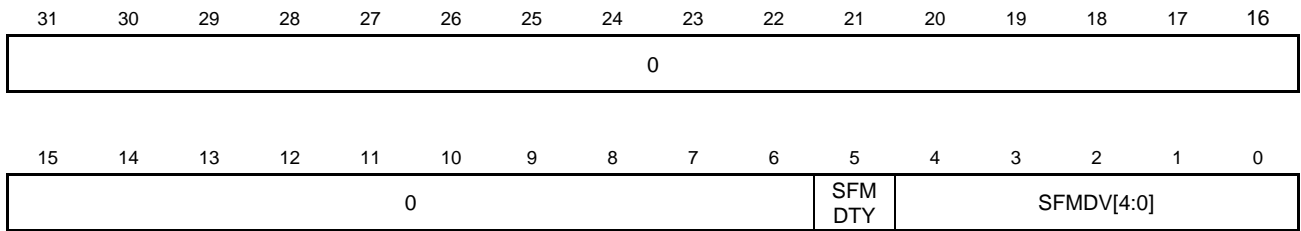


Table 12.5 SFMSKC Register Settings for Standard Reading

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. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>SFMDV4</th> <th>SFMDV3</th> <th>SFMDV2</th> <th>SFMDV1</th> <th>SFMDV0</th> <th>Serial clock selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>HCLK/4</td> </tr> </tbody> </table>	SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection	0	0	0	1	0	HCLK/4
SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection								
0	0	0	1	0	HCLK/4								

12.5.2 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 12.4.3, 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

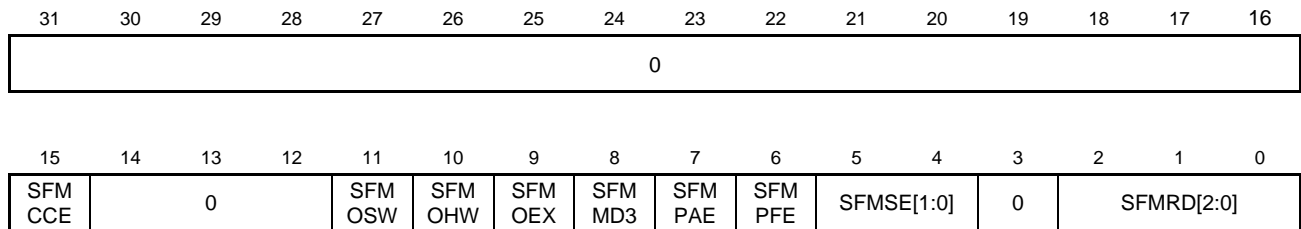


Table 12.6 SFMSMD Register Settings for Fast Read Dual I/O

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

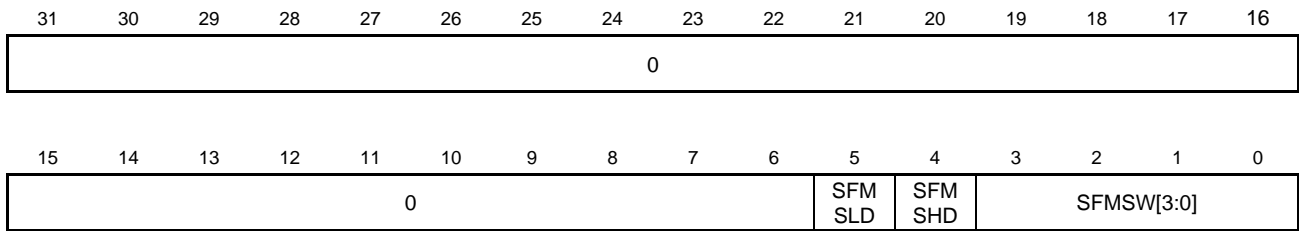


Table 12.7 SFMSSC Register Settings for Fast Read Dual I/O

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. <table border="1" style="margin-left: 20px; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">SFMSW3</th> <th style="text-align: center;">SFMSW2</th> <th style="text-align: center;">SFMSW1</th> <th style="text-align: center;">SFMSW0</th> <th style="text-align: center;">Minimum width at high level of SMCSZ signal</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">8 SMSCK cycles (initial value)</td> </tr> </tbody> </table>	SFMSW3	SFMSW2	SFMSW1	SFMSW0	Minimum width at high level of SMCSZ signal	0	1	1	1	8 SMSCK cycles (initial value)
SFMSW3	SFMSW2	SFMSW1	SFMSW0	Minimum width at high level of SMCSZ signal							
0	1	1	1	8 SMSCK cycles (initial value)							

(c) SFMSKC Register

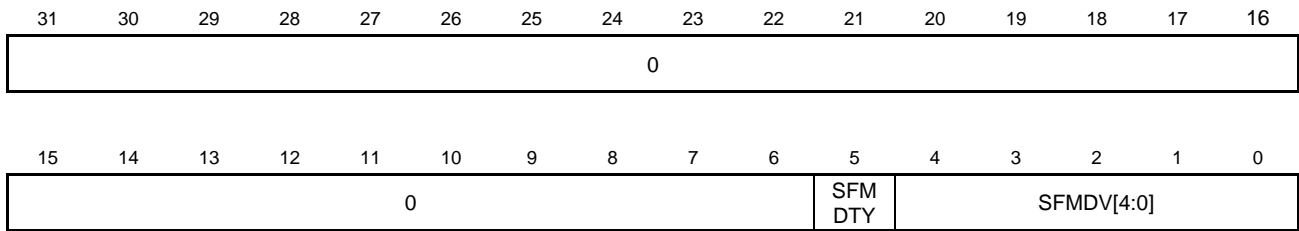


Table 12.8 SFMSKC Register Settings for Fast Read Dual I/O

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 border="1" style="margin-left: 20px;"> <thead> <tr> <th>SFMDV4</th> <th>SFMDV3</th> <th>SFMDV2</th> <th>SFMDV1</th> <th>SFMDV0</th> <th>Serial clock selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>HCLK/2</td> </tr> </tbody> </table>	SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection	0	0	0	0	0	HCLK/2
SFMDV4	SFMDV3	SFMDV2	SFMDV1	SFMDV0	Serial clock selection								
0	0	0	0	0	HCLK/2								

13. DMA Controllers

An R-IN32M3 incorporates the two DMA controllers for the AHB bus, which consist of a total of five channels.

Table 13.1 R-IN32M3 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-IN32M3, 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-IN32M3 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.

13.1 Features

13.1.1 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 13.2 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 ^{Note4}	✓	✓
APB internal peripheral modules ^{Note1}	✓	✓
Real-time ports	—	✓
General ports	✓	—
HWOS ^{Note2}	—	—
DMA controller for real-time ports ^{Note3}	—	—
General DMA controller ^{Note3}	—	—

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.
Their respective symbol names are TAUJ2, WDT, UARTE, CSIH, IICB, and AFC.

2. Hardware real-time OS

3. The register area of each DMA controller

4. The target module differs with each R-IN32M3 series product.
[R-IN32M3-EC] : EtherCAT
[R-IN32M3-CL] : 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 13.3 Relation between DMA Units/Channels and External DMA Interface Pins

Type	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**
- 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).**
 - 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-byte) boundary should set as the start address.**
 - Write to the addresses continuously in the increment direction.**

13.2 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.

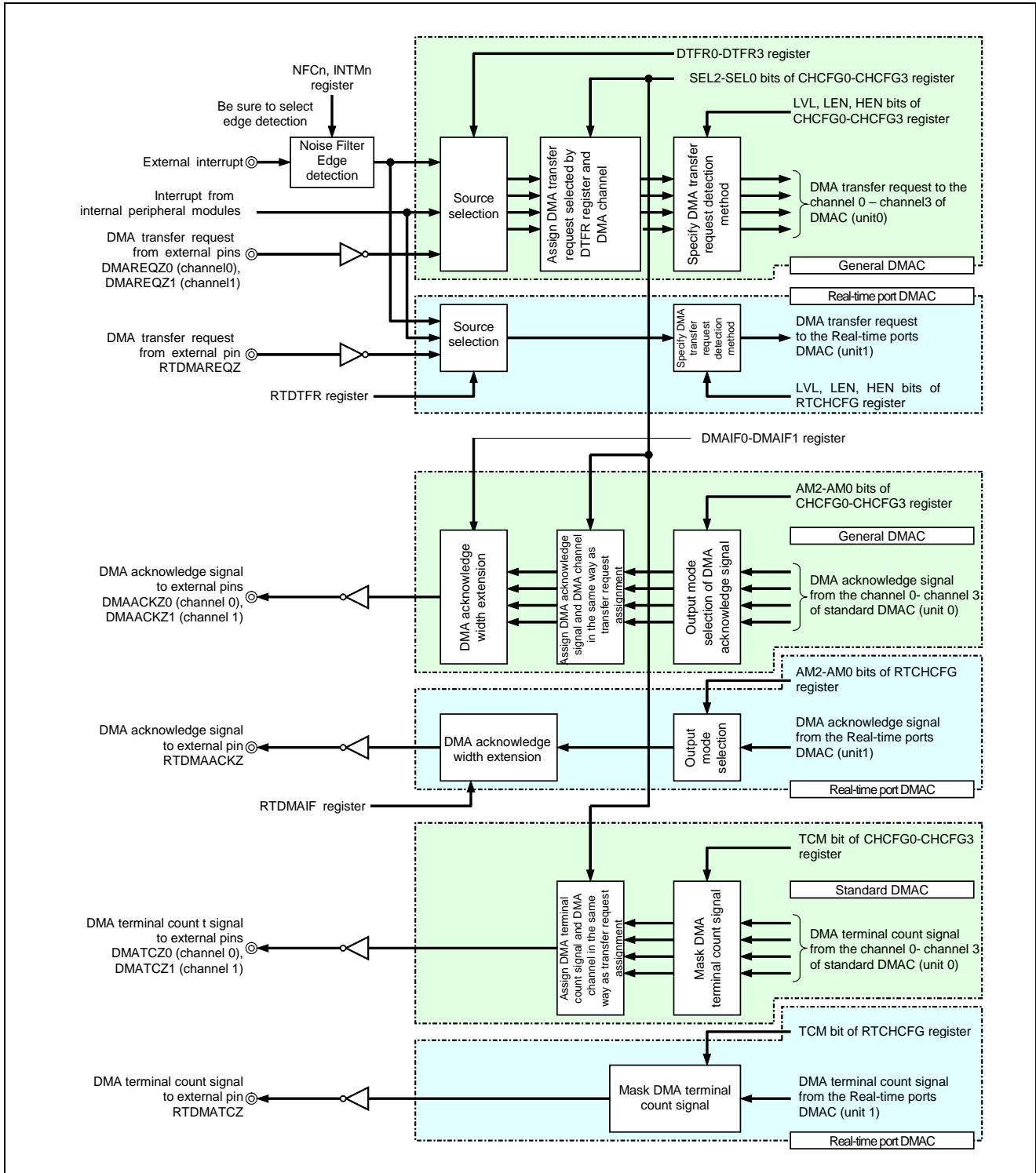


Figure 13.1 Relation between DMA Units/Channels and DMA Triggers

13.3 Terms and Definition

The terms used for the DMA controller are defined below.

Table 13.4 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 $[(\log_2\text{SIZE}-1): 0]$ 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 $[(\log_2\text{SIZE}-1): 0]$ 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.

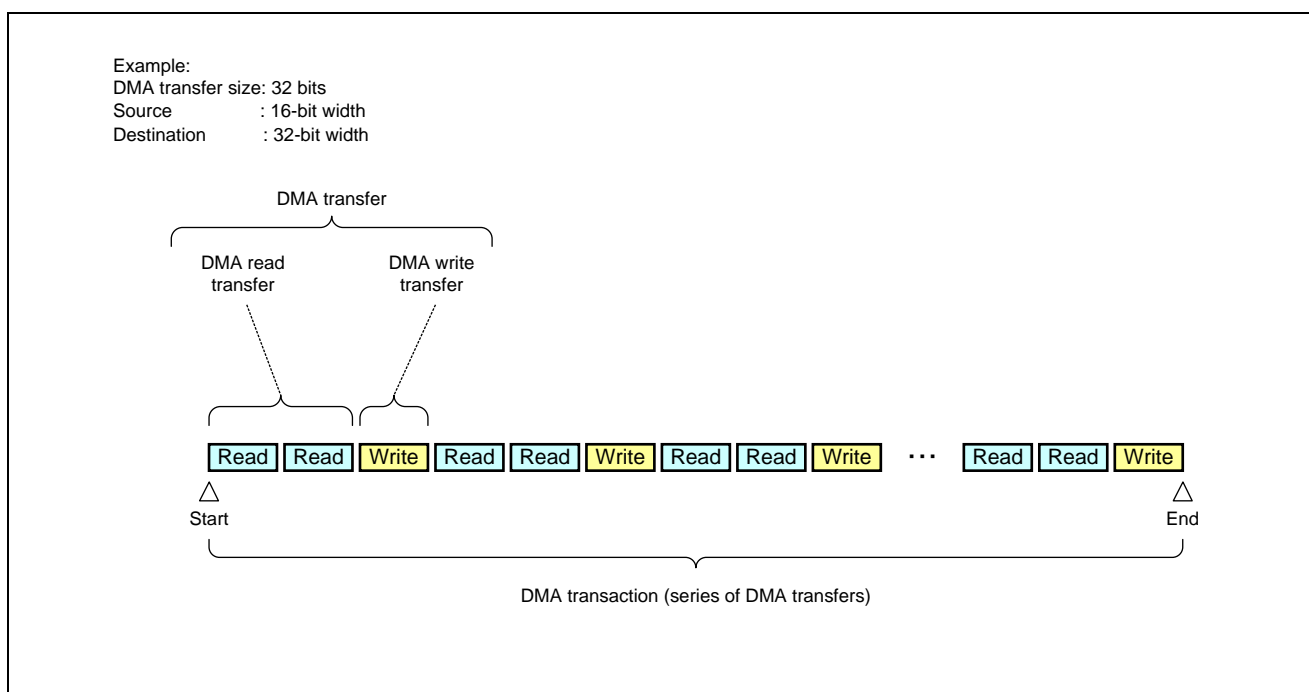


Figure 13.2 Name of Transfers

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.

13.4 DMA Controller Registers

13.4.1 Register Configuration

An R-IN32M3 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 13.5 DMA Controller Register Configuration

Register	Function
Next register set	<p>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.</p> <p>In register mode, set this register set using software.</p> <p>In link mode, the descriptor read data is automatically set in the Next 0 register set.</p> <p>The values of these register sets are loaded to Current register set and used for DMA transfer.</p>
Current register set	<p>This register set shows the source address, destination address, and the number of transfer bytes of the currently executed DMA transaction.</p> <p>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.</p> <p>The register set is automatically updated each time a DMA transaction is executed.</p>
Channel register set	<p>This register set is used to make DMA transfer settings.</p> <p>The settings made in this register set include the channel status indication, channel control, DMA transaction setting, and DMA transaction interval.</p>
Link register set	<p>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.</p> <p>The Current link address register is automatically updated at the time of descriptor read and cannot be written directly.</p>
DMA control register	<p>This register consists of the register for controlling the entire DMA unit and the register for indicating the status of each channel.</p> <p>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.</p>
DMA interface register	<p>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.</p>

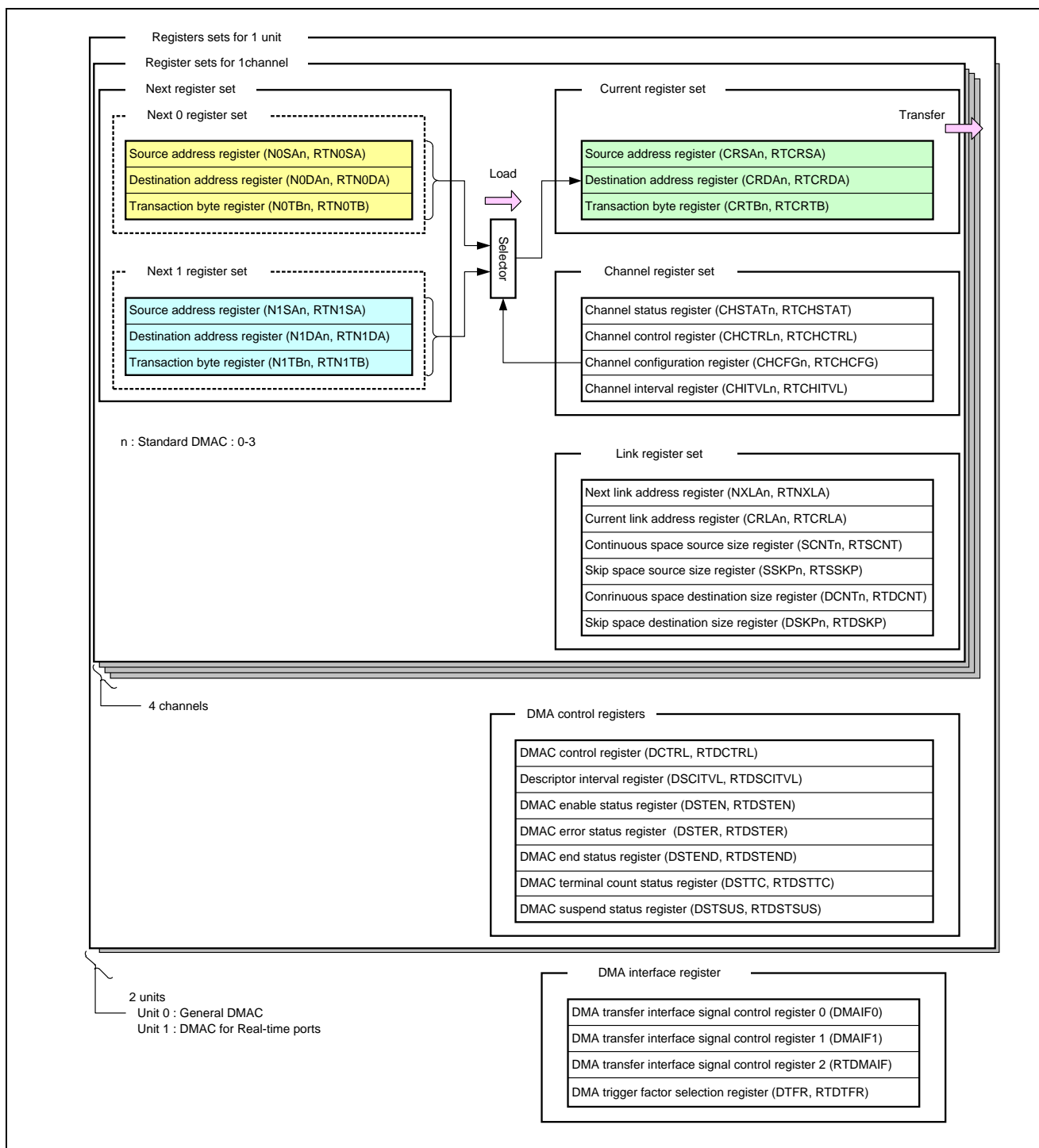


Figure 13.3 Register Block Diagram of DMA

Remark: n: General-purpose DMAC: 0 to 3

13.4.2 Control Register Outline

Table 13.6 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	RTC RSA	400A 2C18H
RTDMAC Current destination address register	RTC RDA	400A 2C1CH
RTDMAC Current transaction byte register	RTC RTB	400A 2C20H
RTDMAC Channel status register	RTC HSTAT	400A 2C24H
RTDMAC Channel control register	RTC HCTRL	400A 2C28H
RTDMAC Channel configuration register	RTC HCFG	400A 2C2CH
RTDMAC Channel interval register	RTC HITVL	400A 2C30H
RTDMAC Next link address register	RTNXLA	400A 2C38H
RTDMAC Current link address register	RTC RLA	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 selection register 0	DTFR0	4001 0730H
DMA trigger source selection register 1	DTFR1	4001 0734H
DMA trigger source selection register 2	DTFR2	4001 0738H
DMA trigger source selection register 3	DTFR3	4001 073CH
DMA trigger source selection register 4	RTDTFR	4001 0740H

13.4.3 General DMA Controller Register Set

13.4.3.1 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 13.7.4, Write-Only Mode).

- Access These registers can be read or written in units of 32 bits.

N0SAn	31		0	Address	Initial value
	<div style="border: 1px solid black; padding: 2px; text-align: center;"> Normal mode: Source address Write-only mode: Write data </div>			400A 2800H + 40H × n	0000 0000H
R/W				R/W	
N1SAn	31		0	Address	Initial value
	<div style="border: 1px solid black; padding: 2px; text-align: center;"> Normal mode: Source address Write-only mode: Write data </div>			400A 280CH + 40H × n	0000 0000H
R/W				R/W	

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.

N0DAn	31	0	Address	Initial value
	Destination address		400A 2804H + 40H × n	0000 0000H
	R/W			
N1DAn	31	0	Address	Initial value
	Destination address		400A 2810H + 40H × n	0000 0000H
	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.

N0TBn	31	0	Address	Initial value
	Transaction byte		400A 2808H + 40H × n	0000 0000H
R/W		R/W		
N1TBn	31	0	Address	Initial value
	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

13.4.3.2 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 (CRSA_n)

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
CRSA _n	Source address	400A 2818H + 40H × n	0000 0000H
R/W	R		
Bit Position	Bit Name	Description	
31 to 0	CRSA31- CRSA0	<p>Current source address registers.</p> <p>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).</p> <p>The initial value is loaded from one of the following registers.</p> <p style="padding-left: 20px;">In register mode: Loads the source address from NOSAn / N1SAn.</p> <p style="padding-left: 20px;">In link mode: Loads the source address from the descriptor. (The descriptor read data is assigned to the NOSAn register and, at the time of transfer, assigned to the CRSA_n register.)</p> <p>This register is updated when the read operation for the DMA transfer is completed.</p> <p>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.</p>	

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.

CRDAn	31	0	Address	Initial value
	Destination address		400A 281CH + 40H × n	0000 0000H
R/W	R			
Bit Position	Bit Name	Description		
31 to 0	CRDA31- CRDA0	<p>Current destination address registers.</p> <p>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).</p> <p>The initial value is loaded from one of the following registers.</p> <p style="padding-left: 20px;">In register mode: Loads the destination address from N0DAn / N1DAn.</p> <p style="padding-left: 20px;">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.)</p> <p>This register is updated when the write operation for the DMA transfer is completed.</p> <p>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.</p>		

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	Transaction byte data	0	Address	Initial value
CRTBn			400A 2820H + 40H × n	0000 0000H
R/W		R		

Bit Position	Bit Name	Description
31 to 0	CRTB31-CRTB0	<p>Current transaction byte registers.</p> <p>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.</p> <p>The initial value is loaded from one of the following registers.</p> <p style="margin-left: 20px;">In register mode: Loads the number of transfer bytes from N0TBn/N1TBn.</p> <p style="margin-left: 20px;">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.)</p> <p>This register is updated when the write operation for the DMA transfer is completed.</p> <p>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.</p>

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 n (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)

CHSTAT n	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 2824H + 40H x 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 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.						
		<table border="1"> <tr> <th>Increment condition</th> <th>Decrement condition</th> <th>Condition for clearing this bit to 0</th> </tr> <tr> <td> <ul style="list-style-type: none"> • The DMA read transfer is completed. </td> <td> <ul style="list-style-type: none"> • The DMA write transfer is completed. </td> <td> <ul style="list-style-type: none"> • CHSTATn.EN bit clearing condition. • The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.) </td> </tr> </table>	Increment condition	Decrement condition	Condition for clearing this bit to 0	<ul style="list-style-type: none"> • The DMA read transfer is completed. 	<ul style="list-style-type: none"> • The DMA write transfer is completed. 	<ul style="list-style-type: none"> • CHSTATn.EN bit clearing condition. • The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.)
		Increment condition	Decrement condition	Condition for clearing this bit to 0				
<ul style="list-style-type: none"> • The DMA read transfer is completed. 	<ul style="list-style-type: none"> • The DMA write transfer is completed. 	<ul style="list-style-type: none"> • CHSTATn.EN bit clearing condition. • The CHCTRLn.SWRST bit is set to 1. (The channel status register (CHSTATn, i.e., this register) is cleared.) 						
Reserved. These bits return 0 when read.								
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.				
				<table border="1"> <tr> <th>Condition for setting this bit to 1</th> <th>Condition for clearing this bit to 0</th> </tr> <tr> <td> <ul style="list-style-type: none"> • The CHCTRLn.SETSSWPRQ bit is set to 1. </td> <td> <ul style="list-style-type: none"> • 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.) </td> </tr> </table>	Condition for setting this bit to 1	Condition for clearing this bit to 0	<ul style="list-style-type: none"> • The CHCTRLn.SETSSWPRQ bit is set to 1. 	<ul style="list-style-type: none"> • 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.)
Condition for setting this bit to 1	Condition for clearing this bit to 0							
<ul style="list-style-type: none"> • The CHCTRLn.SETSSWPRQ bit is set to 1. 	<ul style="list-style-type: none"> • 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
		<ul style="list-style-type: none"> The CHCTRLn.SETDMARQM bit is set to 1. 	<ul style="list-style-type: none"> 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
		<ul style="list-style-type: none"> The CHCTRLn.SETINTM bit is set to 1. 	<ul style="list-style-type: none"> 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
		<ul style="list-style-type: none"> 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)). 	<ul style="list-style-type: none"> 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)

Bit Position	Bit Name	Description	
9	DW	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.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> The writeback of the header starts in link mode. 	<ul style="list-style-type: none"> 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.)
8	DL	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.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> The descriptor is being loaded in link mode. 	<ul style="list-style-type: none"> 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.)
7	SR	Indicates the register set selected in register mode. 0: Next 0 register set. 1: Next 1 register set	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> CHCFGn.RSEL is set to 1. (when Next 1 register set is selected) 	<ul style="list-style-type: none"> CHCFGn.RSEL is set to 0. (when Next 1 register set is selected)
6	TC	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).	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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.)

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 INTDMA _n occurs.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> The condition for setting the TC bit to 1 and the following condition are met: CHCFG_n.DEM = 0 (INTDMA_n) DMA transfer completion interrupt output is enabled) The following condition are all met in link mode: <ul style="list-style-type: none"> The LV bit of the descriptor header is set to 0 (descriptor disabled). CHCFG_n.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.) CHCFG_n.DIM is set to 0. (When the LV bit of the descriptor header is set to 0, the descriptor error interrupt (INTDMA_n) is enabled.) 	<ul style="list-style-type: none"> The CHCTRL_n.CLREND bit is set to 1. (The END_n bit, i.e., this bit is cleared.) The CHCTRL_n.SWRST bit is set to 1. (The channel status register (CHSTAT_n, i.e., this register) is cleared.)
4	ER ^{Note1}	This bit is set to 1 when a transfer error ^{Note2} occurs during DMA transfer and the INTDMAERR0 interrupt occurs.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> A DMA transfer error occurs ^{Note2} 	<ul style="list-style-type: none"> The CHCTRL_n.SWRST_n bit is set to 1. (The channel status register (CHSTAT_n, 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
		<ul style="list-style-type: none"> During the DMA transaction (the series of DMA transfers) for DMA channel n, the CHCTRL_n.SETSUS bit is set to 1 and the DMA transaction for DMA channel n is suspended. 	<ul style="list-style-type: none"> The CHCTRL_n.CLRSUS bit is set to 1. (Release from the suspended state) The CHCTRL_n.CLREN bit is set to 1. The condition for clearing CHSTAT_n.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
		<ul style="list-style-type: none"> The CHCTRL_n.SETEN bit is set to 1. (The system waits for the start of descriptor read or DMA trigger.) 	<ul style="list-style-type: none"> The CHSTAT_n.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
		<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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 (INTDMA_n) 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)

Bit Position	Bit Name	Description	
0	EN	Indicates whether the operation of DMA channel n is enabled or disabled. 0: Operation disabled. 1: Operation enabled	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> CHCTRLn.SETEN is set to 1. 	<ul style="list-style-type: none"> 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.

Note: A bus error occurs during access to an undefined area, etc.

Remark: n = 0 to 3

- Cautions**
- If transfer proceeds while the ER bit is set to 1, use processing to handle the series of associated DMA transfers as invalid.
 - 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 13.8.13, Suspending Transfer).
 - 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 CHCTRLn.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.
 - 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)

CHCTRLn	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
	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	400A 2828H + 40H x n Initial value 0000 0000H
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 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 INTDMA _n output. When this bit is set to 1, the temporary mask status is set for INTDMA _n output. This sets the CHSTAT _n .INTM (temporary mask status for INTDMA _n output) bit to 1. 0: Does not affect the operation. 1: Masks INTDMA _n 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 13.8.7, Forced Dumping). Note that, when CHCFG _n .REQD is set to 1 and DMAACKZ _p 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 CHCFG _n .RSEL bit after a DMA transaction (the series of DMA transfers) is completed in register mode. This sets the CHCFG _n .REN bit to 1. For details, see the description of the REN bit of the channel configuration register (CHCFG _n). 0: Does not affect the operation. 1: Sets CHCFG _n .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 <i>n</i> . If this bit is set to 1 when CHSTAT _n .SUS is set to 1, DMA channel <i>n</i> 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 <i>n</i> . If this bit is set to 1 when CHSTAT _n .EN is set to 1 (the operation of DMA channel <i>n</i> is enabled), the active DMA channel <i>n</i> 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 CHSTAT _n .DER (descriptor error) bit is cleared to 0. 0: Does not affect the operation. 1: CHSTAT _n .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 n (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 13.8.13, 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	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	
	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- SELO		400A 282CH + 40H x n							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	R/W	Initial value
																																		0000 0000H

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
		<ul style="list-style-type: none"> This bit is set to 1. The CHCTRLn.SETREN bit is set to 1. 	<ul style="list-style-type: none"> 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)

Bit Position	Bit Name	Description				
28	RSEL	<p>Selects the Next register set to be used for the next DMA transfer.</p> <p>This setting is valid only in register mode.</p> <p>When RSW is set to 1, the bit is automatically inverted upon the completion of a DMA transaction (the series of DMA transfers).</p> <p>0: Uses the Next 0 register set (initial value). 1: Uses the Next 1 register set</p>				
27	SBE	<p>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 DMAACKZ is output at the time of writing, this bit cannot be set to 1.</p> <p>0: Stops the transfer without dumping (writing) the buffer data (initial value). 1: Stops the transfer after dumping (writing) the buffer data.</p>				
26	DIM	<p>Selects how the descriptor error interrupt (INTDMAERR0) behaves if the LV bit of the descriptor header is set to 0 in link mode.</p> <p>0: Does not mask INTDMAERR0 (initial value). 1: Masks INTDMAERR0.</p>				
25	TCM	<p>Masks terminal count output (DMATCZp).</p> <p>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.</p> <p>Use this bit when controlling DMA transfers by software.</p> <p>0: Does not mask (enables terminal count output (DMATCZp); initial value). 1: Masks (disables terminal count output (DMATCZp).</p>				
		<table border="1"> <thead> <tr> <th>Condition for setting this bit to 1</th> <th>Condition for clearing this bit to 0</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> This bit is set to 1. </td> <td> <ul style="list-style-type: none"> 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> </tbody> </table>	Condition for setting this bit to 1	Condition for clearing this bit to 0	<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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.
		Condition for setting this bit to 1	Condition for clearing this bit to 0			
<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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. 					
<table border="1"> <thead> <tr> <th>Condition for setting this bit to 1</th> <th>Condition for clearing this bit to 0</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> This bit is set to 1. </td> <td> <ul style="list-style-type: none"> 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> </tbody> </table>	Condition for setting this bit to 1	Condition for clearing this bit to 0	<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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. 		
Condition for setting this bit to 1	Condition for clearing this bit to 0					
<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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. 					
24	DEM	<p>Selects how INTDMA_n behaves when a DMA transaction (the series of DMA transfers) is completed.</p> <p>If this bit is set to 1 when INTDMA_n occurs, INTDMA_n 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.</p> <p>0: Does not mask (enable INTDMA_n output, initial value). 1: Masks (disables INTDMA_n output).</p>				
		<table border="1"> <thead> <tr> <th>Condition for setting this bit to 1</th> <th>Condition for clearing this bit to 0</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> This bit is set to 1. </td> <td> <ul style="list-style-type: none"> 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> </tbody> </table>	Condition for setting this bit to 1	Condition for clearing this bit to 0	<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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.
		Condition for setting this bit to 1	Condition for clearing this bit to 0			
<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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. 					
<table border="1"> <thead> <tr> <th>Condition for setting this bit to 1</th> <th>Condition for clearing this bit to 0</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> This bit is set to 1. </td> <td> <ul style="list-style-type: none"> 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> </tbody> </table>	Condition for setting this bit to 1	Condition for clearing this bit to 0	<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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. 		
Condition for setting this bit to 1	Condition for clearing this bit to 0					
<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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. 					

Remark: n = 0 to 3; p = 0, 1

(3/7)

Bit Position	Bit Name	Description
23	WONLY	<p>Selects normal mode or write-only mode.</p> <p>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.</p> <p>0: Normal mode (initial value) 1: Write-only mode.</p>
22	TM	<p>Selects the DMA transfer mode.</p> <p>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).</p>
21	DAD	<p>Sets the counting direction of the destination address of DMA channel n.</p> <p>0: Increment (initial value) 1: Fixed</p> <p>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.</p>
20	SAD	<p>Sets the counting direction of the source address of DMA channel n.</p> <p>0: Increment (initial value) 1: Fixed</p> <p>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.</p>
19	DDS3	<p>Selects normal mode or skip mode for DMA destination addressing.</p> <p>0: Normal mode (initial value) 1: Skip mode</p>

Remark: n = 0 to 3

(4/7)

Bit Position	Bit Name	Description																																				
18 to 16	DDS2- DDS0	<p>Sets the transfer size of the DMA destination.</p> <table border="1"> <thead> <tr> <th>DDS2</th> <th>DDS1</th> <th>DDS0</th> <th>DMA Destination Transfer Size</th> </tr> </thead> <tbody> <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</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>256 bits</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>512 bits ^{Note}</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Setting prohibited</td> </tr> </tbody> </table> <p>Note: These bits can be set only when addresses are aligned in units of the transfer size.</p>	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
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																																			
15	SDS3	<p>Selects normal mode or skip mode for DMA source addressing.</p> <p>0: Normal mode (initial value) 1: Skip mode</p>																																				
14 to 12	SDS2- SDS0	<p>Sets the transfer size of the DMA source.</p> <table border="1"> <thead> <tr> <th>SDS2</th> <th>SDS1</th> <th>SDS0</th> <th>DMA Source Transfer Size</th> </tr> </thead> <tbody> <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</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>256 bits</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>512 bits ^{Note}</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Setting prohibited</td> </tr> </tbody> </table> <p>Note: These bits can be set only when addresses are aligned in units of the transfer size.</p>	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
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																																			

Remark: n = 0 to 3

(5/7)

Bit Position	Bit Name	Description																				
11	DRRP	<p>Selects the operation if the descriptor header is disabled (LV = 0) in link mode.</p> <p>0: Sets the CHSTATn.DER (descriptor error) bit to 1 and stops the operation (initial value).</p> <p>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).</p>																				
10 to 8	AM2-AM0	<p>Selects the output mode of the DMA acknowledge signal.</p> <table border="1"> <thead> <tr> <th>AM2</th> <th>AM1</th> <th>AM0</th> <th>DMA Acknowledge Signal (DMAACKZp) Output Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Pulse mode ^{Note1} (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 ^{Note2} 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> </tbody> </table> <p>Notes 1. A pulse of one BUSCLK cycle is output as the DMAACKZp signal.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Remark: X: Don't Care</p>	AM2	AM1	AM0	DMA Acknowledge Signal (DMAACKZp) Output Mode	0	0	0	Pulse mode ^{Note1} (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 ^{Note2} The active level is maintained during a DMA transfer bus cycle.	1	X	X	DMA acknowledge signal (DMAACKZp) output disabled
AM2	AM1	AM0	DMA Acknowledge Signal (DMAACKZp) Output Mode																			
0	0	0	Pulse mode ^{Note1} (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 ^{Note2} The active level is maintained during a DMA transfer bus cycle.																			
1	X	X	DMA acknowledge signal (DMAACKZp) output disabled																			
7	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.																				

Remark: n = 0 to 3; p = 0, 1

(6/7)

Bit Position	Bit Name	Description																																																																																			
6	LVL	<p>Selects the method of detection of the DMA transfer request signal.</p> <p>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.</p> <p>[When the DMA transfer request signal is a DMA request signal of an external pin]</p> <p>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.</p> <table border="1"> <thead> <tr> <th colspan="3"></th> <th colspan="2">Detection Method of DMA Transfer Request Signal (DMAREQZp)</th> </tr> <tr> <th>LVLn</th> <th>HENn</th> <th>LENn</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td rowspan="4">Edge detection</td> <td>Detection disabled</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Falling edge detection</td> <td>Rising edge detection</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Rising edge detection</td> <td>Falling edge detection</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td colspan="2">Rising/falling edge detection (Does not recommend)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td rowspan="4">Level detection</td> <td>Detection disabled</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Low level detection</td> <td>High level detection</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>High level detection</td> <td>Low level detection</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td colspan="2">Setting prohibited</td> </tr> </tbody> </table> <p>[When DMA transfer request signal is an interrupt signal (The signal which starts with INT).]</p> <table border="1"> <thead> <tr> <th colspan="3"></th> <th colspan="2">Detection Procedure of the DMA Transfer Request Signal by an Interrupt Signal</th> </tr> <tr> <th>LVLn</th> <th>HENn</th> <th>LENn</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td rowspan="3">Edge detection</td> <td>Detection disabled</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Low level detection</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>High level detection</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td></td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>x</td> <td>x</td> <td>Level detection</td> <td>Setting prohibited</td> </tr> </tbody> </table>				Detection Method of DMA Transfer Request Signal (DMAREQZp)		LVLn	HENn	LENn			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					Detection Procedure of the DMA Transfer Request Signal by an Interrupt Signal		LVLn	HENn	LENn			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	x	x	Level detection	Setting prohibited
			Detection Method of DMA Transfer Request Signal (DMAREQZp)																																																																																		
LVLn	HENn		LENn																																																																																		
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																																																																																	
			Detection Procedure of the DMA Transfer Request Signal by an Interrupt Signal																																																																																		
LVLn	HENn	LENn																																																																																			
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	x	x	Level detection	Setting prohibited																																																																																	
5	HEN																																																																																				
4	LEN																																																																																				
3	REQD	<p>Selects when DMAACKZp is to become active.</p> <p>Usually, set this bit so that DMAACKZp is output to the side on which DMAREQZp is asserted.</p> <p>0: DMAACKZp is active when reading (DMAREQZp is the source). 1: DMAACKZp is active when writing (DMAREQZp is the destination).</p>																																																																																			

Remark: n = 0 to 3; p = 0, 1

(7/7)

Bit Position	Bit Name	Description																								
2 to 0	SEL2-SEL0	<p>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.</p> <table border="1"> <thead> <tr> <th>SEL2</th> <th>SEL1</th> <th>SEL0</th> <th>DMA Trigger Selection</th> </tr> </thead> <tbody> <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> </tbody> </table>	SEL2	SEL1	SEL0	DMA Trigger Selection	0	0	0	The DMA transfer source selected by DTFR0 is chosen.	0	0	1	The DMA transfer source selected by DTFR1 is chosen.	0	1	0	The DMA transfer source selected by DTFR2 is chosen.	0	1	1	The DMA transfer source selected by DTFR3 is chosen.	Other than the above			Setting prohibited
SEL2	SEL1	SEL0	DMA Trigger Selection																							
0	0	0	The DMA transfer source selected by DTFR0 is chosen.																							
0	0	1	The DMA transfer source selected by DTFR1 is chosen.																							
0	1	0	The DMA transfer source selected by DTFR2 is chosen.																							
0	1	1	The DMA transfer source selected by DTFR3 is chosen.																							
Other than the above			Setting prohibited																							

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 13.8.9, Interval Counting.

CHITVLn	31	16	15	0	Address 400A 2830H + 40H × n	Initial value 0000 0000H	
	0		ITVL15-ITVL0				
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 the link mode, see section 13.7.3, Link Mode.

NXLAN	31		2 1 0		Address 400A 2838H + 40H × n	Initial value 0000 0000H
	NXLA31-NXLA2					
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.

CRLAN	31		0		Address 400A 283CH + 40H × n	Initial value 0000 0000H
	CRLA31-CRLA0					
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.

SCNTn	31	0	Address	Initial value
	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.

SSKPn	31	0	Address	Address
	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

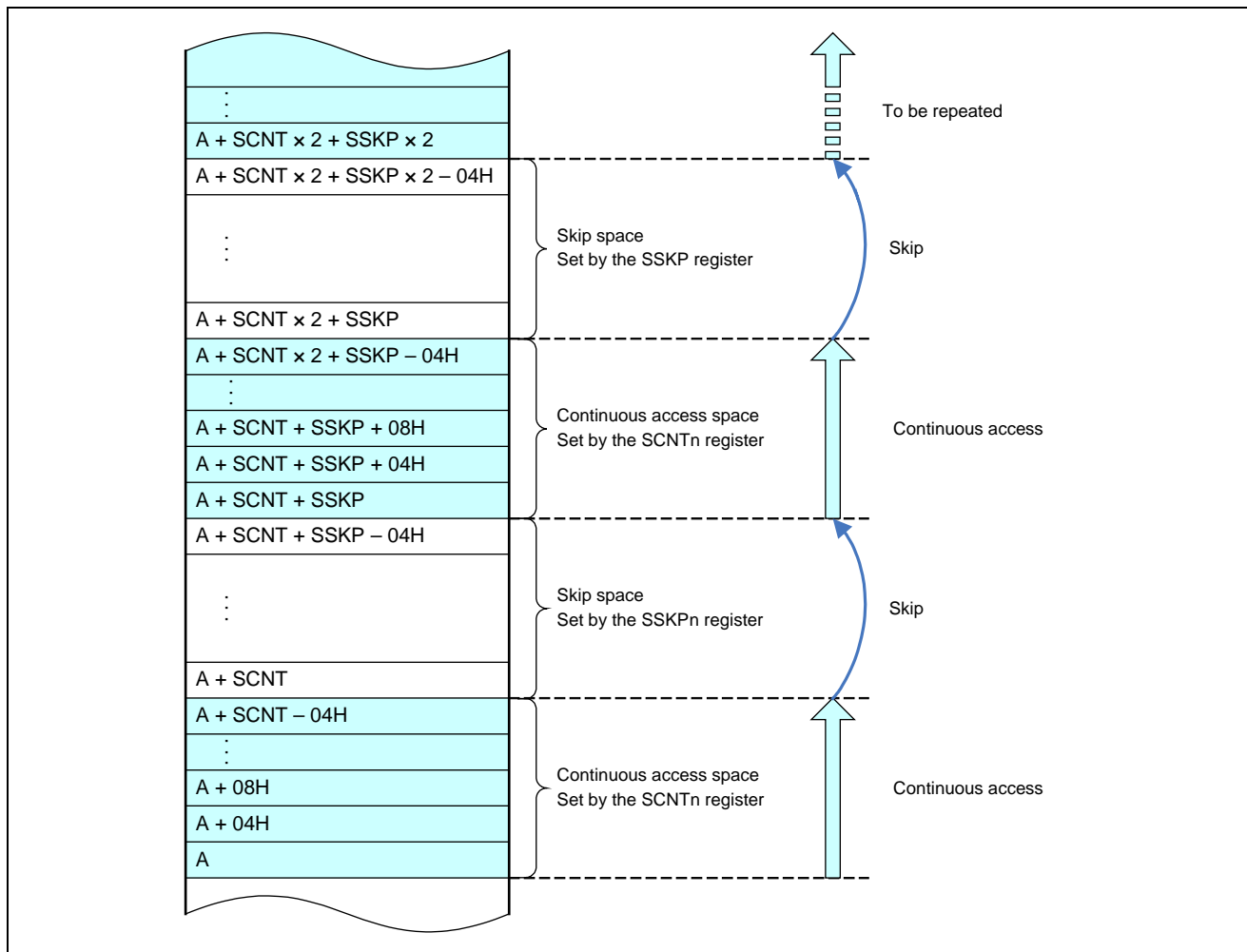


Figure 13.4 Relationship between the SSKPn and SCNTn Registers in Skip Mode

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.

DCNTn	31	0	Address	Initial value
	DCNT31-DCNT0		400A 2A08H + 20H × n	0000 0000H
R/W	R/W			
Bit Position	Bit Name	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.

DSKPn	31	0	Address	Initial value
	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

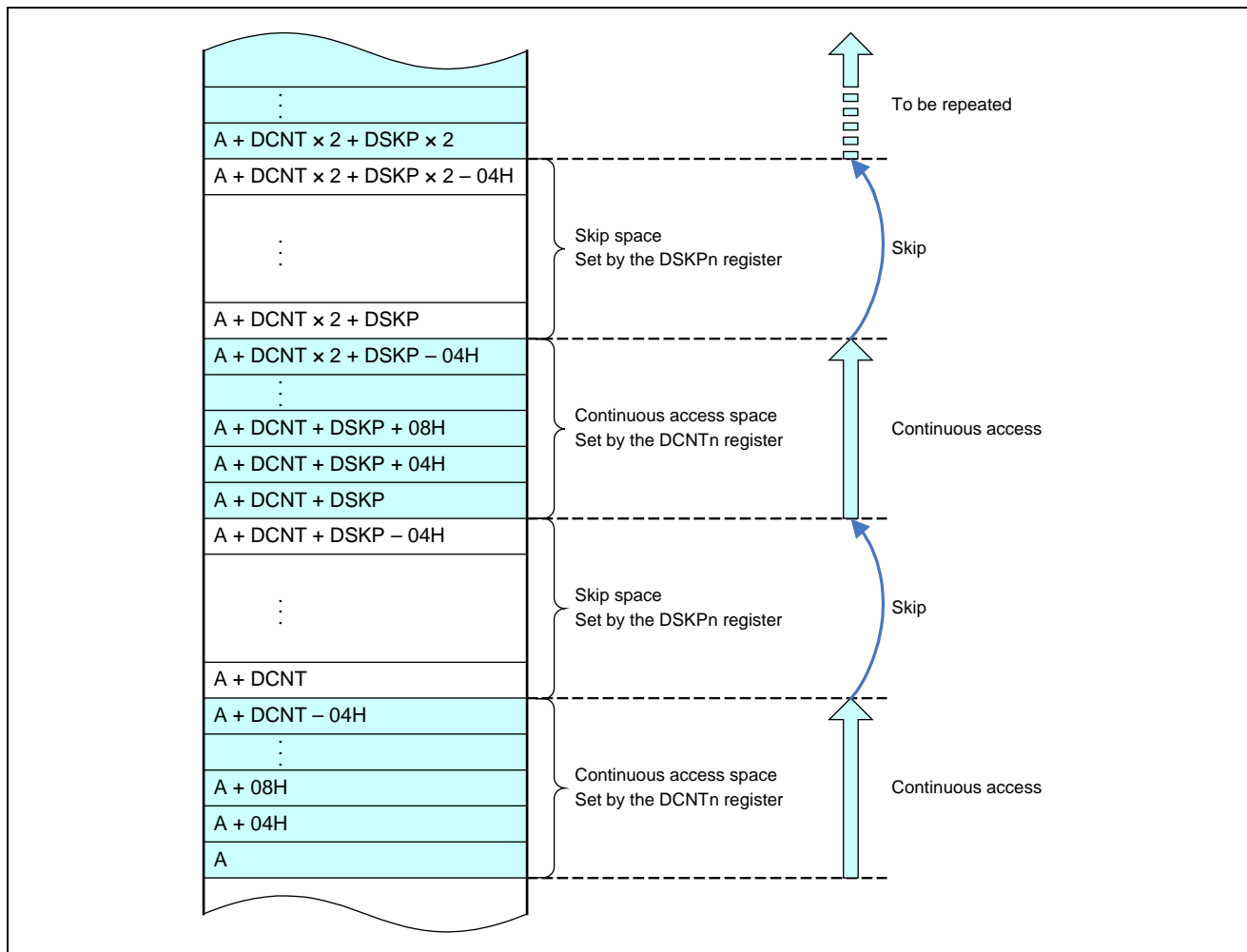


Figure 13.5 Relationship between the DSKPn and DCNTn Registers in Skip Mode

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 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
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 13.8.2, 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 \times 256.

- Access The register can be read or written in units of 32 bits.

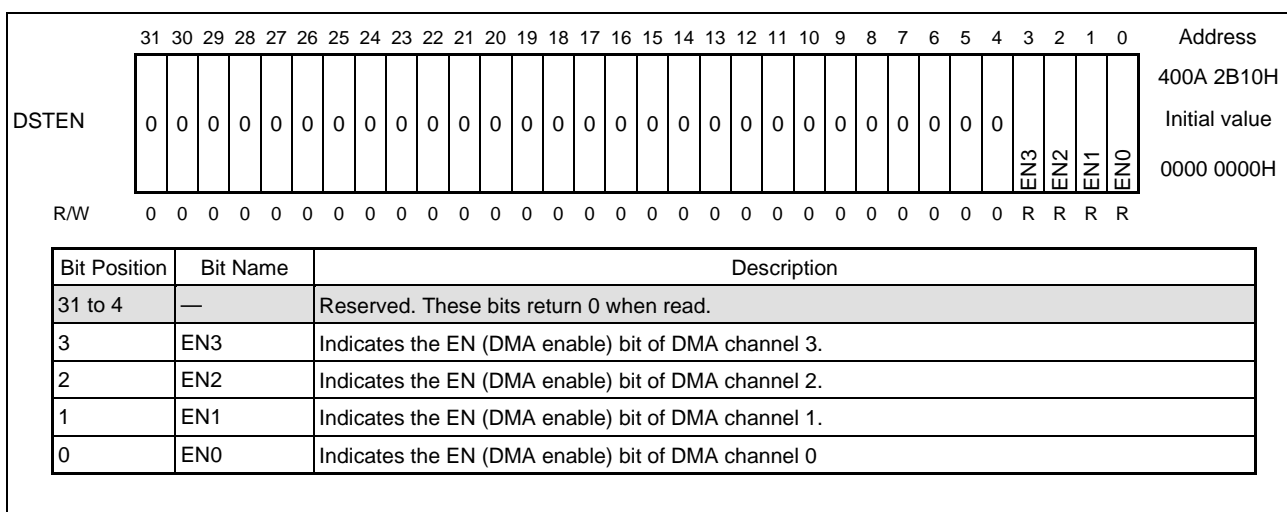
	31		16	15		8	7	0	Address	Initial value
DSCITVL	0			DITVL15-DITVL8		0			400A 2B04H	0000 0000H
R/W	0			R/W		0				
Bit Position	Bit Name	Description								
31 to 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.								
15 to 8	DITVL15-DITVL8	Sets the interval at which the descriptor header continues to be read until the LV bit becomes 1. The descriptor is read in the (DITVL15-DITVL8 value) \times 256 \times 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.



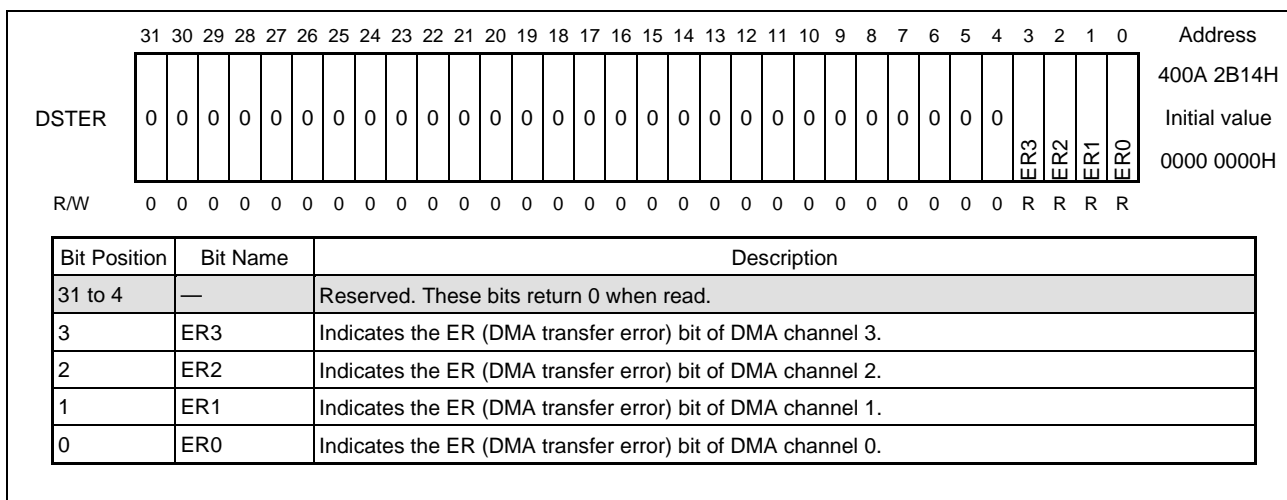
Remarks 1. The EN bit of each DMA Channel is the 0th bit of the channel status register n (CHSTATn).
2. n = 0 to 3

(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.

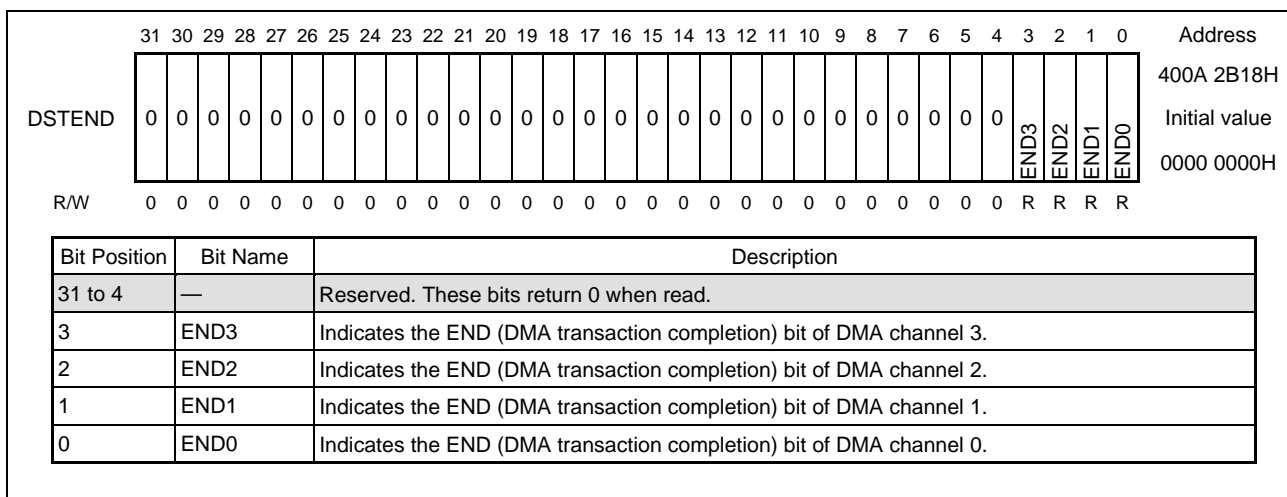


Remarks 1. The ER bit of each DMA channel is the 4th bit of the channel status register n (CHSTATn).
2. n = 0 to 3

(e) DMAC end status register (DSTEND)

This register indicates the state of the END bit (indicating generation of INTDMA_n 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 n (CHSTAT_n).



Remarks 1. The END bit of each DMA channel is the 5th bit of the channel status register n (CHSTAT_n).
2. n = 0 to 3

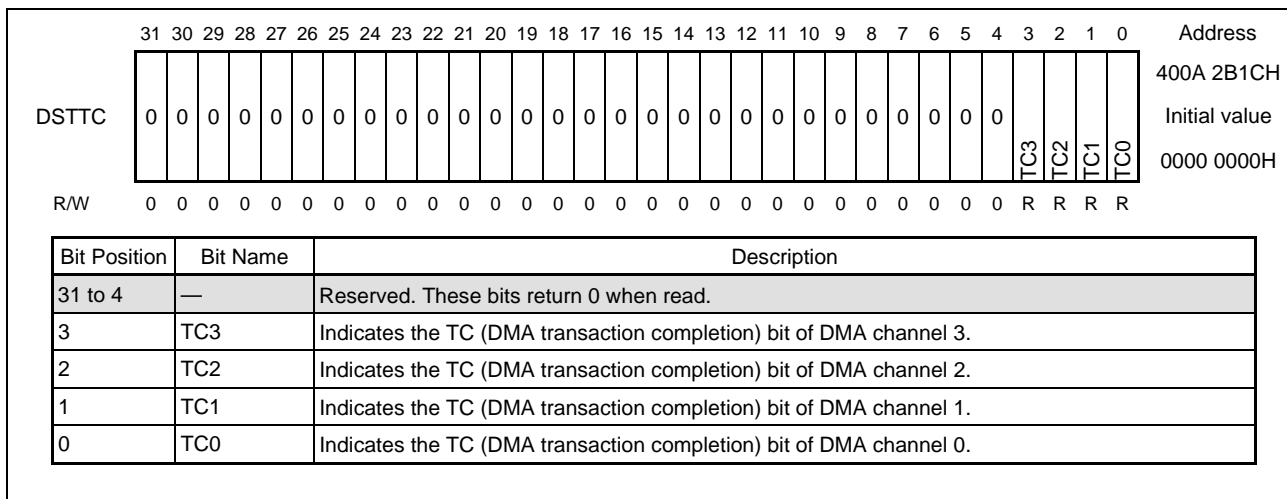
Table 13.7 Correspondence between DMA End Status Registers and Interrupt Signals

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 n (CHSTATn).



Remarks 1. The TC bit of each DMA channel is the 6th bit of the channel status register n (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
DSTSUS	0 0	400A 2B20H
	R/R/R/R	Initial value
		0000 0000H
R/W	0 0	

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 n (CHSTATn).

2. n = 0 to 3

13.4.4 Register Set of DMA Controller for Real-Time Ports

13.4.4.1 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 13.7.4, Write-Only Mode).

- Access These registers can be read or written in units of 32 bits.

RTN0SA	31	0	Address	Initial value
	<div style="border: 1px solid black; padding: 2px; text-align: center;"> Normal mode: Source address Write-only mode: Write data </div>		400A 2C00H	0000 0000H
R/W	R/W			
RTN1SA	31	0	Address	Initial value
	<div style="border: 1px solid black; padding: 2px; text-align: center;"> Normal mode: Source address Write-only mode: Write data </div>		400A 2C0CH	0000 0000H
R/W	R/W			

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.

RTN0DA is for the Next 0 register set, and RTN1DA is for the Next 1 register set.

- Access These registers can be read or written in units of 32 bits.

RTN0DA	31	0	Address	Initial value
	Destination address		400A 2C04H	0000 0000H
R/W		R/W		
RTN1DA	31	0	Address	Initial value
	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.

RTN0TB	31	0	Address	Initial value
	Transaction byte		400A 2C08H	0000 0000H
R/W		R/W		
RTN1TB	31	0	Address	Initial value
	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.

13.4.4.2 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
Source address		400A 2C18H	0000 0000H
R/W	R		

Bit Position	Bit Name	Description
31 to 0	CRSA31-CRSA0	<p>Current source address register.</p> <p>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).</p> <p>The initial value is loaded from one of the following registers.</p> <p style="margin-left: 20px;">In register mode: Loads the source address from RTN0SA / RTN1SA.</p> <p style="margin-left: 20px;">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.)</p> <p>This register is updated when the read operation for the DMA transfer is completed.</p> <p>The register should be read when DMA is not in progress (when RTCHSTAT.TACT = 0).</p> <p>The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.</p>

(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	Destination address	0	Address	Initial value
RTCRDA			400A 2C1CH	0000 0000H
R/W		R		

Bit Position	Bit Name	Description
31 to 0	CRDA31- CRDA0	<p>Current destination address register.</p> <p>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).</p> <p>The initial value is loaded from one of the following registers.</p> <p style="padding-left: 20px;">In register mode: Loads the destination address from RTN0DA/RTN1DA.</p> <p style="padding-left: 20px;">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.)</p> <p>This register is updated when the write operation for the DMA transfer is completed.</p> <p>The register should be read when DMA is not in progress (when RTCHSTAT.TACT = 0).</p> <p>The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.</p>

(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	Transaction byte data	0	Address	Initial value
RTCRTB			400A 2C20H	0000 0000H
R/W		R		

Bit Position	Bit Name	Description
31 to 0	CRTB31-CRTB0	<p>Current transaction byte register.</p> <p>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.</p> <p>The initial value is loaded from one of the following registers.</p> <p style="padding-left: 20px;">In register mode: Loads the number of transfer bytes from RTN0TBA / RTN1TB.</p> <p style="padding-left: 20px;">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.)</p> <p>This register is updated when the write operation for the DMA transfer is completed.</p> <p>The register should be read when DMA is not in progress (when RTCHSTAT.TACT = 0).</p> <p>The value obtained during the DMA operation is a reference value and is not guaranteed to be valid.</p>

(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									
	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	
RTCHSTAT																											Address														
																										400A 2C24H															
																										Initial value															
																										0000 0000H															
R/W	R	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	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	Decrement condition	Condition for clearing this bit to 0																																					
		<ul style="list-style-type: none"> The DMA read transfer is completed. 	<ul style="list-style-type: none"> The DMA write transfer is completed. 	<ul style="list-style-type: none"> The RTCHSTAT.EN bit clearing condition. The RTCHCTRL.SWRST bit is set to 1. (Channel status register (RTCHSTAT, 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 RTCHCTRL.SETSSWPRQ bit is indicated. 0: Forced dump request not asserted. 1: Forced dump request asserted.																																							
		Condition for setting this bit to 1	Condition for clearing this bit to 0																																						
		<ul style="list-style-type: none"> The CHCTRLn.SETSSWPRQ bit is set to 1. 	<ul style="list-style-type: none"> Forced dumping clears all data from the buffer. RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.) 																																						

(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
		<ul style="list-style-type: none"> The RTCHCTRL.SETDMARQM bit is set to 1. 	<ul style="list-style-type: none"> The RTCHCTRL.SETDMARQM bit is set to 1. The RTCHCTRL.SWRST bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)
16	INTM	Indicates the temporary mask status of the INTRTDMA interrupt output. 0: Temporarily mask released. 1: Temporarily mask applied	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> The RTCHCTRL.SETINTM bit is set to 1. 	<ul style="list-style-type: none"> 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.)
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 RTCHCFG.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
		<ul style="list-style-type: none"> 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)). 	<ul style="list-style-type: none"> 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.)

(3/6)

Bit Position	Bit Name	Description	
9	DW	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.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> The writeback of the header starts in link mode. 	<ul style="list-style-type: none"> 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.)
8	DL	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.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> The descriptor is being loaded in link mode. 	<ul style="list-style-type: none"> 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.)
7	SR	Indicates the register set selected in register mode. 0: Next 0 register set. 1: Next 1 register set	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> RTCHCFG.RSEL is set to 1. (when Next 1 register set is selected) 	<ul style="list-style-type: none"> RTCHCFG.RSEL is set to 0. (when Next 1 register set is selected)
6	TC	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).	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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.)

**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)

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 INTRTDMA occurs.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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.) 	<ul style="list-style-type: none"> 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.)
4	ER ^{Note1}	This bit is set to 1 when a transfer error ^{Note2} occurs during DMA transfer and the INTRTDMAERR interrupt occurs.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> A DMA transfer error occurs ^{Note2} 	<ul style="list-style-type: none"> The RTCHCTRL.SWRSTn bit is set to 1. (The channel status register (RTCHSTAT, i.e., this register) is cleared.)
3	SUS	Indicates the suspend 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
		<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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.
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. 1: DMA is active.	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> The RTCHCTRL.SETEN bit is set to 1. (The system waits for the start of descriptor read or DMA trigger.) 	<ul style="list-style-type: none"> The RTCHSTAT.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.

(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
		<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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 (INTRTDMA) 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)

Bit Position	Bit Name	Description	
0	EN	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	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> RTCHCTRL.SETEN is set to 1. 	<ul style="list-style-type: none"> 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}

Note: A bus error occurs during access to an undefined area, etc.

- Cautions**
- 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.**
 - 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 13.8.13, Suspending Transfer).**
 - 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 RTCHCTRL.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.**
 - 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	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		
	0	0	0	0	0	0	0	0	0	0	0	0	CLRDMARQM	SETDMARQM	CLRINTM	SETINTM	0	SETSSWPRQ	0	SETREN	0	0	0	CLRSUS	SETSUS	CLRDER	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN	400A 2C28H	
	Initial value																																0000 0000H		
R/W	0	0	0	0	0	0	0	0	0	0	0	0	W	W	W	W	0	W	0	W	0	0	0	W	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 INTDMA 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 13.8.7, 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 13.8.13, 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.

(1/6)

RTCHCFG	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	
	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											
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	R/W	Initial value 0000 0000H

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 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).
		Condition for setting this bit to 1
		Condition for clearing this bit to 0
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.
28	RSEL	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
27	SBE	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.

(2/6)

Bit Position	Bit Name	Description	
26	DIM	Selects how the descriptor error interrupt (INTRDMAERR) behaves if the LV bit of the descriptor header is set to 0 in link mode. 0: Does not mask INTRDMAERR (initial value). 1: Masks INTRDMAERR.	
25	TCM	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).)	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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.
24	DEM	Selects how INTRDMA behaves when a DMA transaction (the series of DMA transfers) is completed. If this bit is set to 1 when INTRDMA occurs, INTRDMA 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 INTRDMA output, initial value). 1: Masks (disables INTRDMA output).	
		Condition for setting this bit to 1	Condition for clearing this bit to 0
		<ul style="list-style-type: none"> This bit is set to 1. 	<ul style="list-style-type: none"> 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.
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.	

Bit Position	Bit Name	Description																												
20	SAD	<p>Sets the counting direction of the source address of DMA channel n.</p> <p>0: Increment (initial value) 1: Fixed</p> <p>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.</p>																												
19	DDS3	<p>Selects normal mode or skip mode for DMA destination addressing.</p> <p>0: Normal mode (initial value) 1: Skip mode</p>																												
18 to 16	DDS2- DDS0	<p>Sets the transfer size of the DMA destination.</p> <table border="1"> <thead> <tr> <th>DDS2</th> <th>DDS1</th> <th>DDS0</th> <th>DMA destination transfer size</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Note: These bits can be set only when addresses are aligned in units of the transfer size.</p>	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 ^{Note}	Other than the above			Setting prohibited
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 ^{Note}																											
Other than the above			Setting prohibited																											
15	SDS3	<p>Selects normal mode or skip mode for DMA source addressing.</p> <p>0: Normal mode (initial value) 1: Skip mode</p>																												
14 to 12	SDS2- SDS0	<p>Sets the transfer size of the DMA source.</p> <table border="1"> <thead> <tr> <th>SDS2</th> <th>SDS1</th> <th>SDS0</th> <th>DMA source transfer size</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Note: These bits can be set only when addresses are aligned in units of the transfer size.</p>	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 ^{Note}	Other than the above			Setting prohibited
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 ^{Note}																											
Other than the above			Setting prohibited																											

(4/6)

Bit Position	Bit Name	Description																				
11	DRRP	<p>Selects the operation if the descriptor header is disabled (LV = 0) in link mode.</p> <p>0: Sets the RTCHSTAT.DER (descriptor error) bit to 1 and stops the operation (initial value).</p> <p>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).</p>																				
10 to 8	AM2-AM0	<p>Selects the output mode of the DMA acknowledge signal.</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>AM2</th> <th>AM1</th> <th>AM0</th> <th>DMA Acknowledge Signal (RTDMAACKZ) Output Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Pulse mode^{Note1} (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^{Note2} 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> </tbody> </table> <p>Notes 1. A pulse of one BUSCLK cycle is output as the RTDMAACKZ signal.</p> <p>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.</p> <p>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.</p> <p>Remark: X: Don't Care</p>	AM2	AM1	AM0	DMA Acknowledge Signal (RTDMAACKZ) Output Mode	0	0	0	Pulse mode ^{Note1} (initial value)	0	0	1	Level mode The active level is maintained until the DMA transfer request (RTDMAREQZ) becomes inactive.	0	1	X	Bus cycle mode ^{Note2} The active level is maintained during a DMA transfer bus cycle.	1	X	X	DMA acknowledge signal (RTDMAACKZ) output disabled.
AM2	AM1	AM0	DMA Acknowledge Signal (RTDMAACKZ) Output Mode																			
0	0	0	Pulse mode ^{Note1} (initial value)																			
0	0	1	Level mode The active level is maintained until the DMA transfer request (RTDMAREQZ) becomes inactive.																			
0	1	X	Bus cycle mode ^{Note2} The active level is maintained during a DMA transfer bus cycle.																			
1	X	X	DMA acknowledge signal (RTDMAACKZ) output disabled.																			
7	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.																				

Bit Position	Bit Name	Description																																																																				
6	LVL	<p>Selects the method of detecting a DMA transfer request signal.</p> <p>A DMA transfer request is chosen by using the DMA trigger source register 4 (RTDTRFR). The procedure for detecting a DMA transfer request signal differs with the selected DMA transfer request.</p> <p>[In the case where the DMA transfer request signals are the DMA request signals of the external pins]</p> <p>The internal DMA interface is positive logic.</p> <p>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 HEN and LEN bits is chosen.</p> <table border="1"> <thead> <tr> <th colspan="4">Detection Method of DMA Transfer Request Signal (RTDMAREQZ)</th> </tr> <tr> <th>LVL</th> <th>HEN</th> <th>LEN</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Edge detection</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Falling edge detection</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Rising edge detection</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Rising/falling edge detection (Does not recommend)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Level detection</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Low level detection</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>High level detection</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Setting prohibited</td> </tr> </tbody> </table> <p>[In the case where the DMA transfer request signals are interrupt signals (signals which start with INT).]</p> <table border="1"> <thead> <tr> <th colspan="4">Detection Procedure of the DMA Transfer Request Signal by an Interrupt Signal</th> </tr> <tr> <th>LVL</th> <th>HEN</th> <th>LEN</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Edge detection</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Low level detection</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>High level detection</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>x</td> <td>x</td> <td>Level detection</td> </tr> </tbody> </table>	Detection Method of DMA Transfer Request Signal (RTDMAREQZ)				LVL	HEN	LEN		0	0	0	Edge detection	0	0	1	Falling edge detection	0	1	0	Rising edge detection	0	1	1	Rising/falling edge detection (Does not recommend)	1	0	0	Level detection	1	0	1	Low level detection	1	1	0	High level detection	1	1	1	Setting prohibited	Detection Procedure of the DMA Transfer Request Signal by an Interrupt Signal				LVL	HEN	LEN		0	0	0	Edge detection	0	0	1	Low level detection	0	1	0	High level detection	0	1	1	Setting prohibited	1	x	x	Level detection
Detection Method of DMA Transfer Request Signal (RTDMAREQZ)																																																																						
LVL	HEN		LEN																																																																			
0	0	0	Edge detection																																																																			
0	0	1	Falling edge detection																																																																			
0	1	0	Rising edge detection																																																																			
0	1	1	Rising/falling edge detection (Does not recommend)																																																																			
1	0	0	Level detection																																																																			
1	0	1	Low level detection																																																																			
1	1	0	High level detection																																																																			
1	1	1	Setting prohibited																																																																			
Detection Procedure of the DMA Transfer Request Signal by an Interrupt Signal																																																																						
LVL	HEN	LEN																																																																				
0	0	0	Edge detection																																																																			
0	0	1	Low level detection																																																																			
0	1	0	High level detection																																																																			
0	1	1	Setting prohibited																																																																			
1	x	x	Level detection																																																																			
5	HEN																																																																					
4	LEN																																																																					
3	REQD	<p>Selects when RTDMAACKZ is to become active.</p> <p>Usually, set this bit so that RTDMAACKZ is output to the side on which RTDMAREQZ is asserted.</p> <p>0: Makes RTDMAACKZ active when reading (RTDMAREQZ is the source).</p> <p>1: Makes RTDMAACKZ active when writing (RTDMAREQZ is the destination).</p>																																																																				

(6/6)

Bit Position	Bit Name	Description												
2 to 0	SEL2-SEL0	<p>Selects the DMA interface signal for each channel.</p> <p>Since the DMA controller for real-time ports (unit 1) only has one channel, the only available setting is for RTDTFR.</p> <table border="1" data-bbox="485 506 1366 618"> <thead> <tr> <th>SEL2</th> <th>SEL1</th> <th>SEL0</th> <th>DMA interface signal selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>The DMA transfer source selected by RTDTFR is chosen.</td> </tr> <tr> <td colspan="3">Other than the above</td> <td>Setting prohibited</td> </tr> </tbody> </table>	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
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 13.8.9, Interval Counting.

	31		16 15		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 the link mode, see section 13.7.3, Link Mode.

RTNXLA	31	NXLA31-NXLA2	2 1 0	Address	Initial value
			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.

RTCRLA	31	CRLA31-CRLA0	0	Address	Initial value
				200A 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 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
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	200A 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.	

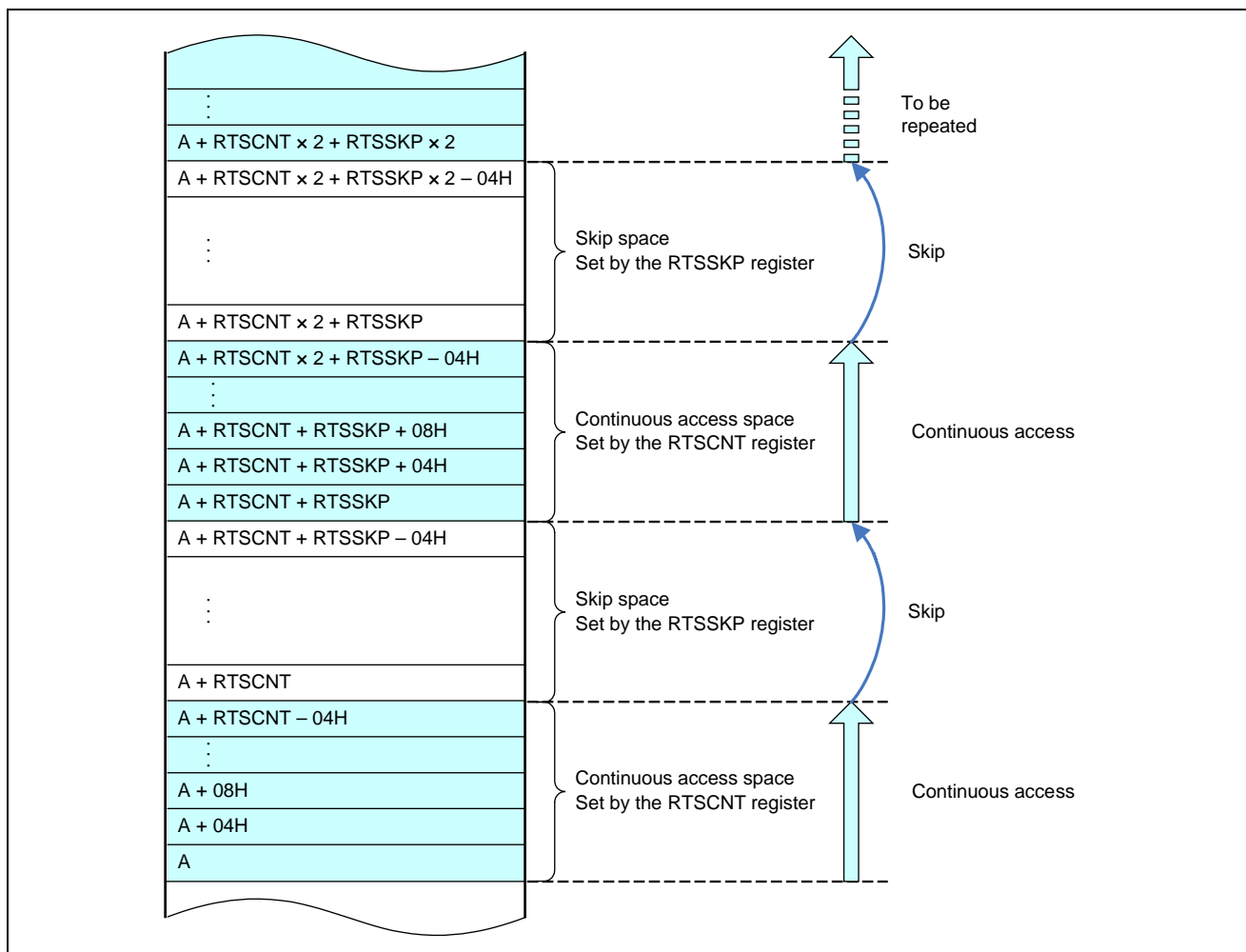


Figure 13.6 Relationship between the RTSSKP and RTSCNT Registers in Skip Mode

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
DCNT31-DCNT0		400A 2E08H	0000 0000H
R/W	R/W		
Bit Position	Bit Name	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
DSKP31-DSKP0		200A 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.	

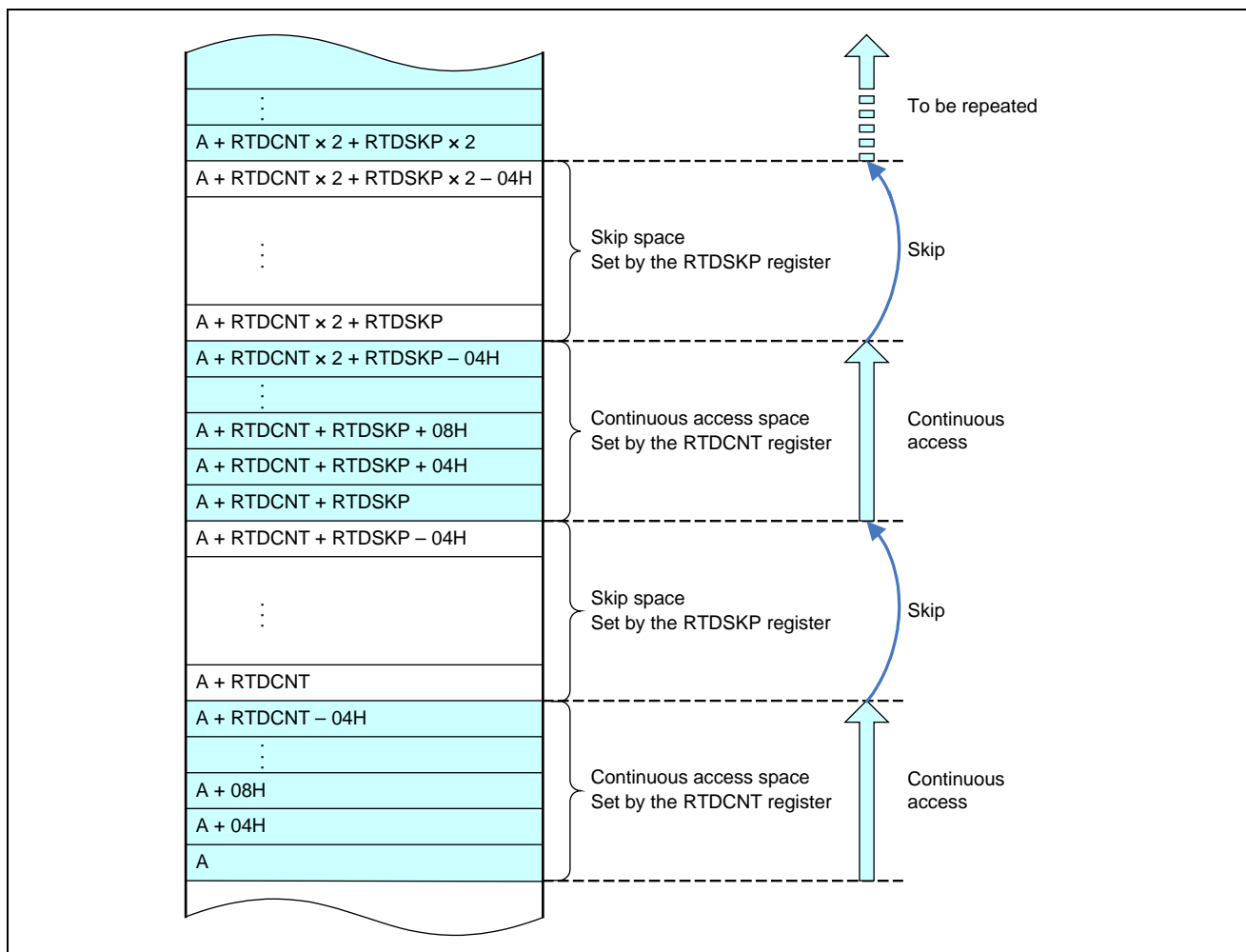


Figure 13.7 Relationship between the RTDSCP and RTDCNT Registers in Skip Mode

Remark: The values of RTDCNT and RTDSCP 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.

RTDCTRL	31	0	1	0	Address	Initial value
			PR			
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.

RTDSCITVL	31	0	16	15	DITVL15-DITVL8	8	7	0	Address	Initial value
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).

	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
	0 0	400A 2F10H
RTDSTEN	0 0	Initial value
	0 0	0000 0000H
R/W	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	Address
RTDSTER	0 0	400A 2F14H
		Initial value
		0000 0000H
R/W	0 0	R

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).

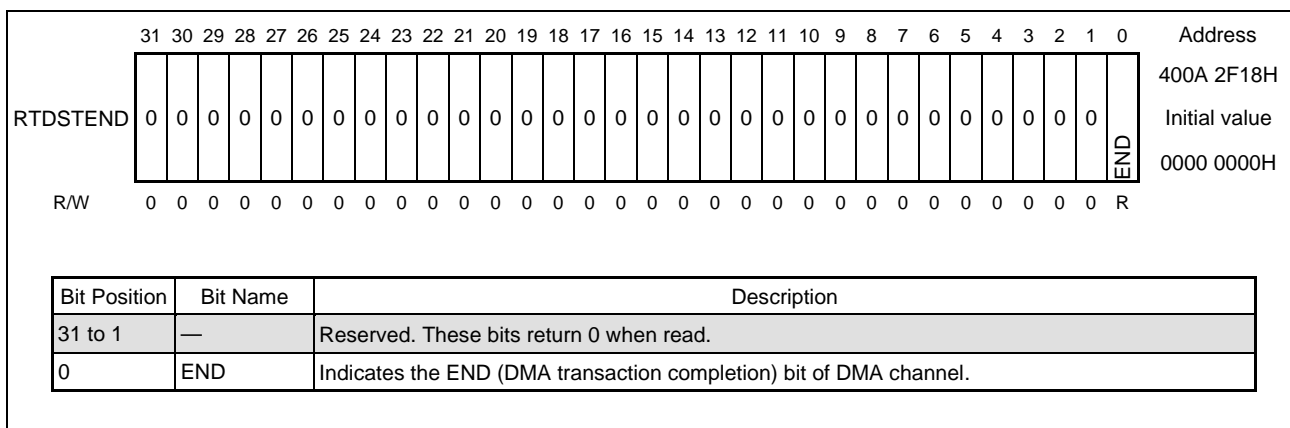


Table 13.8 Correspondence between DMA End Status Register and Interrupt Signal

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).

RTDSTTC	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			
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	400A 2F1CH
																																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	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).

RTDSTSUS	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			
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	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.

13.4.5 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 (DMAIFCn, 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 DMAIFCn 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 how to release protection, see the description of the system protection 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 4001 0720H 4001 0724H	
DMAIFC0	DIF	0																	
DMAIFC1	EN	0																	
R/W	R/W	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	Initial value 0000 0000H
		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		
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 ^{Note1} of the DMA transfer request signal (DMAREQZp) in units of BUSCLK. <table border="1" style="width:100%; border-collapse: collapse; margin: 5px 0;"> <thead> <tr> <th style="width:5%;">RQ MK4</th> <th style="width:5%;">RQ MK3</th> <th style="width:5%;">RQ MK2</th> <th style="width:5%;">RQ MK1</th> <th style="width:5%;">RQ MK0</th> <th style="width:70%;">DMAREQZp Signal Mask Width^{Note1}</th> </tr> </thead> <tbody> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td>0 BUSCLK cycles (initial value)</td> </tr> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td>1 BUSCLK cycle</td> </tr> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>2 BUSCLK cycles</td> </tr> <tr> <td colspan="5" style="text-align:center;">:</td> <td style="text-align:center;">:</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>30 BUSCLK cycles</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td>31 BUSCLK cycles</td> </tr> </tbody> </table>	RQ MK4	RQ MK3	RQ MK2	RQ MK1	RQ MK0	DMAREQZp Signal Mask Width ^{Note1}	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
RQ MK4	RQ MK3	RQ MK2	RQ MK1	RQ MK0	DMAREQZp Signal Mask Width ^{Note1}																																							
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																																							
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 ^{Note2} of the DMA acknowledge signal (DMAACKZp) in units of BUSCLK. <table border="1" style="width:100%; border-collapse: collapse; margin: 5px 0;"> <thead> <tr> <th style="width:5%;">AK WD4</th> <th style="width:5%;">AK WD3</th> <th style="width:5%;">AK WD2</th> <th style="width:5%;">AK WD1</th> <th style="width:5%;">AK WD0</th> <th style="width:70%;">DMAACKZp Signal Active Level Width^{Note2}</th> </tr> </thead> <tbody> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td>+ 0 BUSCLK cycles (initial value)</td> </tr> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td>+ 1 BUSCLK cycle</td> </tr> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>+ 2 BUSCLK cycles</td> </tr> <tr> <td colspan="5" style="text-align:center;">:</td> <td style="text-align:center;">:</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>+ 30 BUSCLK cycles</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td>+ 31 BUSCLK cycles</td> </tr> </tbody> </table>	AK WD4	AK WD3	AK WD2	AK WD1	AK WD0	DMAACKZp Signal Active Level Width ^{Note2}	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 ^{Note2}																																							
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 ^{Note1} of the DMA transfer request signal (RTDMAREQZ) in units of BUSCLK. <table border="1"> <thead> <tr> <th>RQ MK4</th><th>RQ MK3</th><th>RQ MK2</th><th>RQ MK1</th><th>RQ MK0</th><th>RTDMAREQZ Signal Mask Width ^{Note1}</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0 BUSCLK cycles (initial value)</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1 BUSCLK cycle</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2 BUSCLK cycles</td> </tr> <tr> <td colspan="5" style="text-align:center">:</td><td style="text-align:center">:</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>30 BUSCLK cycles</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>31 BUSCLK cycles</td> </tr> </tbody> </table>	RQ MK4	RQ MK3	RQ MK2	RQ MK1	RQ MK0	RTDMAREQZ Signal Mask Width ^{Note1}	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
RQ MK4	RQ MK3	RQ MK2	RQ MK1	RQ MK0	RTDMAREQZ Signal Mask Width ^{Note1}																																							
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																																							
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 ^{Note2} of the DMA acknowledge signal (RTDMAACKZ) in units of BUSCLK. <table border="1"> <thead> <tr> <th>AK WD4</th><th>AK WD3</th><th>AK WD2</th><th>AK WD1</th><th>AK WD0</th><th>RTDMAACKZ Signal Active Level Width ^{Note2}</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>+ 0 BUSCLK cycles (initial value)</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>+ 1 BUSCLK cycle</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>+ 2 BUSCLK cycles</td> </tr> <tr> <td colspan="5" style="text-align:center">:</td><td style="text-align:center">:</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>+ 30 BUSCLK cycles</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>+ 31 BUSCLK cycles</td> </tr> </tbody> </table>	AK WD4	AK WD3	AK WD2	AK WD1	AK WD0	RTDMAACKZ Signal Active Level Width ^{Note2}	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	RTDMAACKZ Signal Active Level Width ^{Note2}																																							
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.

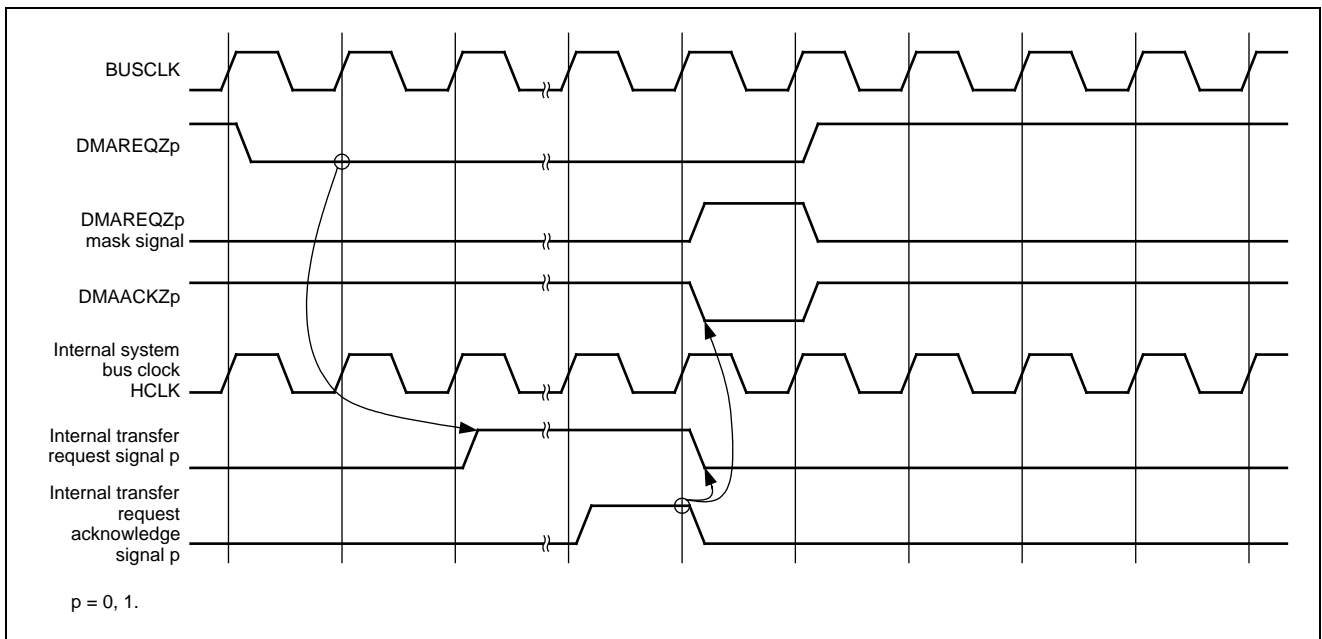


Figure 13.8 DMA Pin Signals and Internal Signals (1) (DMAIFCp = 8000 0000H)

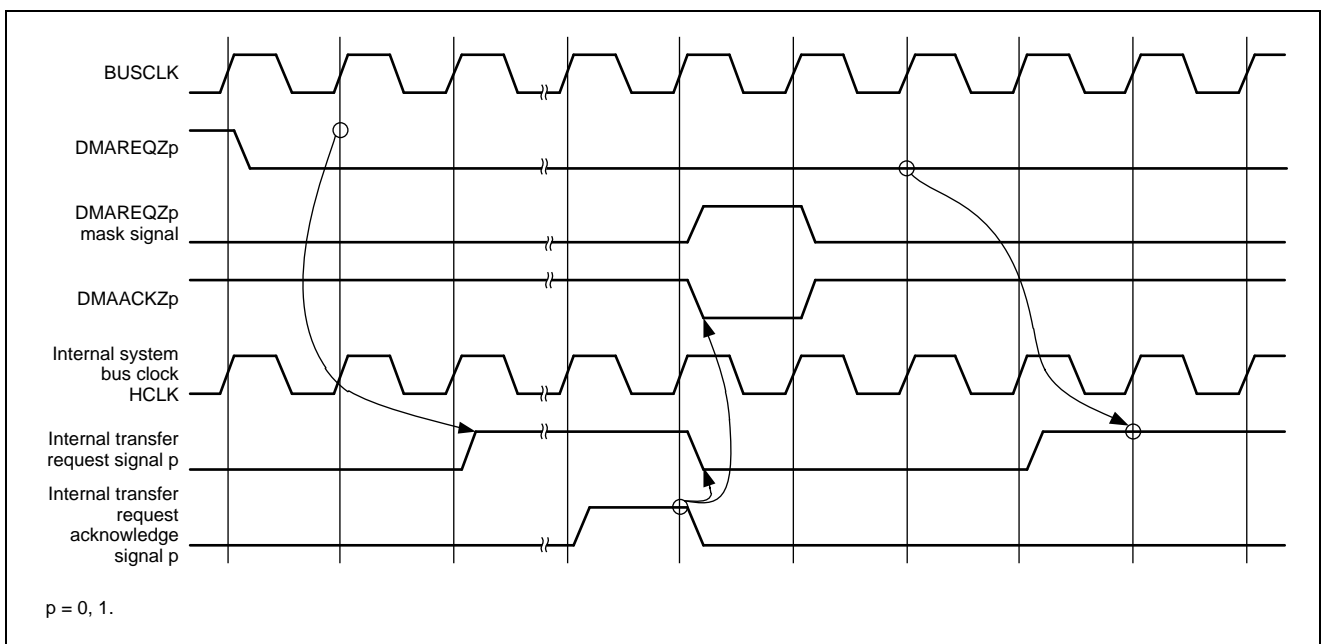


Figure 13.9 DMA Pin Signals and Internal Signals (2) (DMAIFCp = 8000 0000H)

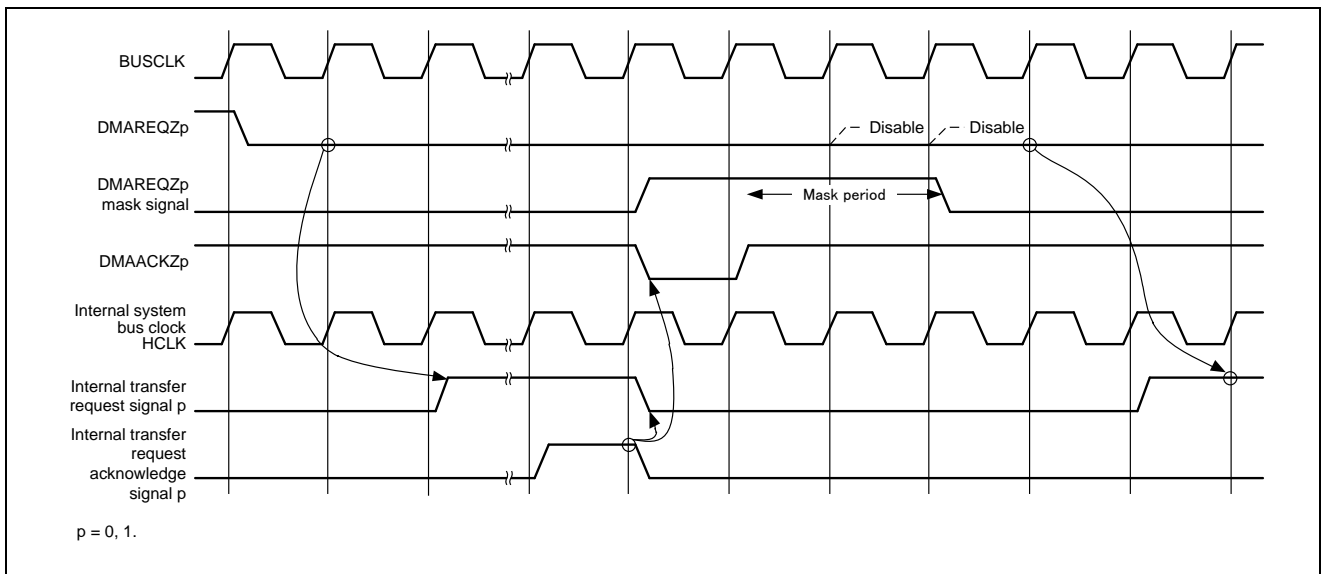


Figure 13.10 DMA Pin Signals and Internal Signals (3) (DMAIFCp = 8000 0200H)

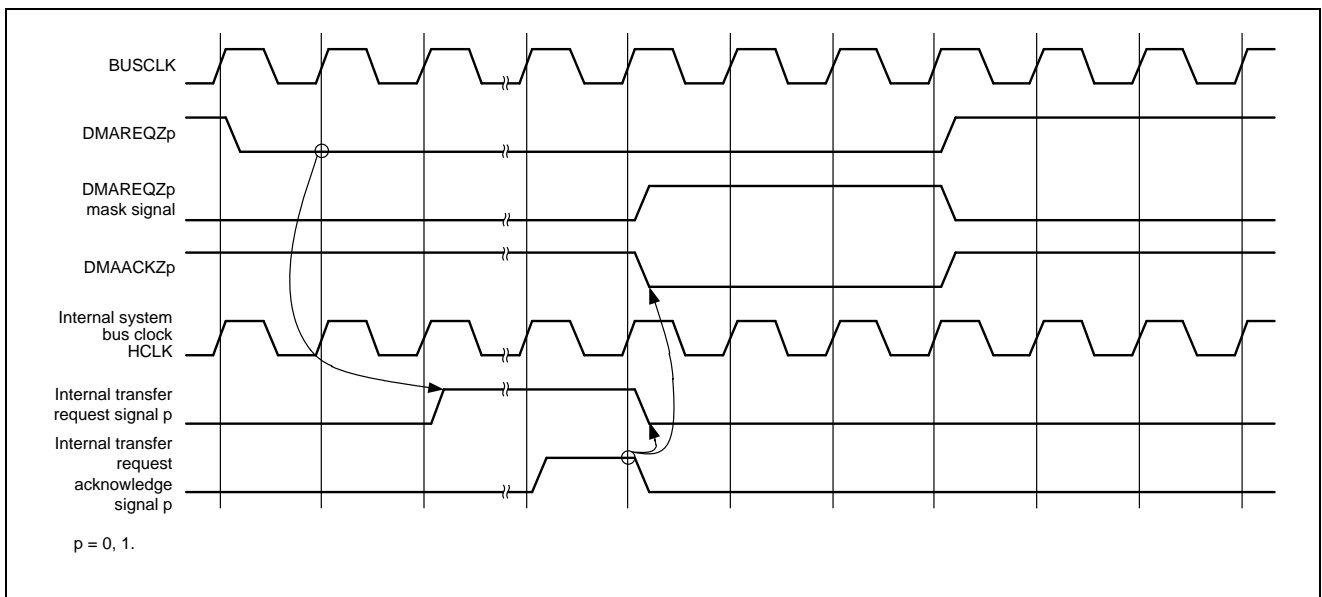


Figure 13.11 DMA Pin Signals and Internal Signals (4) (DMAIFCp = 8000 0002H)

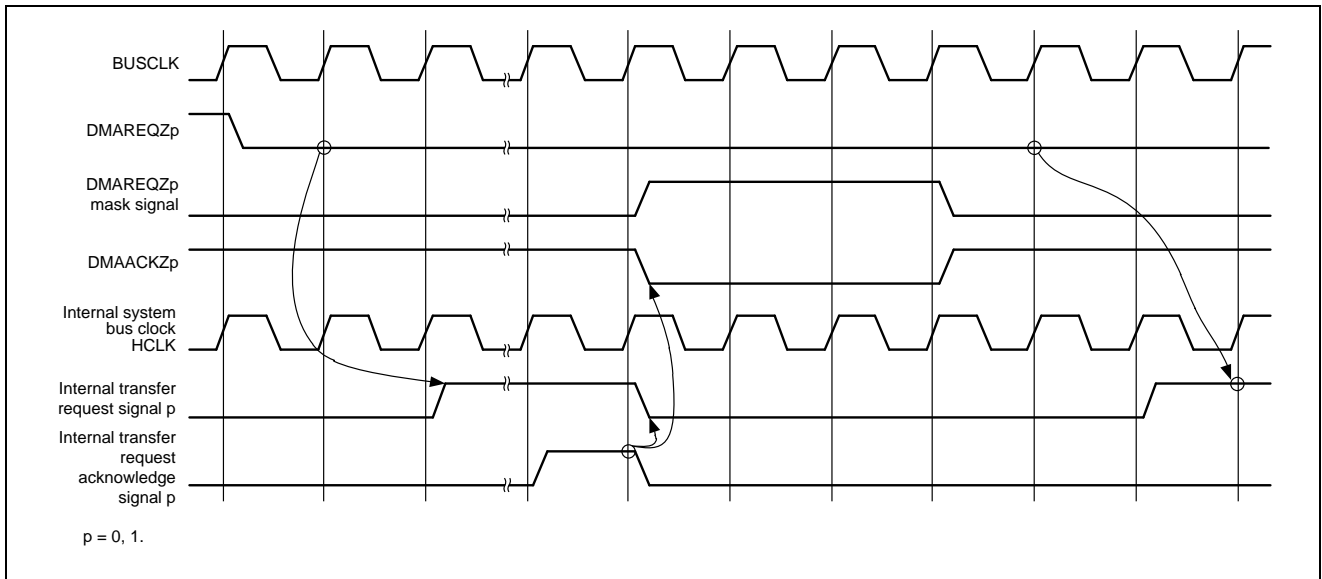


Figure 13.12 DMA Pin Signals and Internal Signals (5) (DMAIFCp = 8000 0002H)

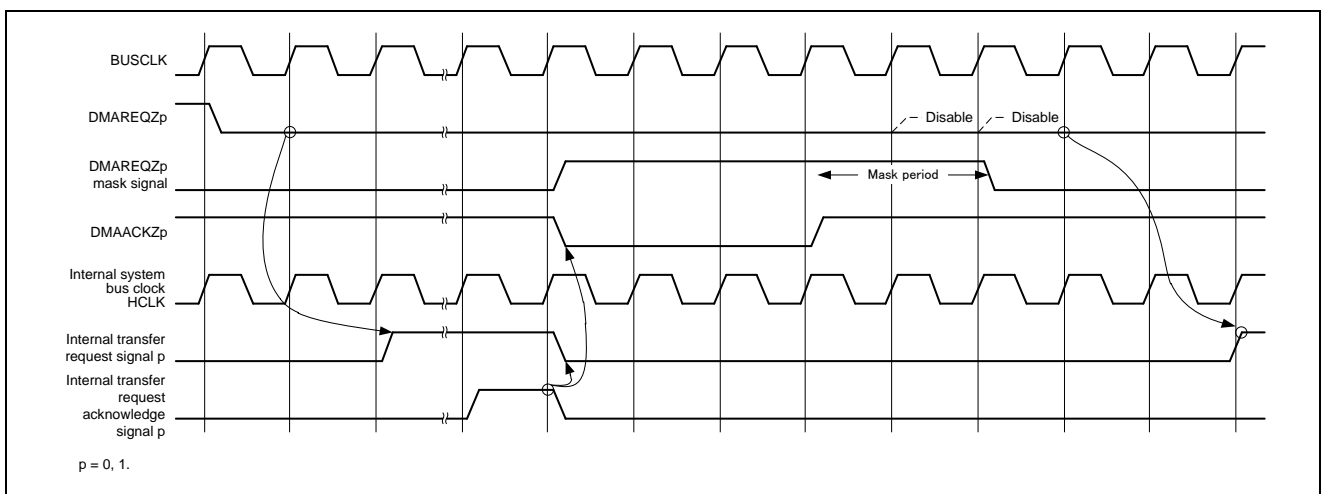


Figure 13.13 DMA Pin Signals and Internal Signals (6) (DMAIFCp = 8000 0202H)

13.4.6 DMA Trigger Source Selection Registers (DTFRn, RTDTFR)

These registers select the DMA transfer request by the interrupt request from DMAREQZ_p, RTDMAREQZ (DMA transfer demand terminal), and internal peripheral modules, and the interrupt request by the pint of a signal to the external interrupt pin as a DMA transfer request. The source selected by this register becomes a trigger for starting DMA transfer.

There are a total of five DTFR_n 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 (CHCFG_n and RTCHCFG).

These registers can be read or written in units of 32 bits.

- Cautions**
1. When you change the setting of the DTFR_n 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 how to release protection, see the description of the system protection 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).

DTFR _n	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4001 0730H +4n	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Initial value
	0	0	0	0	0	0	0	0	0	0	IFC6	IFC5	IFC4	IFC3	IFC2	IFC1	IFC0	0000 0000H
	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
RTDTFR	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4001 0740H	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Initial value
	0	0	0	0	0	0	0	0	0	0	IFC6	IFC5	IFC4	IFC3	IFC2	IFC1	IFC0	0000 0000H
	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: n = 0 to 3; p = 0, 1

Bit Position	Bit Name	Description																																																										
6 to 0	IFC6-IFC0	Trigger source of a DMA channel is chosen. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>IFCn6-IFCn0</th> <th>Selection of a DMA Transfer Trigger Source</th> </tr> </thead> <tbody> <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>UART0 transmission interrupt</td> </tr> <tr> <td>09H</td> <td>UART0 reception interrupt</td> </tr> <tr> <td>0AH</td> <td>UART1 transmission interrupt</td> </tr> <tr> <td>0BH</td> <td>UART1 reception interrupt</td> </tr> <tr> <td>0CH</td> <td>CSI0 communications status interrupt</td> </tr> <tr> <td>0DH</td> <td>CSI0 reception status interrupt</td> </tr> <tr> <td>0EH</td> <td>CSI0 end of job interrupt</td> </tr> <tr> <td>0FH</td> <td>CSI1 communications status interrupt</td> </tr> <tr> <td>10H</td> <td>CSI1 reception status interrupt</td> </tr> <tr> <td>11H</td> <td>CSI1 end of job interrupt</td> </tr> <tr> <td>12H</td> <td>IICB0 data transmission/reception interrupt signal</td> </tr> <tr> <td>13H</td> <td>IICB1 data transmission/reception interrupt signal</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> <tr> <td>1AH</td> <td>General-purpose DMAC channel 0 transfer completion interrupt</td> </tr> <tr> <td>1BH</td> <td>General-purpose DMAC channel 1 transfer completion interrupt</td> </tr> </tbody> </table>	IFCn6-IFCn0	Selection of a DMA Transfer Trigger Source	00H	Mask the DMA transfer trigger source (transmission request is not generated).	01H	DMAREQZ0 pin (DMA transfer request) input ^{Note}	02H	DMAREQZ1 pin (DMA transfer request) input ^{Note}	03H	RTDMAREQZ0 pin (DMA transfer request) input ^{Note}	04H	TAUJ2 channel 0 interrupt	05H	TAUJ2 channel 1 interrupt	06H	TAUJ2 channel 2 interrupt	07H	TAUJ2 channel 3 interrupt	08H	UART0 transmission interrupt	09H	UART0 reception interrupt	0AH	UART1 transmission interrupt	0BH	UART1 reception interrupt	0CH	CSI0 communications status interrupt	0DH	CSI0 reception status interrupt	0EH	CSI0 end of job interrupt	0FH	CSI1 communications status interrupt	10H	CSI1 reception status interrupt	11H	CSI1 end of job interrupt	12H	IICB0 data transmission/reception interrupt signal	13H	IICB1 data transmission/reception interrupt signal	14H	FCN0 reception completion interrupt	15H	FCN0 transmission completion interrupt	16H	FCN0 sleep and wakeup / transmission suspension interrupt	17H	FCN1 reception completion	18H	FCN1 transmission completion	19H	FCN1 sleep and wakeup / transmission suspension interrupt	1AH	General-purpose DMAC channel 0 transfer completion interrupt	1BH	General-purpose DMAC channel 1 transfer completion interrupt
IFCn6-IFCn0	Selection of a DMA Transfer Trigger Source																																																											
00H	Mask the DMA transfer trigger source (transmission request is not generated).																																																											
01H	DMAREQZ0 pin (DMA transfer request) input ^{Note}																																																											
02H	DMAREQZ1 pin (DMA transfer request) input ^{Note}																																																											
03H	RTDMAREQZ0 pin (DMA transfer request) input ^{Note}																																																											
04H	TAUJ2 channel 0 interrupt																																																											
05H	TAUJ2 channel 1 interrupt																																																											
06H	TAUJ2 channel 2 interrupt																																																											
07H	TAUJ2 channel 3 interrupt																																																											
08H	UART0 transmission interrupt																																																											
09H	UART0 reception interrupt																																																											
0AH	UART1 transmission interrupt																																																											
0BH	UART1 reception interrupt																																																											
0CH	CSI0 communications status interrupt																																																											
0DH	CSI0 reception status interrupt																																																											
0EH	CSI0 end of job interrupt																																																											
0FH	CSI1 communications status interrupt																																																											
10H	CSI1 reception status interrupt																																																											
11H	CSI1 end of job interrupt																																																											
12H	IICB0 data transmission/reception interrupt signal																																																											
13H	IICB1 data transmission/reception interrupt signal																																																											
14H	FCN0 reception completion interrupt																																																											
15H	FCN0 transmission completion interrupt																																																											
16H	FCN0 sleep and wakeup / transmission suspension interrupt																																																											
17H	FCN1 reception completion																																																											
18H	FCN1 transmission completion																																																											
19H	FCN1 sleep and wakeup / transmission suspension interrupt																																																											
1AH	General-purpose DMAC channel 0 transfer completion interrupt																																																											
1BH	General-purpose DMAC channel 1 transfer completion interrupt																																																											

Note: External DMA transfer request inputs (inputs on the DMAREQZ0, DMAREQZ1, and RTDMAREQZ pins) can be individually set as DMA transfer trigger requests with the corresponding registers listed below.

DMAREQZ0 pin: DTFR0 register

DMAREQZ1 pin: DTFR1 register

RTDMAREQZ pin: RTDTFR register

Bit Position	Bit Name	Description																																																																
6 to 0	IFC6-IFC0	Trigger source of a DMA channel is chosen.																																																																
		<table border="1"> <thead> <tr> <th>IFCn6-IFCn0</th> <th>Selection of a DMA Transfer Trigger Source</th> </tr> </thead> <tbody> <tr><td>1CH</td><td>General-purpose DMAC channel 2 transfer completion interrupt</td></tr> <tr><td>1DH</td><td>General-purpose DMAC channel 3 transfer completion interrupt</td></tr> <tr><td>1EH</td><td>Real-time ports DMAC transfer completion interrupt</td></tr> <tr><td>1FH</td><td>EtherCAT Sync0 interrupt</td></tr> <tr><td>20H</td><td>EtherCAT Sync1 interrupt</td></tr> <tr><td>21H</td><td>EtherCAT interrupt</td></tr> <tr><td>22H</td><td>EtherCAT SOF interrupt</td></tr> <tr><td>23H</td><td>EtherCAT EOF interrupt</td></tr> <tr><td>24H</td><td>Inter-Buffer DMA transfer completion</td></tr> <tr><td>25H</td><td>Ethernet PHY interrupt 0</td></tr> <tr><td>26H</td><td>Ethernet PHY interrupt 1</td></tr> <tr><td>27H</td><td>Ethernet MII management access completion interrupt</td></tr> <tr><td>28H</td><td>Ethernet pause packet transmission completion</td></tr> <tr><td>29H</td><td>Ethernet transmission completion interrupt</td></tr> <tr><td>2AH</td><td>Ethernet SWITCH interrupt</td></tr> <tr><td>2BH</td><td>Ethernet SWITCH DLR interrupt</td></tr> <tr><td>2CH</td><td>Ethernet SWITCH SEC interrupt</td></tr> <tr><td>2DH</td><td>Reserved (setting prohibited)</td></tr> <tr><td>2EH</td><td>Reserved (setting prohibited)</td></tr> <tr><td>2FH</td><td>Ethernet MACDMA reception completion</td></tr> <tr><td>30H</td><td>Ethernet MACDMA transmission completion</td></tr> <tr><td>31H</td><td>Receive frame successfully interrupt</td></tr> <tr><td>32H</td><td>Reserved (setting prohibited)</td></tr> <tr><td>33H</td><td>INTPZ0 input ^{Note}</td></tr> <tr><td>34H</td><td>INTPZ1 input ^{Note}</td></tr> <tr><td>35H</td><td>INTPZ2 input ^{Note}</td></tr> <tr><td>36H</td><td>INTPZ3 input ^{Note}</td></tr> <tr><td>37H</td><td>INTPZ4 input ^{Note}</td></tr> <tr><td>38H</td><td>INTPZ5 input ^{Note}</td></tr> <tr><td>39H</td><td>INTPZ6 input ^{Note}</td></tr> <tr><td>3AH</td><td>INTPZ7 input ^{Note}</td></tr> </tbody> </table>	IFCn6-IFCn0	Selection of a DMA Transfer Trigger Source	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	EtherCAT Sync0 interrupt	20H	EtherCAT Sync1 interrupt	21H	EtherCAT interrupt	22H	EtherCAT SOF interrupt	23H	EtherCAT EOF interrupt	24H	Inter-Buffer DMA transfer completion	25H	Ethernet PHY interrupt 0	26H	Ethernet PHY interrupt 1	27H	Ethernet MII management access completion interrupt	28H	Ethernet pause packet transmission completion	29H	Ethernet transmission completion interrupt	2AH	Ethernet SWITCH interrupt	2BH	Ethernet SWITCH DLR interrupt	2CH	Ethernet SWITCH SEC interrupt	2DH	Reserved (setting prohibited)	2EH	Reserved (setting prohibited)	2FH	Ethernet MACDMA reception completion	30H	Ethernet MACDMA transmission completion	31H	Receive frame successfully interrupt	32H	Reserved (setting prohibited)	33H	INTPZ0 input ^{Note}	34H	INTPZ1 input ^{Note}	35H	INTPZ2 input ^{Note}	36H	INTPZ3 input ^{Note}	37H	INTPZ4 input ^{Note}	38H	INTPZ5 input ^{Note}	39H	INTPZ6 input ^{Note}	3AH	INTPZ7 input ^{Note}
IFCn6-IFCn0	Selection of a DMA Transfer Trigger Source																																																																	
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	EtherCAT Sync0 interrupt																																																																	
20H	EtherCAT Sync1 interrupt																																																																	
21H	EtherCAT interrupt																																																																	
22H	EtherCAT SOF interrupt																																																																	
23H	EtherCAT EOF interrupt																																																																	
24H	Inter-Buffer DMA transfer completion																																																																	
25H	Ethernet PHY interrupt 0																																																																	
26H	Ethernet PHY interrupt 1																																																																	
27H	Ethernet MII management access completion interrupt																																																																	
28H	Ethernet pause packet transmission completion																																																																	
29H	Ethernet transmission completion interrupt																																																																	
2AH	Ethernet SWITCH interrupt																																																																	
2BH	Ethernet SWITCH DLR interrupt																																																																	
2CH	Ethernet SWITCH SEC interrupt																																																																	
2DH	Reserved (setting prohibited)																																																																	
2EH	Reserved (setting prohibited)																																																																	
2FH	Ethernet MACDMA reception completion																																																																	
30H	Ethernet MACDMA transmission completion																																																																	
31H	Receive frame successfully interrupt																																																																	
32H	Reserved (setting prohibited)																																																																	
33H	INTPZ0 input ^{Note}																																																																	
34H	INTPZ1 input ^{Note}																																																																	
35H	INTPZ2 input ^{Note}																																																																	
36H	INTPZ3 input ^{Note}																																																																	
37H	INTPZ4 input ^{Note}																																																																	
38H	INTPZ5 input ^{Note}																																																																	
39H	INTPZ6 input ^{Note}																																																																	
3AH	INTPZ7 input ^{Note}																																																																	

Note: When the external interrupt is used as the DMA trigger source, be sure to specify edges (do not set level detection).

Bit Position	Bit Name	Description																																																																
6 to 0	IFC6-IFC0	Trigger source of a DMA channel is chosen. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>IFCn6-IFCn0</th> <th>Selection of a DMA Transfer Trigger Source</th> </tr> </thead> <tbody> <tr><td>3BH</td><td>INTPZ8 input ^{Note}</td></tr> <tr><td>3CH</td><td>INTPZ9 input ^{Note}</td></tr> <tr><td>3DH</td><td>INTPZ10 input ^{Note}</td></tr> <tr><td>3EH</td><td>INTPZ11 input ^{Note}</td></tr> <tr><td>3FH</td><td>INTPZ12 input ^{Note}</td></tr> <tr><td>40H</td><td>INTPZ13 input ^{Note}</td></tr> <tr><td>41H</td><td>INTPZ14 input ^{Note}</td></tr> <tr><td>42H</td><td>INTPZ15 input ^{Note}</td></tr> <tr><td>43H</td><td>INTPZ16 input ^{Note}</td></tr> <tr><td>44H</td><td>INTPZ17 input ^{Note}</td></tr> <tr><td>45H</td><td>INTPZ18 input ^{Note}</td></tr> <tr><td>46H</td><td>INTPZ19 input ^{Note}</td></tr> <tr><td>47H</td><td>INTPZ20 input ^{Note}</td></tr> <tr><td>48H</td><td>INTPZ21 input ^{Note}</td></tr> <tr><td>49H</td><td>INTPZ22 input ^{Note}</td></tr> <tr><td>4AH</td><td>INTPZ23 input ^{Note}</td></tr> <tr><td>4BH</td><td>INTPZ24 input ^{Note}</td></tr> <tr><td>4CH</td><td>INTPZ25 input ^{Note}</td></tr> <tr><td>4DH</td><td>INTPZ26 input ^{Note}</td></tr> <tr><td>4EH</td><td>INTPZ27 input ^{Note}</td></tr> <tr><td>4FH</td><td>INTPZ28 input ^{Note}</td></tr> <tr><td>50H-6EH</td><td>Reserved (setting prohibited)</td></tr> <tr><td>6FH</td><td>CC-Link IE Field NMIZ interrupt</td></tr> <tr><td>70H</td><td>CC-Link IE Field WDTZ interrupt</td></tr> <tr><td>71H</td><td>CC-Link IE Field INTZ interrupt</td></tr> <tr><td>72H</td><td>CC-Link IE Field CLKLOSSZ interrupt</td></tr> <tr><td>73H-76H</td><td>Reserved (setting prohibited)</td></tr> <tr><td>77H</td><td>CC-Link INTRQ interrupt</td></tr> <tr><td>78H</td><td>CC-Link REFSTB interrupt</td></tr> <tr><td>79H</td><td>CC-Link MON3 interrupt</td></tr> <tr><td>Other than the above</td><td>Reserved (setting prohibited)</td></tr> </tbody> </table>	IFCn6-IFCn0	Selection of a DMA Transfer Trigger Source	3BH	INTPZ8 input ^{Note}	3CH	INTPZ9 input ^{Note}	3DH	INTPZ10 input ^{Note}	3EH	INTPZ11 input ^{Note}	3FH	INTPZ12 input ^{Note}	40H	INTPZ13 input ^{Note}	41H	INTPZ14 input ^{Note}	42H	INTPZ15 input ^{Note}	43H	INTPZ16 input ^{Note}	44H	INTPZ17 input ^{Note}	45H	INTPZ18 input ^{Note}	46H	INTPZ19 input ^{Note}	47H	INTPZ20 input ^{Note}	48H	INTPZ21 input ^{Note}	49H	INTPZ22 input ^{Note}	4AH	INTPZ23 input ^{Note}	4BH	INTPZ24 input ^{Note}	4CH	INTPZ25 input ^{Note}	4DH	INTPZ26 input ^{Note}	4EH	INTPZ27 input ^{Note}	4FH	INTPZ28 input ^{Note}	50H-6EH	Reserved (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-76H	Reserved (setting prohibited)	77H	CC-Link INTRQ interrupt	78H	CC-Link REFSTB interrupt	79H	CC-Link MON3 interrupt	Other than the above	Reserved (setting prohibited)
IFCn6-IFCn0	Selection of a DMA Transfer Trigger Source																																																																	
3BH	INTPZ8 input ^{Note}																																																																	
3CH	INTPZ9 input ^{Note}																																																																	
3DH	INTPZ10 input ^{Note}																																																																	
3EH	INTPZ11 input ^{Note}																																																																	
3FH	INTPZ12 input ^{Note}																																																																	
40H	INTPZ13 input ^{Note}																																																																	
41H	INTPZ14 input ^{Note}																																																																	
42H	INTPZ15 input ^{Note}																																																																	
43H	INTPZ16 input ^{Note}																																																																	
44H	INTPZ17 input ^{Note}																																																																	
45H	INTPZ18 input ^{Note}																																																																	
46H	INTPZ19 input ^{Note}																																																																	
47H	INTPZ20 input ^{Note}																																																																	
48H	INTPZ21 input ^{Note}																																																																	
49H	INTPZ22 input ^{Note}																																																																	
4AH	INTPZ23 input ^{Note}																																																																	
4BH	INTPZ24 input ^{Note}																																																																	
4CH	INTPZ25 input ^{Note}																																																																	
4DH	INTPZ26 input ^{Note}																																																																	
4EH	INTPZ27 input ^{Note}																																																																	
4FH	INTPZ28 input ^{Note}																																																																	
50H-6EH	Reserved (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-76H	Reserved (setting prohibited)																																																																	
77H	CC-Link INTRQ interrupt																																																																	
78H	CC-Link REFSTB interrupt																																																																	
79H	CC-Link MON3 interrupt																																																																	
Other than the above	Reserved (setting prohibited)																																																																	

Note: When the external interrupt is used as the DMA trigger source, be sure to specify edges (do not set level detection).

13.5 DMA Interface Pins

13.5.1 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

13.5.2 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

13.6 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 13.9 General DMA Controller Interrupt Output

Interrupt Signal	Interrupt Source	Interrupt Detection Mask	Interrupt Output Mask
INTDMA _n	The DMA transaction is completed.	CHCFG _n register DEM = 1	CHSTAT _n . 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 13.10 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)

13.7 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.

13.7.1 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 13.11 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

13.7.2 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.

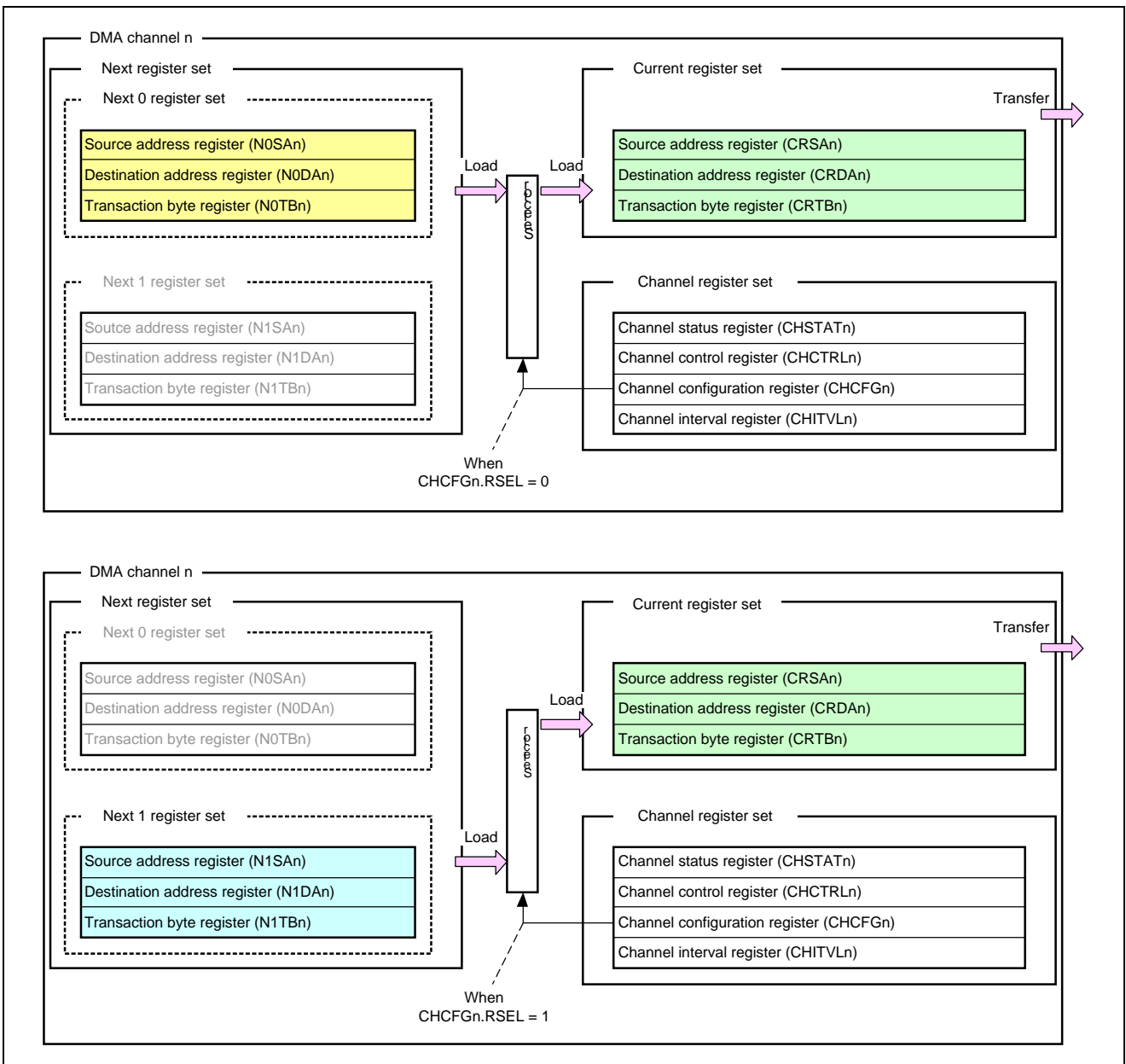
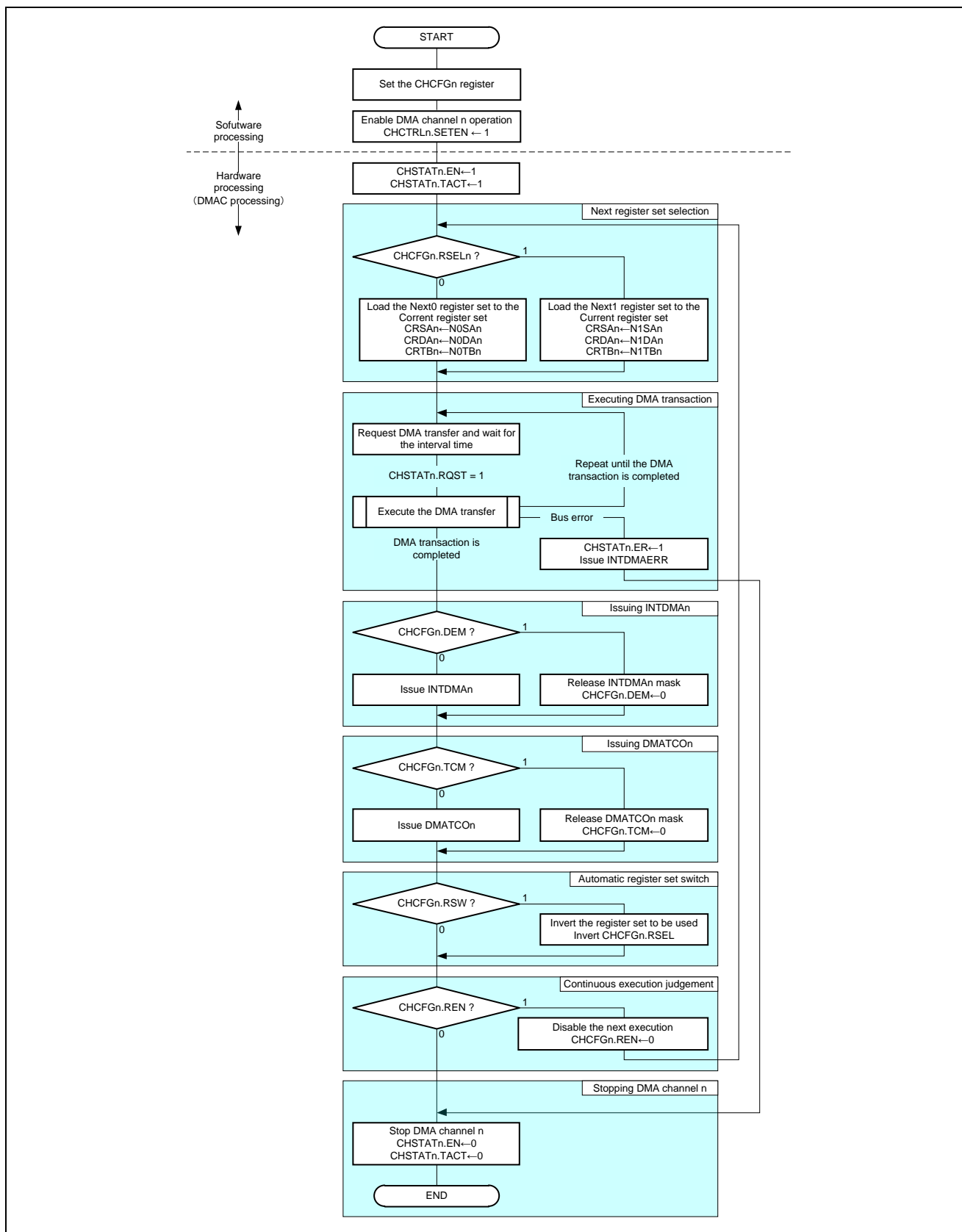


Figure 13.14 Outline of the Register Mode Operation

(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 n (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 13.8, 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 13.12 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 13.13 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 13.14 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 13.15 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 13.16 Automatic Register Set Switch Setting

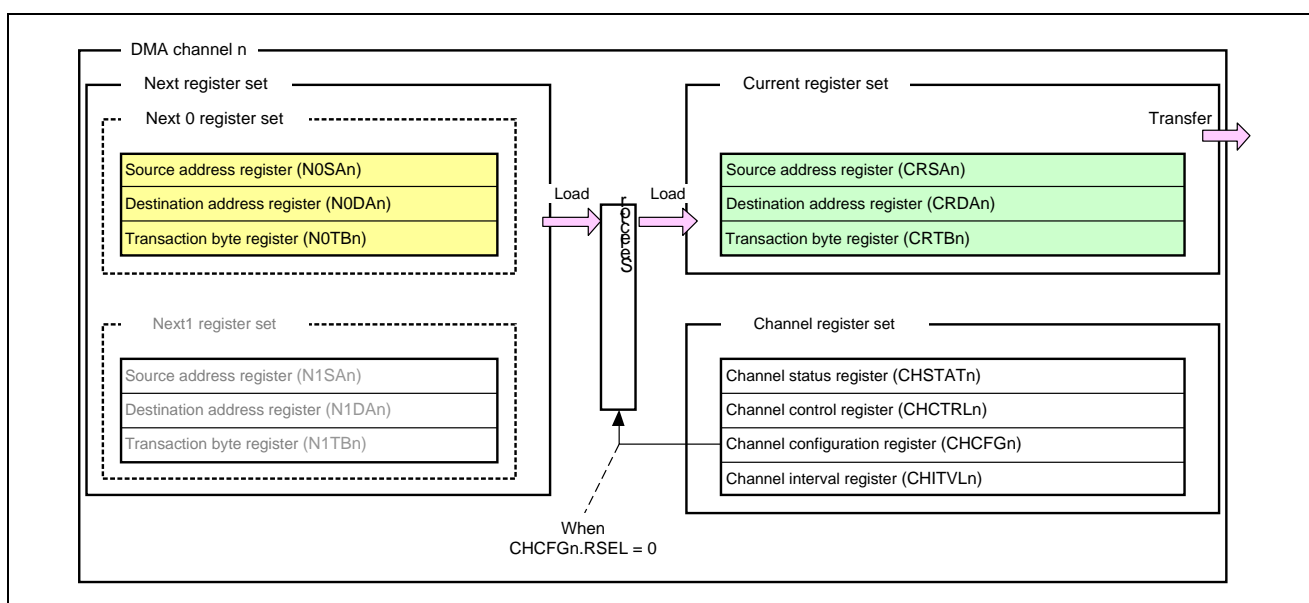
CHCFGn.RSW	Operation	Remark
0	Does not switch the register (invert the RSEL bit) when the DMA transaction (the series of DMA transfers) is completed while REN is set to 1 (continuous execution enabled).	Select this value when using only one register set.
1	Switches the register (inverts the RSEL bit) 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

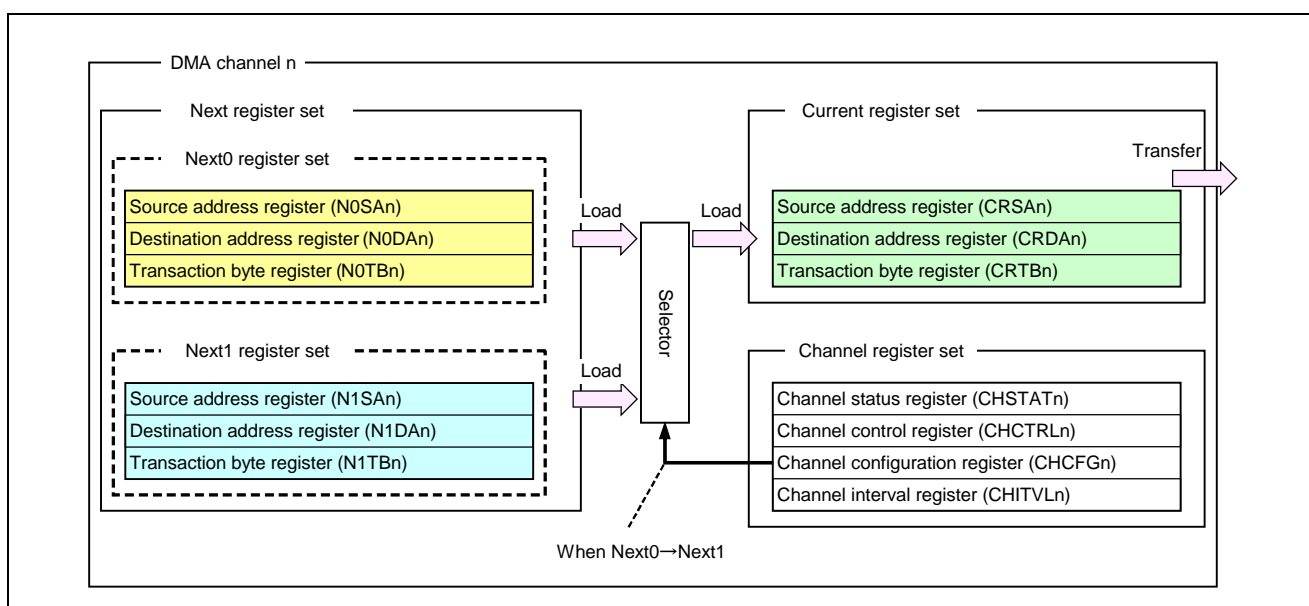


- <1> When CHCTRLn.SETEN is set to 1, both CHSTATn.EN and CHSTATn.TACT are set to 1 and the Next 0 register set is loaded to the Current register set.
- <2> A DMA transaction (the series of DMA transfers) is executed according to the values of the Current register set and channel register set.
- <3> Since CHCFGn.DEM is set to 0, INTDMAn is issued after the DMA transaction (the series of DMA transfers) is completed.
- <4> Since CHCFG.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

13.7.3 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).

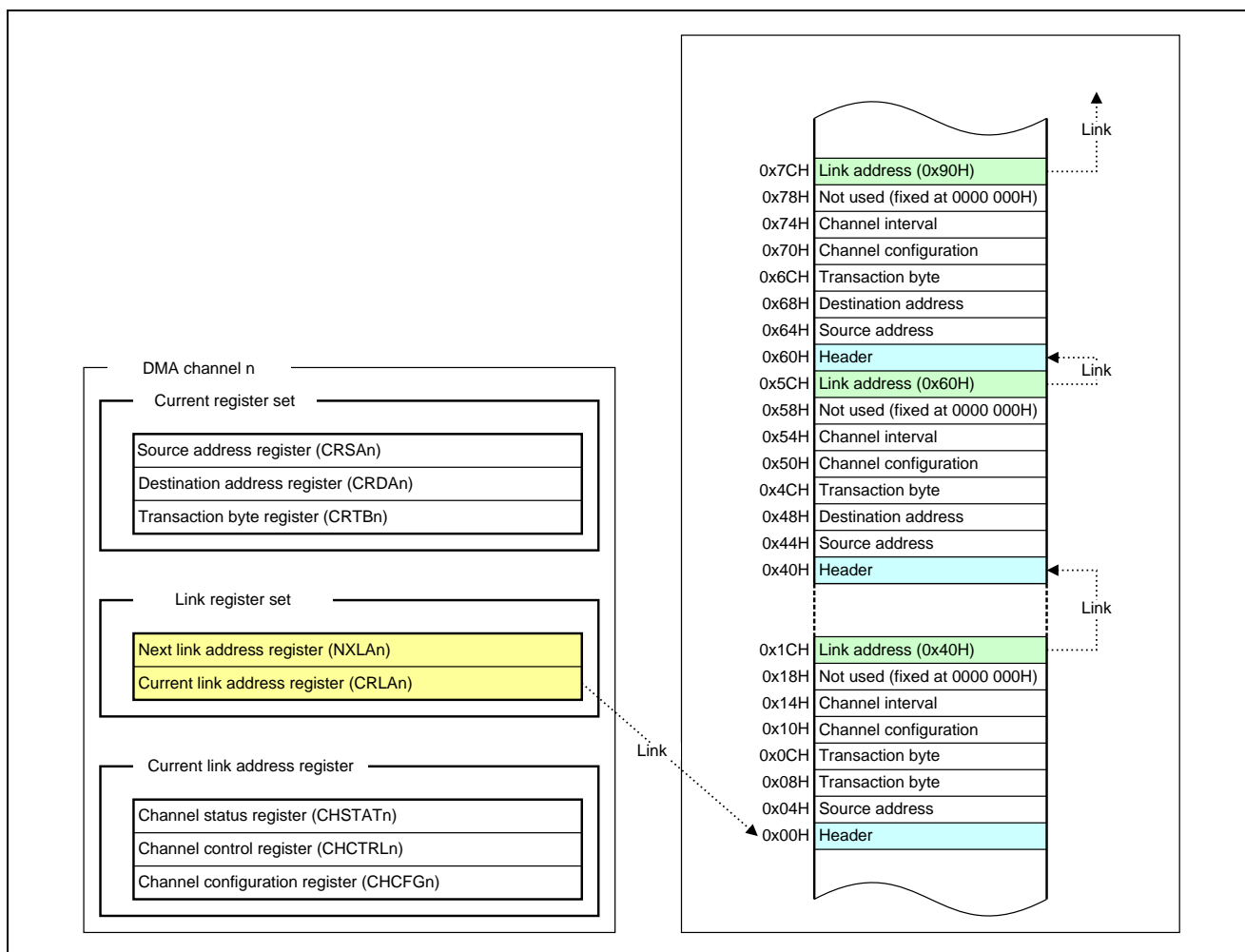
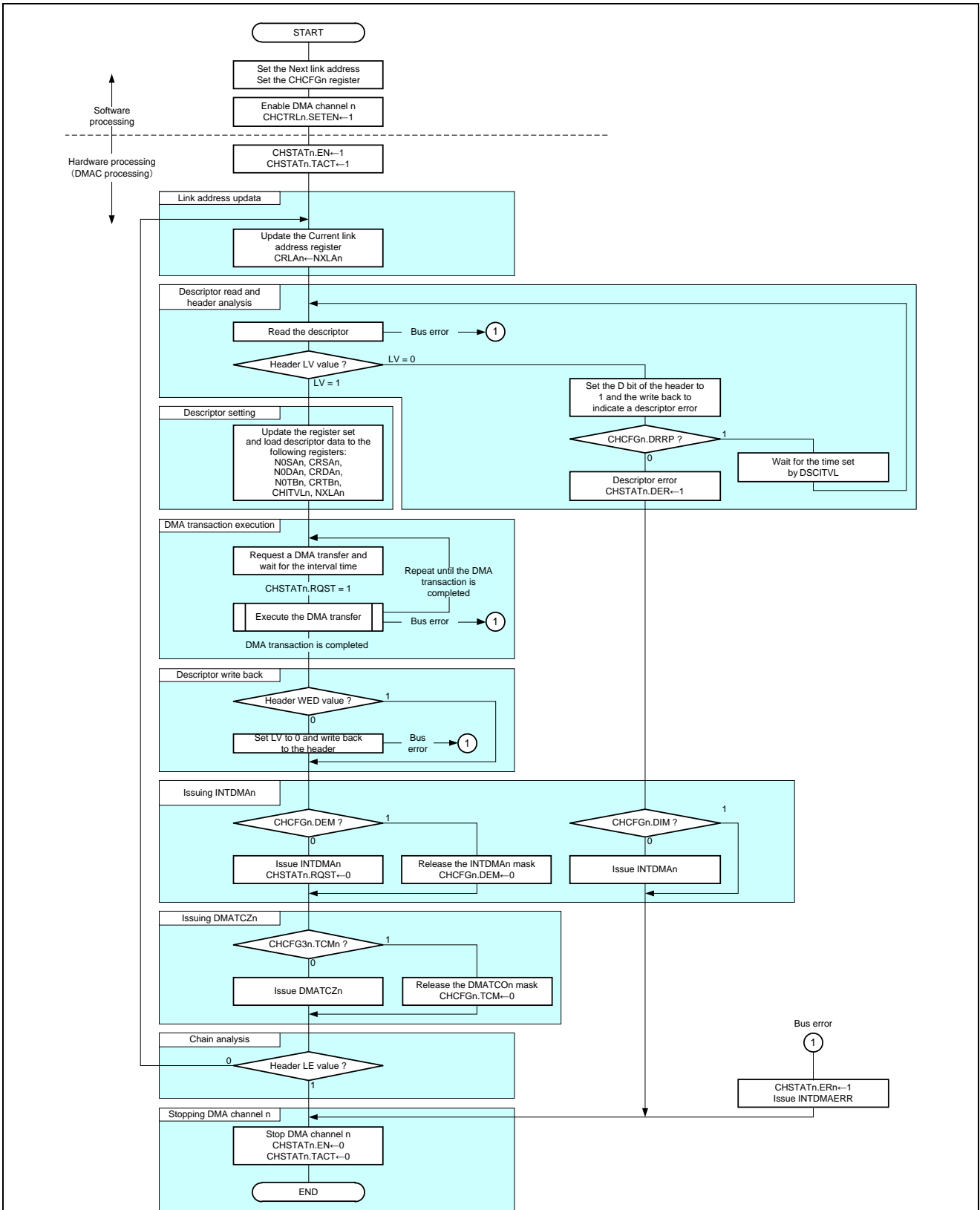


Figure 13.15 Outline of Link Mode

(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, INTDMA_n 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> Executing DMA transaction
A DMA transaction is executed according to the set values.
If a DMA transfer error occurs during this process, INTDMAERR_n 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 INTDMA_n
The INTDMA_n is masked according to the value set by the CHCFGn.DEM bit. INTDMA_n is not issued if DEM = 1.
- <8> Issuing DMATCZ_p
DMATCZ_p output is masked according to the value set by the CHCFGn.TCM bit.
When TCM is set to 1, DMATCZ_p 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 13.17 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 13.18 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 (DMAREQz_p 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 13.19 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 13.20 Description of Each Field of the Descriptor

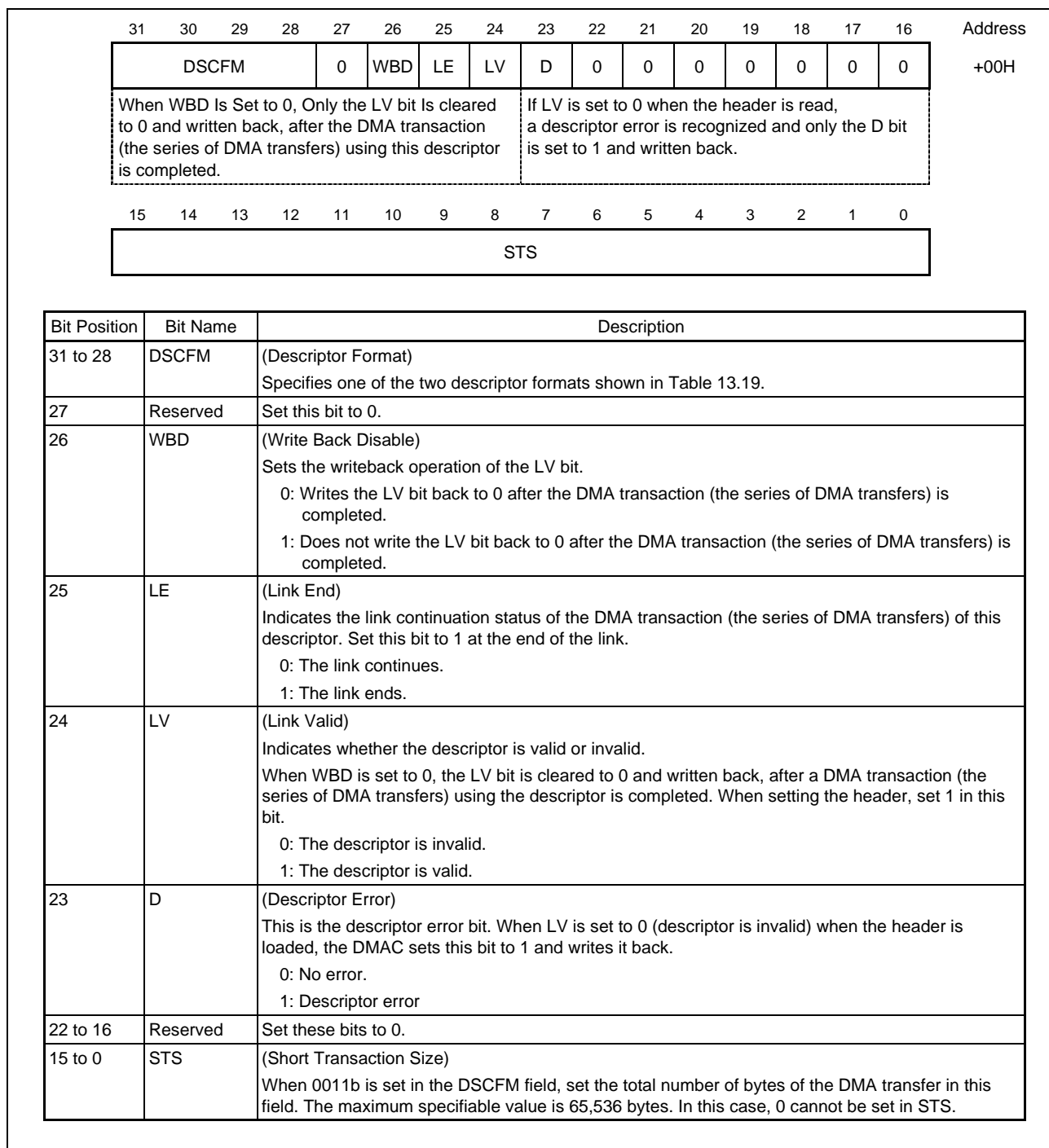
Field	Symbol	Description
Link address	✓	Specifies the address (link address) of the next descriptor to be read after the DMA transfer executed with the current descriptor.
Channel interval, Channel configuration	✓	Specifies the channel interval and channel configuration.
	— (reload)	Omits the specification 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.



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 13.21, Correspondence between the Descriptors Other Than the Header and the DMAC Internal Registers, shows their correspondence. For information about the specifications of the registers in the DMAC, see section 13.4, DMA Controller Registers.

Table 13.21 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

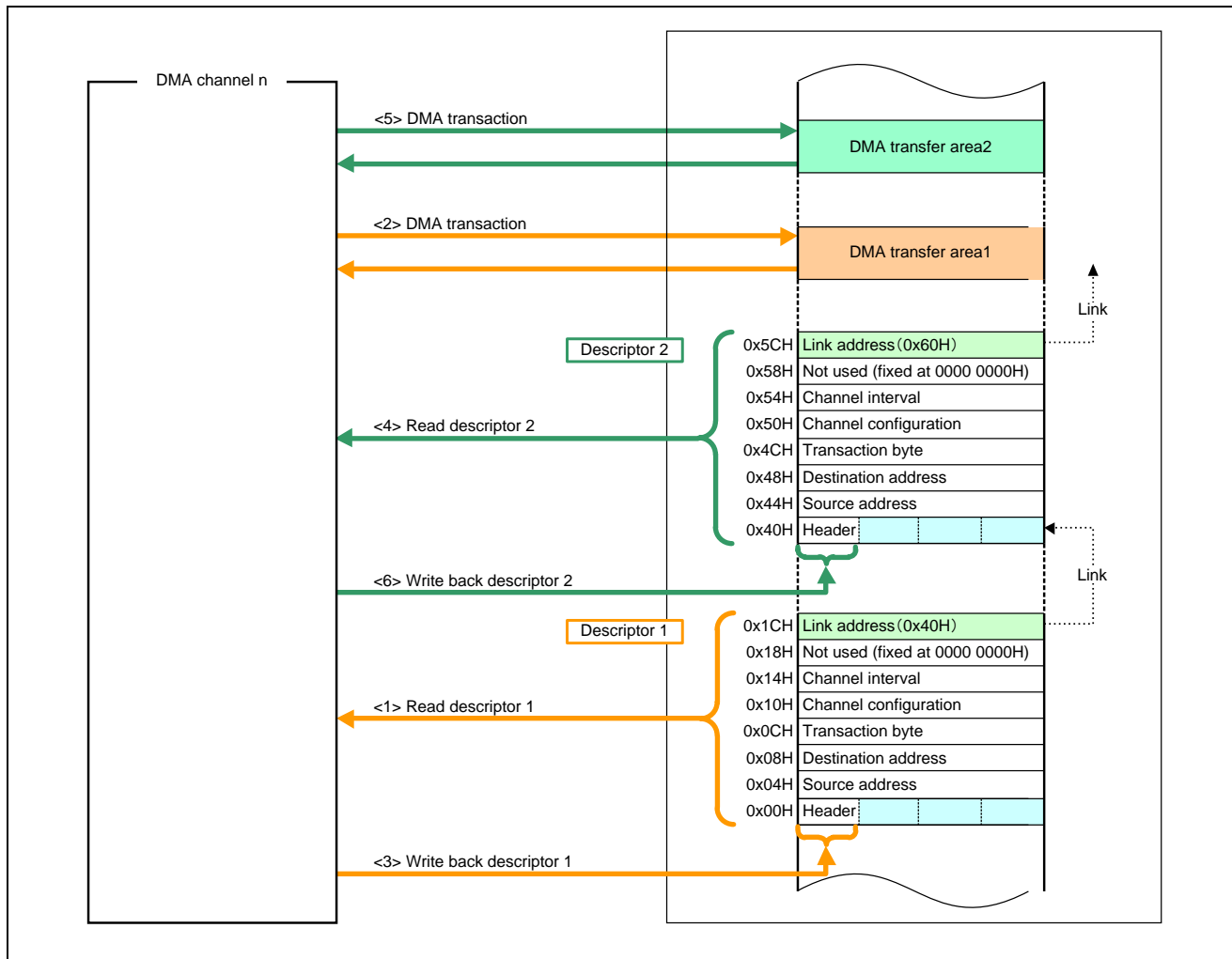


Figure 13.16 Outline of the Descriptor Area and DMA Transfer Area

<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 DSCITVLm 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.

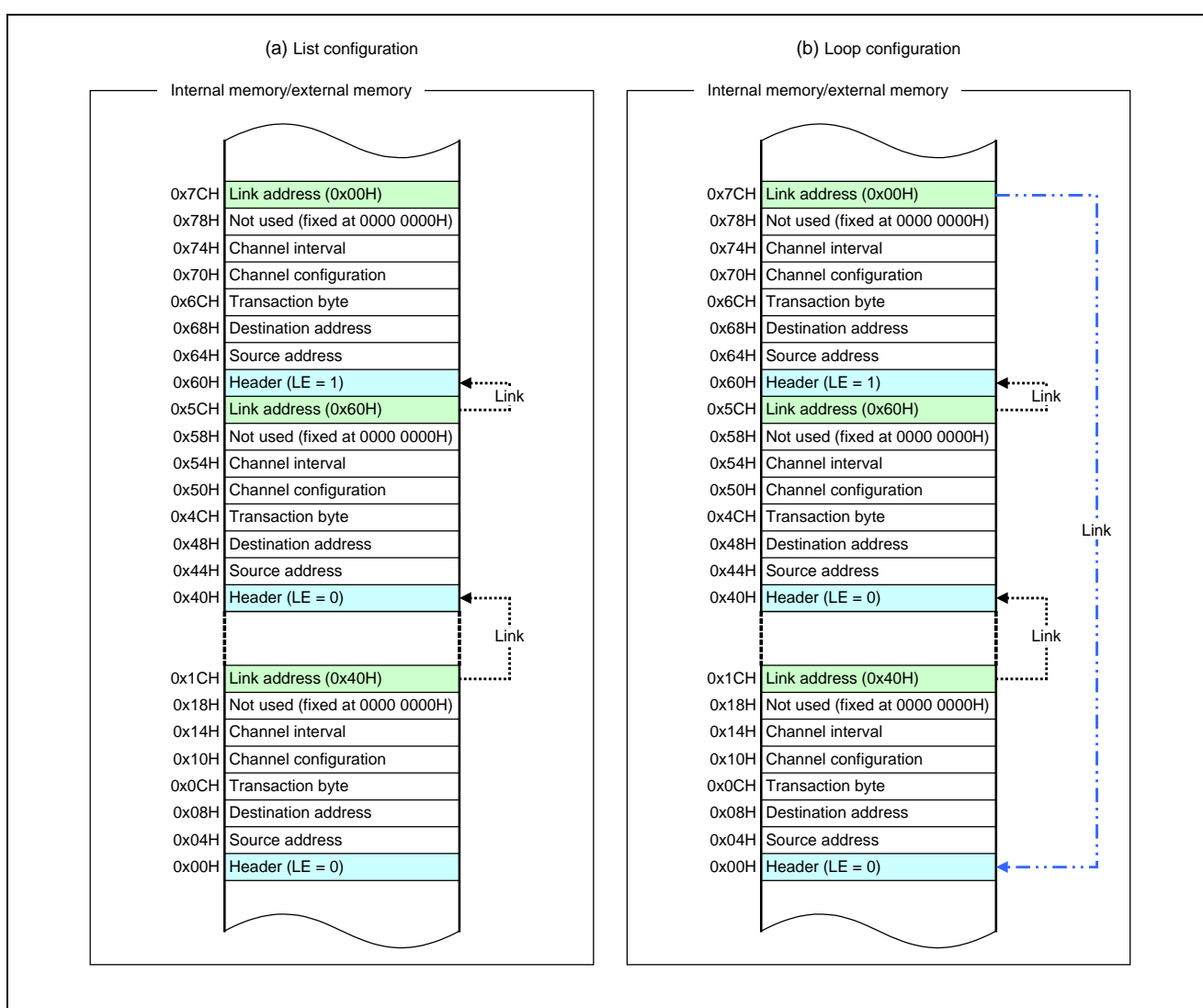


Figure 13.17 Link Mode Configuration Example

13.7.4 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 13.22 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

13.8 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.

13.8.1 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 13.23 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 13.8.10, Differences in Operation by Transfer Size.

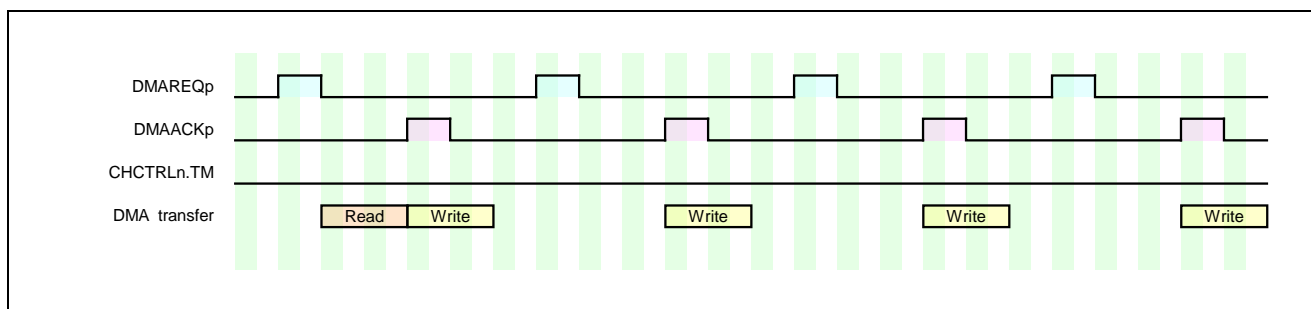


Figure 13.18 Single Transfer Mode Example

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).

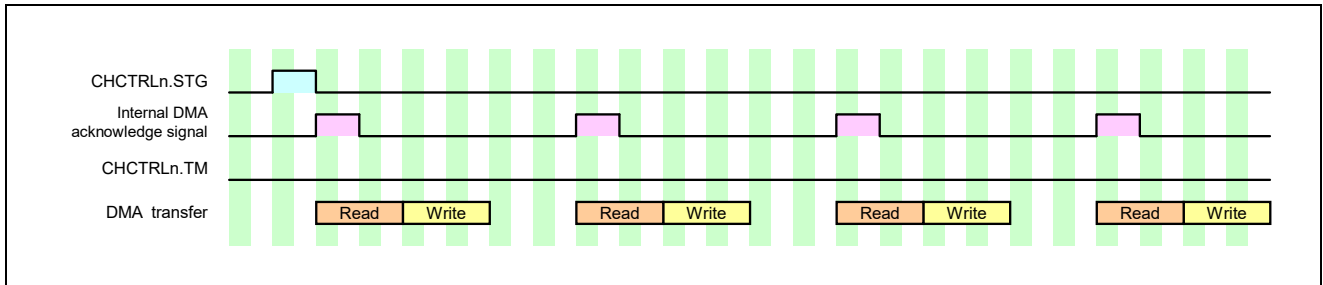


Figure 13.19 Block Transfer Mode Example

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

13.8.2 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 4.1, AHB Internal Buses of an R-IN32M3.

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 13.24 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.

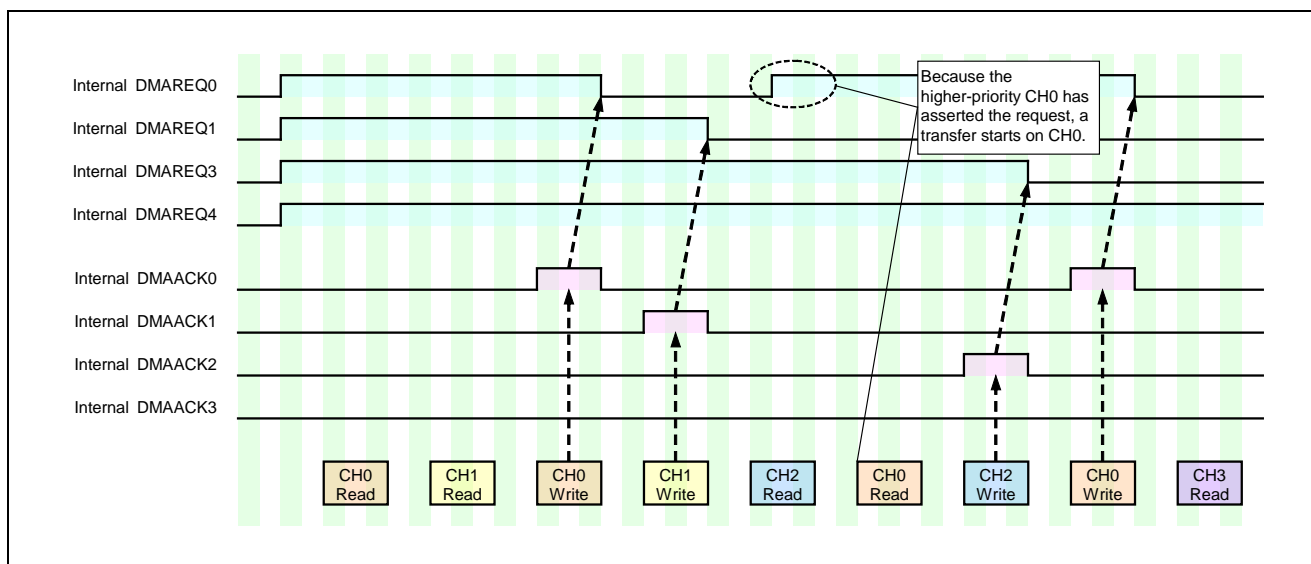


Figure 13.20 Fixed Priority Mode Example

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.

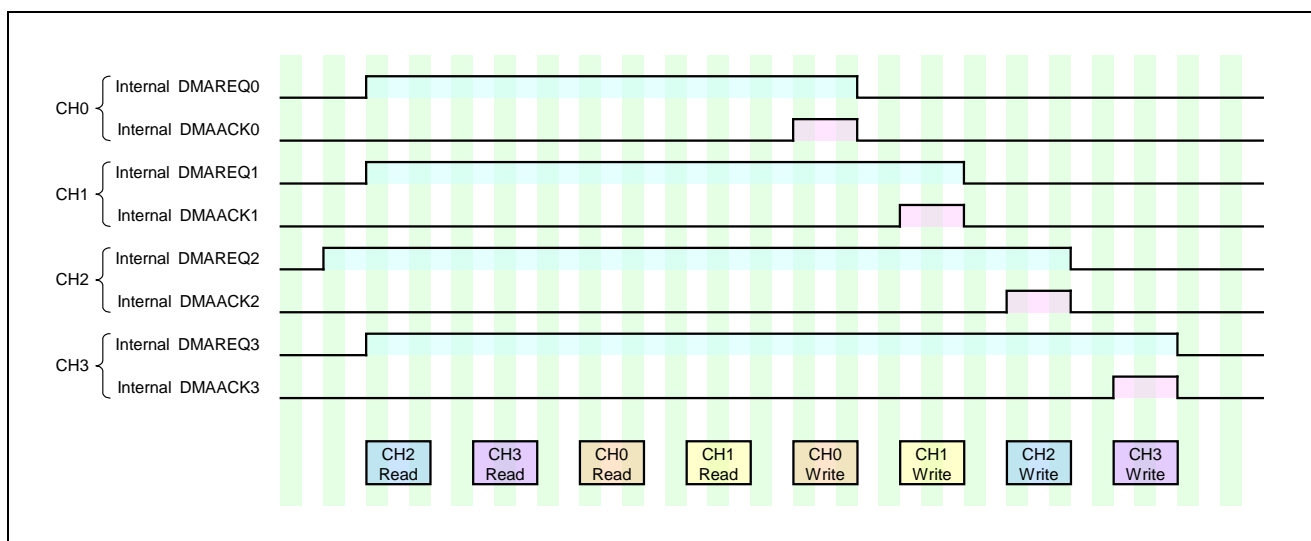


Figure 13.21 Round Robin Mode

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).

13.8.3 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 13.25 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-INTPZ28)	Rising edge detection	To be set arbitrarily according to the specification of the DMA transfer request source.
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 13.26 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.

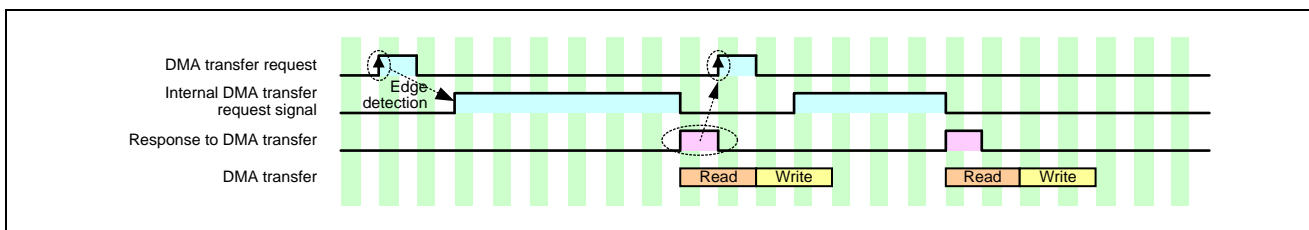


Figure 13.22 Edge Detection Mode Operation Example 1

DMA transfer request: Rising edge detection
 Request from the source (CHCFGn.REQD = 0)
 DMA acknowledge output: Pulse mode

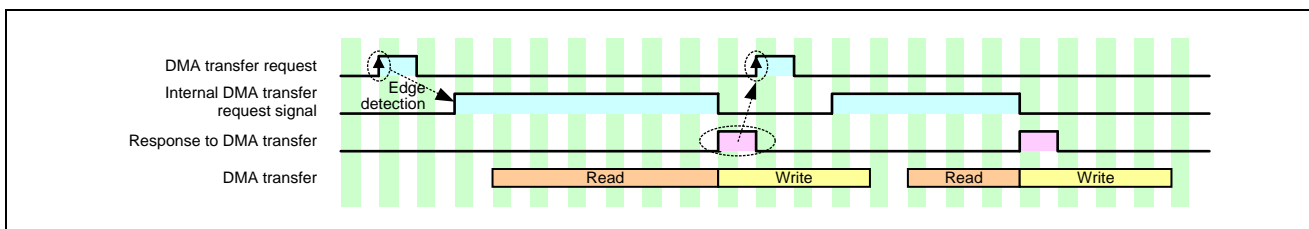


Figure 13.23 Edge Detection Mode Operation Example 2

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 $BUSCLK \times 2$ 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 $1 \times BUSCLK$.

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.

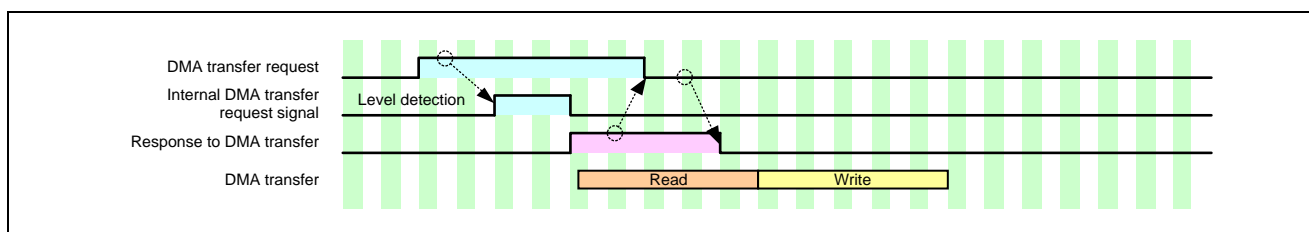


Figure 13.24 Level Detection Mode Operation Example 1

DMA transfer request: High level detection
 Request from the source (CHCFGn.REQD = 0)
 DMA acknowledge output: Level mode

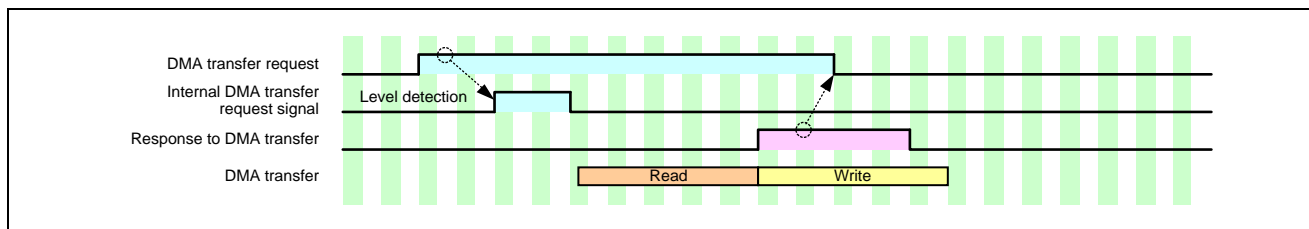


Figure 13.25 Level Detection Mode Operation Example 2

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

13.8.4 DMA Acknowledge Output

The DMA acknowledge signal is output as an acknowledge response signal for a DMA transfer request.

When the DMAREQZ_p signal is used as a DMA transfer request, use DMAACKZ_p 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 (CHCFG_n).

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 (CHCFG_n). For information about the relationship between DMA transfer requests and DMA units, see Figure 13.1, Relation between DMA Units/Channels and DMA Triggers.

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 13.27 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-INTPZ31)	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 pin (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 13.28 DMA Acknowledge Signal (DMAACKZp) Output Mode

AMn2	AMn1	AMn0	DMA Acknowledge Signal (DMAACKZp) Output Mode
0	0	0	Pulse mode ^{Note1} (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 ^{Note2} 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.

- 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.

- 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 $1 \times \text{BUSCLK}$ is output.

If the pulse width is insufficient for the DMA transfer request source, the width of DMAACKZp can be set from $1 \times \text{BUSCLK}$ to $32 \times \text{BUSCLK}$ by using the AKWD4 to AKWD0 bits of the DMA transfer interface signal control registers 0 to 3 (DMAIFC0 to DMAIFC3).

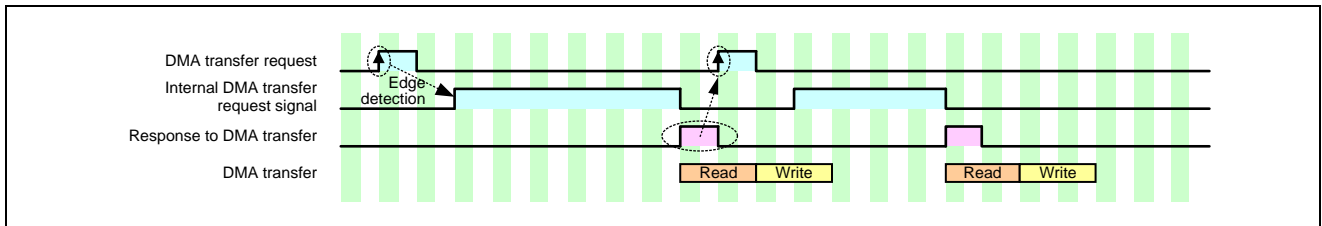


Figure 13.26 Pulse Output Mode Operation Example 1

DMA transfer request: Rising edge detection
 Request from the source (CHCFGn.REQD = 0)
 DMA acknowledge output: Pulse mode

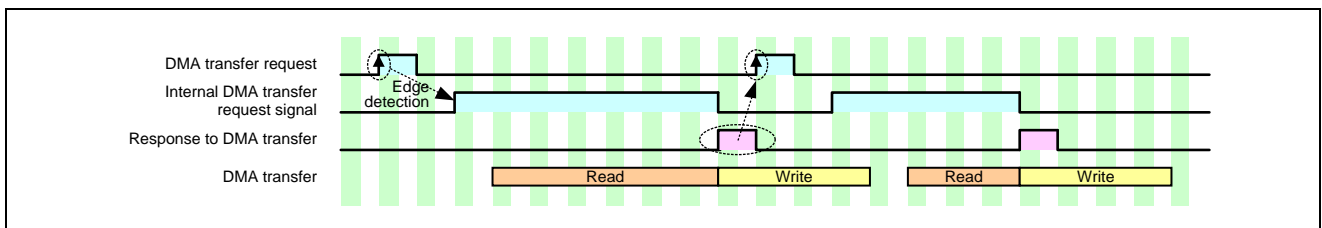


Figure 13.27 Pulse Output Mode Operation Example 2

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.

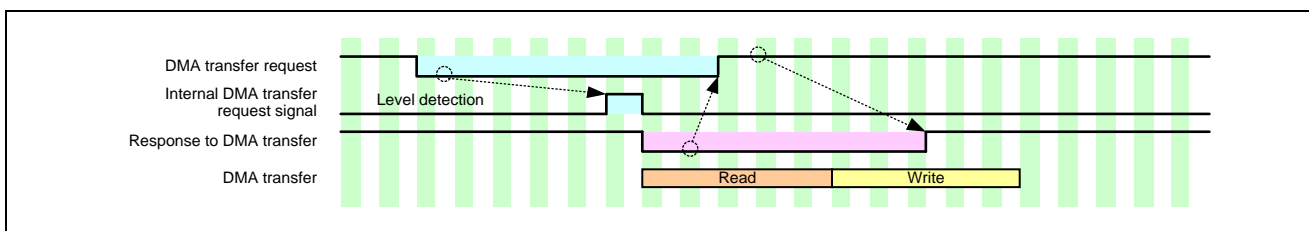


Figure 13.28 Level Output Mode Operation Example 1

DMA transfer request: High level detection
 Request from the source (CHCFGn.REQD = 0)
 DMA acknowledge output: Level mode

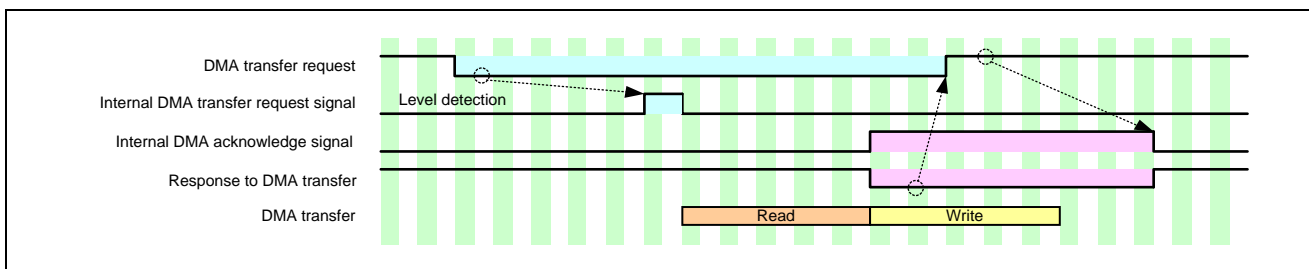


Figure 13.29 Level Output Mode Operation Example 2

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.

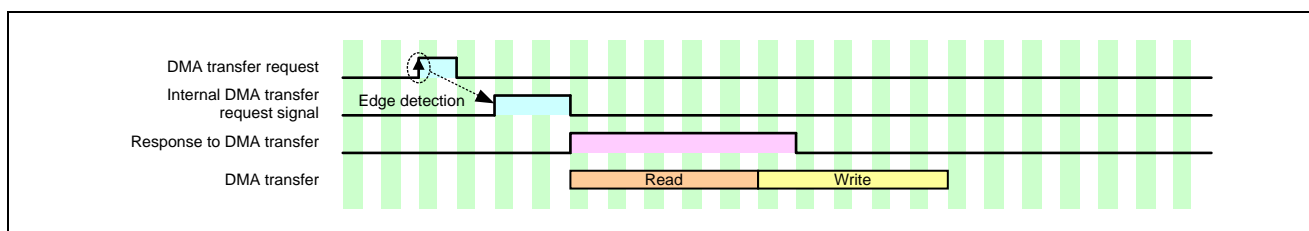


Figure 13.30 Bus Cycle Output Mode Operation Example 1

DMA transfer request: Rising edge detection
 Request from the source (CHCFGn.REQD = 0)
 DMA acknowledge output: Level mode

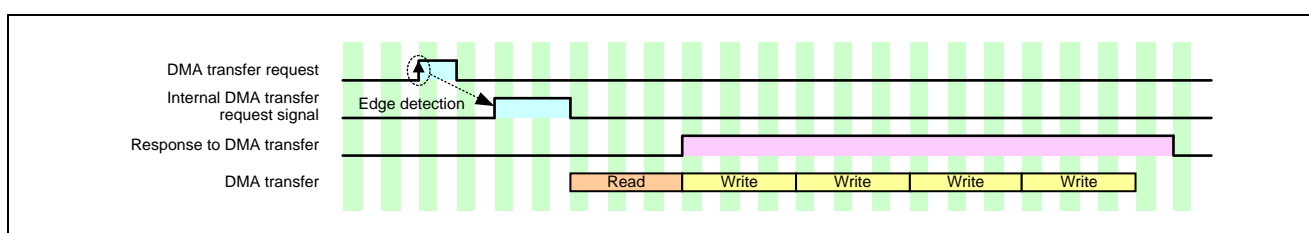


Figure 13.31 Bus Cycle Output Mode Operation Example 2

DMA transfer request: High level 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

13.8.5 DMA Transfer Completion Interrupt

When a DMA transaction (the series of DMA transfers) is completed, the pulse output of INTDMA_n occurs. For the relationship between INTDMA_n and units/channels, see Table 13.29, Relationship between DMA Transfer Completion Interrupts and Units/Channels.

When the total numbers of transfer bytes loaded to the Current transaction byte register (CRTB_n) have successfully been transferred, the END bit of the channel status register n (CHSTAT_n) is set to 1. In this case, when the DEM bit of the channel configuration register (CHCFG_n) is cleared to 0, INTDMA_n occurs.

When writeback is performed in link mode, INTDMA_n occurs after the writeback operation. Also, when a descriptor is read in link mode and CHCFG_n.DRRP is set to 0, the CHSTAT_n.DER bit is set to 1 if LV of the read descriptor header is set to 0. In this case, when CHCFG_n.DIM is set to 0, INTDMA_n occurs.

Table 13.29 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 13.30 DMA Transfer Completion Interrupt Asserting Conditions

Source	Condition	INTDMA _n Mask Setting Bit
DMA transaction completion	The total numbers of transfer bytes loaded to the Current transaction byte register (CRTB _n) have been successfully transferred. (When writeback is performed in link mode, the interrupt occurs after the writeback operation.)	CHCFG _n .DEM
Invalid descriptor (LV of the header = 0)	When DRRP and DIM of the channel configuration register (CHCFG _n) are set to 0 in link mode, the LV of the read descriptor header is set to 0.	CHCFG _n .DIM

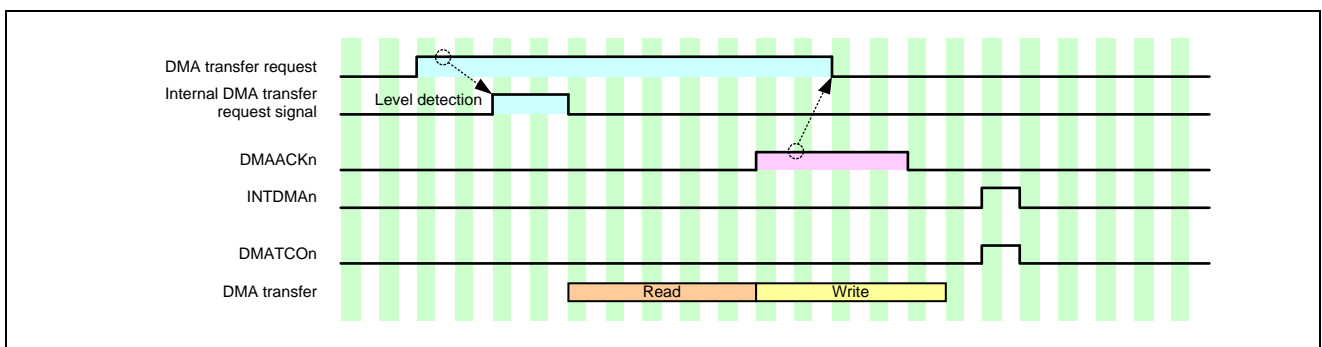


Figure 13.32 DMA Transfer Completion Interrupt Output Operation Example

DMA transfer request: High level detection, request from the destination.
 DMA acknowledge output: Pulse mode

Remark: n = 0 to 3

13.8.6 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 n (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.

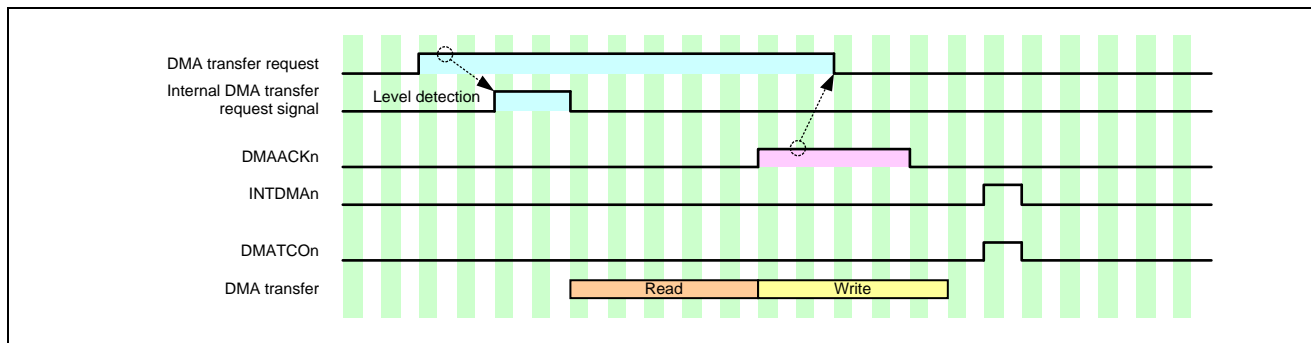


Figure 13.33 DMA Terminal Count Output Operation Example

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 13.31 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: <ul style="list-style-type: none"> • 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 13.1, Relation between DMA Units/Channels and DMA Triggers.

Remark: n = 0 to 3

13.8.7 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 n (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

13.8.8 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 n (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

13.8.9 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 \times 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

13.8.10 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.

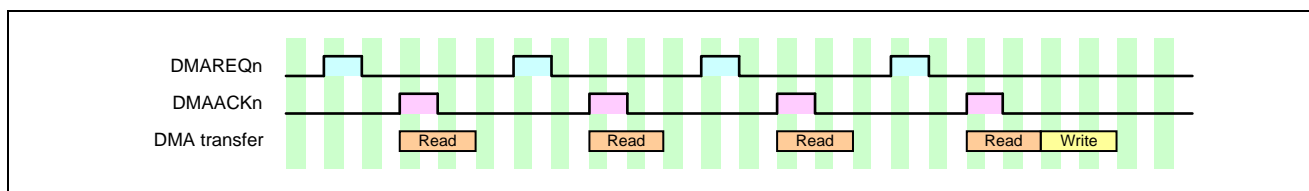


Figure 13.34 When the Transfer Size of the Source Is Smaller Than That of the Destination

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.

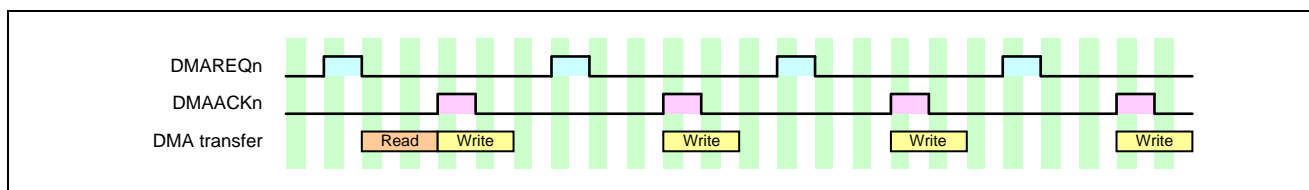


Figure 13.35 When the Transfer Size of the Destination Is Smaller Than That of the Source

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.

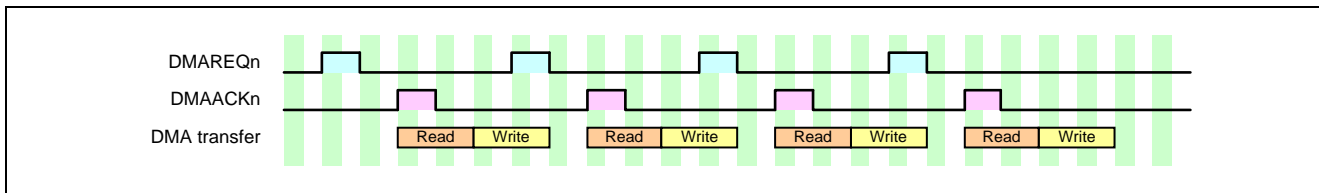


Figure 13.36 When the Transfer Size of the Destination Is Equal to That of the Source

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

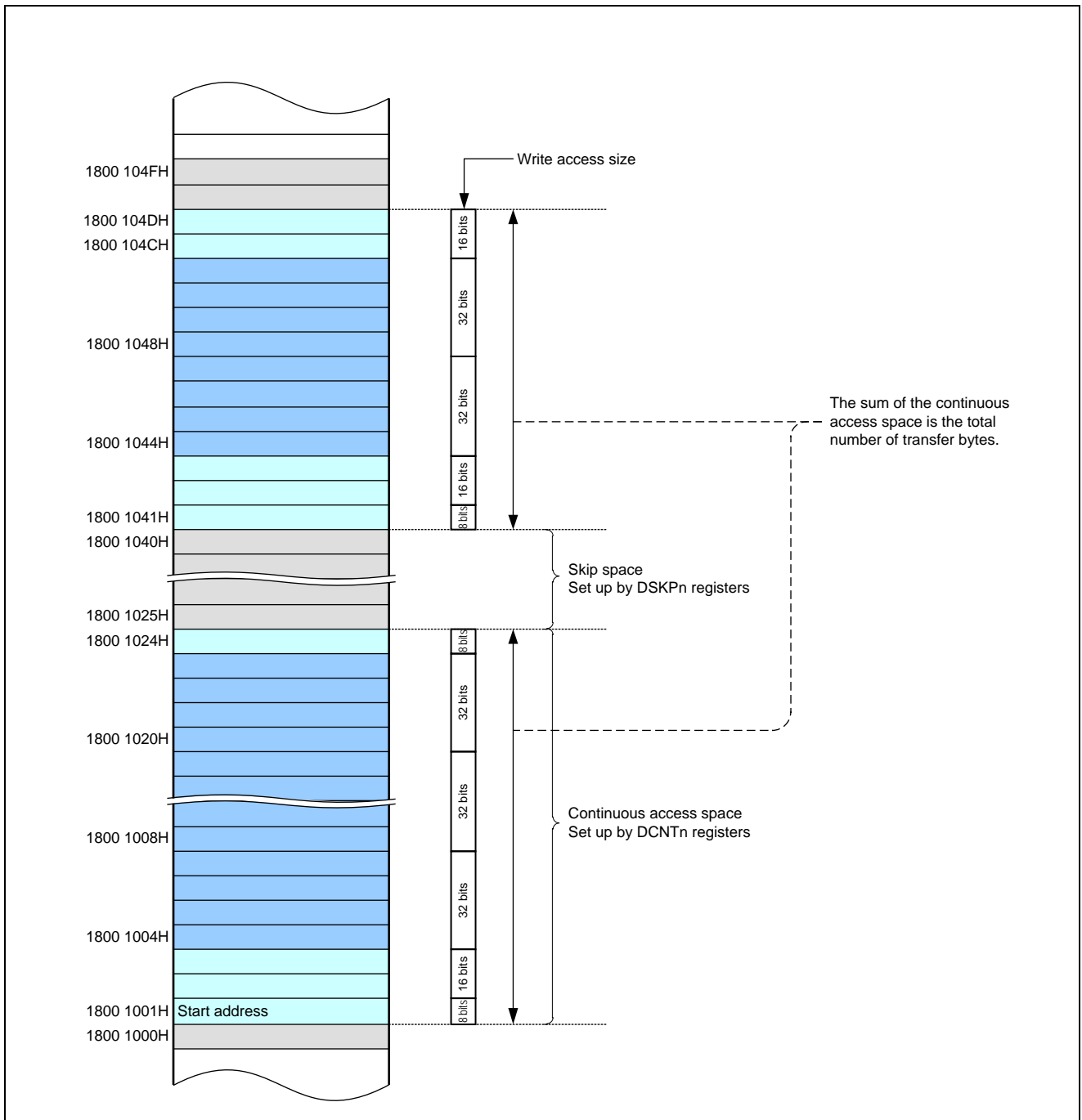


Figure 13.37 DMA Write Access and Access Type Example

13.8.11 Transfer Status

The state of transfer by DMA channel n can be checked by using the channel status register n (CHSTAT n).

The TACT bit of the CHSTAT n register indicates whether channel n is active. When the SETEN bit of the channel control register (CHCTRL n) 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 CHSTAT n 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 CHSTAT n register is not met (for example, CHCFG n .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

13.8.12 Suspension

DMA transfer can be suspended by setting the SETSUS bit of the channel control register (CHCTRL n) 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 (CHCTRL n) to 1 leads to release from the suspended state.

To check the suspended state, set SETSUS to 1 and then read the CHSTAT n register or DSTSUS register to see that the SUS bit is set to 1 for the relevant channel.

Remark: $n = 0$ to 3

13.8.13 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 n (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.

13.9 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 13.32 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

13.9.1 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 13.33.

Table 13.33 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 13.34 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 13.35 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			Address		
	400A 286CH																																
	Initial value																																
	0000 0000H																																
Set value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Bit Position	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	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	0: Enables INTDMA01 output when a DMA transaction is completed.								
23	WONLY	0: Normal mode.								
22	TM	0: Single 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.								
18-16	DDS2- DDS0	<table border="1" style="width: 100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width: 10%;">DDS2</td> <td style="width: 10%;">DDS1</td> <td style="width: 10%;">DDS0</td> <td style="width: 70%;">DMA Transfer Destination Transfer Size</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>32 bits</td> </tr> </table>	DDS2	DDS1	DDS0	DMA Transfer Destination Transfer Size	0	1	0	32 bits
DDS2	DDS1	DDS0	DMA Transfer Destination Transfer Size							
0	1	0	32 bits							
15	SDS3	0: Uses the normal addressing mode for the source.								
14 to 12	SDS2- SDS0	<table border="1" style="width: 100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width: 10%;">SDS2</td> <td style="width: 10%;">SDS1</td> <td style="width: 10%;">SDS0</td> <td style="width: 70%;">Uses the Normal Addressing Mode for the Source</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>32 bits</td> </tr> </table>	SDS2	SDS1	SDS0	Uses the Normal Addressing Mode for the Source	0	1	0	32 bits
SDS2	SDS1	SDS0	Uses the Normal Addressing Mode for the Source							
0	1	0	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	<table border="1" style="width: 100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width: 10%;">AM2</td> <td style="width: 10%;">AM1</td> <td style="width: 10%;">AM0</td> <td style="width: 70%;">DMA Acknowledge Signal (DMAACKZ1) Output Mode</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Pulse mode (initial value)</td> </tr> </table>	AM2	AM1	AM0	DMA Acknowledge Signal (DMAACKZ1) Output Mode	0	0	0	Pulse mode (initial value)
AM2	AM1	AM0	DMA Acknowledge Signal (DMAACKZ1) Output Mode							
0	0	0	Pulse mode (initial value)							
6	LVL	<table border="1" style="width: 100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width: 10%;">LVL</td> <td style="width: 10%;">HEN</td> <td style="width: 10%;">LEN</td> <td style="width: 70%;">DMA Transfer Request Signal Detection Method</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Rising edge detection</td> </tr> </table>	LVL	HEN	LEN	DMA Transfer Request Signal Detection Method	0	1	0	Rising edge detection
LVL	HEN		LEN	DMA Transfer Request Signal Detection Method						
0	1		0	Rising edge detection						
5	LEN									
4	HEN									
3	REQD	0: DMAACKZ1 becomes active when read (DMAREQZ1 is the source)								
2 to 0	SEL2- SEL0	<table border="1" style="width: 100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width: 10%;">SEL2</td> <td style="width: 10%;">SEL1</td> <td style="width: 10%;">SEL0</td> <td style="width: 70%;">DMA Interface Signal Selection</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>The DMA transfer source selected by DTFR1 is chosen.</td> </tr> </table>	SEL2	SEL1	SEL0	DMA Interface Signal Selection	0	0	1	The DMA transfer source selected by DTFR1 is chosen.
SEL2	SEL1	SEL0	DMA Interface Signal Selection							
0	0	1	The DMA transfer source selected by DTFR1 is chosen.							

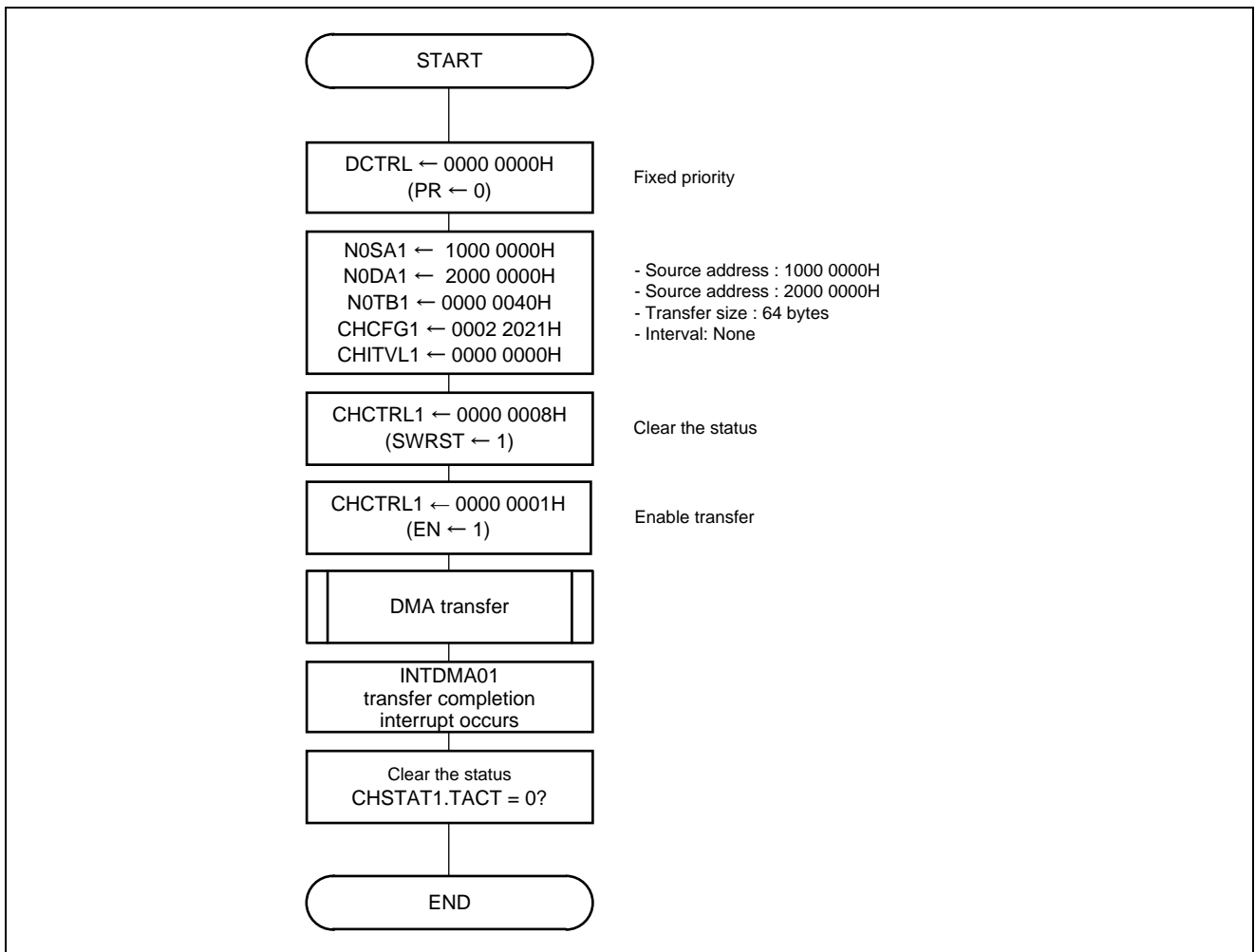


Figure 13.38 Operation Flow of Setting Example 1

13.9.2 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 13.36.

Table 13.36 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 13.37 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 13.38 Channel Configuration Register (CHCFG2) Settings of Setting Example 2

CHCFG2	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																					
	<table border="1" style="width:100%; border-collapse: collapse; font-size: small;"> <tr> <td style="width: 2.5%;">DMS</td><td style="width: 2.5%;">REN</td><td style="width: 2.5%;">RSW</td><td style="width: 2.5%;">RSEL</td><td style="width: 2.5%;">SBE</td><td style="width: 2.5%;">DIM</td><td style="width: 2.5%;">TCM</td><td style="width: 2.5%;">DEM</td><td style="width: 2.5%;">WONLY</td><td style="width: 2.5%;">TM</td><td style="width: 2.5%;">DAD</td><td style="width: 2.5%;">SAD</td><td style="width: 5%;">DDS3- DDS0</td><td style="width: 5%;">SDS3- SDS0</td><td style="width: 2.5%;">DRRP</td><td style="width: 2.5%;">AM2- AM0</td><td style="width: 2.5%;">0</td><td style="width: 2.5%;">LVL</td><td style="width: 2.5%;">LEN</td><td style="width: 2.5%;">HEN</td><td style="width: 2.5%;">REQD</td><td style="width: 5%;">SEL2- SEL0</td> </tr> </table>	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
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		
	Setting	Initial value																					
	0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0	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	<table border="1" style="width:100%; border-collapse: collapse; font-size: x-small;"> <tr> <td style="width: 10%;">DDS2</td> <td style="width: 10%;">DDS1</td> <td style="width: 10%;">DDS0</td> <td style="width: 70%;">DMA Transfer Destination Transfer Size</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>256 bits</td> </tr> </table>	DDS2	DDS1	DDS0	DMA Transfer Destination Transfer Size	1	0	1	256 bits
DDS2	DDS1	DDS0	DMA Transfer Destination Transfer Size							
1	0	1	256 bits							
15	SDS3	0: Uses the normal addressing mode for the source.								
14 to 12	SDS2- SDS0	<table border="1" style="width:100%; border-collapse: collapse; font-size: x-small;"> <tr> <td style="width: 10%;">SDS2</td> <td style="width: 10%;">SDS1</td> <td style="width: 10%;">SDS0</td> <td style="width: 70%;">DMA Transfer Source Transfer Size</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>8 bits</td> </tr> </table>	SDS2	SDS1	SDS0	DMA Transfer Source Transfer Size	0	0	0	8 bits
SDS2	SDS1	SDS0	DMA Transfer Source Transfer Size							
0	0	0	8 bits							
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	<table border="1" style="width:100%; border-collapse: collapse; font-size: x-small;"> <tr> <td style="width: 10%;">AM2</td> <td style="width: 10%;">AM1</td> <td style="width: 10%;">AM0</td> <td style="width: 70%;">DMA Acknowledge Signal Output Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">X</td> <td style="text-align: center;">X</td> <td>Disables DMA acknowledge signal output.</td> </tr> </table>	AM2	AM1	AM0	DMA Acknowledge Signal Output Mode	1	X	X	Disables DMA acknowledge signal output.
AM2	AM1	AM0	DMA Acknowledge Signal Output Mode							
1	X	X	Disables DMA acknowledge signal output.							
6	LVL	<table border="1" style="width:100%; border-collapse: collapse; font-size: x-small;"> <tr> <td style="width: 10%;">LVL</td> <td style="width: 10%;">HEN</td> <td style="width: 10%;">LEN</td> <td style="width: 70%;">DMA Transfer Request Signal Detection Method</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Disables detection.</td> </tr> </table>	LVL	HEN	LEN	DMA Transfer Request Signal Detection Method	0	0	0	Disables detection.
LVL	HEN		LEN	DMA Transfer Request Signal Detection Method						
0	0		0	Disables detection.						
5	LEN									
4	HEN									
3	REQD	0: Acknowledge becomes active when read.								
2 to 0	SEL2- SEL0	<table border="1" style="width:100%; border-collapse: collapse; font-size: x-small;"> <tr> <td style="width: 10%;">SEL2</td> <td style="width: 10%;">SEL1</td> <td style="width: 10%;">SEL0</td> <td style="width: 70%;">DMA Interface Signal Selection</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>DMA transfer source selected by DTFR2 is chosen.</td> </tr> </table>	SEL2	SEL1	SEL0	DMA Interface Signal Selection	0	1	0	DMA transfer source selected by DTFR2 is chosen.
SEL2	SEL1	SEL0	DMA Interface Signal Selection							
0	1	0	DMA transfer source selected by DTFR2 is chosen.							

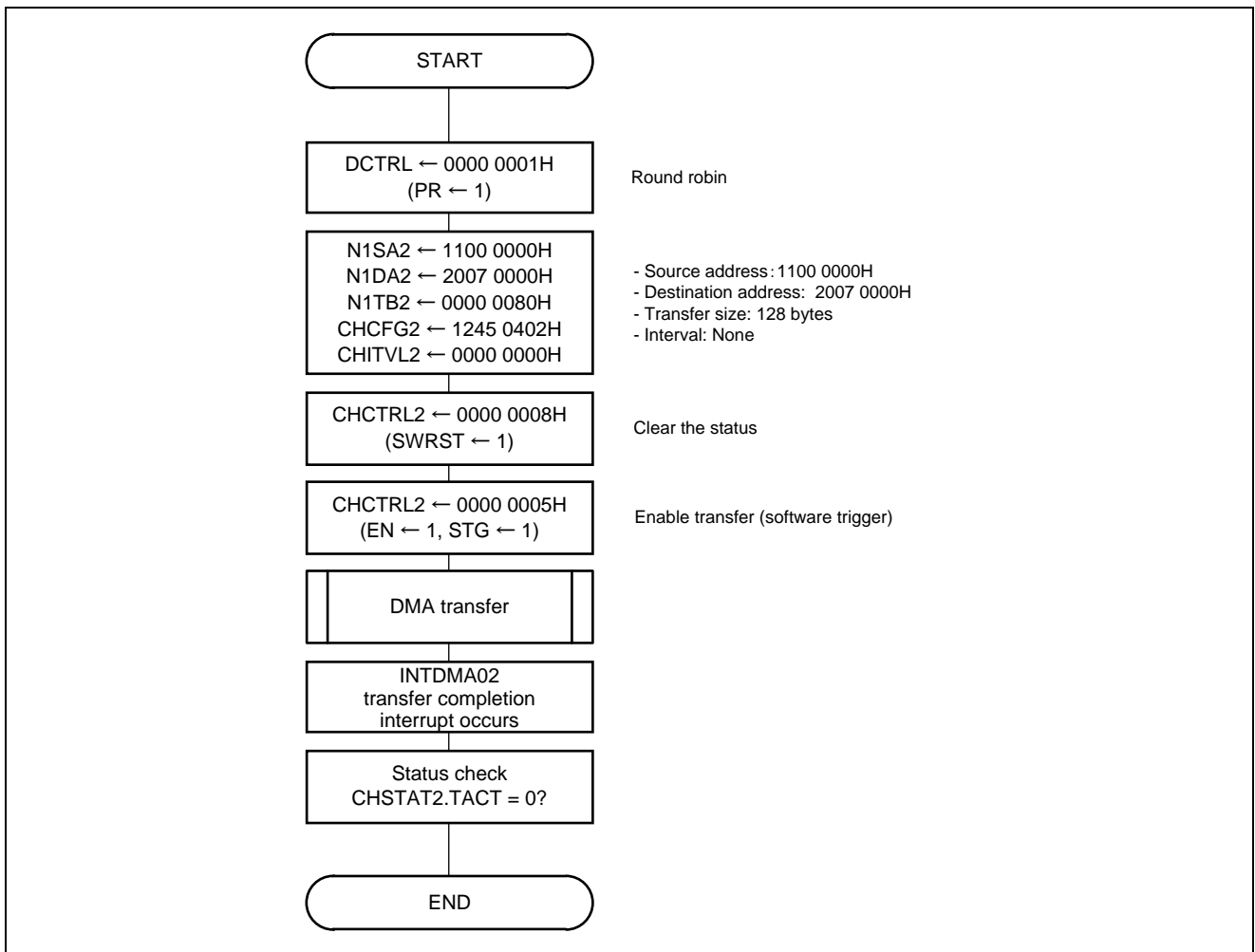


Figure 13.39 Operation Flow of Setting Example 2

13.9.3 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 13.39.

Table 13.39 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 13.40 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 13.41 Channel Configuration Register (CHCFG1) Settings of Setting Example 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	Address
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- SELO	400A 286CH										
	0	1	0	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	Initial value 0000 0000H
Setting	0	1	0	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	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	<table border="1" style="width:100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width:10%;">DDS2</td> <td style="width:10%;">DDS1</td> <td style="width:10%;">DDS0</td> <td style="width:70%;">DMA Destination Transfer Size</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>512 bits</td> </tr> </table>	DDS2	DDS1	DDS0	DMA Destination Transfer Size	1	1	0	512 bits
DDS2	DDS1	DDS0	DMA Destination Transfer Size							
1	1	0	512 bits							
15	SDS3	0: Uses the normal addressing mode for the source.								
14 to 12	SDS2- SDS0	<table border="1" style="width:100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width:10%;">SDS2</td> <td style="width:10%;">SDS1</td> <td style="width:10%;">SDS0</td> <td style="width:70%;">DMA Source Transfer Size</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>32 bits</td> </tr> </table>	SDS2	SDS1	SDS0	DMA Source Transfer Size	0	1	0	32 bits
SDS2	SDS1	SDS0	DMA Source Transfer Size							
0	1	0	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	<table border="1" style="width:100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width:10%;">AM2</td> <td style="width:10%;">AM1</td> <td style="width:10%;">AM0</td> <td style="width:70%;">DMA Acknowledge Signal (DMAACKZ1) Output Mode</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Pulse mode</td> </tr> </table>	AM2	AM1	AM0	DMA Acknowledge Signal (DMAACKZ1) Output Mode	0	0	0	Pulse mode
AM2	AM1	AM0	DMA Acknowledge Signal (DMAACKZ1) Output Mode							
0	0	0	Pulse mode							
6	LVL	<table border="1" style="width:100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width:10%;">LVL</td> <td style="width:10%;">HEN</td> <td style="width:10%;">LEN</td> <td style="width:70%;">DMA Transfer Request Signal Detection Method</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Disables detection.</td> </tr> </table>	LVL	HEN	LEN	DMA Transfer Request Signal Detection Method	0	0	0	Disables detection.
LVL	HEN		LEN	DMA Transfer Request Signal Detection Method						
0	0		0	Disables detection.						
5	LEN									
4	HEN									
3	REQD	0: DMAACKZ1 becomes active when read (DMAREQZ1 is the source).								
2 to 0	SEL2- SELO	<table border="1" style="width:100%; border-collapse: collapse; margin: 5px;"> <tr> <td style="width:10%;">SEL2</td> <td style="width:10%;">SEL1</td> <td style="width:10%;">SELO</td> <td style="width:70%;">DMA Interface Signal Selection</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>DMA transfer source selected by DTFR1 is chosen.</td> </tr> </table>	SEL2	SEL1	SELO	DMA Interface Signal Selection	0	0	1	DMA transfer source selected by DTFR1 is chosen.
SEL2	SEL1	SELO	DMA Interface Signal Selection							
0	0	1	DMA transfer source selected by DTFR1 is chosen.							

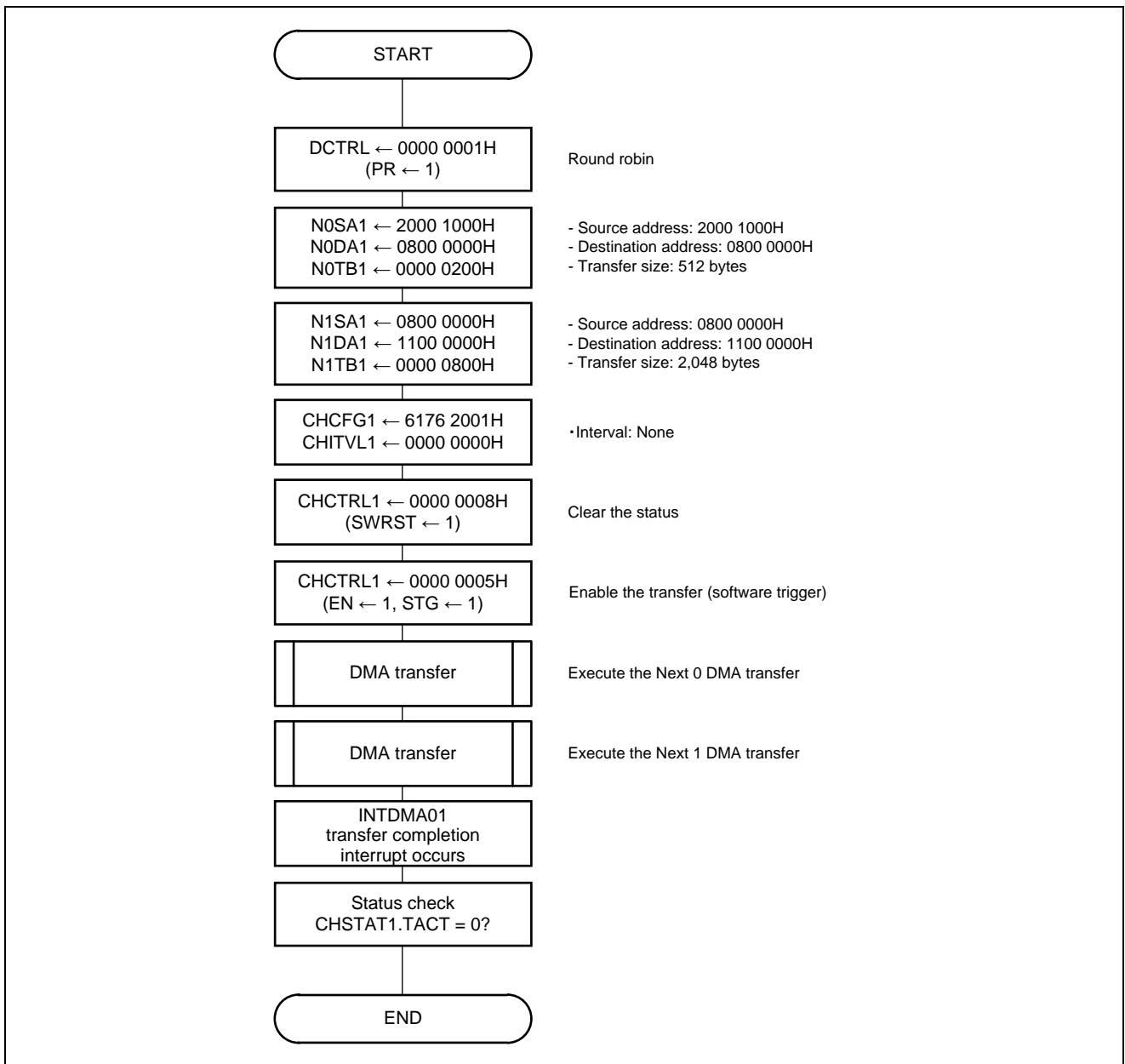


Figure 13.40 Operation Flow of Setting Example 3

13.9.4 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 13.42 to Table 13.47.

Table 13.42 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 13.43 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 13.44 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	64 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 13.45 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 13.46 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 13.47 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

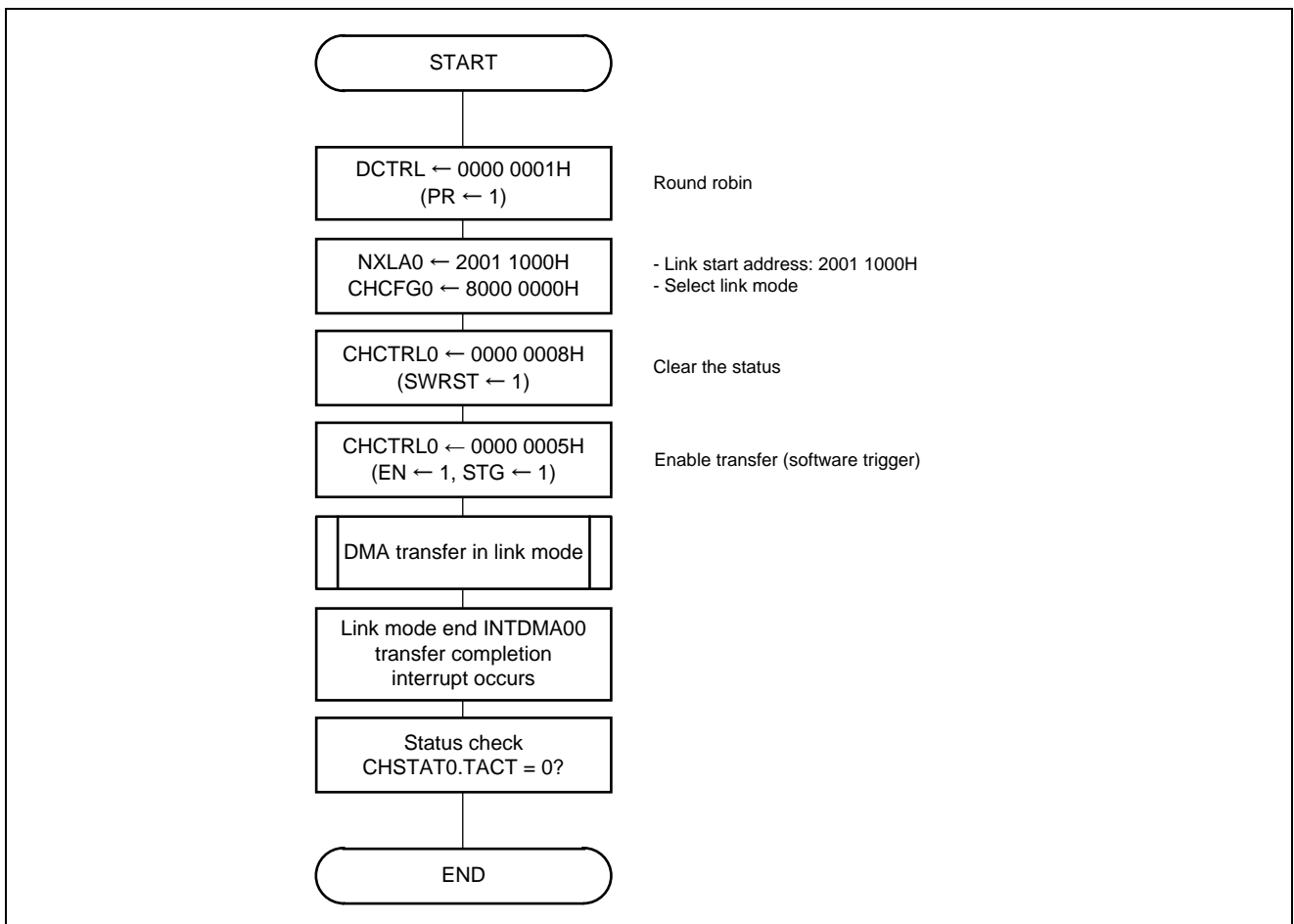


Figure 13.41 Operation Flow of Setting Example 4

13.10 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

14. Timer Array Unit (TAUJ2)

This section explains the timer array unit (TAUJ2).

14.1 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.
- I/O signals: The I/O signals of TAUJ2 are listed in the following table.

Table 14.1 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)

- 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), and for updating the real-time port pins (RP00-RP37).

Table 14.2 TAUJ2 Interrupt Signals and Requests for Peripheral Modules

TAUJ2 signal	Function	Connected to
INTTAUJ2I0	Channel 0 interrupt	<ul style="list-style-type: none"> • Interrupt Controller TAUJ2I0 • DMA Controller trigger (DTFR/RTDTFR) • Timer Capture trigger (TMTFR) • Real-time Port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTTAUJ2I1	Channel 1 interrupt	<ul style="list-style-type: none"> • Interrupt Controller TAUJ2I1 • DMA Controller trigger (DTFR/RTDTFR) • Timer Capture trigger (TMTFR) • Real-time Port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTTAUJ2I2	Channel 2 interrupt	<ul style="list-style-type: none"> • Interrupt Controller TAUJ2I2 • DMA Controller trigger (DTFR/RTDTFR) • Timer Capture trigger (TMTFR) • Real-time Port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTTAUJ2I3	Channel 3 interrupt	<ul style="list-style-type: none"> • Interrupt Controller TAUJ2I3 • DMA Controller trigger (DTFR/RTDTFR) • Timer Capture trigger (TMTFR) • Real-time Port trigger (RPTFR) • HW-RTOS (Hardware ISR)

Caution: Since TIN_m and TOUT_m are multiplexed on the same port pins, the input pin function for TIN_m must be set to a pin other than a port pin (m = 0 to 3).
For details, see section 21.9.1, Timer Input Function Selection Register (SELCNT).

14.1.1 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by operating combinations of multiple channels.

Caution: TAUJ2 only supports usage described in Table 14.3, TAUJ2 Operations. Settings of the registers for usage other than those listed in Table 14.3 are prohibited.

Table 14.3 TAUJ2 Operations

Operation	Description
Independent channel operation	
14.7.1 "Interval Timer"	An interrupt is output at a regular interval.
14.7.2 "TAUJ2TTINm Input Interval Timer"	An interrupt is output at a regular interval or in response to an effective edge of an externally input signal.
14.7.3 "Delay Counting"	Interrupts which have a defined delay relative to input of the effective edge of an externally input signal are output.
14.7.4 "TAUJ2TTINm Input Pulse Interval Measurement"	The time of the input interval of an externally input signal is measured.
14.7.5 "TAUJ2TTINm Input Signal Width Measurement"	The signal width of an externally input signal is measured.
14.7.6 "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.
14.7.7 "TAUJ2TTINm Input Position Detection"	The interval from the start of counting to an effective edge of an externally input signal is measured.
Synchronous channel operation	
14.8.1 "PWM Output"	PWM waveform is output.

14.2 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

14.2.1 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:

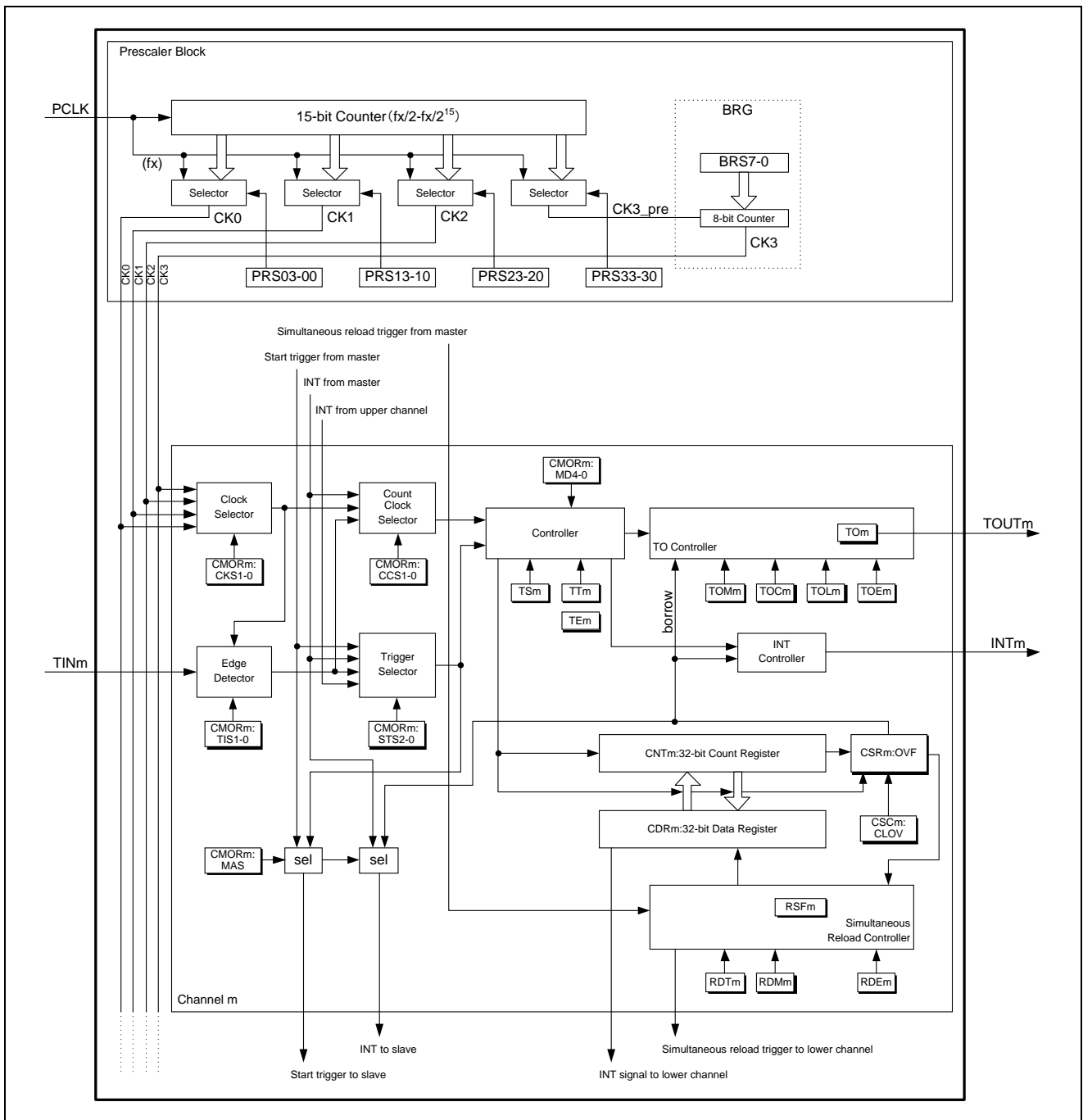


Figure 14.1 Block Diagram of the TAUJ2

14.2.2 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.

14.3 Registers

This section describes all the registers of the 32-bit TAUJ2.

Caution: TAUJ2 only supports usage described in Table 14.3, TAUJ2 Operations. Settings of the registers for usage other than those listed in Table 14.3 are prohibited.

14.3.1 TAUJ2 Registers Overview

The TAUJ2 is controlled and operated by the registers in the following table.

Table 14.4 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

14.3.2 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).

(1/4)

	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	R/W	R/W

Bit Position	Bit Name	Function																																		
15 to 12	TAUJ2PRS3[3:0]	<p>Specifies the CK3_PRE clock.</p> <p>Clock CK3_PRE is the input clock of the BRG unit. The BRG unit supplies the CK3 operation clock for all channels.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <thead> <tr> <th style="width: 50%;">TAUJ2PRS3[3:0]</th> <th style="width: 50%;">CK3_PRE Clock</th> </tr> </thead> <tbody> <tr><td>0000B</td><td>PCLK/2⁰</td></tr> <tr><td>0001B</td><td>PCLK/2¹</td></tr> <tr><td>0010B</td><td>PCLK/2²</td></tr> <tr><td>0011B</td><td>PCLK/2³</td></tr> <tr><td>0100B</td><td>PCLK/2⁴</td></tr> <tr><td>0101B</td><td>PCLK/2⁵</td></tr> <tr><td>0110B</td><td>PCLK/2⁶</td></tr> <tr><td>0111B</td><td>PCLK/2⁷</td></tr> <tr><td>1000B</td><td>PCLK/2⁸</td></tr> <tr><td>1001B</td><td>PCLK/2⁹</td></tr> <tr><td>1010B</td><td>PCLK/2¹⁰</td></tr> <tr><td>1011B</td><td>PCLK/2¹¹</td></tr> <tr><td>1100B</td><td>PCLK/2¹²</td></tr> <tr><td>1101B</td><td>PCLK/2¹³</td></tr> <tr><td>1110B</td><td>PCLK/2¹⁴</td></tr> <tr><td>1111B</td><td>PCLK/2¹⁵</td></tr> </tbody> </table> <p>These bits can only be rewritten while all counters using CK3 are stopped (TAUJ2TE.TAUJ2TEm = 0).</p>	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
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																																			

(2/4)

Bit Position	Bit Name	Function																																		
11 to 8	TAUJ2PRS2[3:0]	Specifies the CK2 clock. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>TAUJ2PRS2[3:0]</th> <th>CK2 Clock</th> </tr> </thead> <tbody> <tr><td>0000B</td><td>PCLK/2⁰</td></tr> <tr><td>0001B</td><td>PCLK/2¹</td></tr> <tr><td>0010B</td><td>PCLK/2²</td></tr> <tr><td>0011B</td><td>PCLK/2³</td></tr> <tr><td>0100B</td><td>PCLK/2⁴</td></tr> <tr><td>0101B</td><td>PCLK/2⁵</td></tr> <tr><td>0110B</td><td>PCLK/2⁶</td></tr> <tr><td>0111B</td><td>PCLK/2⁷</td></tr> <tr><td>1000B</td><td>PCLK/2⁸</td></tr> <tr><td>1001B</td><td>PCLK/2⁹</td></tr> <tr><td>1010B</td><td>PCLK/2¹⁰</td></tr> <tr><td>1011B</td><td>PCLK/2¹¹</td></tr> <tr><td>1100B</td><td>PCLK/2¹²</td></tr> <tr><td>1101B</td><td>PCLK/2¹³</td></tr> <tr><td>1110B</td><td>PCLK/2¹⁴</td></tr> <tr><td>1111B</td><td>PCLK/2¹⁵</td></tr> </tbody> </table> <p style="margin-left: 20px;">These bits can only be rewritten while all counters using CK2 are stopped (TAUJ2TE.TAUJ2TEm = 0).</p>	TAUJ2PRS2[3:0]	CK2 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
TAUJ2PRS2[3:0]	CK2 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																																			

(3/4)

Bit Position	Bit Name	Function																																		
7 to 4	TAUJ2PRS1[3:0]	Specifies the CK1 clock. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>TAUJ2PRS1[3:0]</th> <th>CK1 Clock</th> </tr> </thead> <tbody> <tr><td>0000B</td><td>PCLK/2⁰</td></tr> <tr><td>0001B</td><td>PCLK/2¹</td></tr> <tr><td>0010B</td><td>PCLK/2²</td></tr> <tr><td>0011B</td><td>PCLK/2³</td></tr> <tr><td>0100B</td><td>PCLK/2⁴</td></tr> <tr><td>0101B</td><td>PCLK/2⁵</td></tr> <tr><td>0110B</td><td>PCLK/2⁶</td></tr> <tr><td>0111B</td><td>PCLK/2⁷</td></tr> <tr><td>1000B</td><td>PCLK/2⁸</td></tr> <tr><td>1001B</td><td>PCLK/2⁹</td></tr> <tr><td>1010B</td><td>PCLK/2¹⁰</td></tr> <tr><td>1011B</td><td>PCLK/2¹¹</td></tr> <tr><td>1100B</td><td>PCLK/2¹²</td></tr> <tr><td>1101B</td><td>PCLK/2¹³</td></tr> <tr><td>1110B</td><td>PCLK/2¹⁴</td></tr> <tr><td>1111B</td><td>PCLK/2¹⁵</td></tr> </tbody> </table> <p>These bits can only be rewritten while all counters using CK1 are stopped (TAUJ2TE.TAUJ2TEm = 0).</p>	TAUJ2PRS1[3:0]	CK1 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
TAUJ2PRS1[3:0]	CK1 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																																			

(4/4)

Bit Position	Bit Name	Function																																		
3 to 0	TAUJ2PRS0[3:0]	Specifies the CK0 clock. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>TAUJ2PRS0[3:0]</th> <th>CK0 Clock</th> </tr> </thead> <tbody> <tr><td>0000B</td><td>PCLK/2⁰</td></tr> <tr><td>0001B</td><td>PCLK/2¹</td></tr> <tr><td>0010B</td><td>PCLK/2²</td></tr> <tr><td>0011B</td><td>PCLK/2³</td></tr> <tr><td>0100B</td><td>PCLK/2⁴</td></tr> <tr><td>0101B</td><td>PCLK/2⁵</td></tr> <tr><td>0110B</td><td>PCLK/2⁶</td></tr> <tr><td>0111B</td><td>PCLK/2⁷</td></tr> <tr><td>1000B</td><td>PCLK/2⁸</td></tr> <tr><td>1001B</td><td>PCLK/2⁹</td></tr> <tr><td>1010B</td><td>PCLK/2¹⁰</td></tr> <tr><td>1011B</td><td>PCLK/2¹¹</td></tr> <tr><td>1100B</td><td>PCLK/2¹²</td></tr> <tr><td>1101B</td><td>PCLK/2¹³</td></tr> <tr><td>1110B</td><td>PCLK/2¹⁴</td></tr> <tr><td>1111B</td><td>PCLK/2¹⁵</td></tr> </tbody> </table> <p>These bits can only be rewritten while all counters using CK0 are stopped (TAUJ2TE.TAUJ2TE_m = 0).</p>	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
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																																			

(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).

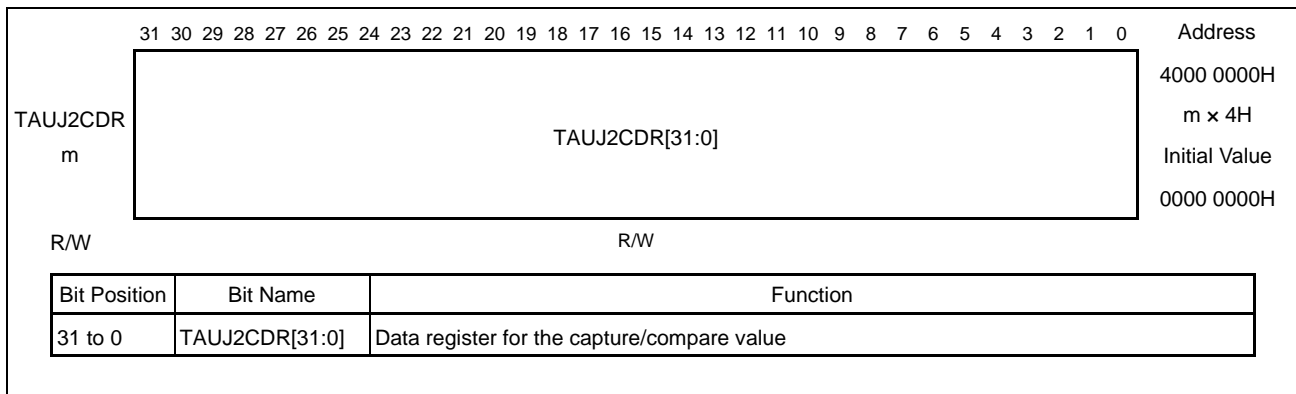
	7	6	5	4	3	2	1	0	Address	Initial Value																
TAUJ2BRS	TAUJ2BRS[7:0]								4000 0094H	00H																
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W																		
Bit Position	Bit Name	Function																								
7 to 0	TAUJ2BRS[7:0]	Specifies the CK3_PRE clock division factor for generating CK3. <table border="1" style="width: 100%; margin-top: 10px; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">TAUJ2BRS[7:0]</th> <th style="width: 50%;">CK3 clock</th> </tr> </thead> <tbody> <tr> <td>0000 0000B</td> <td>CK3_PRE / 1</td> </tr> <tr> <td>0000 0001B</td> <td>CK3_PRE / 2</td> </tr> <tr> <td>0000 0010B</td> <td>CK3_PRE / 3</td> </tr> <tr> <td>0000 0011B</td> <td>CK3_PRE / 4</td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> <tr> <td>1111 1110B</td> <td>CK3_PRE / 255</td> </tr> <tr> <td>1111 1111B</td> <td>CK3_PRE / 256</td> </tr> </tbody> </table>									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
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																									

14.3.3 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].

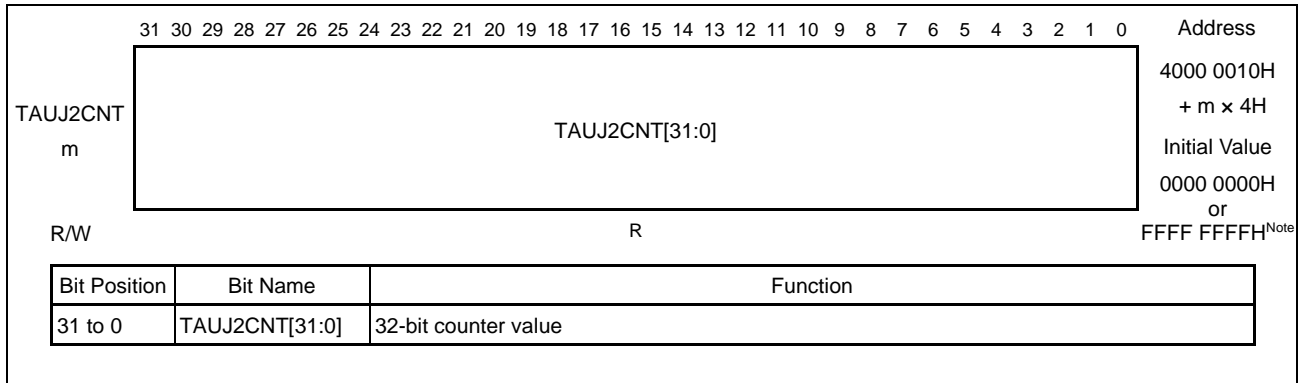
- Access This register can be read/written in 32-bit units.
 In capture mode, only reading is possible. Write operation is ignored.
 In compare mode, reading and writing is possible.



(2) TAUJ2 Channel Counter Register (TAUJ2CNTm)

This register is the channel m counter register.

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



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 14.3.3(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)

										Address	Initial Value							
										4000 0080H	0000H							
										+ m × 4H								
TAUJ2 CMORm	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	TAUJ2 CKS[1:0]		TAUJ2 CCS[1:0]		TAUJ2 MAS	TAUJ2STS[2:0]		TAUJ2 COS[1:0]		0	TAUJ2MD[4:0]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W				

Bit Position	Bit Name	Function															
15, 14	TAUJ2CKS[1:0]	<p>Selects the operation clock.</p> <p>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].</p> <table border="1"> <thead> <tr> <th>TAUJ2CKS1</th> <th>TAUJ2CKS0</th> <th>Selected Prescaler Output</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>CK0</td> </tr> <tr> <td>0</td> <td>1</td> <td>CK1</td> </tr> <tr> <td>1</td> <td>0</td> <td>CK2</td> </tr> <tr> <td>1</td> <td>1</td> <td>CK3</td> </tr> </tbody> </table>	TAUJ2CKS1	TAUJ2CKS0	Selected Prescaler Output	0	0	CK0	0	1	CK1	1	0	CK2	1	1	CK3
TAUJ2CKS1	TAUJ2CKS0	Selected Prescaler Output															
0	0	CK0															
0	1	CK1															
1	0	CK2															
1	1	CK3															
13, 12	TAUJ2CCS[1:0]	<p>Selects the count clock for TAUJ2CNTm counter.</p> <table border="1"> <thead> <tr> <th>TAUJ2CCS1</th> <th>TAUJ2CCS0</th> <th>Selected Count Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Prescaler output specified by TAUJ2CMORm.TAUJ2CKS[1:0]</td> </tr> <tr> <td>0</td> <td>1</td> <td>Effective edge of TAUJ2TTINm input signal</td> </tr> <tr> <td>1</td> <td>0</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>1</td> <td>Setting prohibited</td> </tr> </tbody> </table>	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	Setting prohibited
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	Setting prohibited															
11	TAUJ2MAS	<p>Specifies the channel as master or slave channel during synchronous channel operation.</p> <p>0: Slave 1: Master</p> <p>This bit is only valid for even-numbered channels (CHm_even). For odd-numbered channels (CHm_odd), it is fixed to 0.</p>															

(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)

Bit Position	Bit Name	Function																				
7, 6	TAUJ2COS[1:0]	<p>Specifies when the capture register TAUJ2CDRm and the overflow flag TAUJ2CSRm.TAUJ2OVF of channel m are updated.</p> <p>These bits are only valid if channel m is in capture mode.</p> <ul style="list-style-type: none"> • Capture mode • Capture & One Count mode • Capture & Gate Count mode • Count Capture mode <table border="1"> <thead> <tr> <th>TAUJ2 COS1</th> <th>TAUJ2 COS0</th> <th>TAUJ2CDRm</th> <th>TAUJ2CSRm.TAUJ2OVF</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Updated upon detection of an effective edge of the TAUJ2TTINm input signal</td> <td> <p>Updated (cleared or set) upon detection of an effective edge of the TAUJ2TTINm input signal:</p> <ul style="list-style-type: none"> • 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></td> <td>Invalid</td> </tr> <tr> <td>1</td> <td>0</td> <td> <p>Updated upon detection of an effective edge of the TAUJ2TTINm input signal and upon counter overflow:</p> <ul style="list-style-type: none"> • 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> </tbody> </table>	TAUJ2 COS1	TAUJ2 COS0	TAUJ2CDRm	TAUJ2CSRm.TAUJ2OVF	0	0	Updated upon detection of an effective edge of the TAUJ2TTINm input signal	<p>Updated (cleared or set) upon detection of an effective edge of the TAUJ2TTINm input signal:</p> <ul style="list-style-type: none"> • 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. 	0	1		Invalid	1	0	<p>Updated upon detection of an effective edge of the TAUJ2TTINm input signal and upon counter overflow:</p> <ul style="list-style-type: none"> • 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. 	Invalid	1	1	Setting prohibited	
TAUJ2 COS1	TAUJ2 COS0	TAUJ2CDRm	TAUJ2CSRm.TAUJ2OVF																			
0	0	Updated upon detection of an effective edge of the TAUJ2TTINm input signal	<p>Updated (cleared or set) upon detection of an effective edge of the TAUJ2TTINm input signal:</p> <ul style="list-style-type: none"> • 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. 																			
0	1		Invalid																			
1	0	<p>Updated upon detection of an effective edge of the TAUJ2TTINm input signal and upon counter overflow:</p> <ul style="list-style-type: none"> • 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. 	Invalid																			
1	1	Setting prohibited																				

(4/4)

Bit Position	Bit Name	Function																																																																																		
5	—	Reserved. This bit is read as 0.																																																																																		
4 to 0	TAUJ2MD[4:0]	<p>Specifies the operating mode. Settings not listed in the following table are prohibited.</p> <table border="1"> <thead> <tr> <th>TAUJ2 MD4</th> <th>TAUJ2 MD3</th> <th>TAUJ2 MD2</th> <th>TAUJ2 MD1</th> <th>TAUJ2 MD0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1/0</td> <td>Interval Timer mode</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1/0</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1/0</td> <td>Capture mode</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1/0</td> <td>Event count mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1/0</td> <td>One Count mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1/0</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>Capture & One Count mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td rowspan="3">1/0</td> <td rowspan="3">Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td rowspan="3">1/0</td> <td rowspan="3">Count Capture mode</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>Capture & Gate Count mode</td> </tr> </tbody> </table>	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	Count Capture mode	1	0	1	1	1	1	0	0	1	1	0	1	0	Capture & Gate Count mode
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	Count Capture mode																																																																															
1	0	1	1																																																																																	
1	1	0	0																																																																																	
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]	<table border="1" style="width: 100%;"> <thead> <tr> <th>TAUJ2TIS1</th> <th>TAUJ2TIS0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Falling edge</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Rising edge</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Rising and falling edges (low-width measurement selection). Start trigger: falling edge Stop trigger (capture): rising edge</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Rising and falling edges (high-width measurement selection). Start trigger: rising edge Stop trigger (capture): falling edge</td> </tr> </tbody> </table> <p>Edge detection for TAUJ2TTINm input signals is performed based on the prescaler output selected by TAUJ2CMORm.TAUJ2CKS[1:0].</p>	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
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: <ul style="list-style-type: none"> • 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	TAUJ2CLOV	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 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	TAUJ2TE _m	Indicates whether counter for channel m is enabled or disabled. 0: Counter disabled 1: Counter enabled This bit is set when TAUJ2TSST _m (the synchronous channel start trigger signal) trigger input is detected or when TAUJ2TS.TAUJ2TS _m is set. Setting TAUJ2TT.TAUJ2TT _m to 1 resets this bit 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	TAUJ2TS _m	These bits enable counter operation for channel m. 1: Enables counter operation and sets TAUJ2TE.TAUJ2TE _m = 1. 0: No function

(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.

14.3.4 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.

								Address	Initial Value	
	7	6	5	4	3	2	1	0		
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.

								Address	Initial Value	
	7	6	5	4	3	2	1	0		
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).

14.3.5 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)

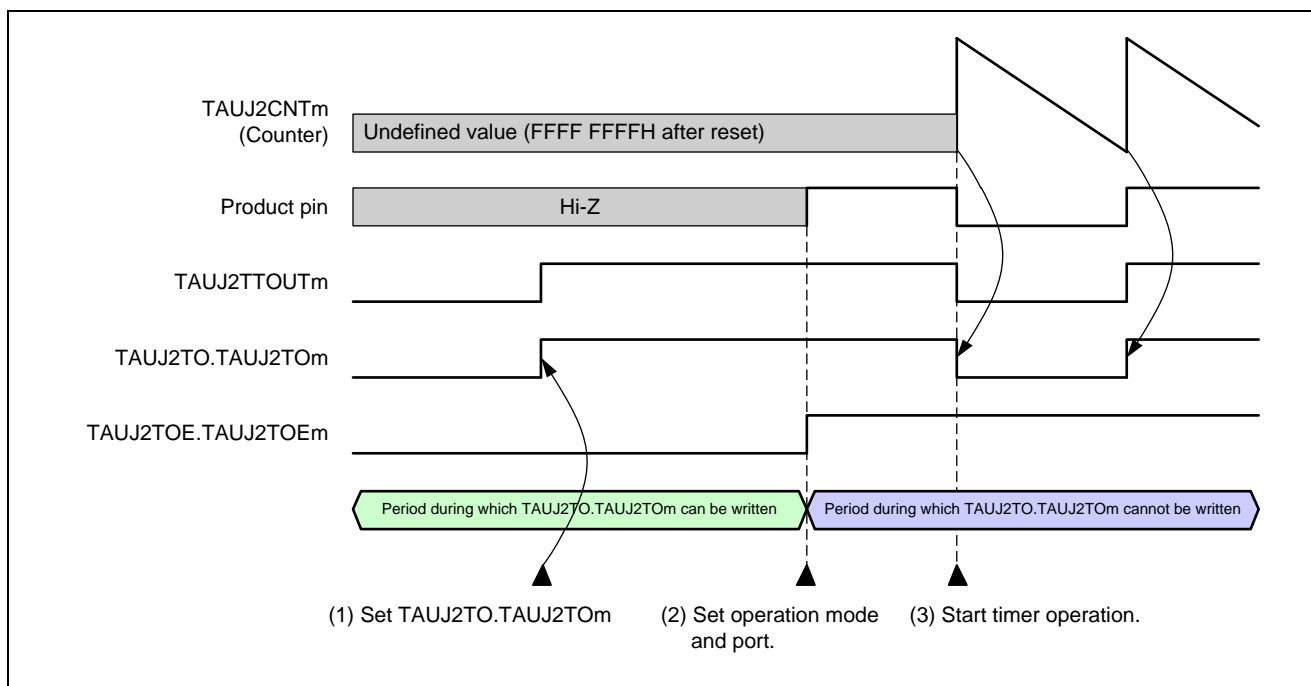


Figure 14.2 General Procedure for Specifying TAUJ2TTOUTm Channel Output Mode

(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.TAUJ2TE_m = 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	TAUJ2TOM _m	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 <ul style="list-style-type: none"> • 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.

14.4 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.

14.5 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 in section 14.5.1, Basic Rules of Synchronous Channel Operation.

14.5.1 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 channels.
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 INTTAUJ2Im and start trigger to lower channels.
8. Slave channels can use INTTAUJ2Im and start trigger of the master channels but cannot transfer their INTTAUJ2Im and start trigger to the lower channels.
9. A master channel cannot use INTTAUJ2Im 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.

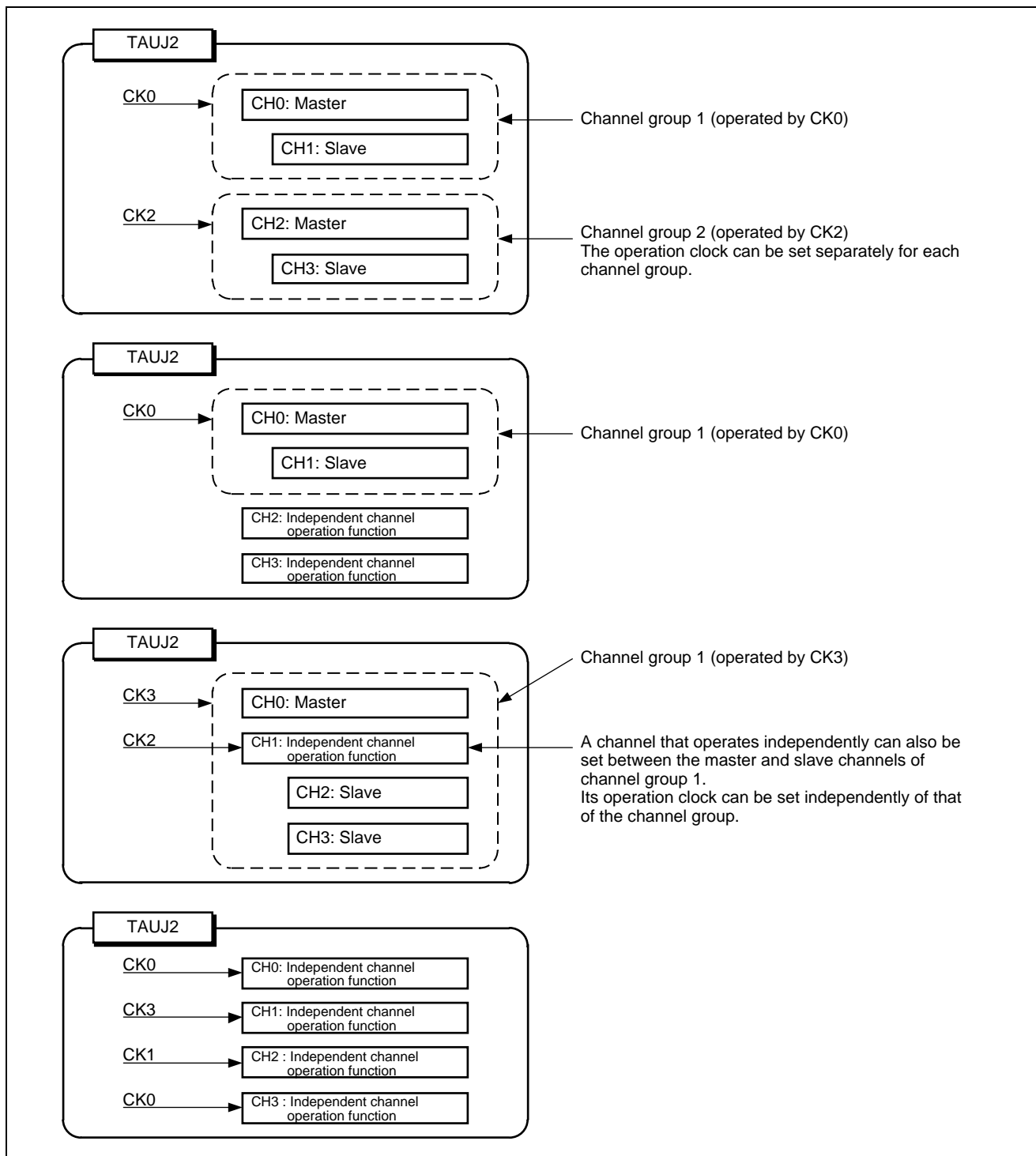


Figure 14.3 Grouping of the Channels and Assignment of Operation Clocks

14.6 Simultaneous Reloading

14.6.1 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.

14.6.2 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.

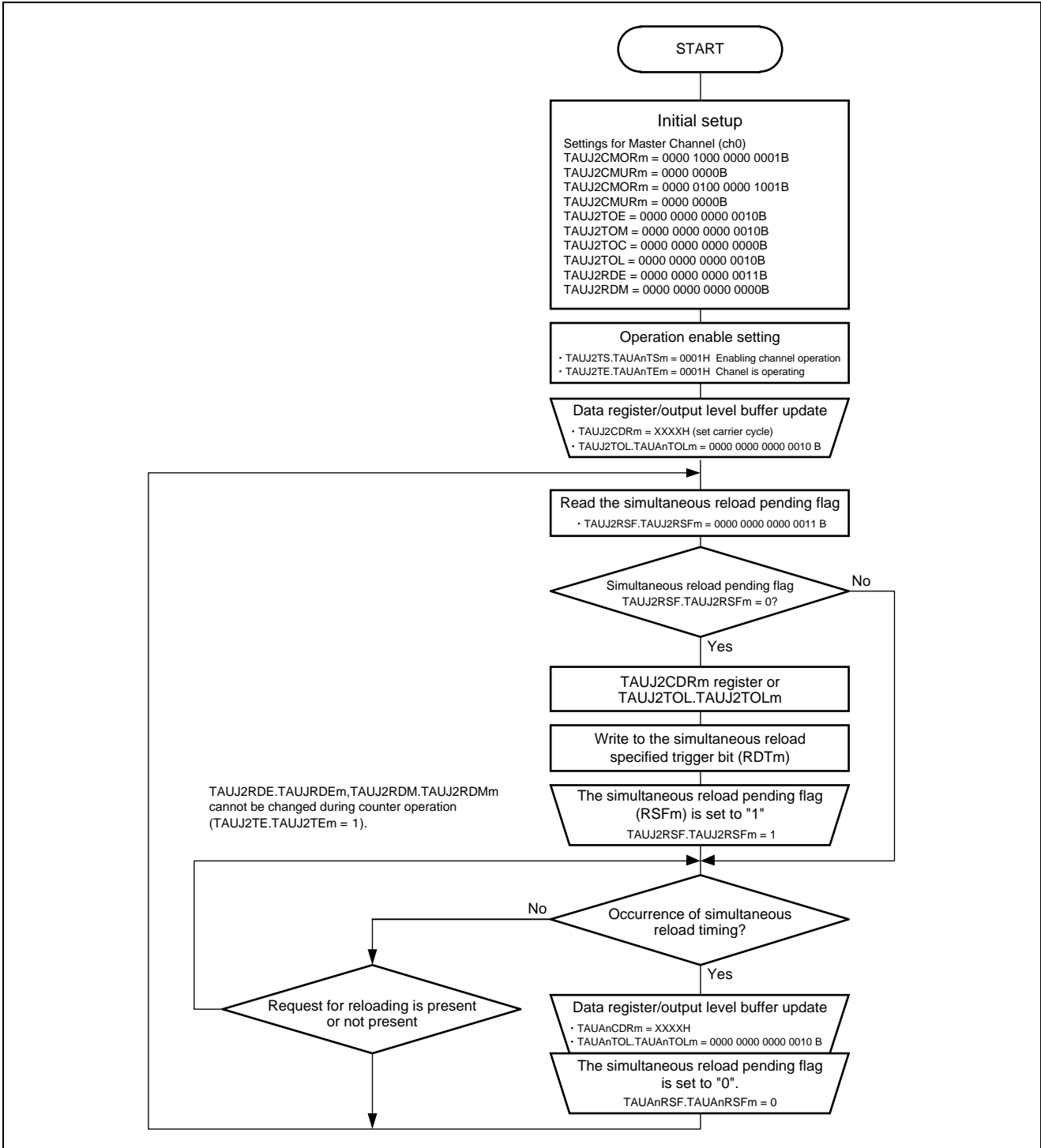


Figure 14.4 Basic Procedure of Simultaneous Reloading

14.6.3 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.TAUJ2TE_m = 0, set the following bits.
 - TAUJ2RDE.TAUJ2RDE_m
 - TAUJ2RDM.TAUJ2RDM_m
- (3) Targets of simultaneous reloading in synchronous operation are TAUJ2CDR_m and TAUJ2TOL.TAUJ2TOL_m.
- (4) The function which allows TAUJ2TOL.TAUJ2TOL_m 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.TAUJ2RDT_m is not set to 1, simultaneous reloading does not proceed.**
 - 3. When TAUJ2RDT.TAUJ2RDT_m is set to 1, TAUJ2RSF.TAUJ2RSF_m is set to 1 and generation of a simultaneous reload trigger leads to TAUJ2RSF.TAUJ2RSF_m to be cleared; accordingly, before changing the value of the register, read TAUJ2RSF.TAUJ2RSF_m and confirm that its value is 0.**

14.7 Independent Channel Operation

The following describes the individual functions of independent channel operation.

- 14.7.1 “Interval Timer”
- 14.7.2 “TAUJ2TTINm Input Interval Timer”
- 14.7.3 “Delay Counting”
- 14.7.4 “TAUJ2TTINm Input Pulse Interval Measurement”
- 14.7.5 “TAUJ2TTINm Input Signal Width Measurement”
- 14.7.6 “External Event Counting”
- 14.7.7 “TAUJ2TTINm Input Position Detection”

14.7.1 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

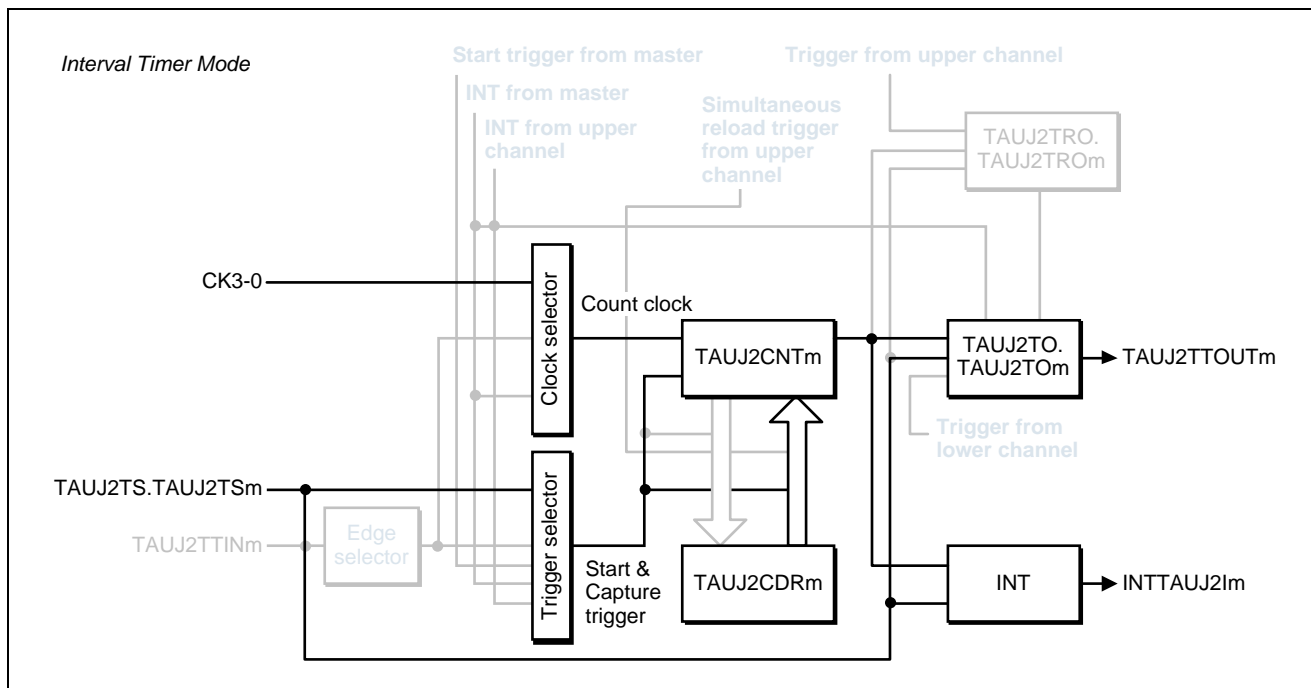


Figure 14.5 Block Diagram of Interval Timer

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- INTTAUJ2Im is generated at the start of operation (TAUJ2CMORm.TAUJ2MD0 = 1)

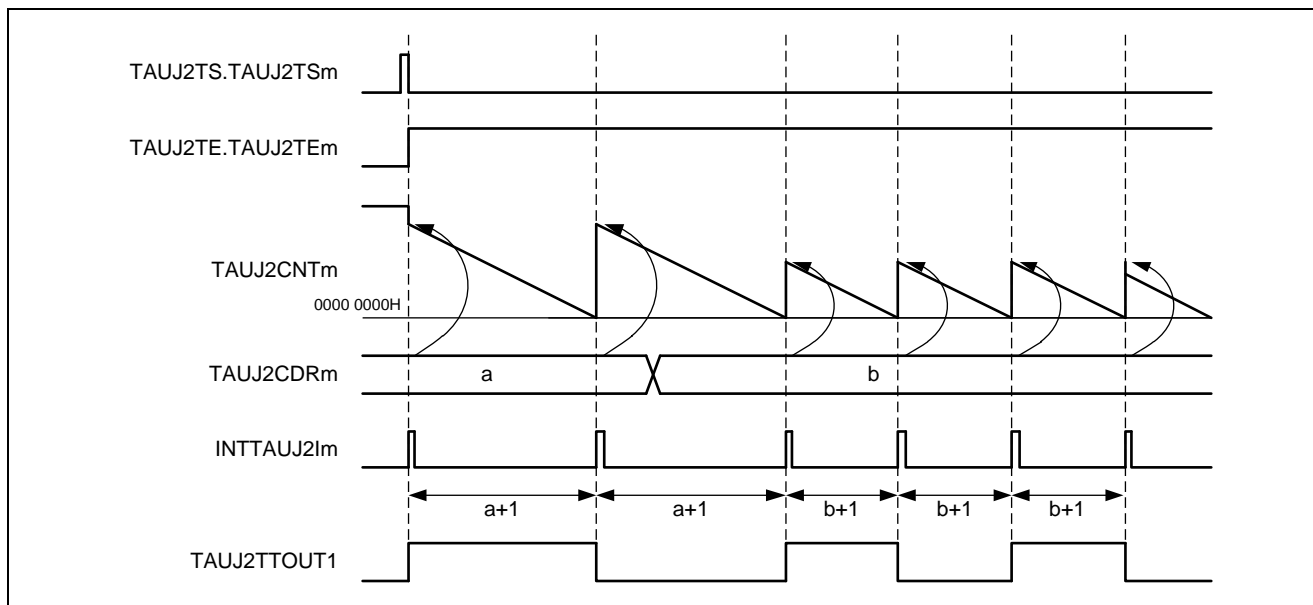


Figure 14.6 General Timing Diagram of Interval Timer

(4) Equations

$INTTAUJ2Im \text{ cycle} = \text{count clock cycle} \times (TAUJ2CDRm + 1)$

$TAUJ2TTOUTm \text{ square wave cycle} = \text{count clock cycle} \times (TAUJ2CDRm + 1) \times 2$

(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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]			TAUJ2MD0	

Table 14.5 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 14.6 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 14.7 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

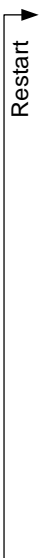
Table 14.8 Control Bit Settings for Independent Channel Output

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 14.9 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	<ul style="list-style-type: none"> 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	Register whose value can be changed at any timing <ul style="list-style-type: none"> TAUJ2CDRm Register which is readable at any timing <ul style="list-style-type: none"> TAUJ2CNTm 	TAUJ2CNTm counts down. When the counter reaches 0000H: <ul style="list-style-type: none"> 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

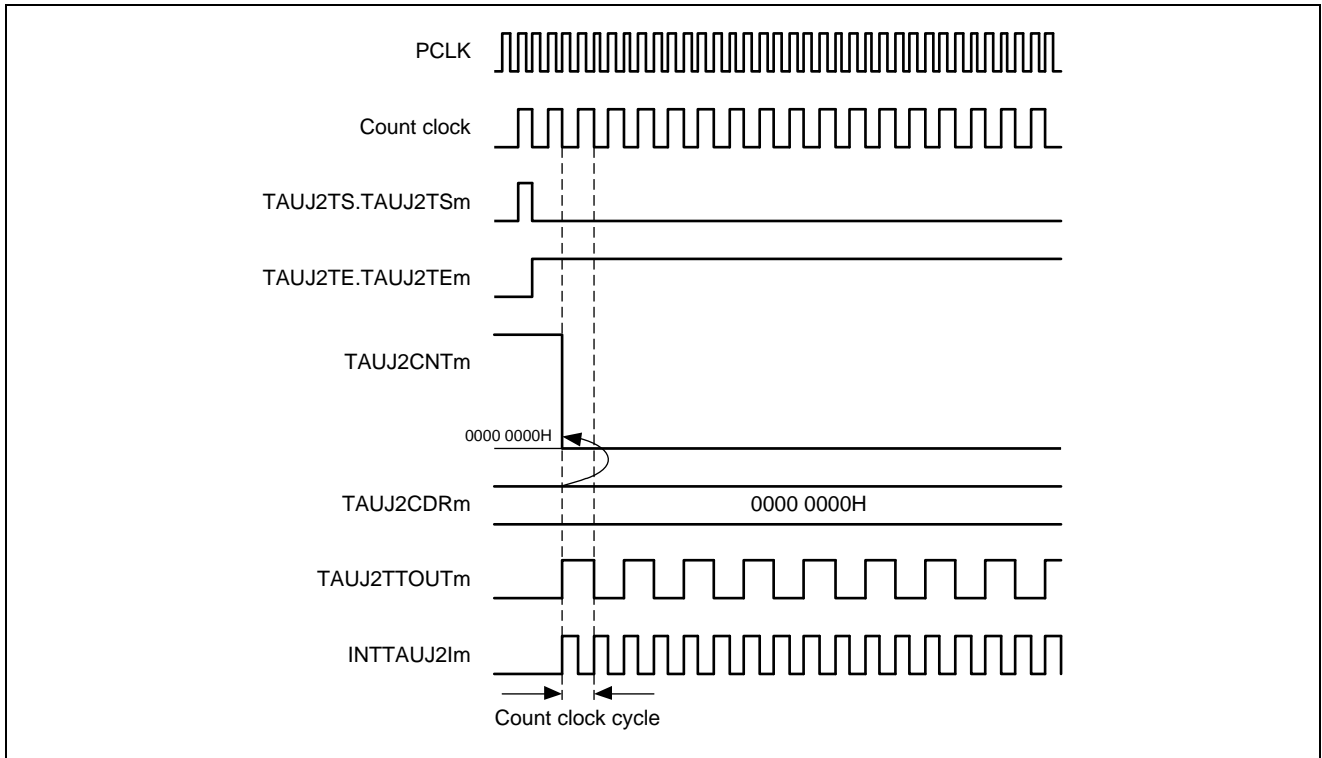


Figure 14.7 Count Clock = PCLK/2

- 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

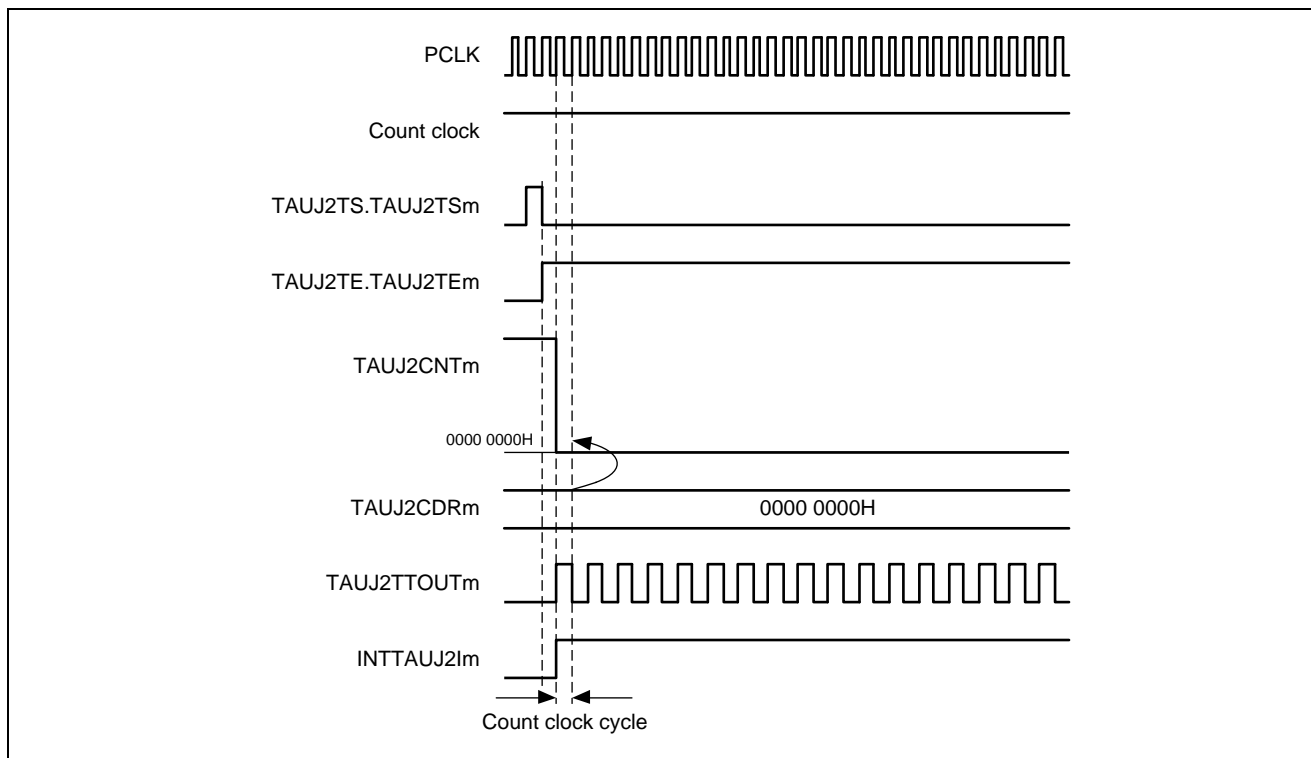


Figure 14.8 Count Clock = PCLK

- 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

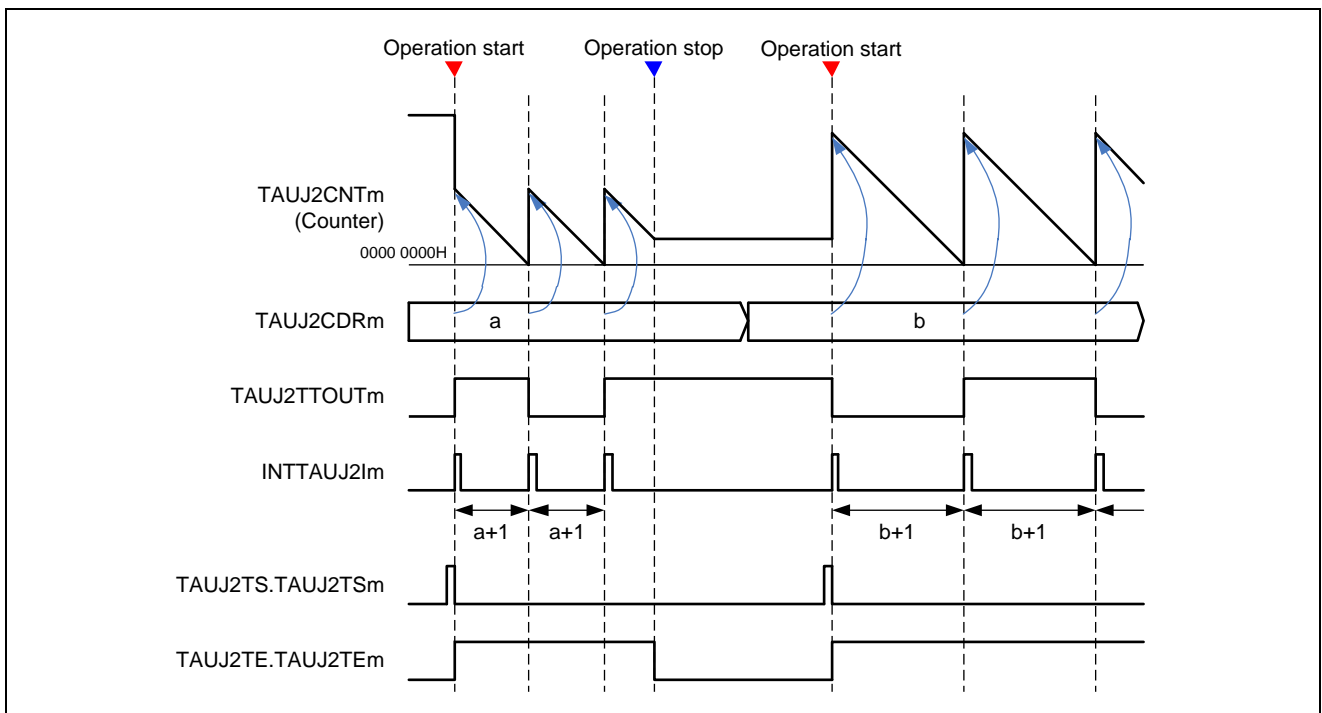


Figure 14.9 Operation Stop and Restart (TAUJ2CMORm.TAUJ2MD0 = 1)

- 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

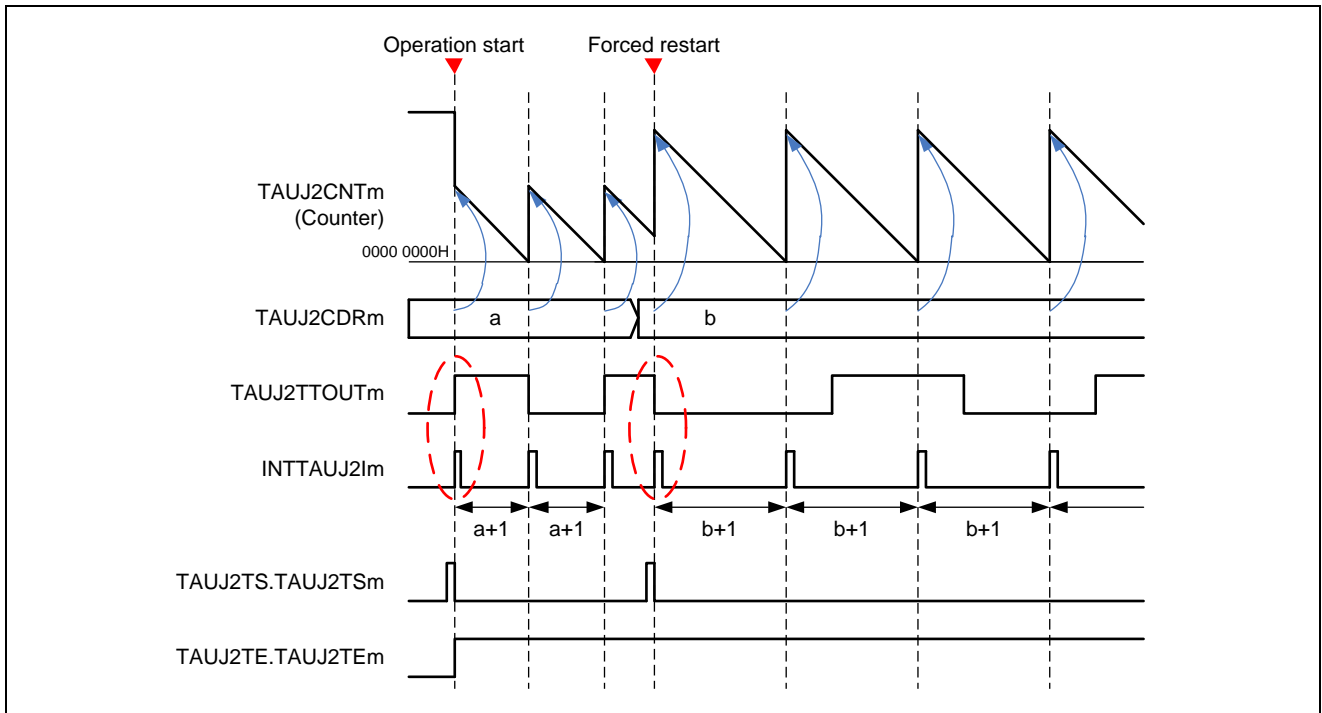


Figure 14.10 Forced Restart Operation, TAUJ2CMORm.TAUJ2MD0 = 1

- 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.

14.7.2 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 TAUJ2TTOUm 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

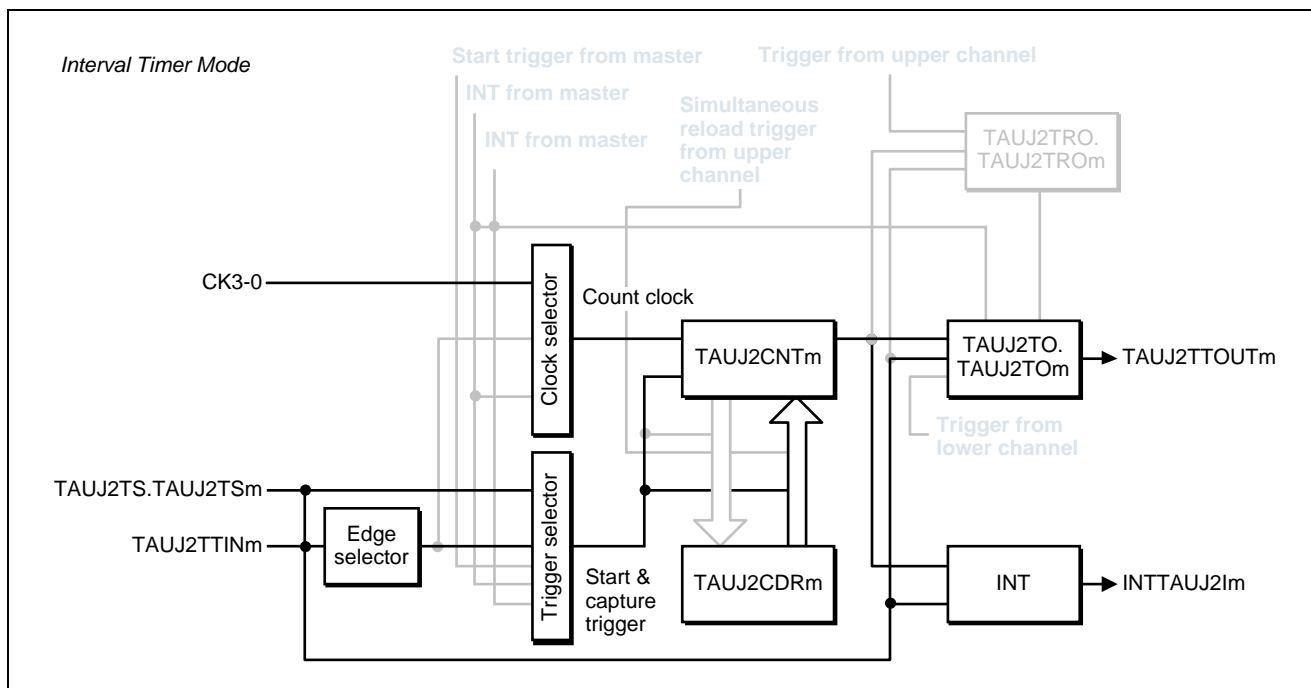


Figure 14.11 TAUJ2TTINm Block Diagram of TAUJ2TTINm Input Interval Timer

(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)

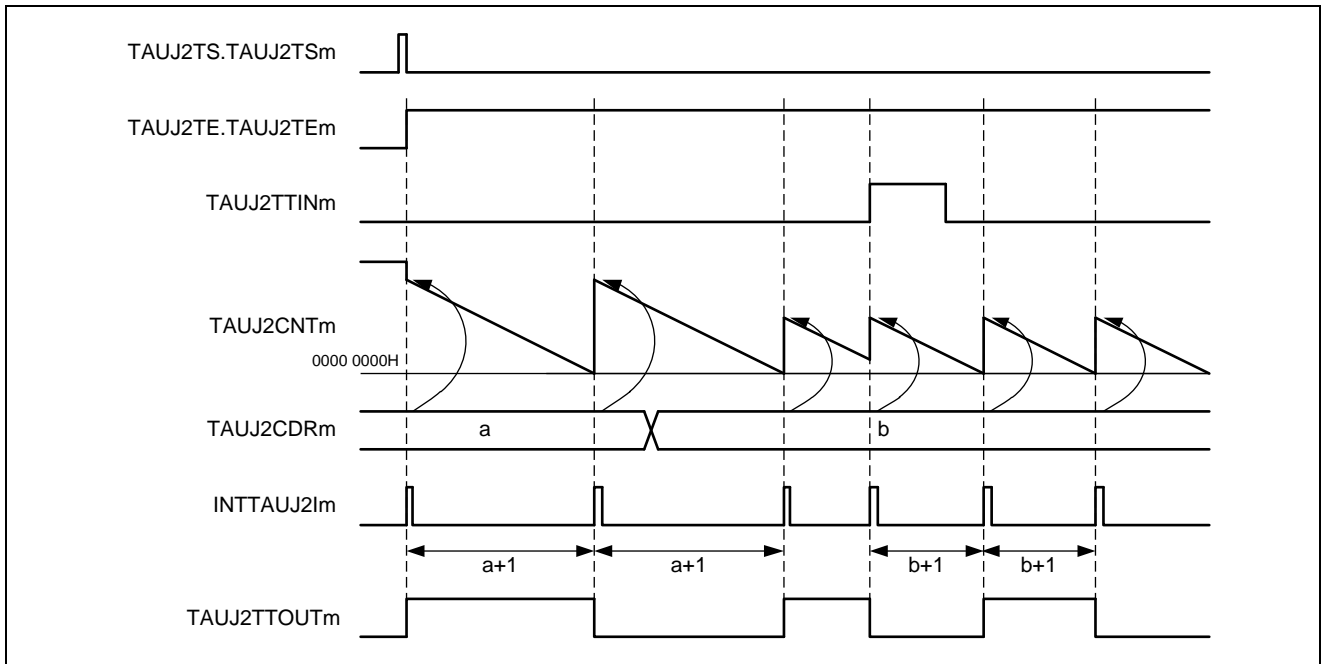


Figure 14.12 TAUJ2TTINm General Timing Diagram of TAUJ2TTINm Input Interval Timer

(4) Equations

$$\text{INTTAUJ2Im cycle} = \text{count clock cycle} \times (\text{TAUJ2CDRm} + 1)$$

$$\text{TAUJ2TTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUJ2CDRm} + 1) \times 2$$

(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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2MD0

Table 14.10 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
0														TAUJ2TIS[1:0]	

Table 14.11 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 14.12 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 14.13 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 14.14 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	<ul style="list-style-type: none"> 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.
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	Edge detection by TAUJ2TTINm input Registers whose value can be changed at any time <ul style="list-style-type: none"> TAUJ2CMURm.TAUJ2TIS[1:0] bits TAUJ2CDRm register Register which is readable at any timing <ul style="list-style-type: none"> TAUJ2CNTm register 	TAUJ2CNTm counts down. When the counter reaches 0000 0000H or an effective edge of the TAUJ2TTINm input signal: <ul style="list-style-type: none"> 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

For the operation of section 14.7.1, Interval Timer, the counter can also be restarted by an effective TAUJ2TTINm input edge.

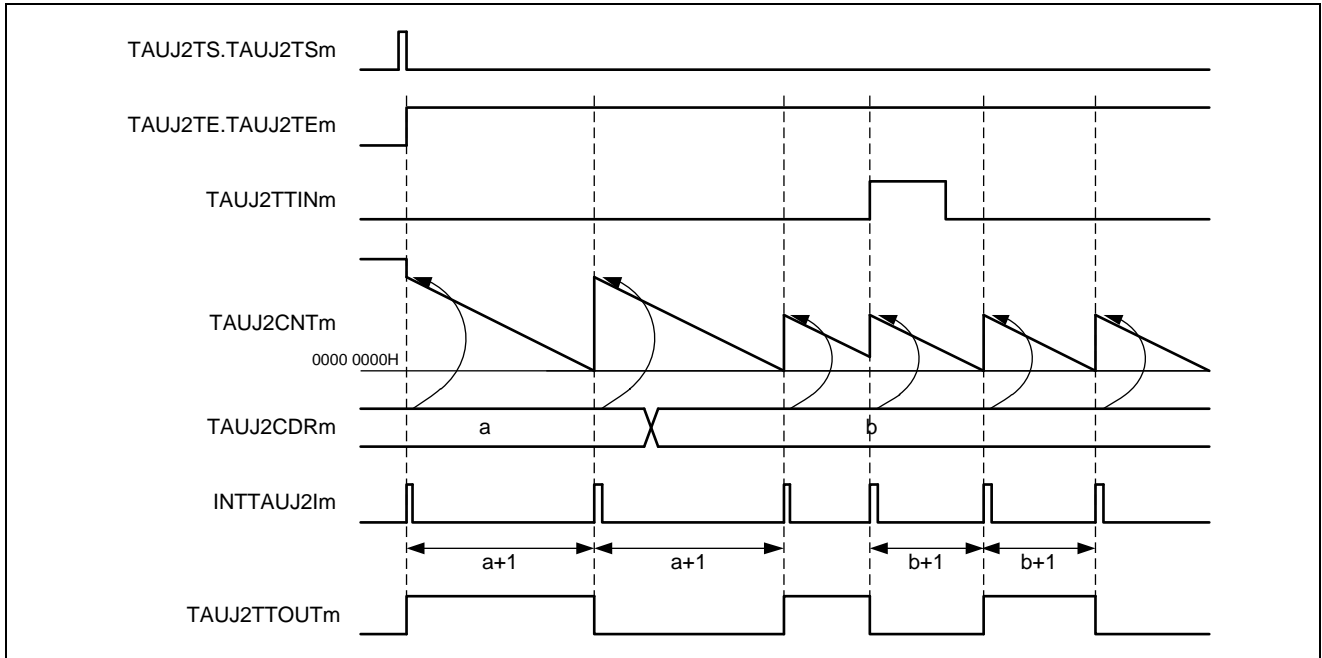


Figure 14.13 Counter Triggered by Rising TAUJ2TTINm Input Edge (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B), TAUJ2CMORm.TAUJ2MD0 = 1

- If an effective TAUJ2TTINm input edge is detected, an interrupt is generated which causes TAUJ2TTOUTm to toggle.

14.7.3 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

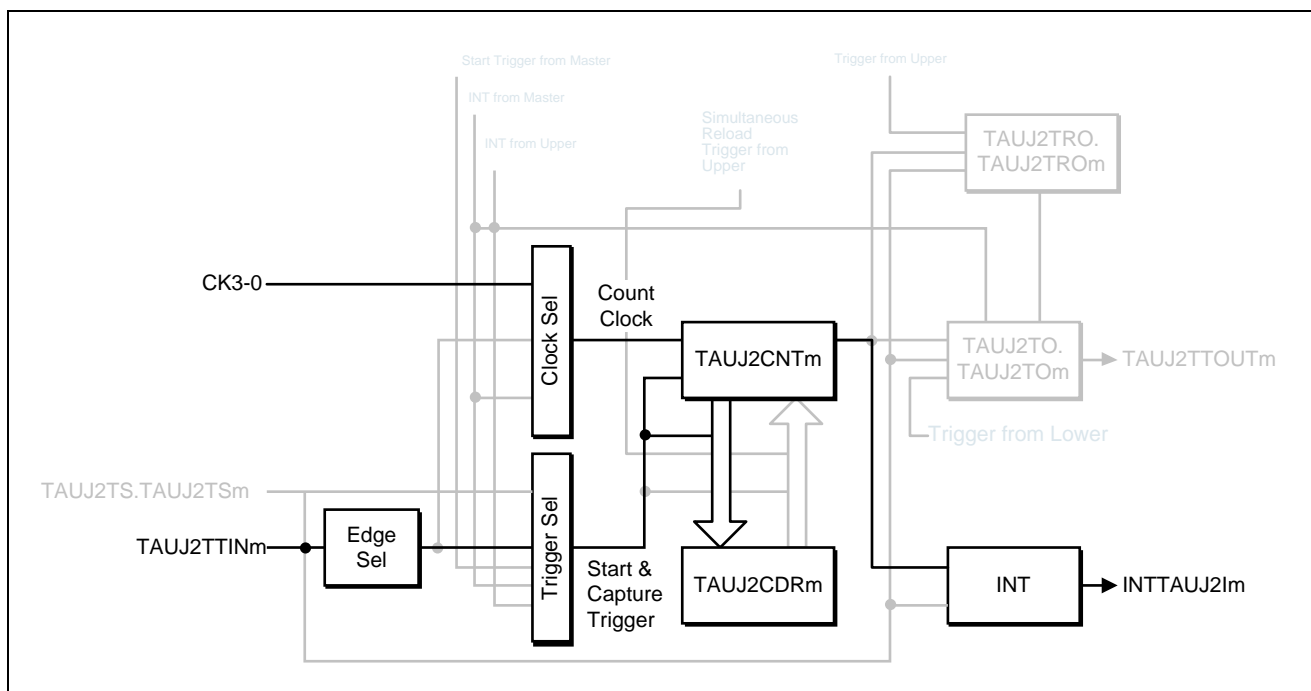


Figure 14.14 Block Diagram of Delay Counting

(3) General Timing Diagram

The following settings apply to the general timing diagram:

- Falling edge detection (TAUJ2CMURm.TAUJ2TIS[1:0] = 00B)

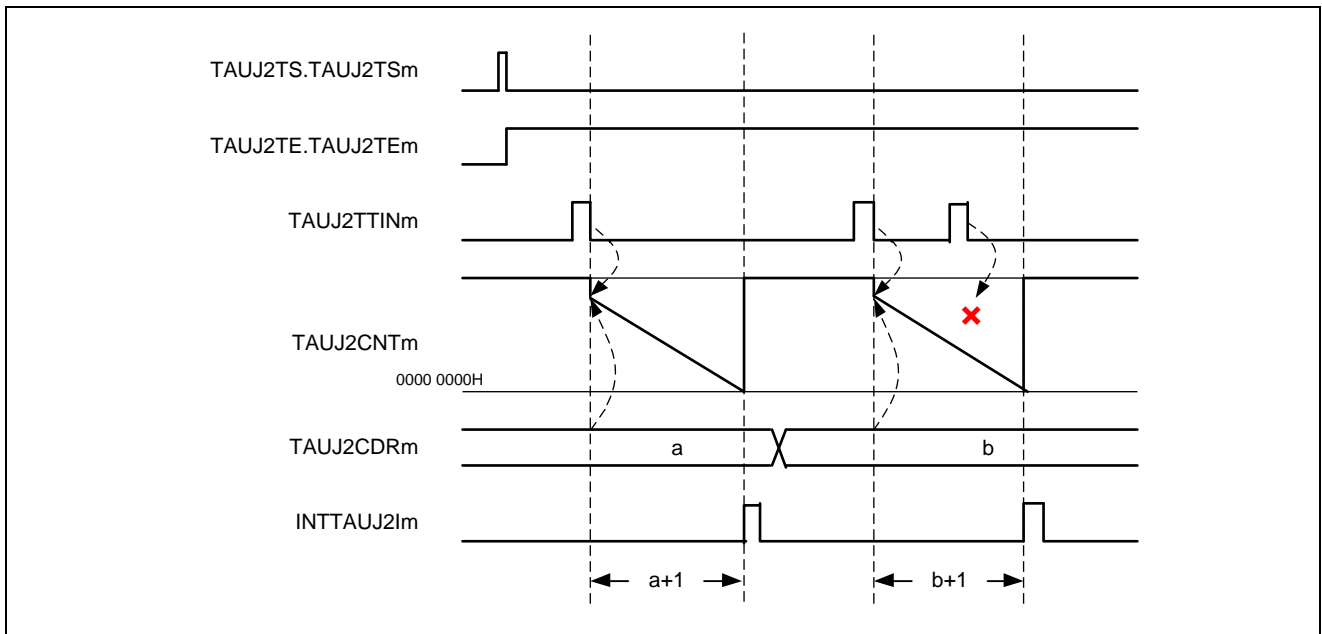


Figure 14.15 General Timing Diagram of Delay Counting

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 \times (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]		TAUJ2MAS	TAUJ2STS[2:0]		TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2MD0	

Table 14.15 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 14.16 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 14.17 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

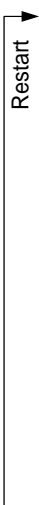
Table 14.18 Control Bit Settings for Independent Channel Output

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 14.19 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	<ul style="list-style-type: none"> 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.
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 <ul style="list-style-type: none"> TAUJ2CDRm Register which is readable at any time <ul style="list-style-type: none"> 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.



14.7.4 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

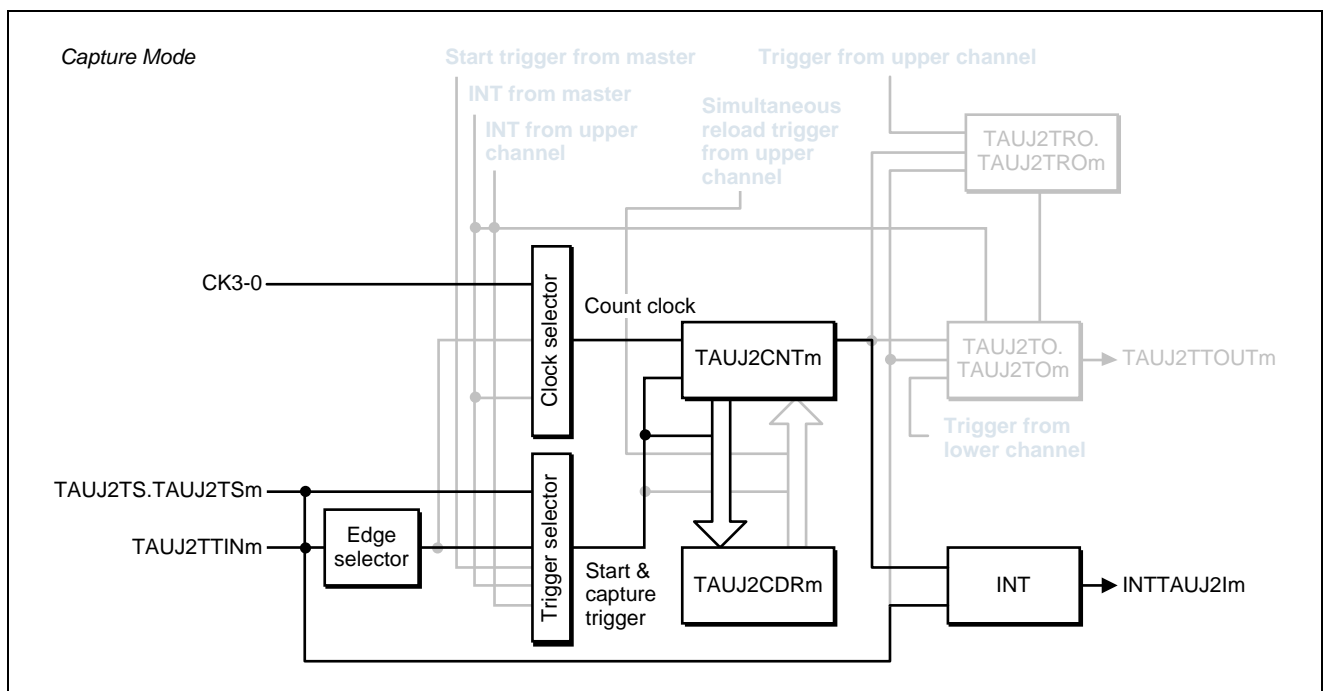


Figure 14.16 Block Diagram of TAUJ2TTINm Input Pulse Interval Measurement

(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)

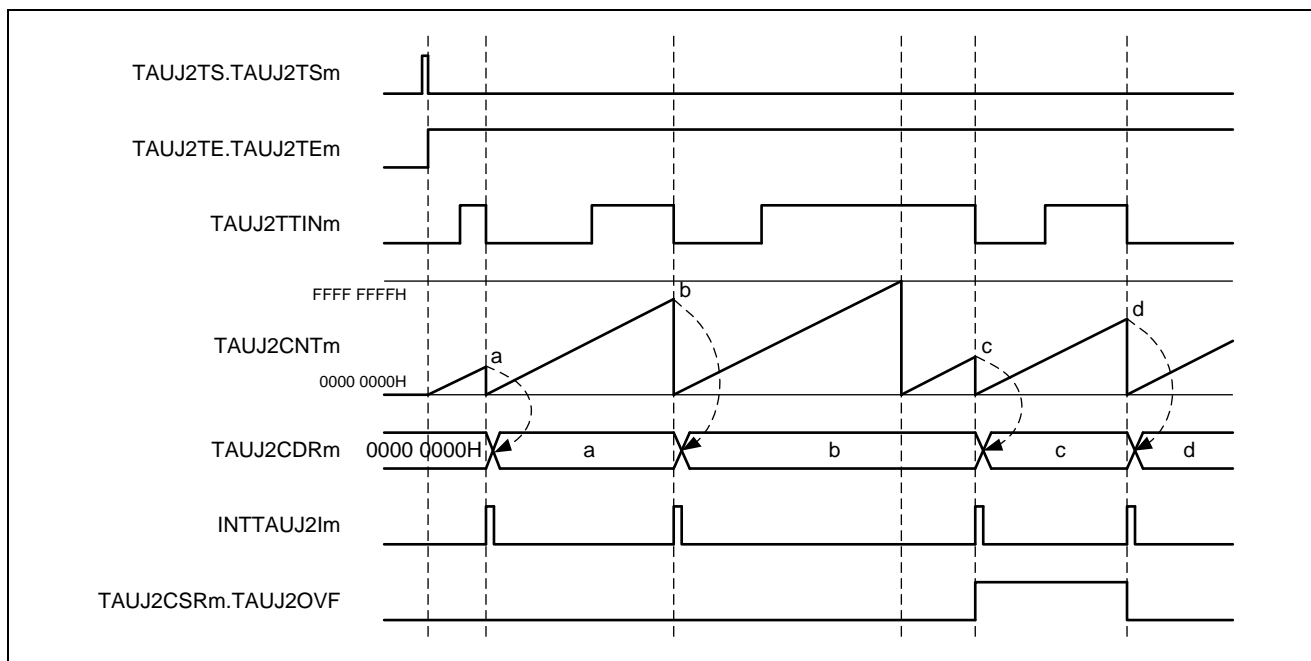


Figure 14.17 General Timing Diagram of TAUJ2TTINm Input Pulse Interval Measurement

(4) Equations

$$\text{TAUJ2TTINm input pulse interval} = \text{count clock cycle} \times [(\text{TAUJ2CSRm.TAUJ2OVF} \times (\text{FFFF FFFFH} + 1)) + \text{TAUJ2CDRm capture value} + 1]$$

(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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2MD0

Table 14.20 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

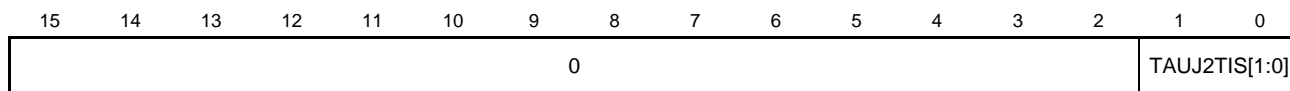


Table 14.21 TAUJ2CMORM Settings for TAUJ2TTINm Input Pulse Interval Measurement

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 14.22 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 14.23 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 14.24 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	<ul style="list-style-type: none"> 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	Registers whose value can be changed at any time <ul style="list-style-type: none"> TAUJ2CMURm.TAUJ2TIS[1:0] bits TAUJ2CDRm register Registers which are readable at any time <ul style="list-style-type: none"> 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 TAUAnCSRm.TAUAnOVF retain their current values.

Restart

(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

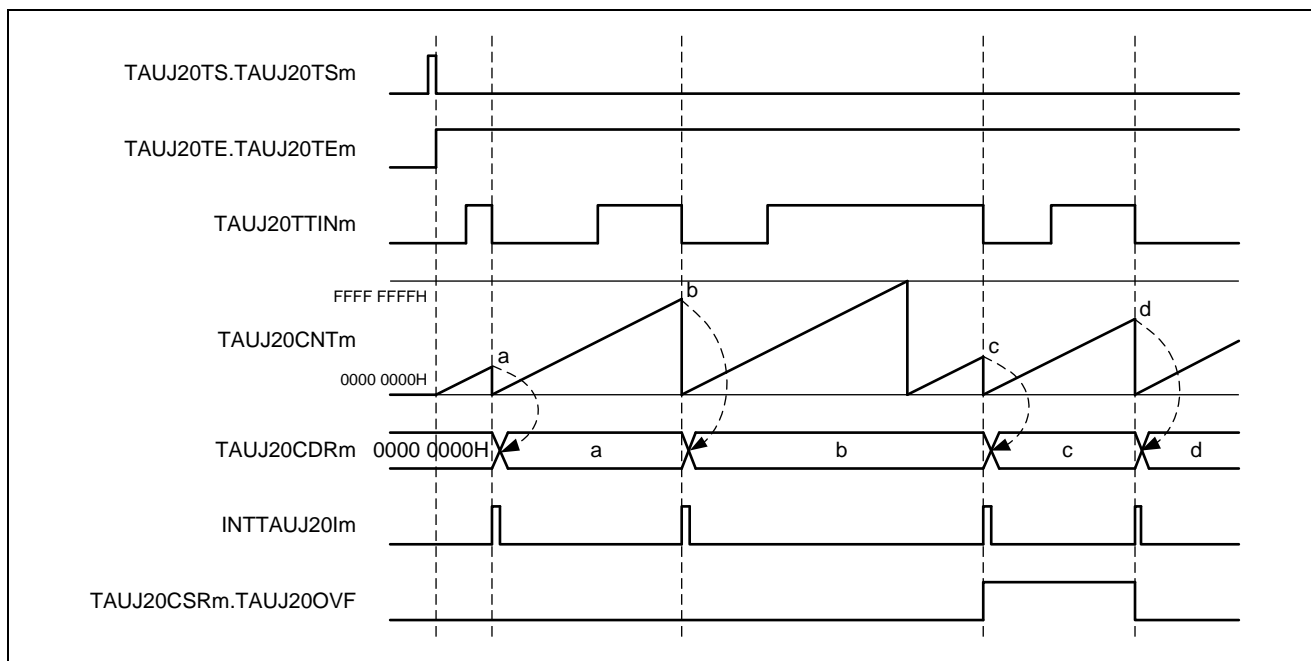


Figure 14.18 TAUJ2CMORm.TAUJ2COS[1:0] = 00B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B

- 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

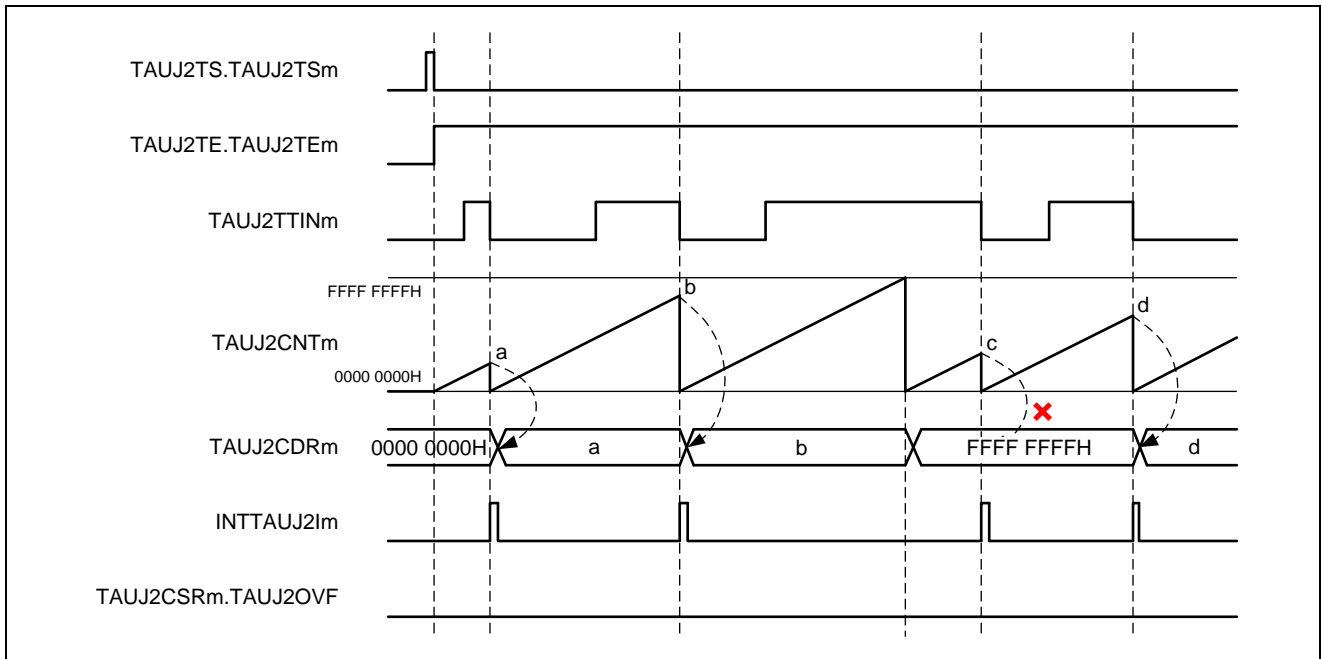


Figure 14.19 TAUJ2CMORm.TAUJ2COS[1:0] = 10B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B

- 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.

14.7.5 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

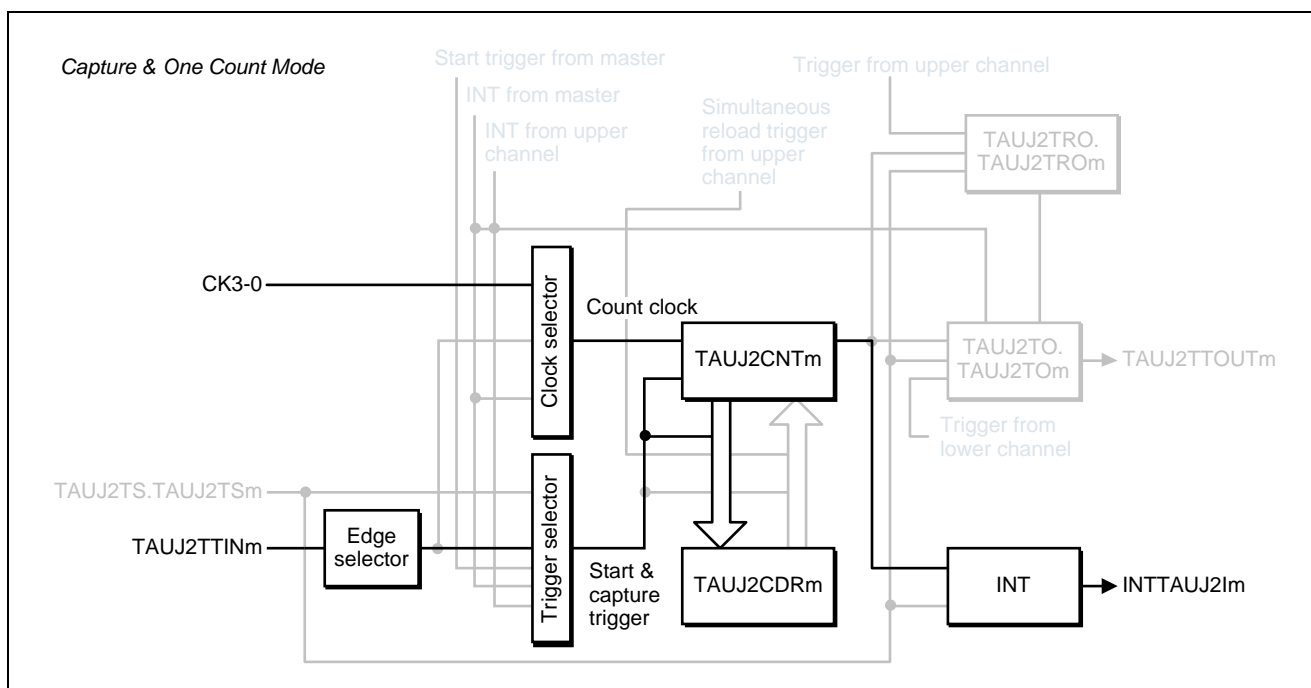


Figure 14.20 Block Diagram of TAUJ2TTINm Input Signal Width Measurement

(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)

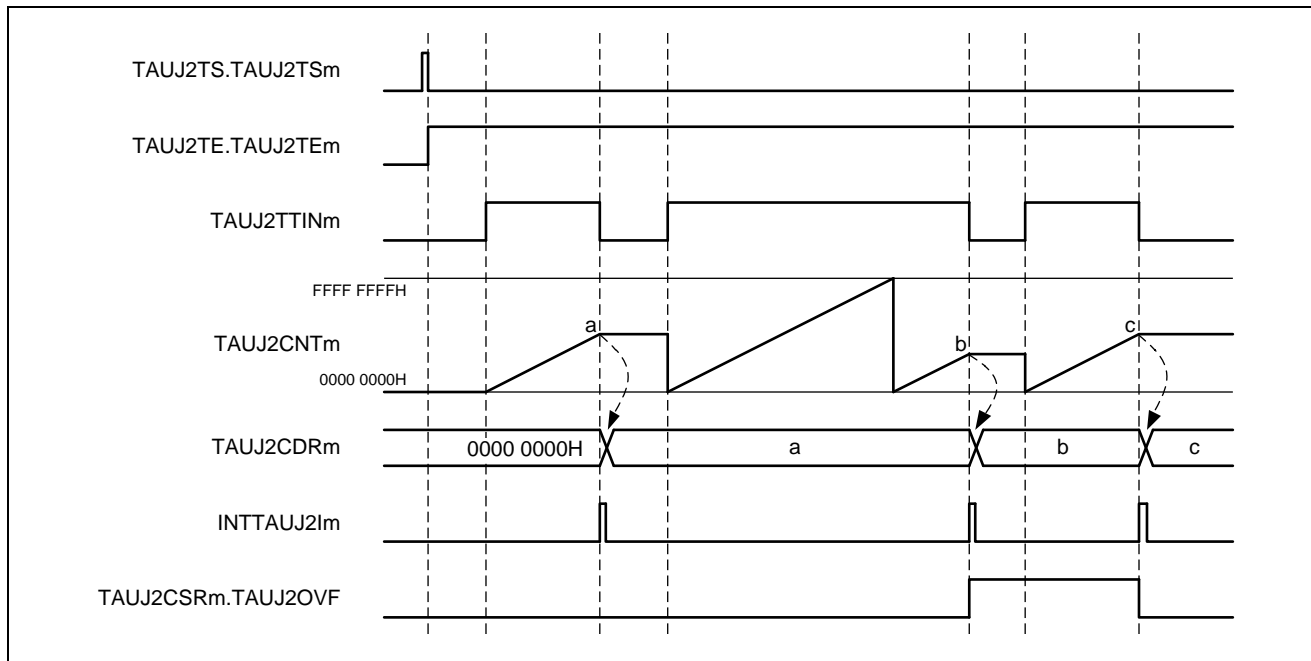


Figure 14.21 General Timing Diagram of TAUJ2TTINm Input Signal Width Measurement

(4) Equations

$$\text{TAUJ2TTINm input signal width} = \text{count clock cycle} \times [(\text{TAUJ2CSRm.TAUJ2OVF} \times (\text{FFFF FFFFH} + 1)) + \text{TAUJ2CDRm capture value} + 1]$$

(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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]			TAUJ2MD0	

Table 14.25 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
—														TAUJ2TIS[1:0]	

Table 14.26 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 14.27 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 14.28 Control Bit Settings for Independent Channel Output Mode

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 14.29 Operating Procedure

	Operation	Status of TAUJ2
Initial Channel Setting	<ul style="list-style-type: none"> • 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 TAUJ2CNTm waits for TAUJ2TTINm start edge detection.
During Operation	Registers which are readable at any time: <ul style="list-style-type: none"> • 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.

Restart →

(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

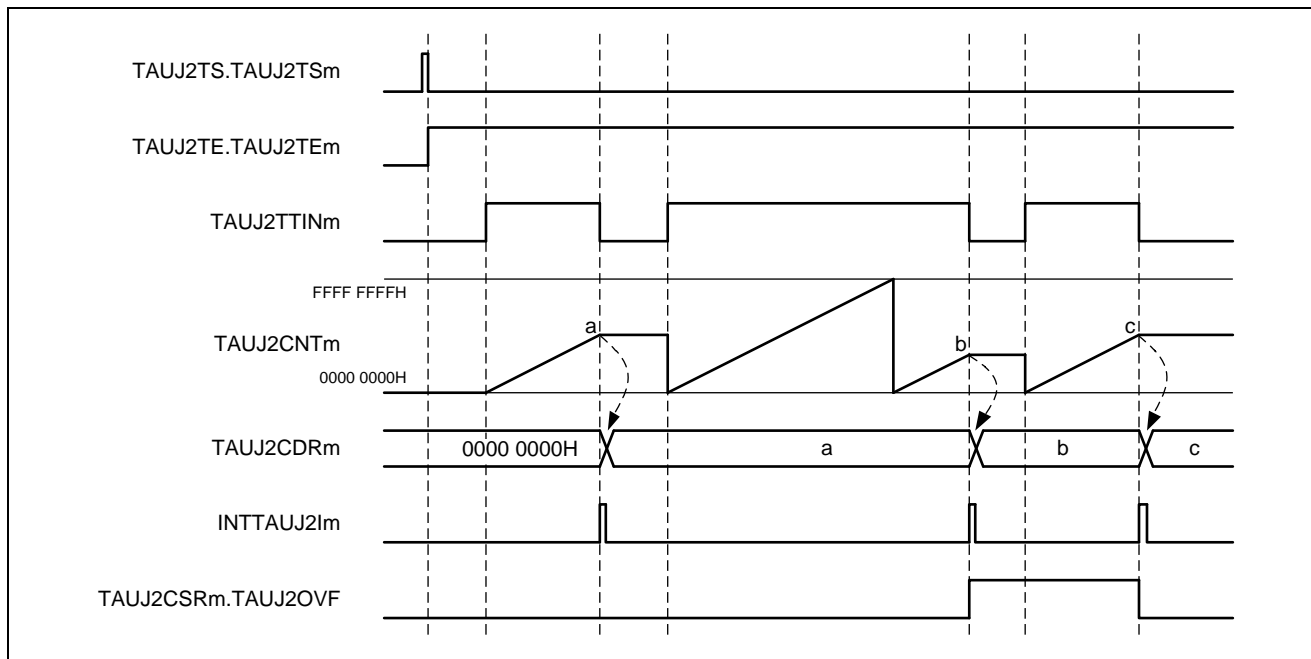


Figure 14.22 TAUJ2CMORm.TAUJ2COS[1:0] = 00B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 11B

- 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

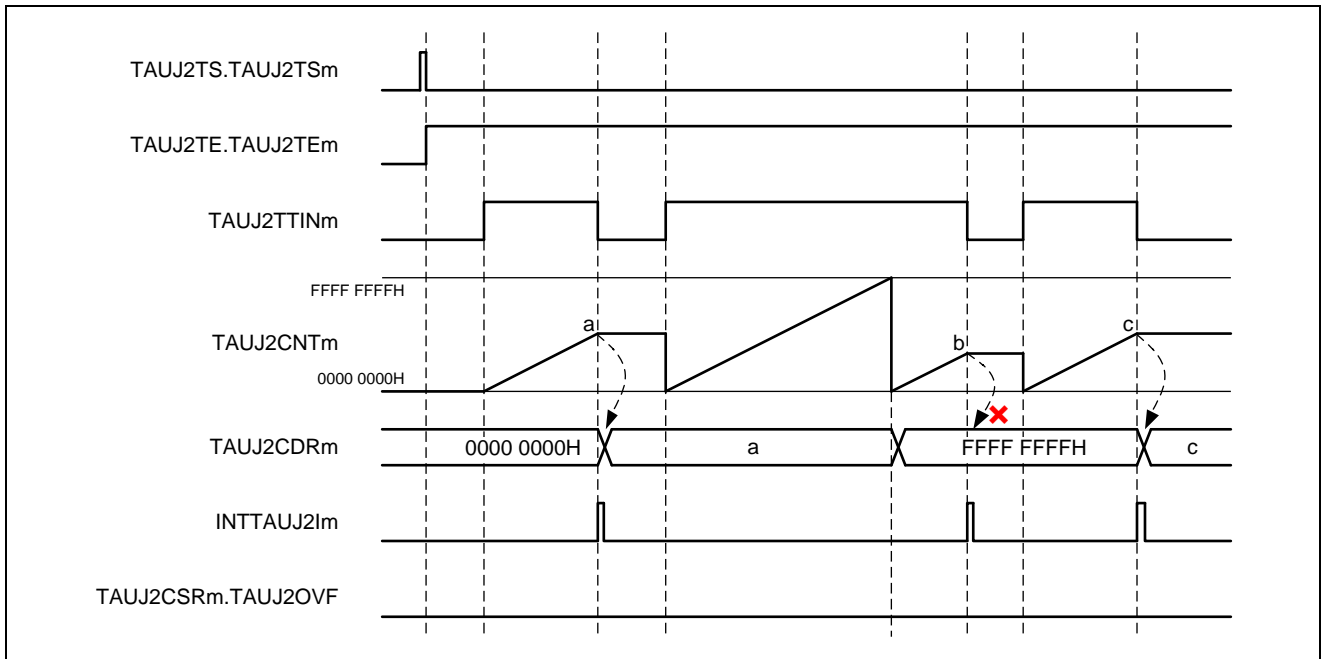


Figure 14.23 TAUJ2CMORm.TAUJ2COS[1:0] = 10B, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 11B

- 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 14.24, Block Diagram of Overflow Interrupt Output (for TAUJ2TTINm Width Measurement), for configuration of channels.

(b) Block diagram

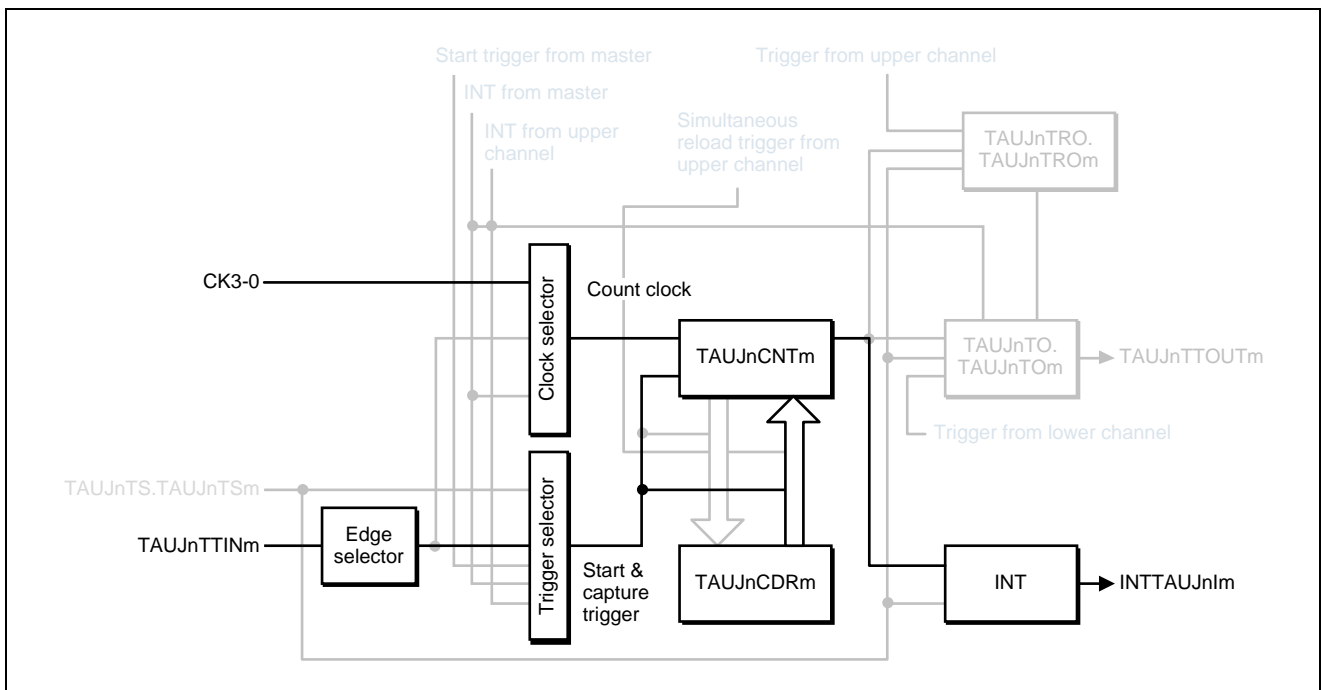


Figure 14.24 Block Diagram of Overflow Interrupt Output (for TAUJ2TTINm Width Measurement)

(c) General timing diagram

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement

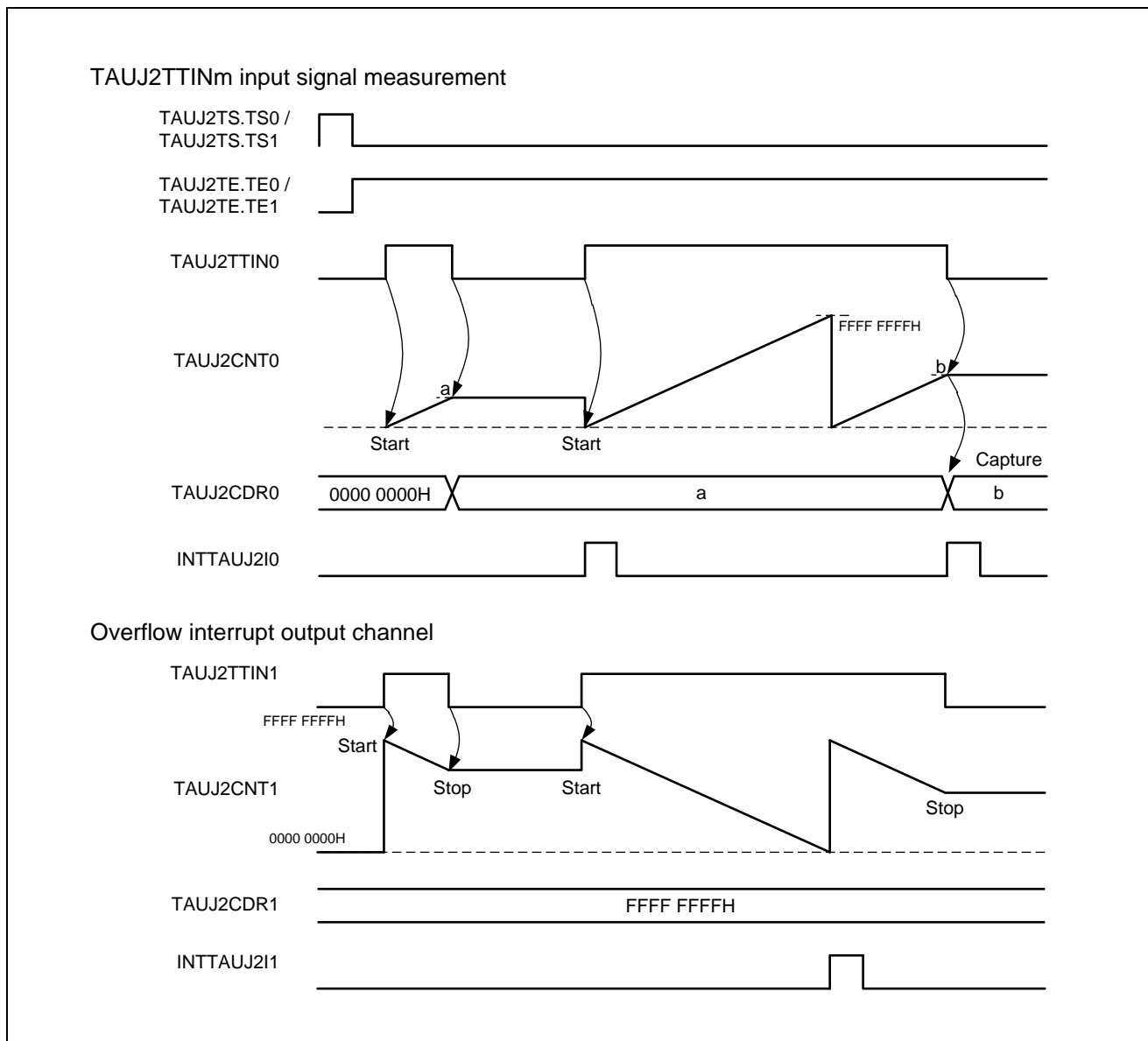


Figure 14.25 General Timing Diagram at the Time of Overflow Interrupt Output

(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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2MD0

Table 14.30 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

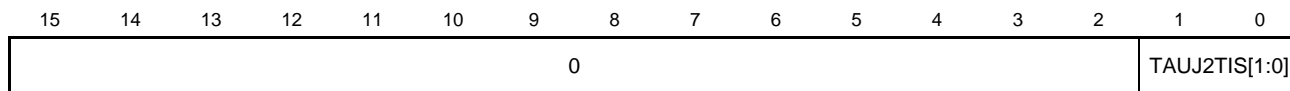


Table 14.31 TAUJ2CMURm Settings for TAUJ2TTINm Input Signal Width Measurement

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 14.32 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 14.33 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 14.34 Operating Procedure

	Operation	Status of TAUJ2
Restart ↓	Initial Channel Setting	Channel operation is stopped.
	Start Operation	TAUJ2TE.TAUJ2TE _m is set to 1, and TAUJ2CNT _m waits for TAUJ2TTIN _m start edge detection. When a start edge is detected, the value of TAUJ2CDR _m (FFFF FFFFH) is updated in TAUJ2CNT _m .
	During Operation	When a start edge of TAUJ2TTIN _m is detected, TAUJ2CNT _m starts counting from FFFF FFFFH. When a stop edge of TAUJ2TTIN _m is detected, it stops counting. When the counter reaches 0000 0000H, INTTAUJ2I _m is generated. Afterwards, this procedure is repeated.
	Stop Operation	TAUJ2TE.TAUJ2TE _m is cleared to 0 and the counter stops. TAUJ2CNT _m stops and retains its current value.

14.7.6 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 14.35, 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 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

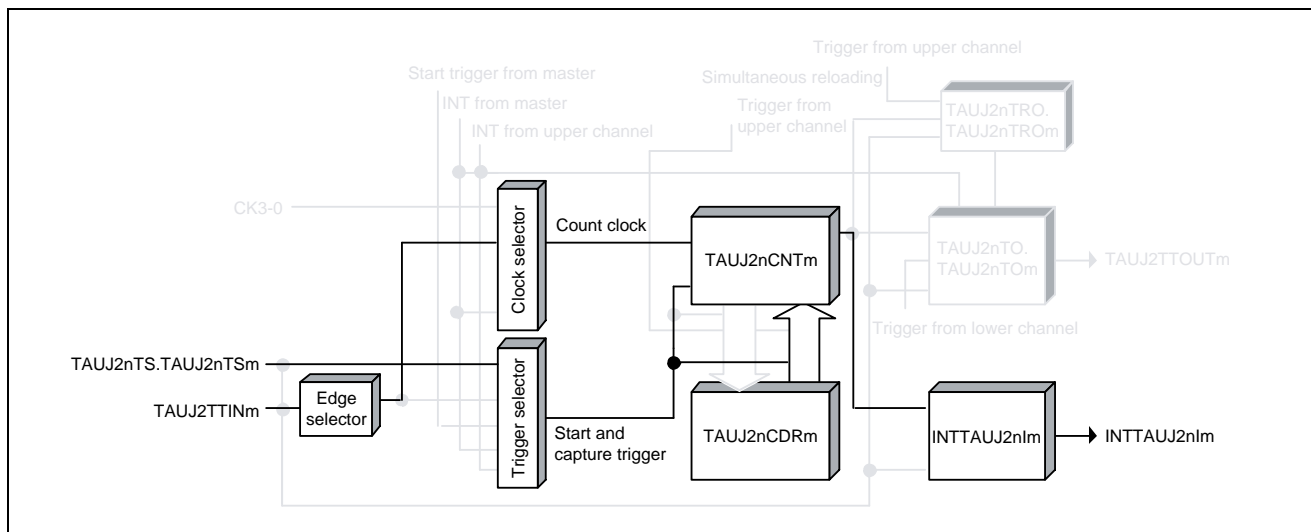


Figure 14.26 Block Diagram of External Event Counting

The following settings apply to the general timing diagram.

- Detection of rising edges (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B)

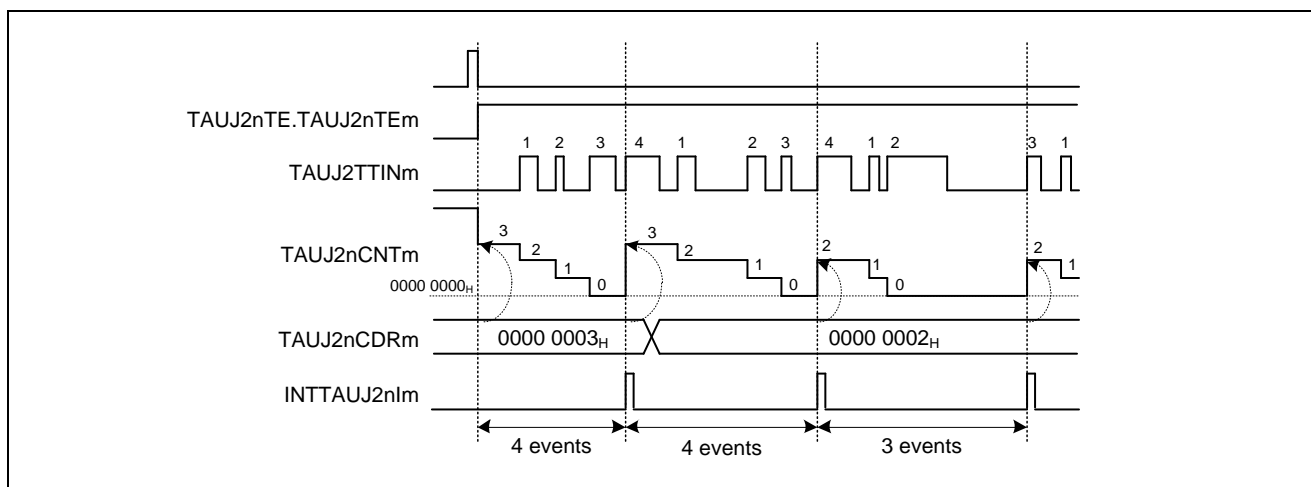


Figure 14.27 General Timing Diagram of External Event Counting

(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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]			TAUJ2MD0	

Table 14.35 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 14.36 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, TAUJ2RDS, TAUJ2RDM, and TAUJ2RDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 14.37 Simultaneous Reload Settings for External Event Counting

Bit Name	Setting
TAUJ2RDE.TAUJ2RDEm	0: Disables simultaneous reloading.
TAUJ2RDM.TAUJ2RDSm	0: When simultaneous reloading is disabled (TAUJ2RDE.TAUJ2RDEm = 0), set these bits to 0.
TAUJ2RDM.TAUJ2RDMm	
TAUJ2RDM.TAUJ2RDCm	

(5) Operating Procedure for External Event Counting

Table 14.38 Operating Procedure for External Event Counting

	Operation	Status of TAUJ2
Restart ↓	Initial channel setting Set TAUJ2CMORm and TAUJ2CMURm registers as described in Table 14.35, Contents of the TAUJ2CMORm Register for External Event Counting and Table 14.36, 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: <ul style="list-style-type: none"> • 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.

(6) Specific Timing Diagrams

(a) TAUJ2CDRm = 0000 0000H

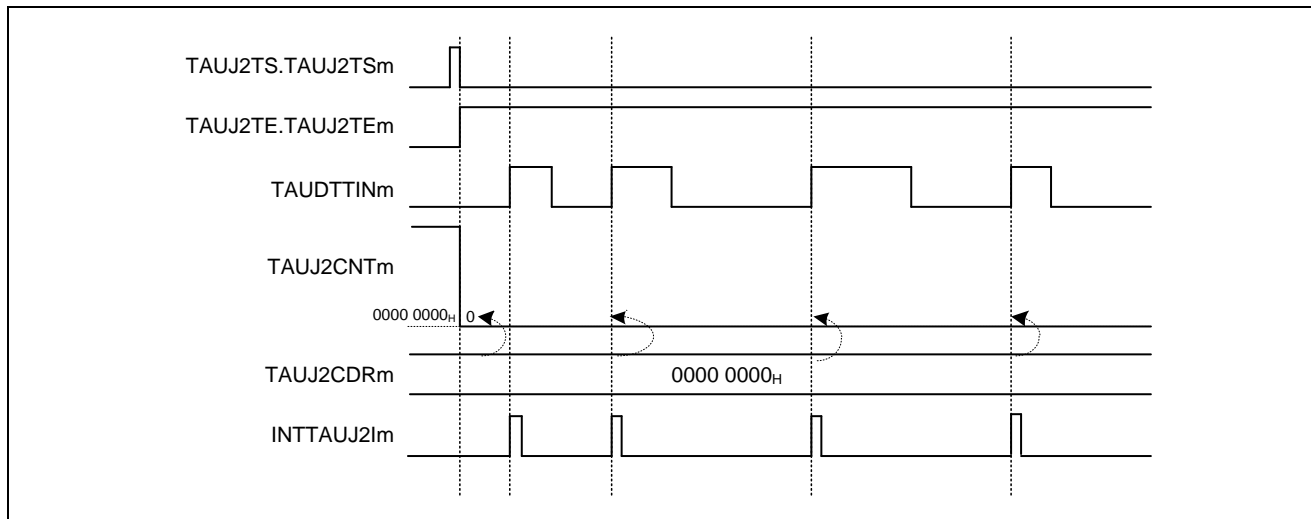


Figure 14.28 TAUJ2CDRm = 0000 0000H, TAUJ2CMURm.TAUJ2TIS[1:0] = 01B

- If 0000 0000H = TAUJ2CDRm, 0000 0000H is loaded to TAUJ2CNTm every time an effective TAUJ2TTINm input edge is detected. In other words, INTTAUJ2Im is generated every time an effective TAUJ2TTINm input edge is detected.

(b) Operation stop and restart

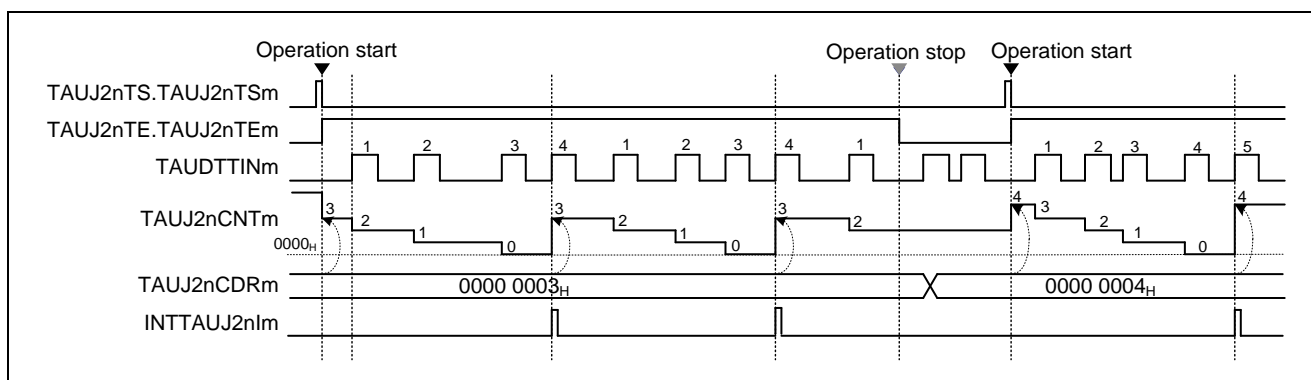


Figure 14.29 Operation Stop and Restart (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B)

- 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

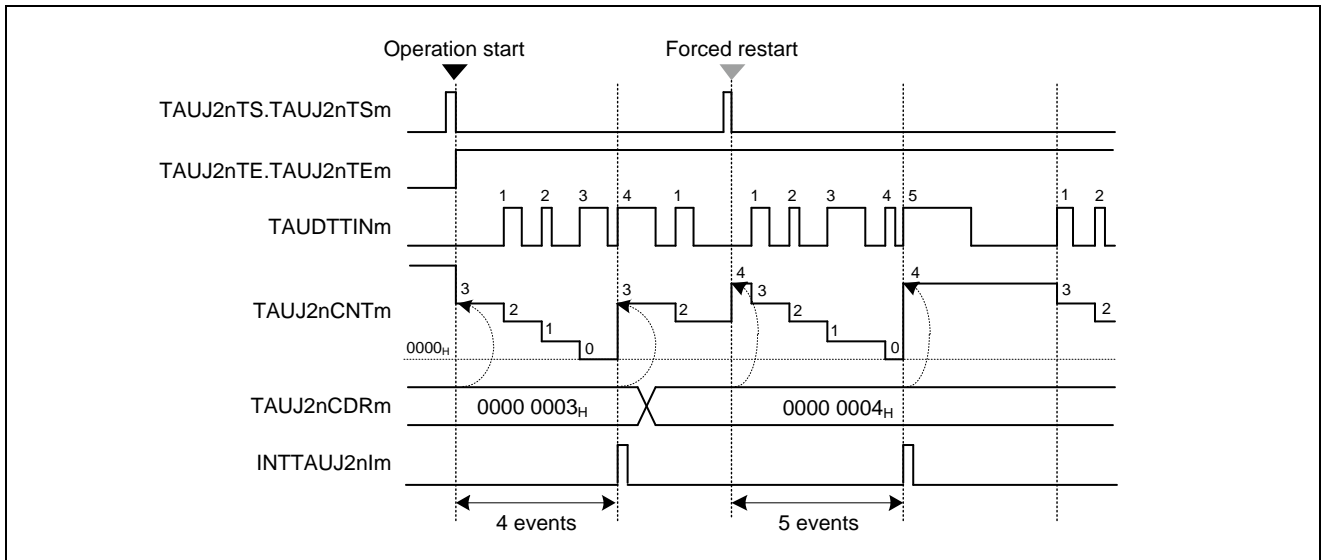


Figure 14.30 Forced Restart (TAUJ2CMURm.TAUJ2TIS[1:0] = 01B)

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.

14.7.7 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 TAUJ2TTOUm.

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

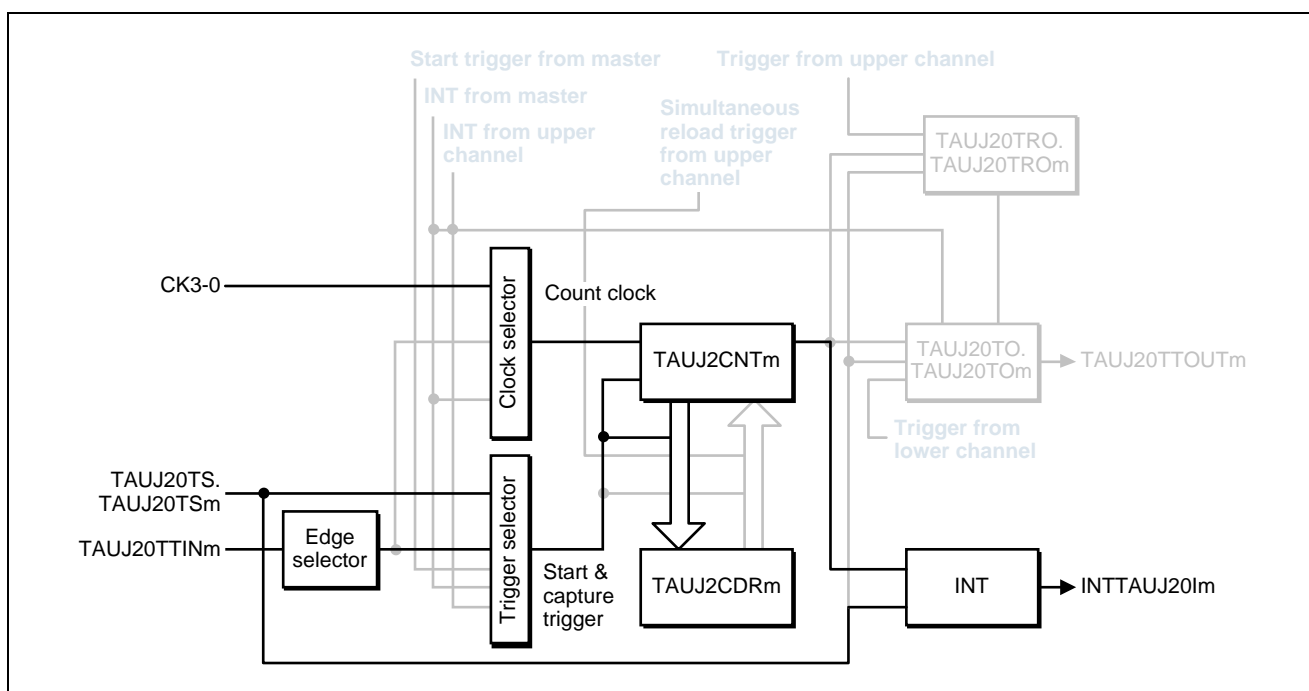


Figure 14.31 Block Diagram of TAUJ2TTINm Input Position Detection

(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)

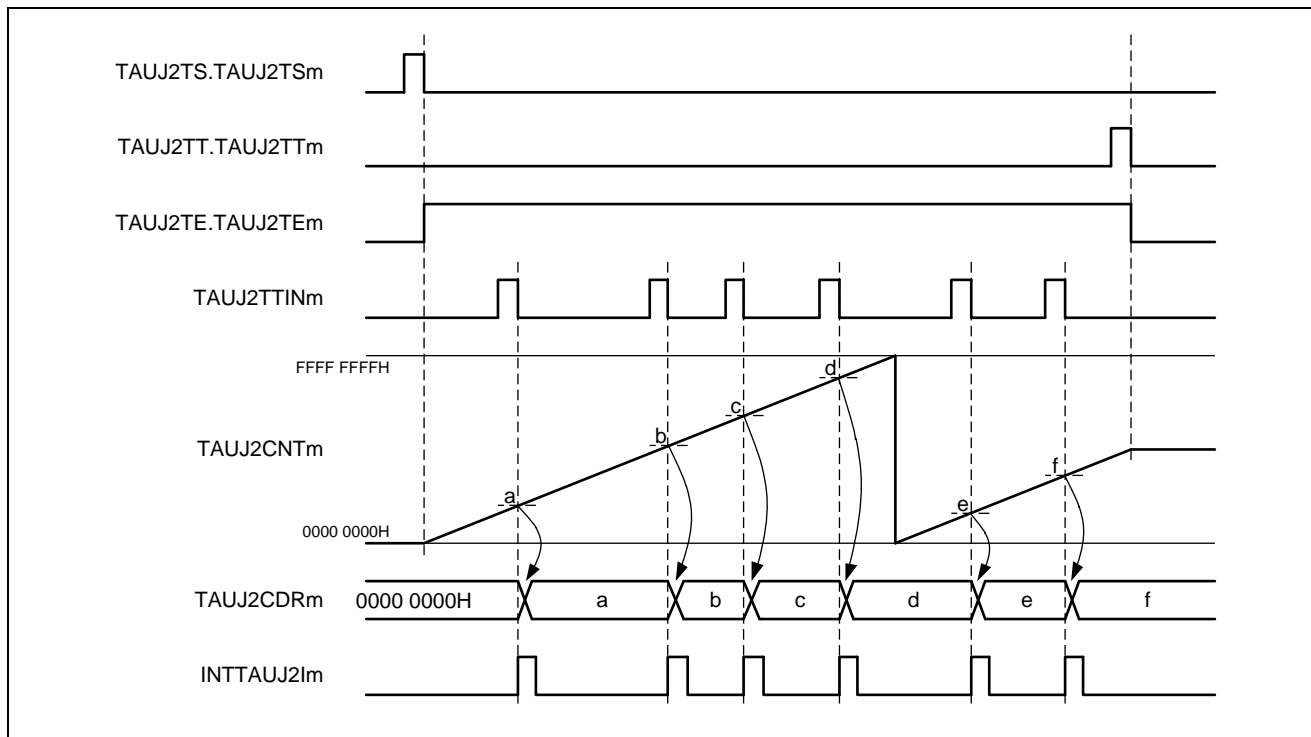


Figure 14.32 General Timing Diagram of TAUJ2TTINm Input Position Detection

(4) Equations

Function duration at a TAUJ2TTINm input pulse =

$$\text{count clock cycle} \times [(\text{FFFF FFFFH} + 1 \times \text{TAUJ2CSRm.TAUJ2OVF}) + (\text{TAUJ2CDRm capture value} + 1)]$$

(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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]			TAUJ2MD0	

Table 14.39 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 14.40 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 14.41 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 14.42 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 14.43 Operating Procedure

	Operation	Status of TAUJ2
Restart →	Initial Channel Setting	Channel operation is stopped.
	Start Operation	TAUJ2TE.TAUJ2TEm is set to 1 and the counter starts. INTTAUJ2Im is generated when TAUJ2CMORm.TAUJ2MD0 is set to 1.
	During Operation	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	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

(a) Operation stop and restart

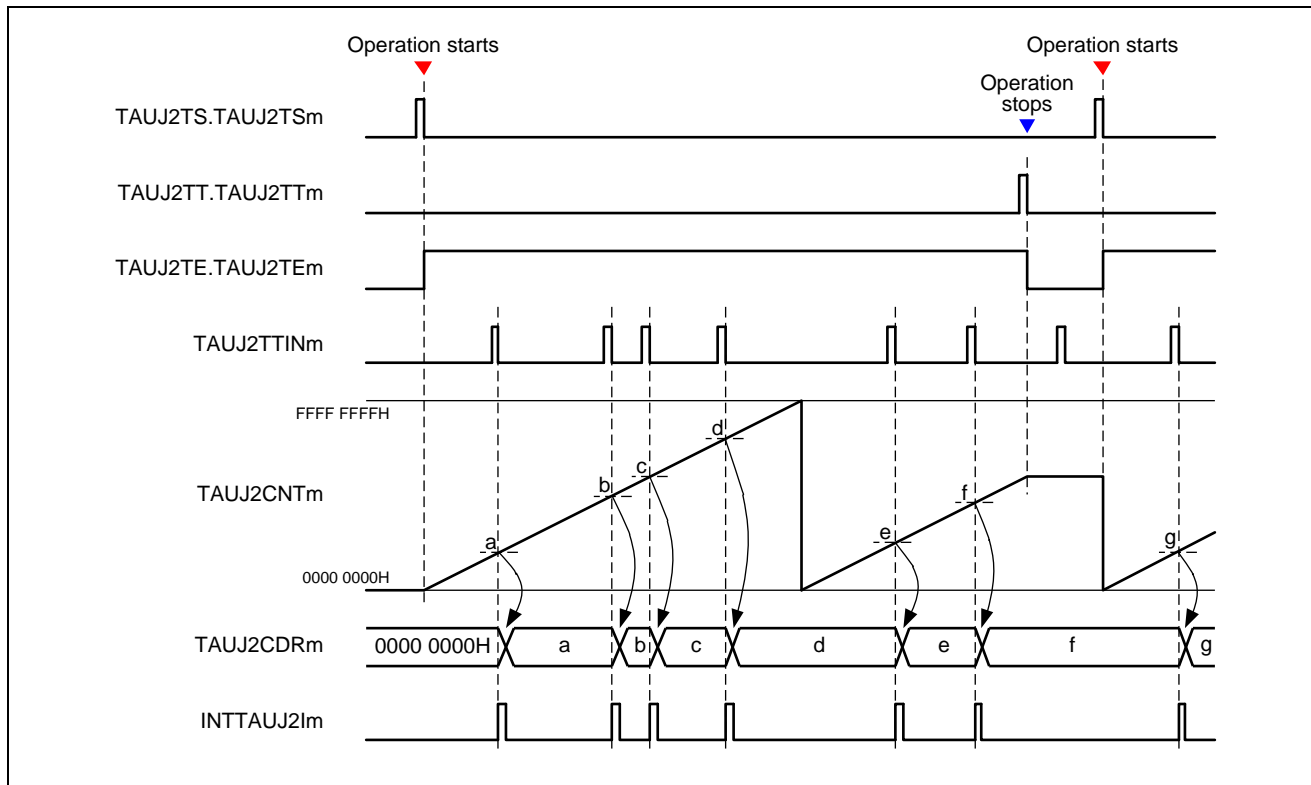


Figure 14.33 Operation Stop and Restart, TAUJ2CMORm.TAUJ2MD0 = 0, TAUJ2CMURm.TAUJ2TIS[1:0] = 00B

- 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 an overflow interrupt output channel (generation of overflow interrupts requires two channels).

(b) Block diagram

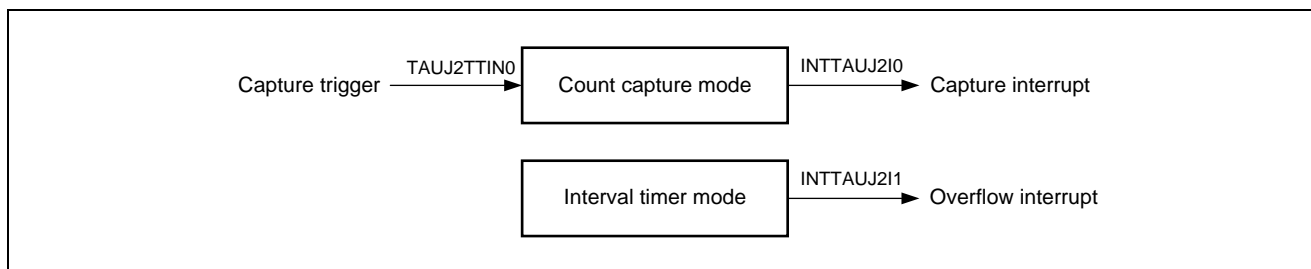


Figure 14.34 Block Diagram of Overflow Interrupt Output (when TAUJ2TTINm Input Position is Detected)

(c) General timing diagram

The following settings apply to the general timing diagram

- Falling edge detection

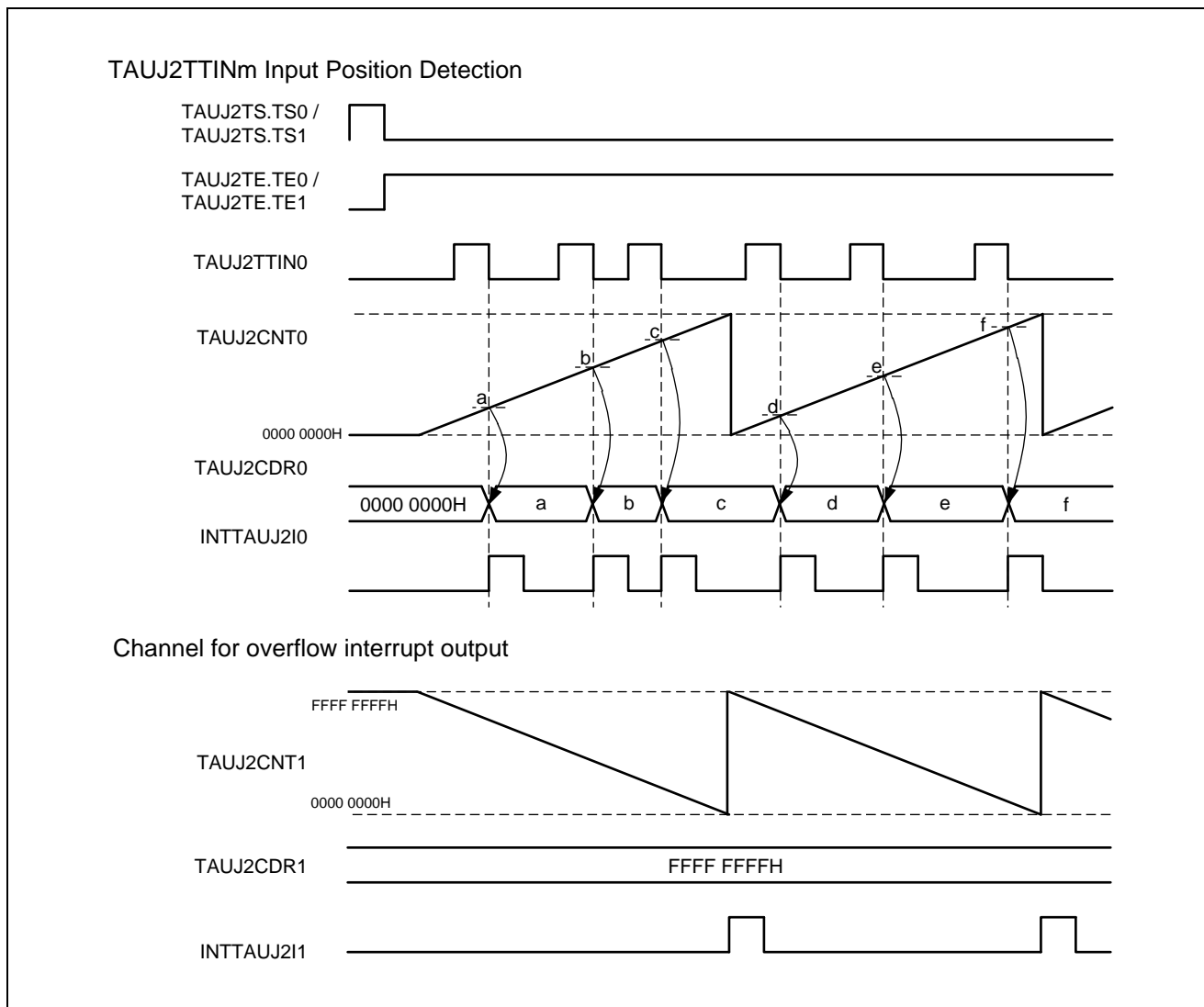


Figure 14.35 General Timing Diagram at the Time of Overflow Interrupt Output (when TAUJ2TTINm Input Position Detection is Used)

(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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]				TAUJ2MD0

Table 14.44 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 14.45 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 14.46 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 14.47 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 14.48 Operating Procedure

	Operation	Status of TAUJ2	
Restart ↓	Initial Channel Setting	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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.

14.8 Synchronous Channel Operation

14.8.1 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 and General Timing Diagram

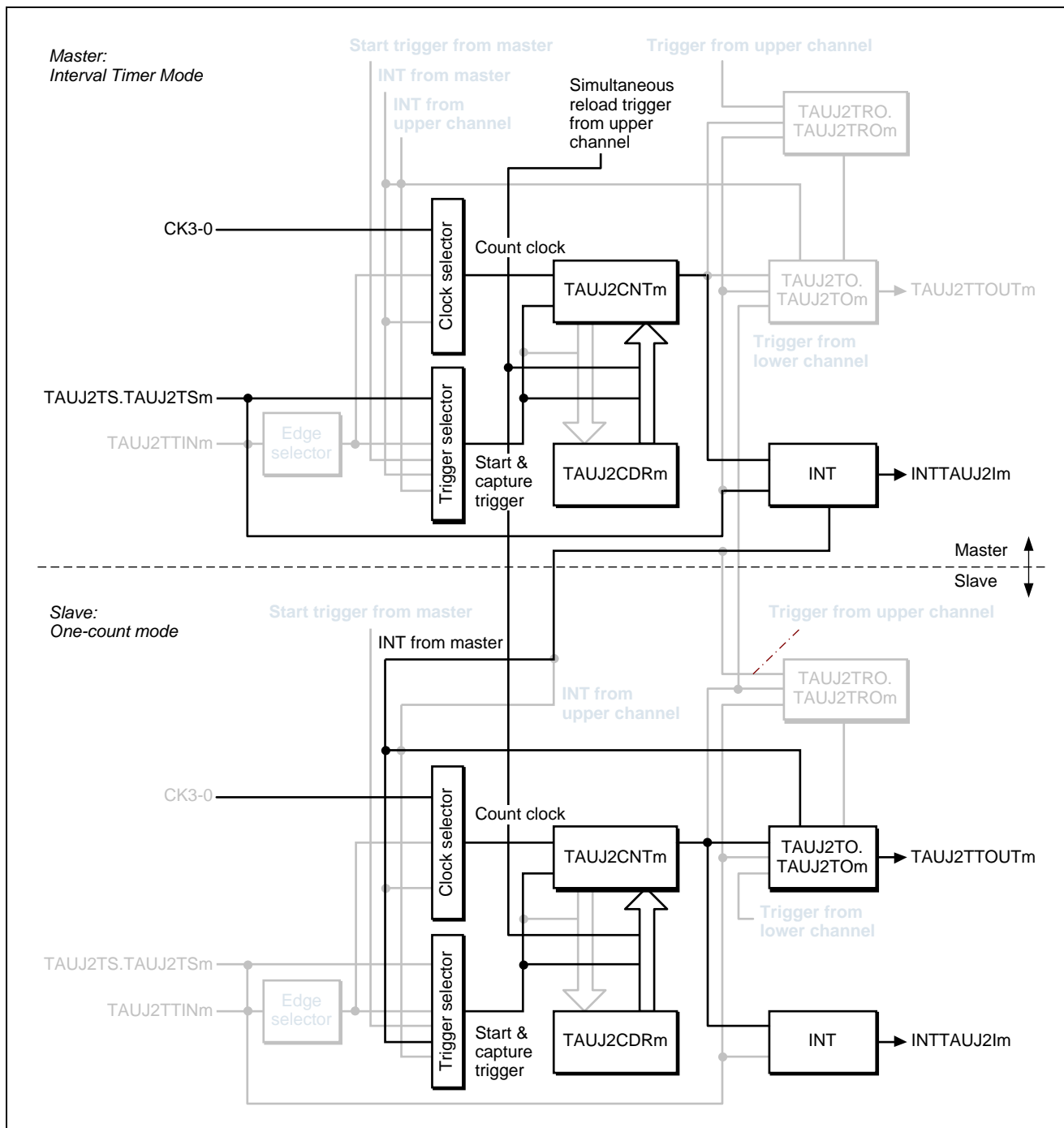


Figure 14.36 Block Diagram of PWM Output

(3) Timing Diagram

The following settings apply to the general timing diagram:

- Slave channel: Positive logic (TAUJ2TOL.TAUJ2TOLm = 0)

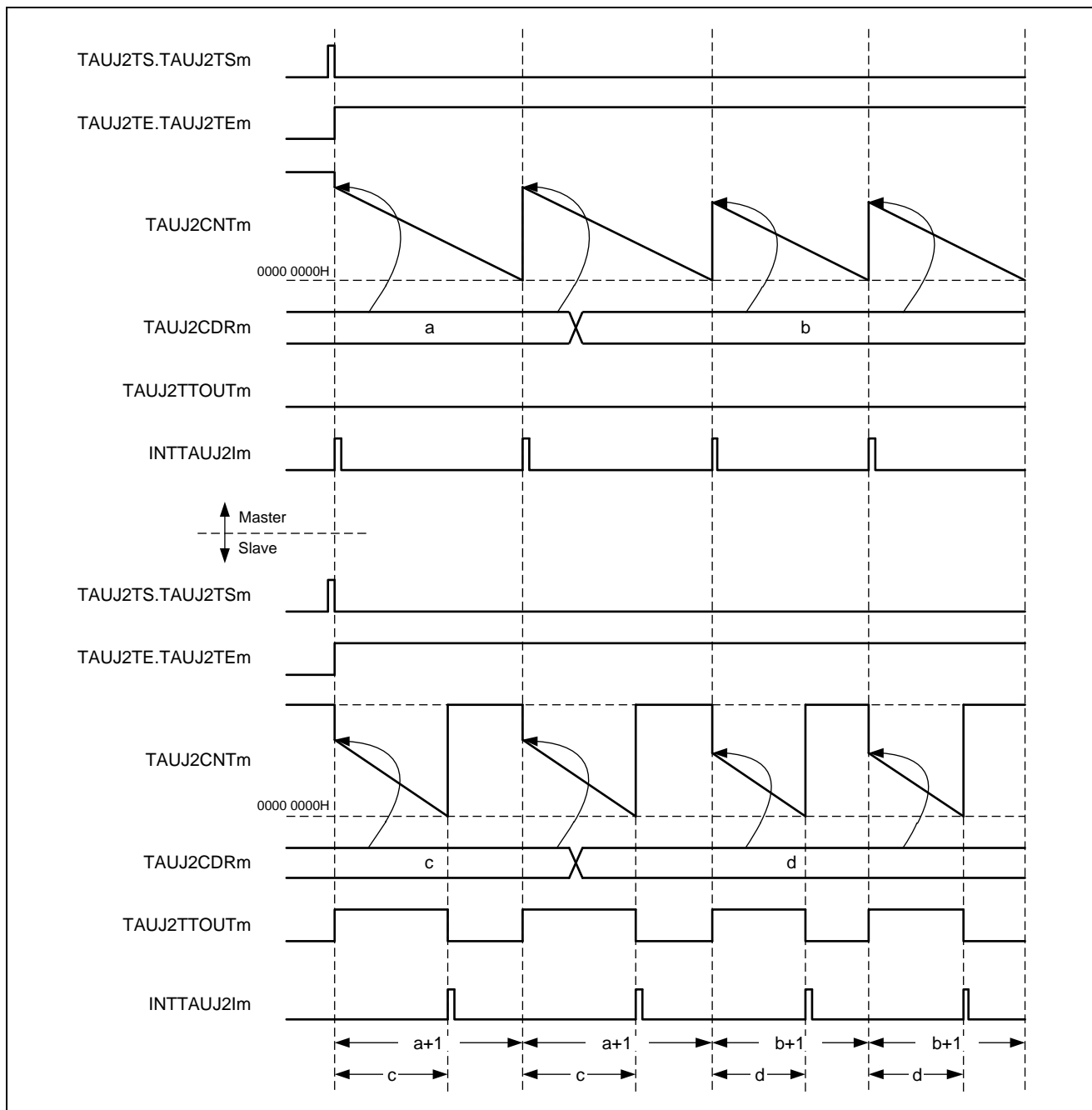


Figure 14.37 General Timing Diagram of PWM Output

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) x count clock cycle

Duty cycle [%] = (TAUJ2CDRm (slave) / (TAUJ2CDRm (master) + 1)) x 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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]			TAUJ2MD0	

Table 14.49 TAUJ2CMORm Settings for the Master Channel of 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

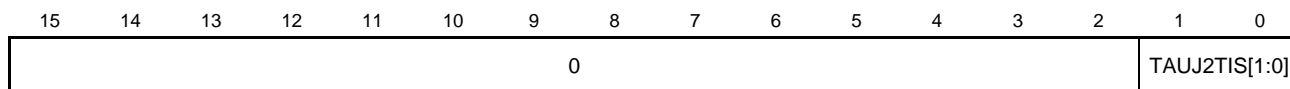


Table 14.50 TAUJ2CMURm Settings for the Master Channel for PWM Output

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 14.51 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 14.52 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]		TAUJ2MAS	TAUJ2STS[2:0]			TAUJ2COS[1:0]		0	TAUJ2MD[4:1]			TAUJ2MD0	

Table 14.53 TAUJ2CMORm Settings for the Slave 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 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 14.54 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 14.55 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 14.56 Control Bit Settings 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 14.57 Operating Procedure for PWM Output

	Operation	Status of TAUJ2	
Restart ↓	Initial Channel Setting	<ul style="list-style-type: none"> • 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.
	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 <ul style="list-style-type: none"> • TAUJ2CDRm register • TAUJ2TOL.TAUJ2TOLm bit • TAUJ2RDT.TAUJ2RDTm bit (When simultaneous reloading is used) Register which is readable at any time <ul style="list-style-type: none"> • 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%

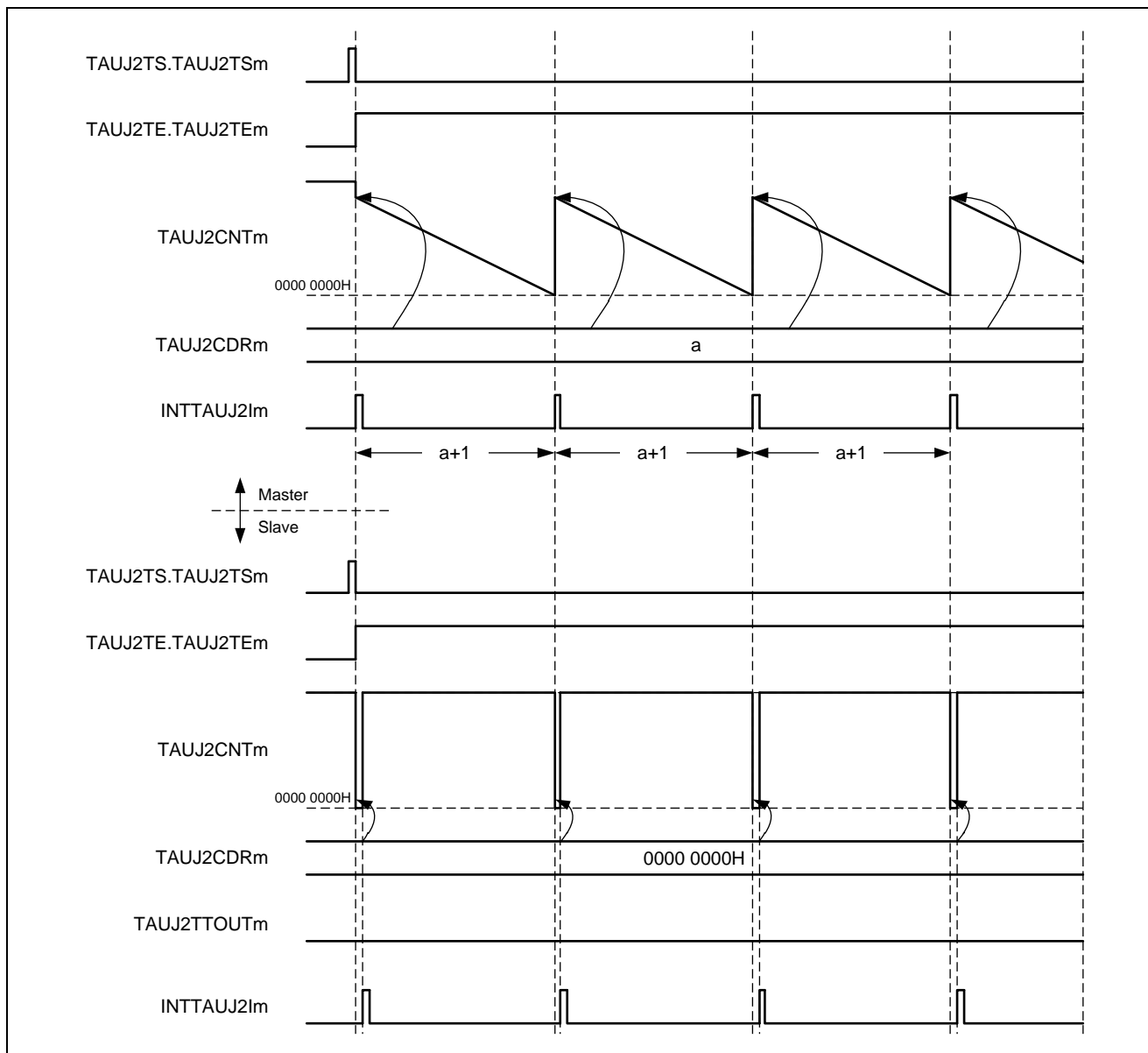


Figure 14.38 TAUJ2CDRm (Slave) = 0000 0000H, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0)

- 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%

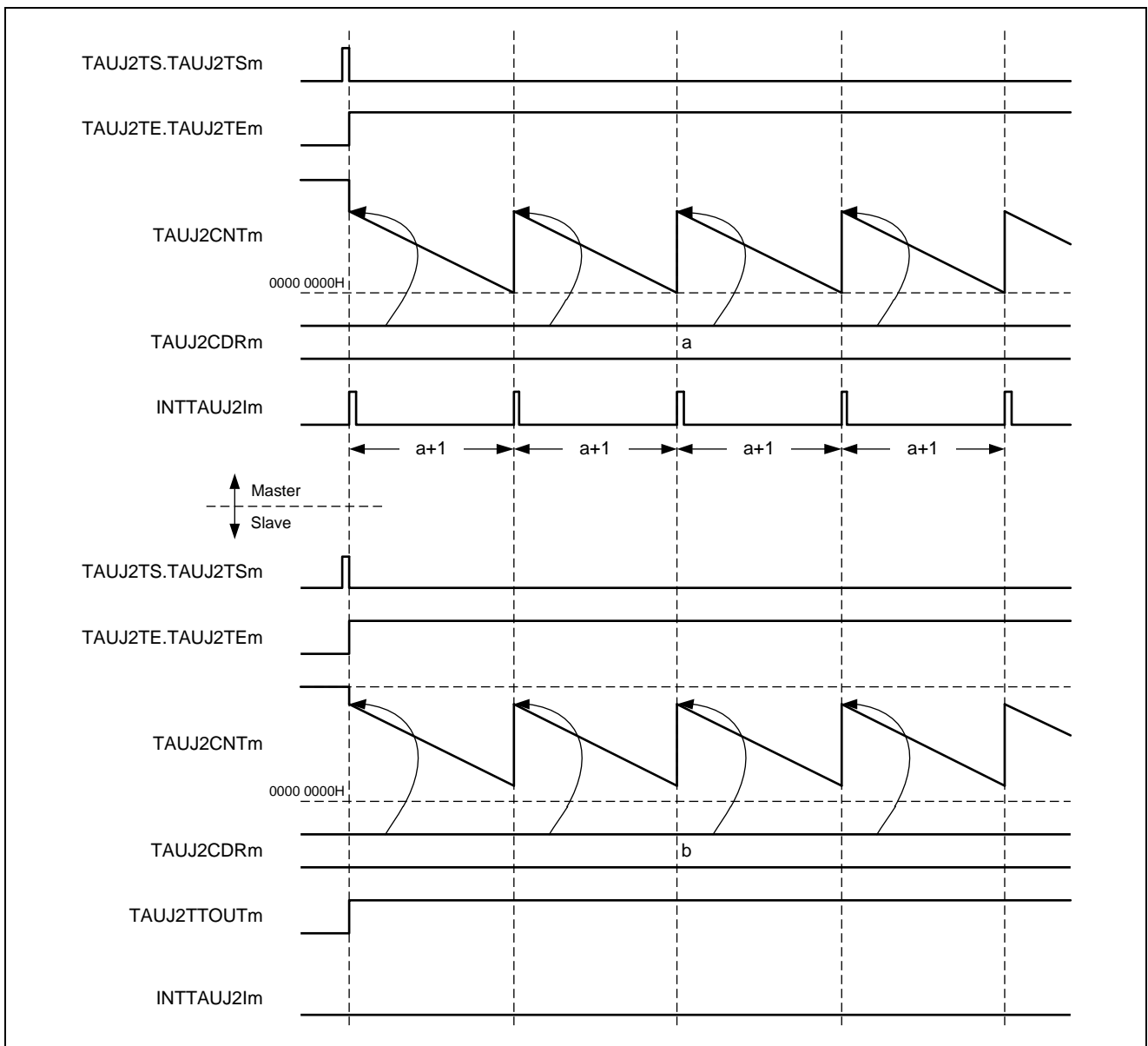


Figure 14.39 TAUJ2CDRm (Slave) ≥ TAUJ2CDRm (Master) + 1, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0)

- 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) Operation stop and restart

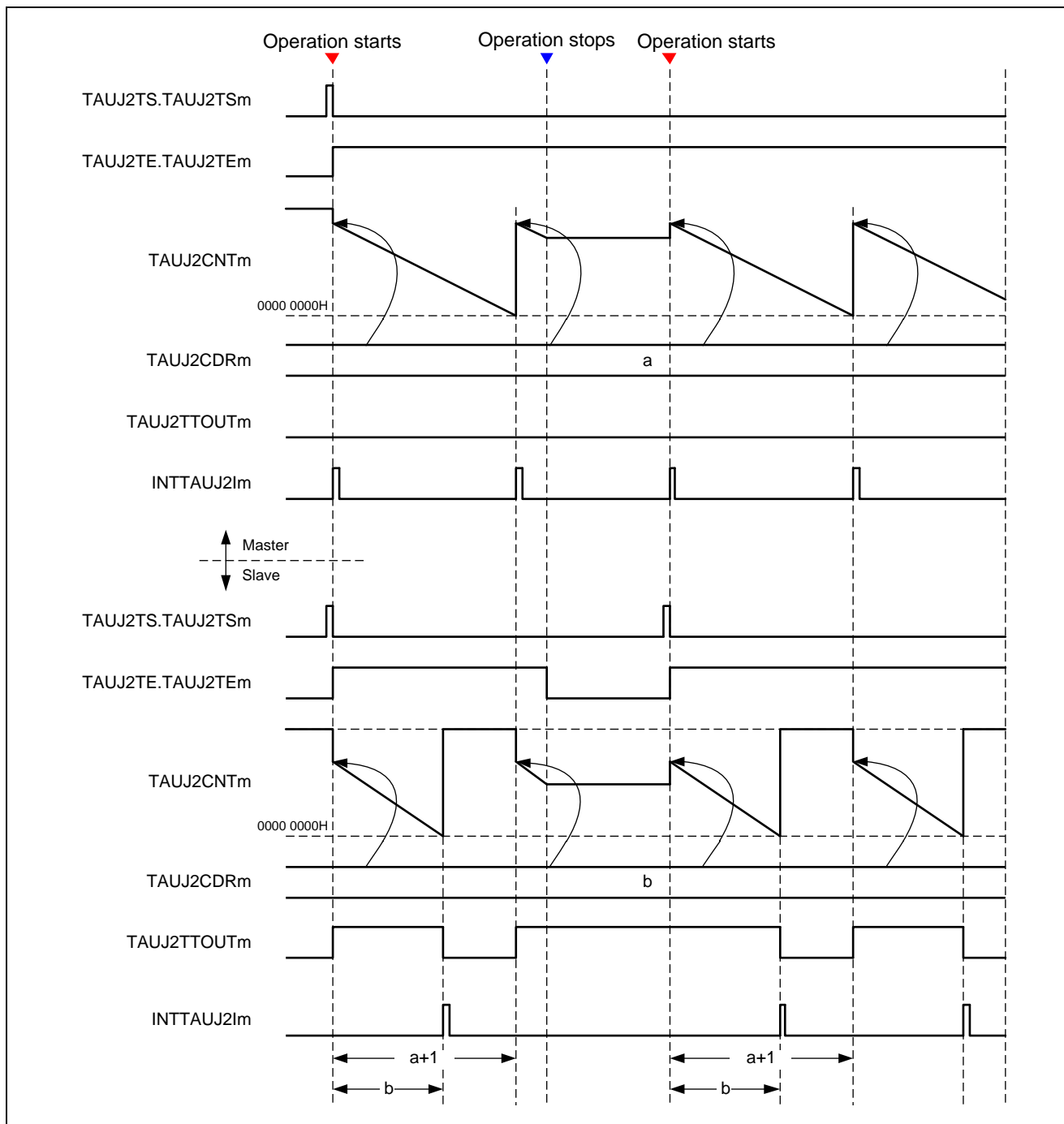


Figure 14.40 Stopping and Restarting Operation, Positive Logic (TAUJ2TOL.TAUJ2TOLm (Slave) = 0)

- 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

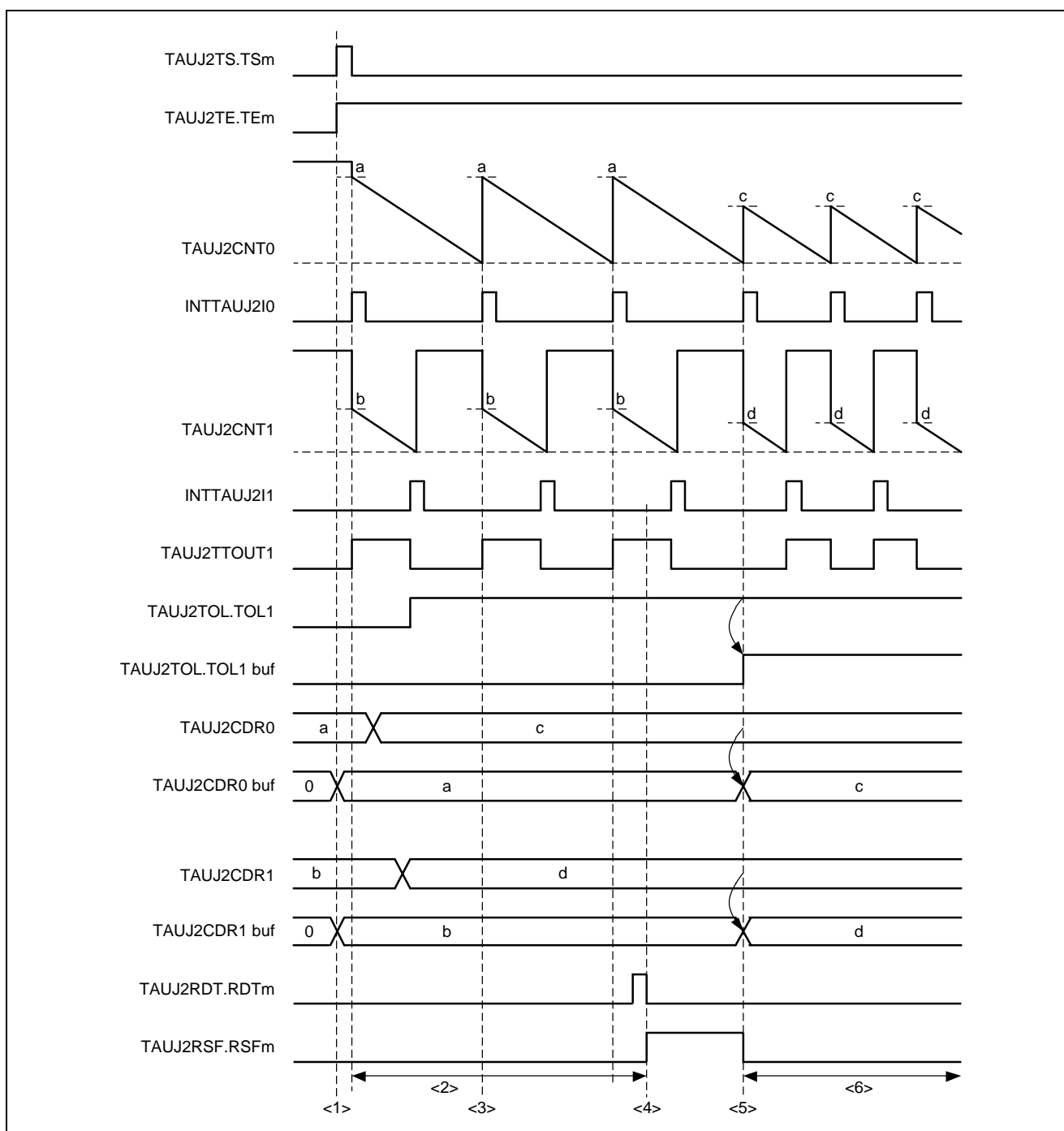


Figure 14.41 Simultaneous Reloading of the Master Channel

- 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.

15. Window Watchdog Timer A (WDTA)

This section explains window watchdog timer A.

15.1 Features of WDTA

This microcontroller has the following number of channels of window watchdog timer A.

Table 15.1 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 15.2 WDTA Interrupts and Reset Outputs

WDTA Signals	Function	Connected to
WDTA0		
WDTA0TRES	WDTA0 error reset	Reset Controller WDTA0RES
WDTA0TNMI	WDTA0 error NMI	Cortex-M3 NMI Input Output as WDTOUTZ through the port (P25) WDTIL input of CC-Link IE Field Network ^{Note} Wdl_n input of CC-Link (remote device station)

Note: This is only provided in the R-IN32M3-CL.

15.2 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
- Window function
- Overflow Time
 - 163 μ s to 5.36 s

The following figure shows the main components of WDTA:

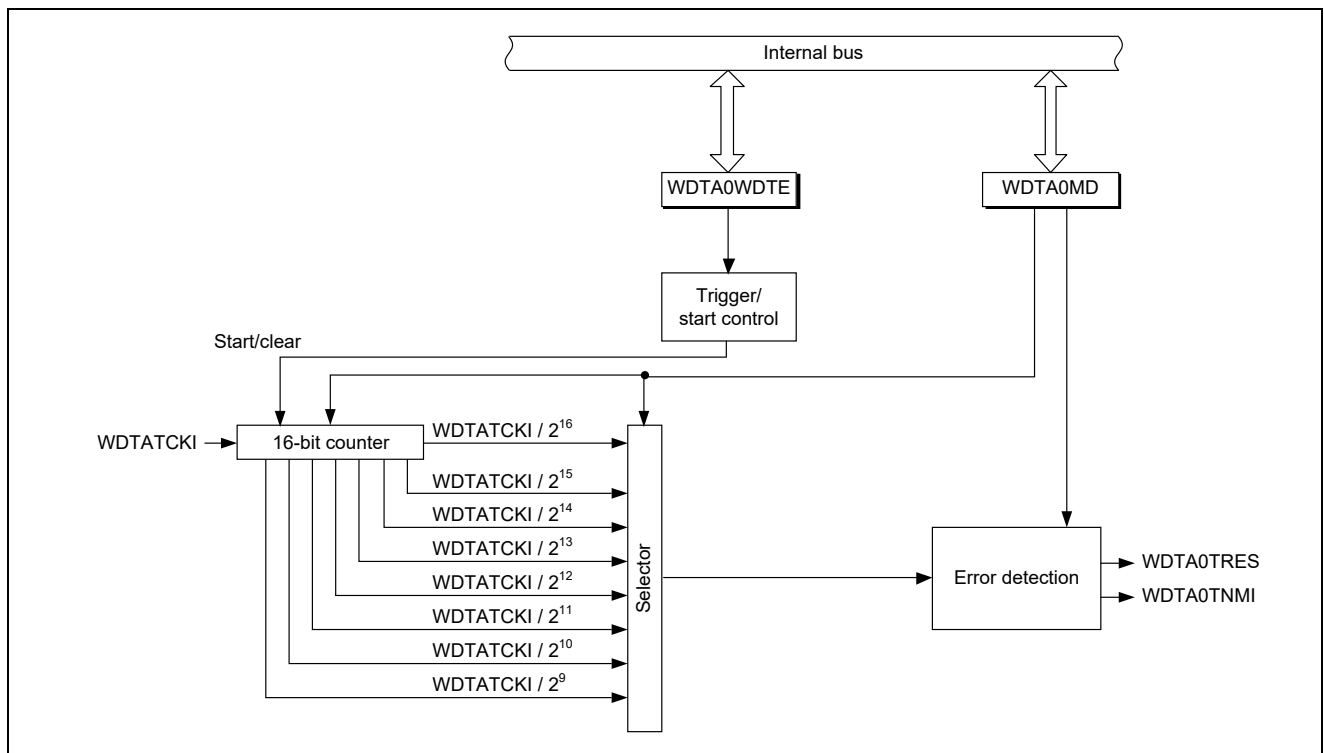


Figure 15.1 Block Diagram of WDTA

15.3 Registers

This section contains a description of all registers of WDTA.

15.3.1 Overview of WDTA Registers

WDTA is controlled and operated by the following registers:

Table 15.3 Overview of WDTA Registers

Register Name	Symbol	Address
WDTA enable register	WDTA0WDTE	4000 0700H
WDTA mode register	WDTA0MD	4000 070CH

15.3.2 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 15.4.2, WDTA Trigger, for details.
- **Access** This register can be read/written in 8-bit units.
- **Initial value** WDTA0 is started while it is disabled.

								Address	Initial Value
	7	6	5	4	3	2	1	0	
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 WDTA0: 0: WDTA0 disabled 1: WDTA0 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.

(1/2)

					Address	Initial Value			
7	6	5	4	3	2	1	0		
0	WDTA0OVF[2:0]			0	WDTA0ERM	WDTA0WS[1:0]		4000 070CH	0FH
R/W	0	R/W			0	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: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>WDTA0OVF2</th> <th>WDTA0OVF1</th> <th>WDTA0OVF0</th> <th>Overflow Interval Time</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>2^9 / WDTATCKI</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>2^{10} / WDTATCKI</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>2^{11} / WDTATCKI</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>2^{12} / WDTATCKI</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>2^{13} / WDTATCKI</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>2^{14} / WDTATCKI</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>2^{15} / WDTATCKI</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>2^{16} / WDTATCKI</td> </tr> </tbody> </table>	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
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																																			
3	—	Reserved																																				
2	WDTA0ERM	Specifies the error mode: 0: NMI request mode 1: Reset mode																																				

(2/2)

Bit Position	Bit Name	Description															
1, 0	WDTA0WS[1:0]	Select the open window size: <table border="1"><thead><tr><th>WDTA0WS1</th><th>WDTA0WS0</th><th>Open Window Size</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>25%</td></tr><tr><td>0</td><td>1</td><td>50%</td></tr><tr><td>1</td><td>0</td><td>75%</td></tr><tr><td>1</td><td>1</td><td>100%</td></tr></tbody></table>	WDTA0WS1	WDTA0WS0	Open Window Size	0	0	25%	0	1	50%	1	0	75%	1	1	100%
WDTA0WS1	WDTA0WS0	Open Window Size															
0	0	25%															
0	1	50%															
1	0	75%															
1	1	100%															

15.4 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 15.4.3, Error Detection.

The counter is cleared and restarted every time a WDTA trigger occurs in the open window period. Refer to section 15.4.2, WDTA Trigger and section 15.4.4, Window Function, for details.

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 15.4.1, WDTA after Release from the Reset State.

15.4.1 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 15.4.1, 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.

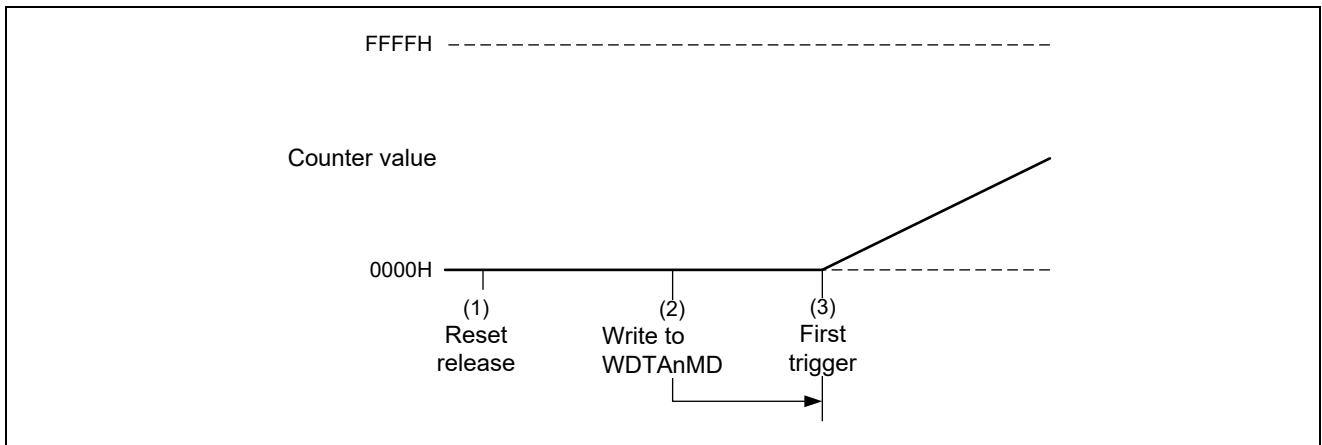


Figure 15.2 Timing Diagram of WDTA Start in Software Trigger Start Mode

The timing diagram above shows the following:

1. After release from 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.

15.4.2 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 15.4 Trigger Register and Activation Code

Trigger register	Activation code
WDTA0WDTE	ACH

15.4.3 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.

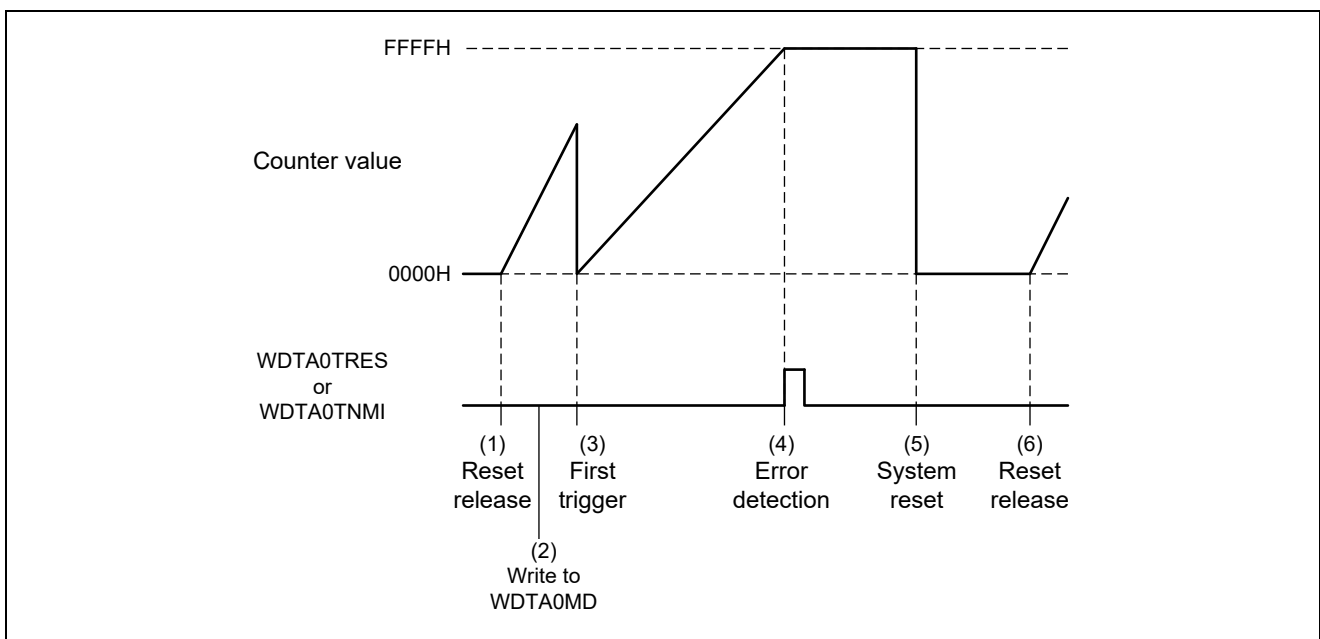


Figure 15.3 Timing Diagram of WDTA NMI Request or Reset Generation

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.

15.4.4 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.

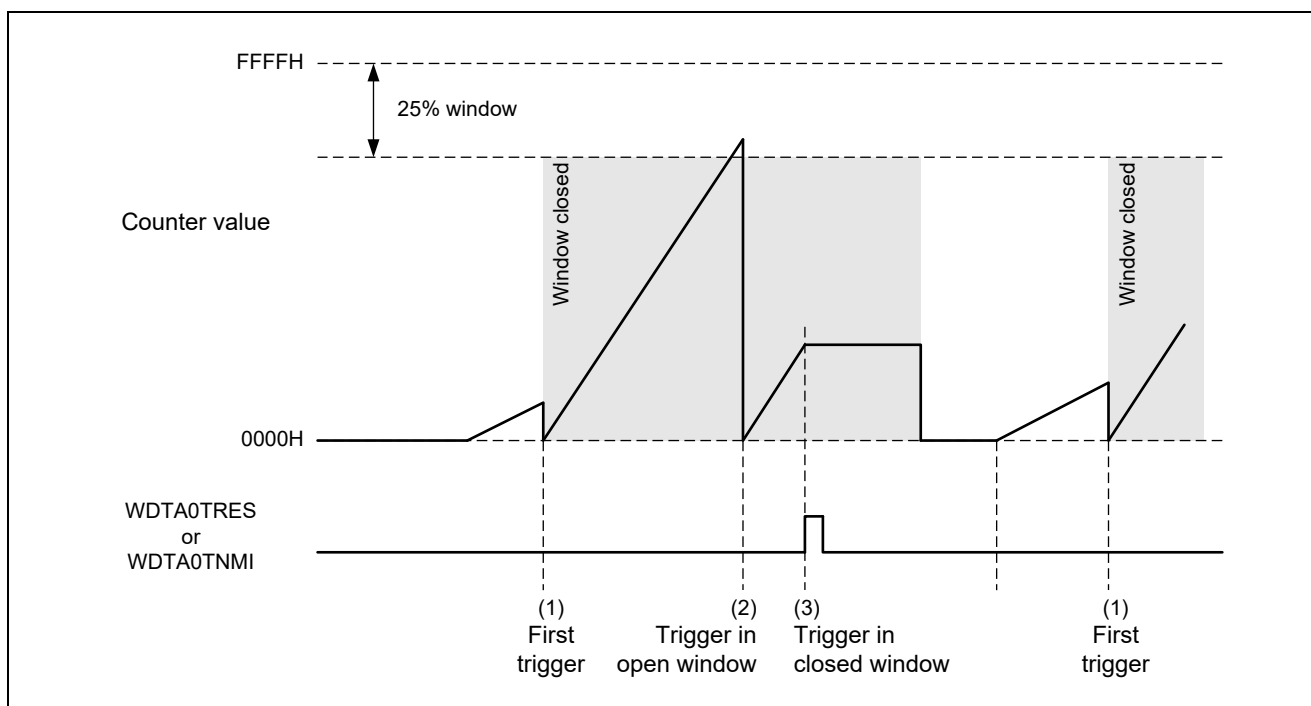


Figure 15.4 Timing Diagram of WDTA Window Function

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.

15.5 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.

15.6 Notes

- (1) Be sure to reset WDTA by using the software reset register (SFTRES1) before accessing each register of WDTA (window watchdog timer A). The reset time of at least 640 ns must be secured.
- (2) 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).

16. Asynchronous Serial Interface J (UARTJ)

This section explains asynchronous serial interface J (UARTJ).

16.1 Features of UARTJn

- Number of channels: This microcontroller has two channels of asynchronous serial interface J (UARTJn).

Table 16.1 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 16.2 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), and for updating the real-time port pins (RP00-RP37).

Table 16.3 UARTJn Interrupts

UARTJn Signals	Function	Connected to
UARTJ0		
INTUAJ0TIT	Transmission interrupt	<ul style="list-style-type: none"> • Interrupt controller INTUAJ0TIT • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR)
INTUAJ0TIR	Reception interrupt	<ul style="list-style-type: none"> • Interrupt controller INTUAJ0TIR • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR)
INTUAJ0TIS	Status interrupt	<ul style="list-style-type: none"> • Interrupt controller INTUAJ0TIS • HW-RTOS (Hardware ISR)
UARTJ1		
INTUAJ1TIT	Transmission interrupt	<ul style="list-style-type: none"> • Interrupt controller INTUAJ1TIT • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR)
INTUAJ1TIR	Reception interrupt	<ul style="list-style-type: none"> • Interrupt controller INTUAJ1TIR • HW-RTOS (Hardware ISR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR)
INTUAJ1TIS	Status interrupt	<ul style="list-style-type: none"> • Interrupt controller INTUAJ1TIS • HW-RTOS (Hardware ISR)

16.2 Functional Overview

- Full-duplex communications via built-in receive and transmit FIFOs:
 - Internal UARTJn 10 bit x 16 receive data FIFO (URTJnFRX)
 - Internal UARTJn 8 bit x 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

16.3 Configuration

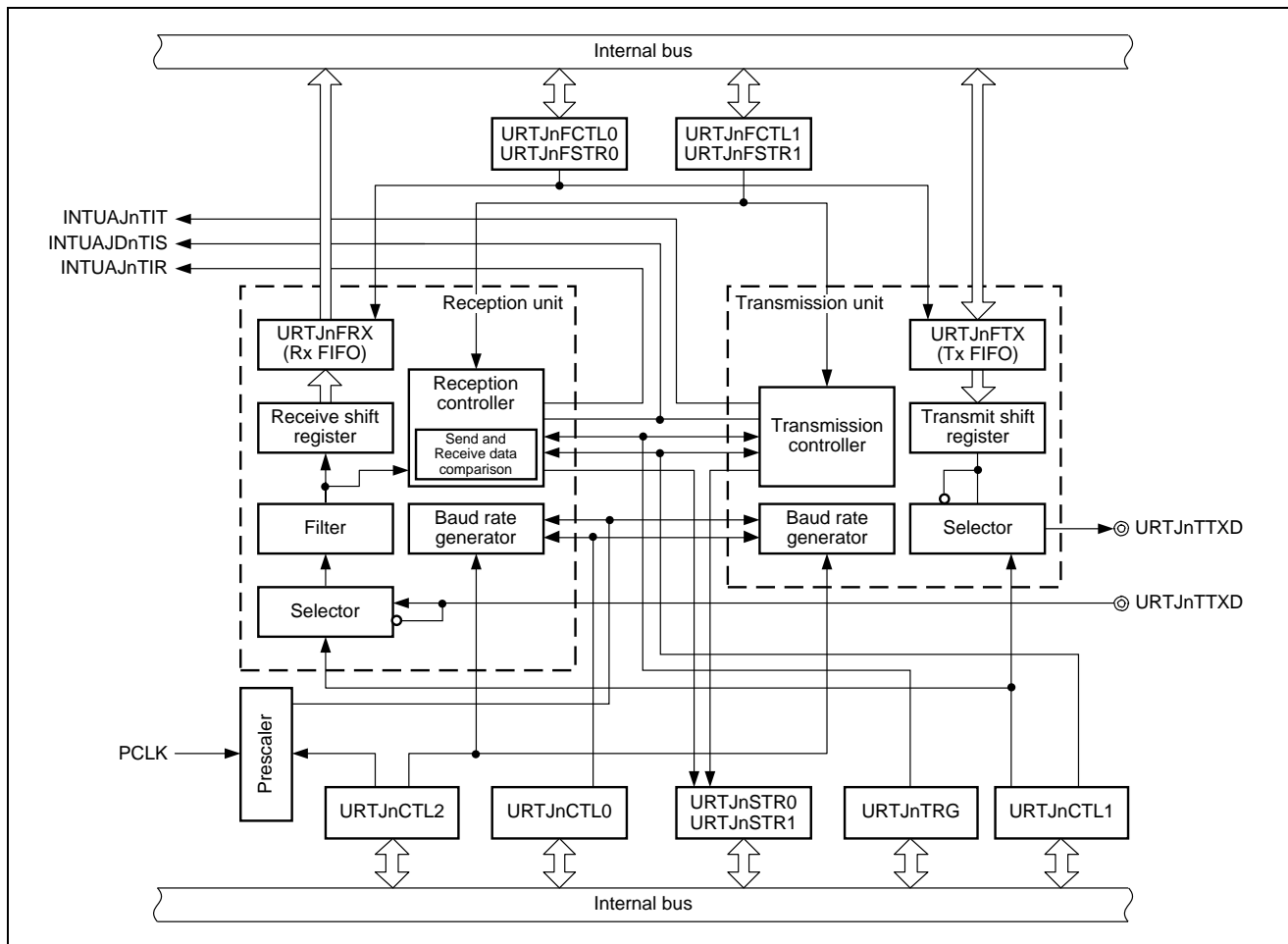


Figure 16.1 Block Diagram of Asynchronous Serial Interface UARTJn

16.4 UARTJn Registers

UARTJn is controlled and operated by using the following registers:

Table 16.4 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 or written in 32- 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
URTJnCTL0	0 0	<div style="display: flex; justify-content: space-around; font-size: 8px;"> URTJnPW URTJnTXE URTJnRXE </div>	4000 0300H +100H × n Initial Value 0000 0000H
R/W	0 0	<div style="display: flex; justify-content: space-around; font-size: 8px;"> R/W R/W R/W </div>	

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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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	<p>Data consistency check enable</p> <p>0: Disable consistency check 1: Enable consistency check</p> <p>This bit selects the handling of data consistency error checks when transmitting data.</p> <p>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.</p> <p>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.</p>

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.

(2/3)

Bit Position	Bit Name	Function																						
7, 6	URTJnSLP[1:0]	Parity bit selection <table border="1" data-bbox="550 459 1348 683" style="margin: 10px auto;"> <thead> <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> </thead> <tbody> <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> </tbody> </table> <ul style="list-style-type: none"> • 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). 	URTJnSLP1	URTJnSLP0	Operation		Transmission	Reception	0	0	Output without parity bit	Received with no parity	0	1	Output 0 parity (0-fixed)	No parity judgment	1	0	Output odd parity	Judged as odd parity	1	1	Output even parity	Judged as even parity
URTJnSLP1	URTJnSLP0	Operation																						
		Transmission	Reception																					
0	0	Output without parity bit	Received with no parity																					
0	1	Output 0 parity (0-fixed)	No parity judgment																					
1	0	Output odd parity	Judged as odd parity																					
1	1	Output even parity	Judged as even parity																					
5	URTJnTDL	Transmission data level control 0: No inverted output of transmit data 1: Inverted output of transmit data <ul style="list-style-type: none"> • 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). 																						
4	URTJnRDL	Reception data level control 0: No inverted output of receive data 1: Inverted output of receive data <ul style="list-style-type: none"> • 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). 																						
3	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.																						
2	URTJnSLG	Stop bit number selection for transmission data 0: 1 bit 1: 2 bits <ul style="list-style-type: none"> • 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). 																						
1	URTJnSLD	Transfer direction selection 0: MSB-first transfer 1: LSB-first transfer <ul style="list-style-type: none"> • 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). 																						

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 to the transmission shift register 1: INTUAJnTIT generated at transmission completion <ul style="list-style-type: none">• 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 or written in 32-bit units.

Caution: Writing to this register is only allowed if the UARTJn operation is disabled (URTJnCTL0.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		Address		
URTJnCTL2	0 0	URTJnPRS[2:0]		4000 0324H +100H x n Initial Value 0000 EFFFH	
R/W	0 0	R/W		R/W	

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 (PRCLK) division value 0: PRCLK = PCLK / 2 ⁰ 1: PRCLK = PCLK / 2 ¹ 2: PRCLK = PCLK / 2 ² 3: PRCLK = PCLK / 2 ³ 4: PRCLK = PCLK / 2 ⁴ 5: PRCLK = PCLK / 2 ⁵ 6: PRCLK = PCLK / 2 ⁶ 7: PRCLK = 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 <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">URTJnBRS[11:0]</th> <th style="width: 40%;">Transmit/receive BRCLK</th> <th style="width: 40%;">BF receive clock</th> </tr> </thead> <tbody> <tr> <td>000H</td> <td rowspan="5" style="text-align: center;">PRCLK/(2 x 4)</td> <td rowspan="5" style="text-align: center;">PRCLK/4</td> </tr> <tr><td>001H</td></tr> <tr><td>002H</td></tr> <tr><td>003H</td></tr> <tr><td>004H</td></tr> <tr> <td>005H</td> <td style="text-align: center;">PRCLK/(2 x 5)</td> <td style="text-align: center;">PRCLK/5</td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">PRCLK/ (2 x URTJnBRS[11:0])</td> <td style="text-align: center;">PRCLK/ URTJnBRS[11:0]</td> </tr> <tr> <td>FFEH</td> <td style="text-align: center;">PRCLK/(2 x 4094)</td> <td style="text-align: center;">PRCLK/4094</td> </tr> <tr> <td>FFFH</td> <td style="text-align: center;">PRCLK/(2 x 4095)</td> <td style="text-align: center;">PRCLK/4095</td> </tr> </tbody> </table>	URTJnBRS[11:0]	Transmit/receive BRCLK	BF receive clock	000H	PRCLK/(2 x 4)	PRCLK/4	001H	002H	003H	004H	005H	PRCLK/(2 x 5)	PRCLK/5	...	PRCLK/ (2 x URTJnBRS[11:0])	PRCLK/ URTJnBRS[11:0]	FFEH	PRCLK/(2 x 4094)	PRCLK/4094	FFFH	PRCLK/(2 x 4095)	PRCLK/4095
URTJnBRS[11:0]	Transmit/receive BRCLK	BF receive clock																						
000H	PRCLK/(2 x 4)	PRCLK/4																						
001H																								
002H																								
003H																								
004H																								
005H	PRCLK/(2 x 5)	PRCLK/5																						
...	PRCLK/ (2 x URTJnBRS[11:0])	PRCLK/ URTJnBRS[11:0]																						
FFEH	PRCLK/(2 x 4094)	PRCLK/4094																						
FFFH	PRCLK/(2 x 4095)	PRCLK/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 or written in 32- 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	4000 0304H +100H × n
URTJnTRG	0 0	Initial Value	0000 0000H
R/W	0 R/W R/W 0 0 0 0 0 0		

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 <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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	<p>BF transmission trigger</p> <p>0: Read value is always 0, writing 0 is ignored</p> <p>1: Enable BF transmission</p> <ul style="list-style-type: none"> • 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- or 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.

URTJnSTR0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">31</td><td style="width: 20px; text-align: center;">30</td><td style="width: 20px; text-align: center;">29</td><td style="width: 20px; text-align: center;">28</td><td style="width: 20px; text-align: center;">27</td><td style="width: 20px; text-align: center;">26</td><td style="width: 20px; text-align: center;">25</td><td style="width: 20px; text-align: center;">24</td><td style="width: 20px; text-align: center;">23</td><td style="width: 20px; text-align: center;">22</td><td style="width: 20px; text-align: center;">21</td><td style="width: 20px; text-align: center;">20</td><td style="width: 20px; text-align: center;">19</td><td style="width: 20px; text-align: center;">18</td><td style="width: 20px; text-align: center;">17</td><td style="width: 20px; text-align: center;">16</td><td style="width: 20px; text-align: center;">15</td><td style="width: 20px; text-align: center;">14</td><td style="width: 20px; text-align: center;">13</td><td style="width: 20px; text-align: center;">12</td><td style="width: 20px; text-align: center;">11</td><td style="width: 20px; text-align: center;">10</td><td style="width: 20px; text-align: center;">9</td><td style="width: 20px; text-align: center;">8</td><td style="width: 20px; text-align: center;">7</td><td style="width: 20px; text-align: center;">6</td><td style="width: 20px; text-align: center;">5</td><td style="width: 20px; text-align: center;">4</td><td style="width: 20px; text-align: center;">3</td><td style="width: 20px; text-align: center;">2</td><td style="width: 20px; text-align: center;">1</td><td style="width: 20px; text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> </table>	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	0	0	0	0	0	0	0	0	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 4000 0308H +100H × 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																																				
0	0	0	0	0	0	0	0	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 border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td> </tr> </table>	0	0	0	0	0	0	0	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	0	0	0	0	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	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. These bits are 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnRXE = 0 • URTJnSTC.URTJnCLF = 1
0	—	Reserved. This bit is read as 0.

Note 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 16.6.5, UARTJn Transmission and section 16.6.7, 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 or written in 32- or 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	Address																															
URTJnSTC	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> <td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td> </tr> </table>		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 0310H +100H × n	Initial Value 0000 0000H
	0	0	0	0	0	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 border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td><td style="width: 10%; text-align: center;">0</td> </tr> </table>		0	0	0	0	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	
	0	0	0	0	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	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

(9) FIFO control register 1 (URTJnFCTL1)

This register controls the Rx time-out detection.

- Access This register can be read or written in 32- or 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	URTJnSLRT[5:0]	4000 03A0H +100H × n Initial Value 0000 003FH
R/W	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

(11) FIFO status register 1 (URTJnFSTR1)

This register controls the Rx time-out detection.

- Access This register can be read in 32-bit units, and can only be written if URTJnPW = 0

(1/2)

URTJn FSTR1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td><td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> </table>	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	0	0	0	0	0	0	0	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 4000 0388H +100H x n Initial Value 0000 0005H
	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																																	
0	0	0	0	0	0	0	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 border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																	
0	0	0	0	0	0	0	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	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.URTJnCLRO 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

(14) FIFO transmit data register (URTJnFTX)

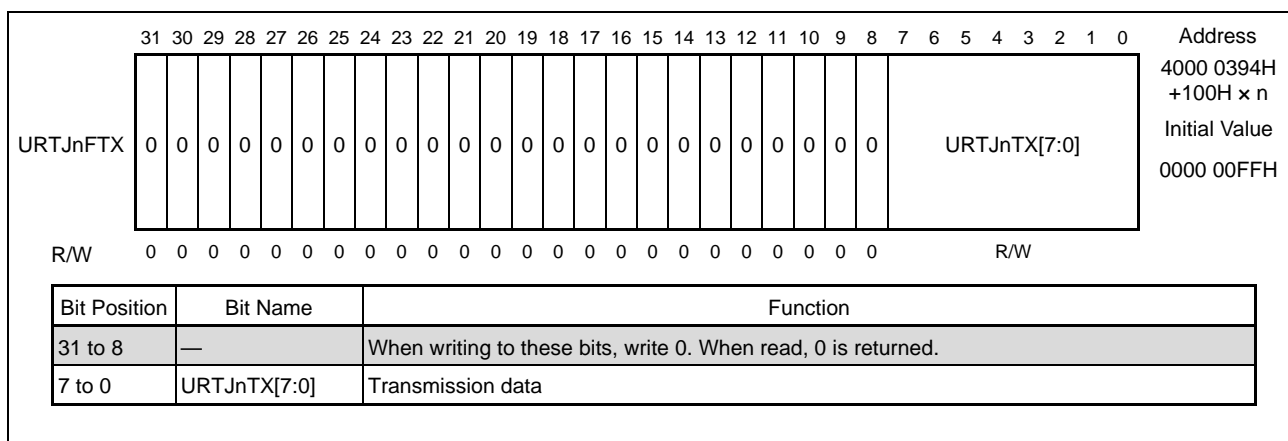
By using this register, the transmission data is written to the Tx FIFO.

- 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 16.6.1, Data Formats.
- Access**

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



- 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.

16.5 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

16.5.1 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.

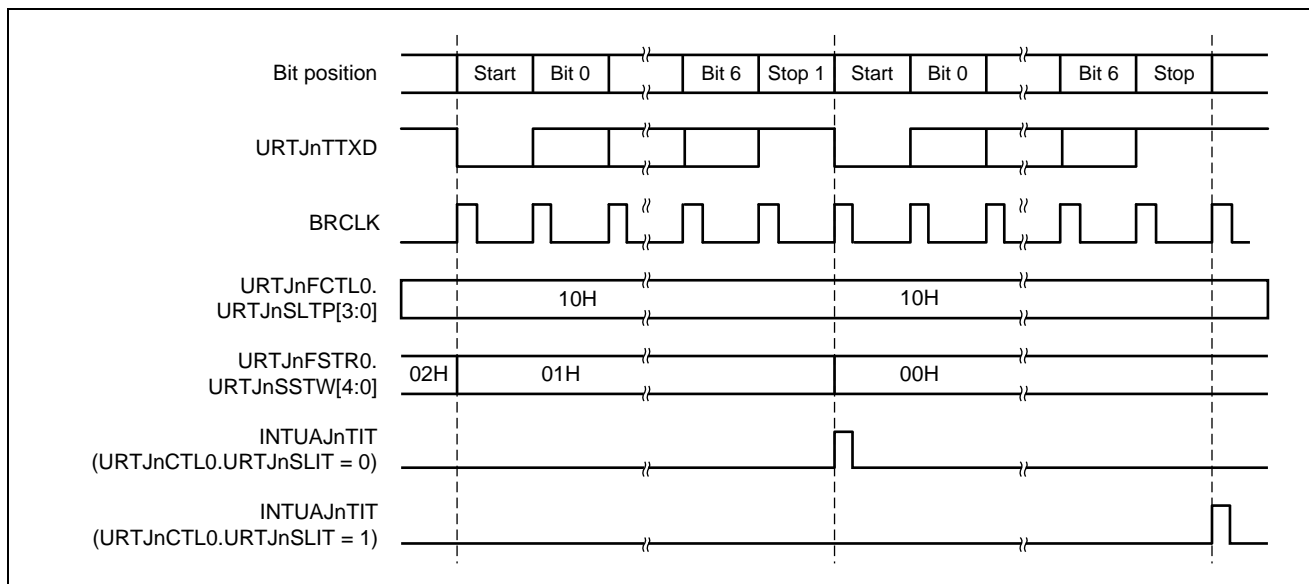


Figure 16.2 Transmission Interrupt Request Timing

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.

16.5.2 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.

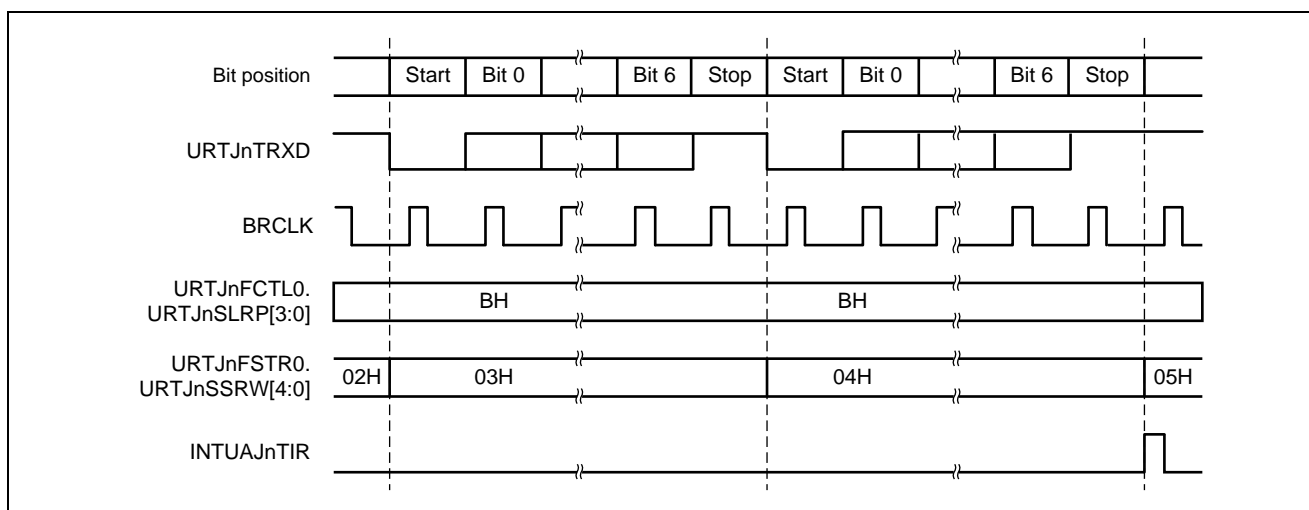


Figure 16.3 Reception Interrupt Request Timing

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.

16.5.3 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).

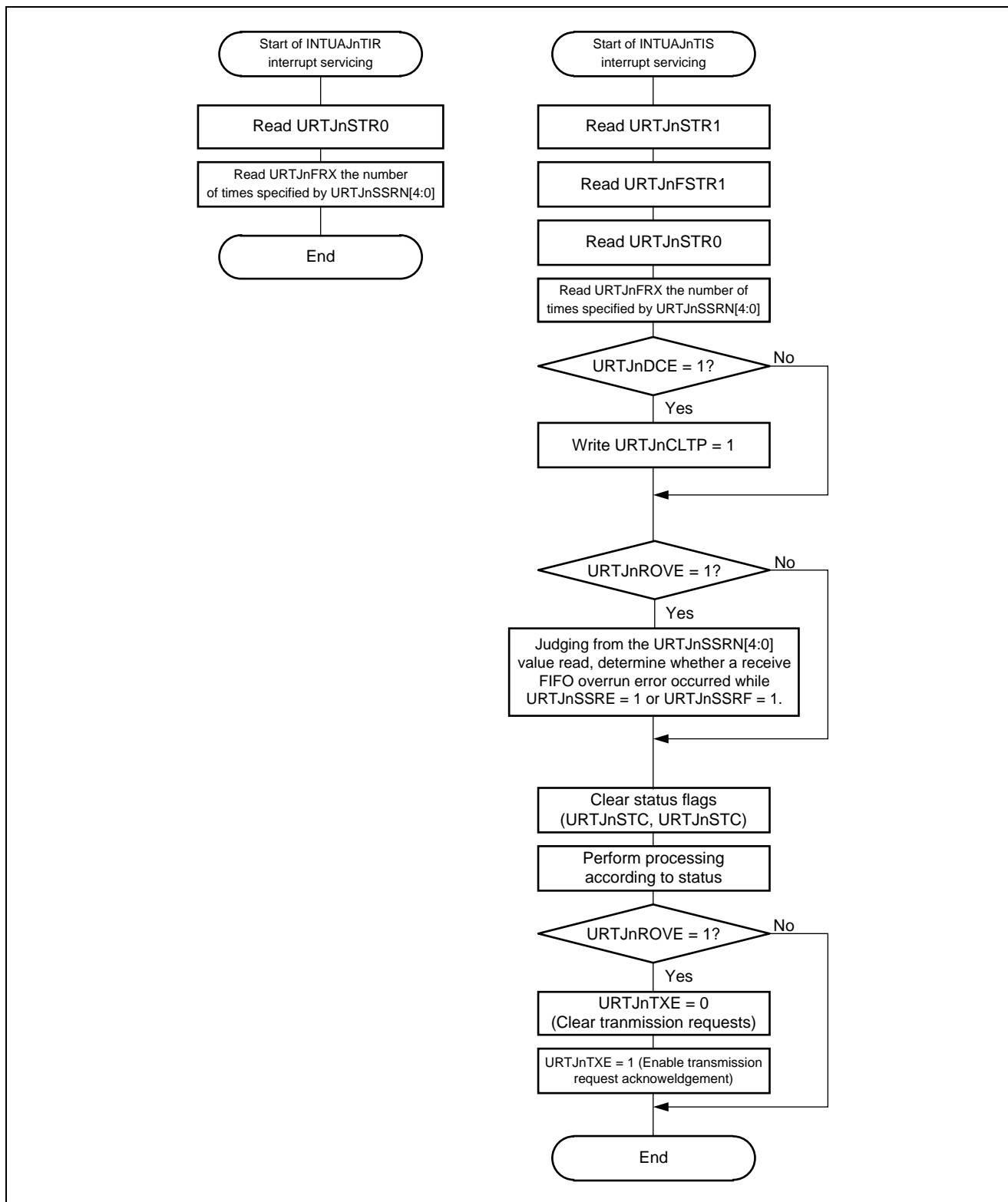


Figure 16.4 Processing Flow after Interrupt Generation

16.6 Operation

16.6.1 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).

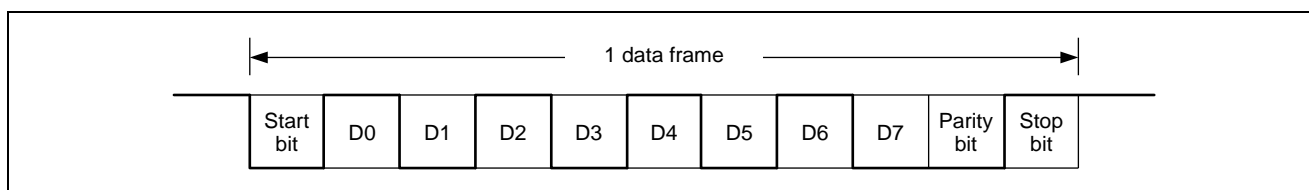
Several properties of a transmit/receive data frame can be specified by control bits of the URTJnCTL1 register:

Table 16.5 Data Format Specification

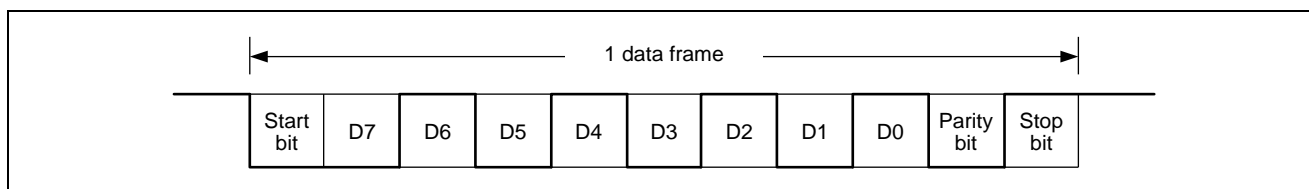
Item	Options	Control Bits
Start bit	1 bit	Fixed
Character bits	7 bits / 8 bits	URTJnCTL1.URTJnCLG
Parity	Even parity / odd parity / 0 parity / no parity	URTJnCTL1.URTJnSLP[1:0]
Stop bit	1 bit / 2 bits	URTJnCTL1.URTJnSLG
Data order	MSB first / LSB first	URTJnCTL1.URTJnSLD
Tx data level	inverted / not inverted	URTJnCTL1.URTJnTDL
Tx data level	inverted / not inverted	URTJnCTL1.URTJnRDL

(1) UARTJn transmit/receive data format

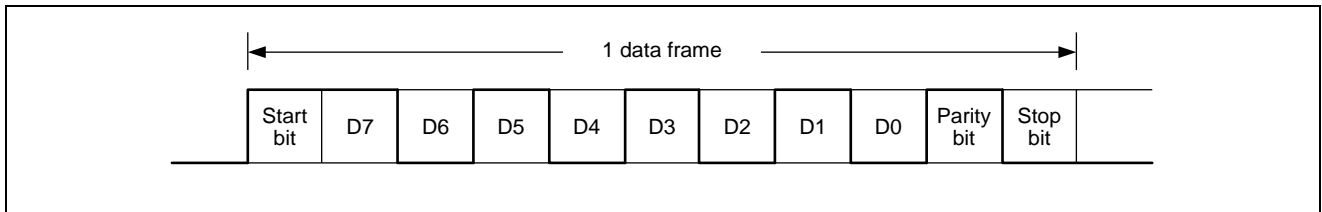
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



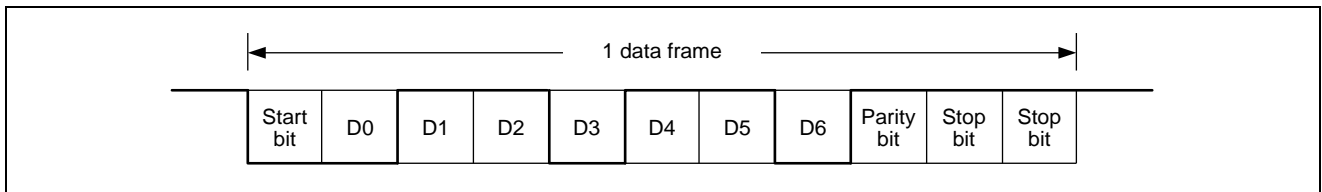
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



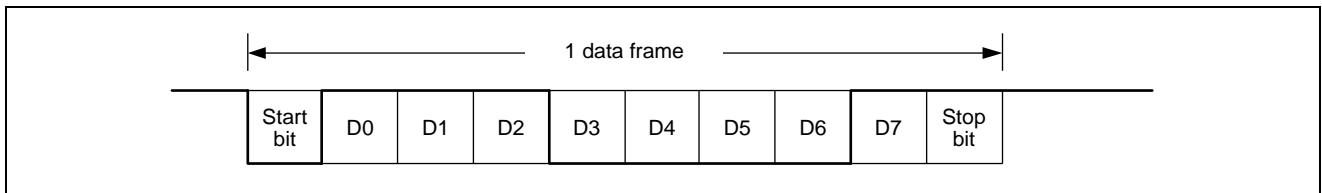
(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, URTJnTTXD inversion



(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



16.6.2 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 $\pm 14\%$ or less.

Figure 16.5, LIN Transmission Outline, and Figure 16.6, LIN Reception Outline outline the transmission and reception operations of LIN.

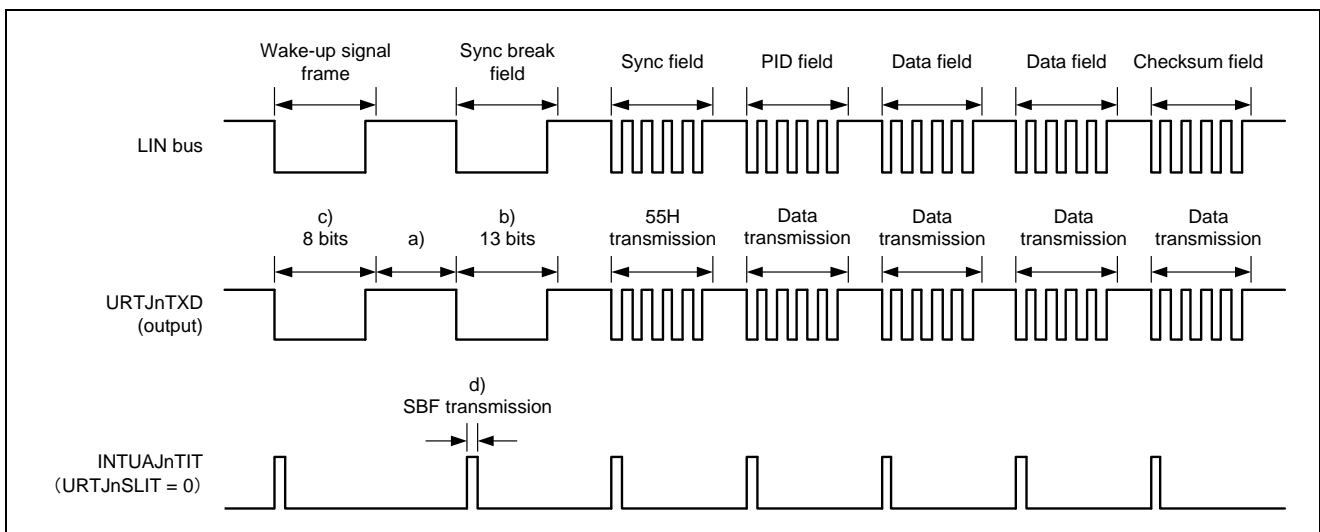


Figure 16.5 LIN Transmission Outline

- The interval between fields is controlled by software.
- 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].
- 80H transfer in the 8-bit mode is substituted for the wakeup signal frame.
- A transmission enable interrupt INTUAJnTIT is generated at the start of each transmission. INTUAJnTIT is also generated at the start of each BF transmission.

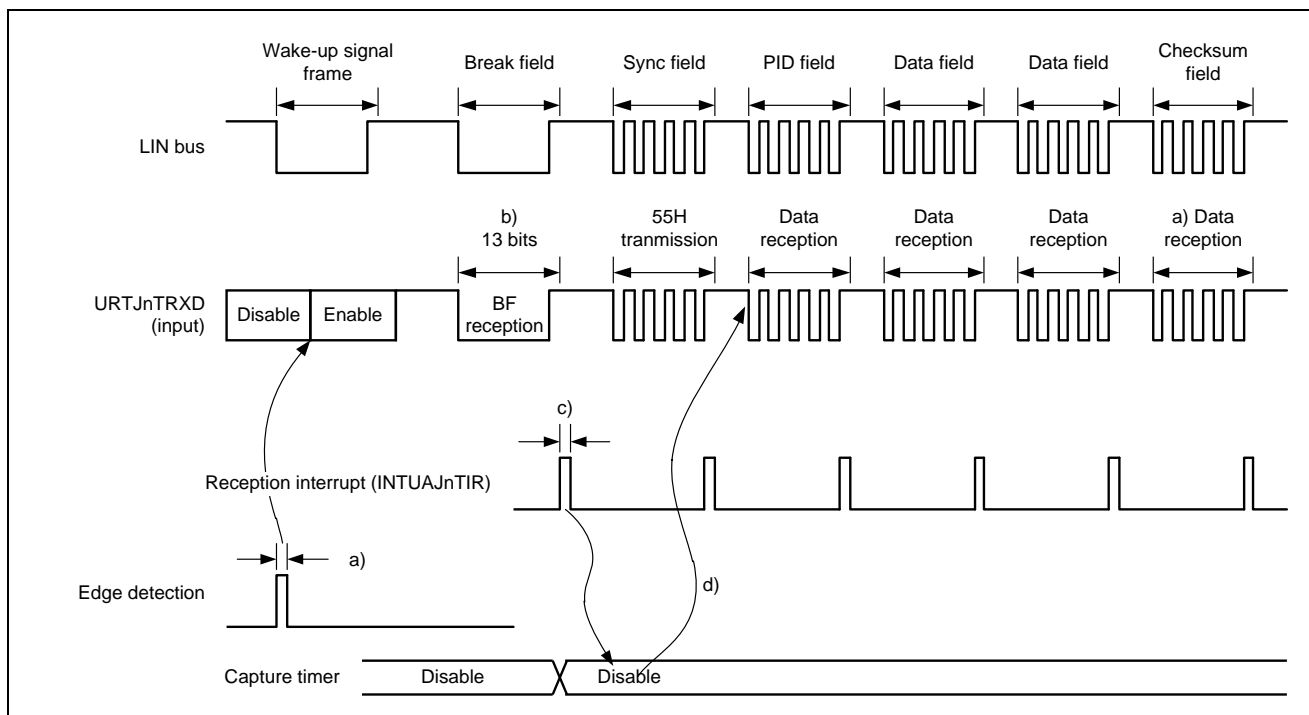


Figure 16.6 LIN Reception Outline

- 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	x	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 URTJnCTL1.URTJnSLBM is 0, the reception interrupt INTUAJnTIR is generated.
 - When URTJnCTL1.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.

16.6.3 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.

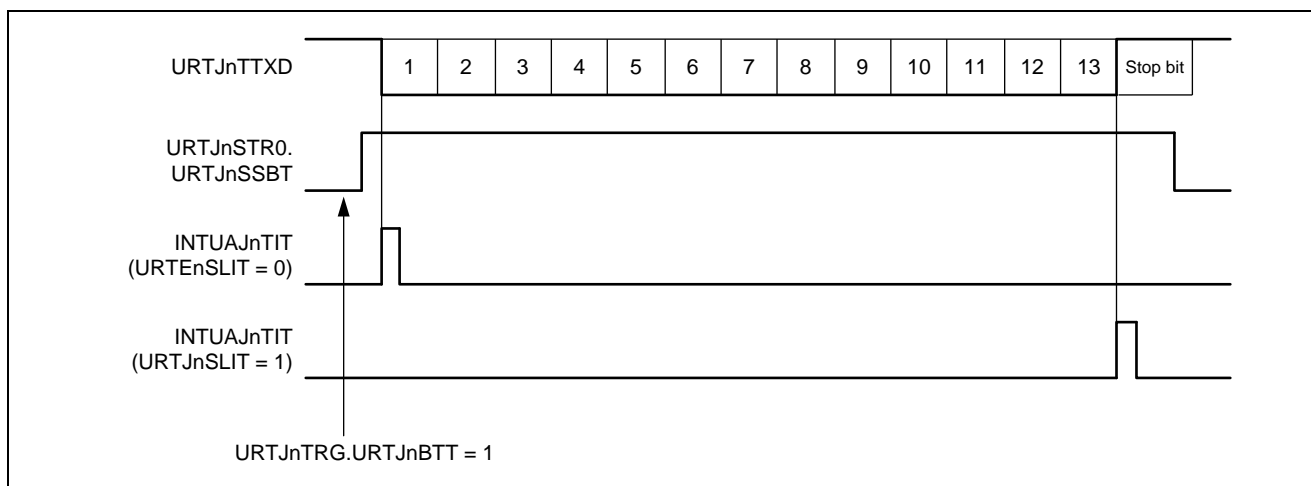


Figure 16.7 BF Transmission

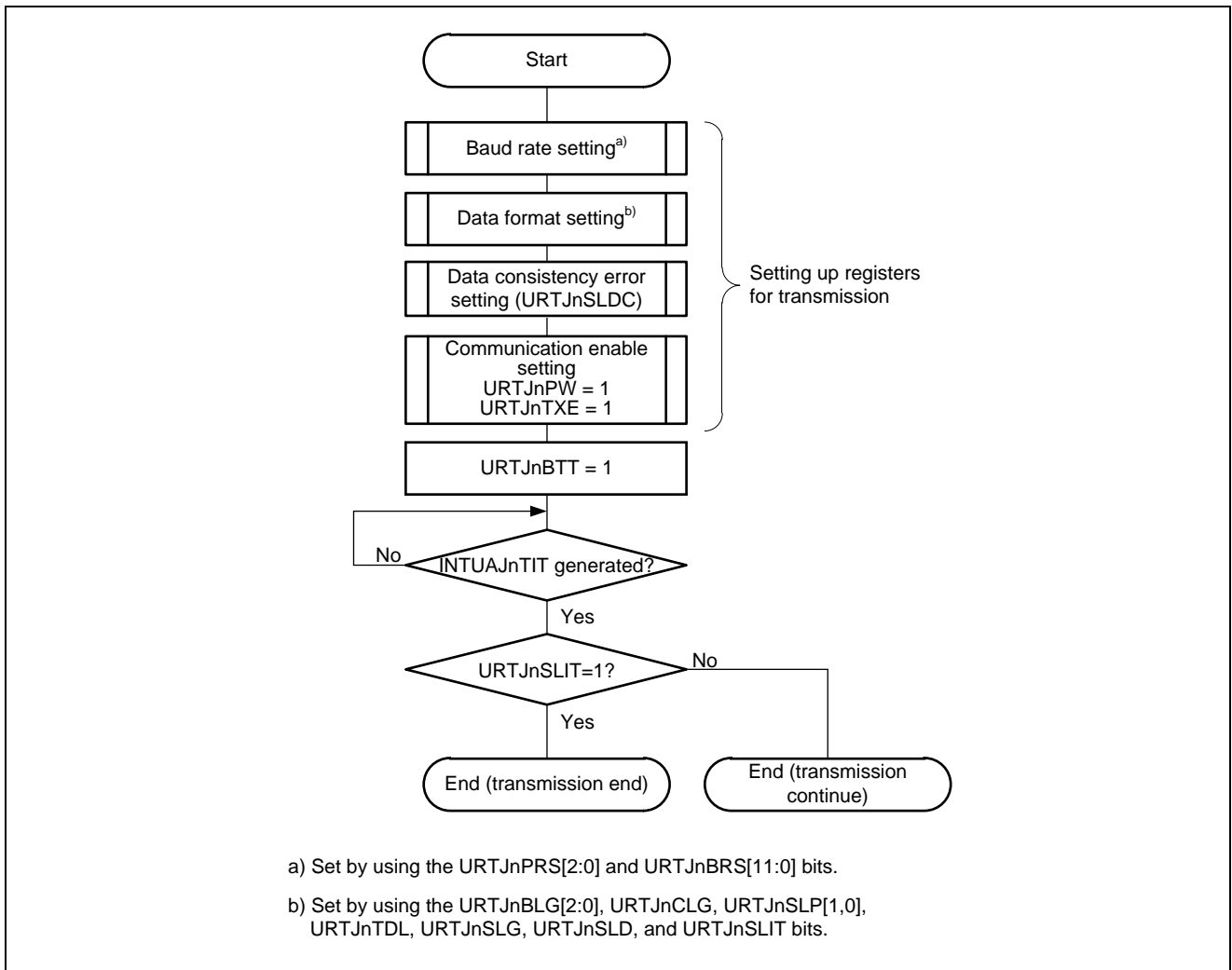


Figure 16.8 Flowchart of BF Transmission

16.6.4 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.

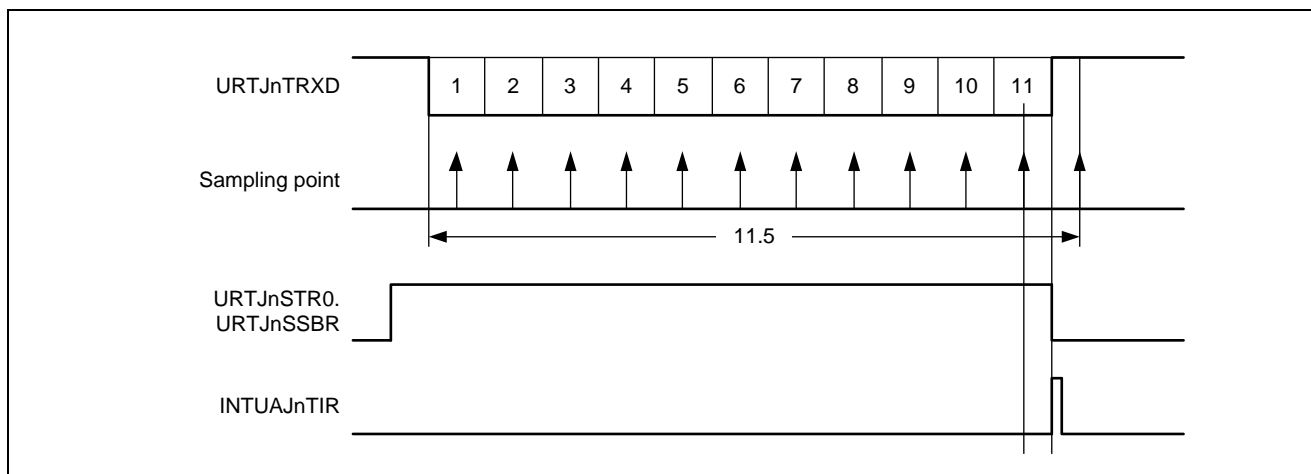


Figure 16.9 Normal BF Reception (Stop Bit after More Than 10.5 “L” Bits)

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.

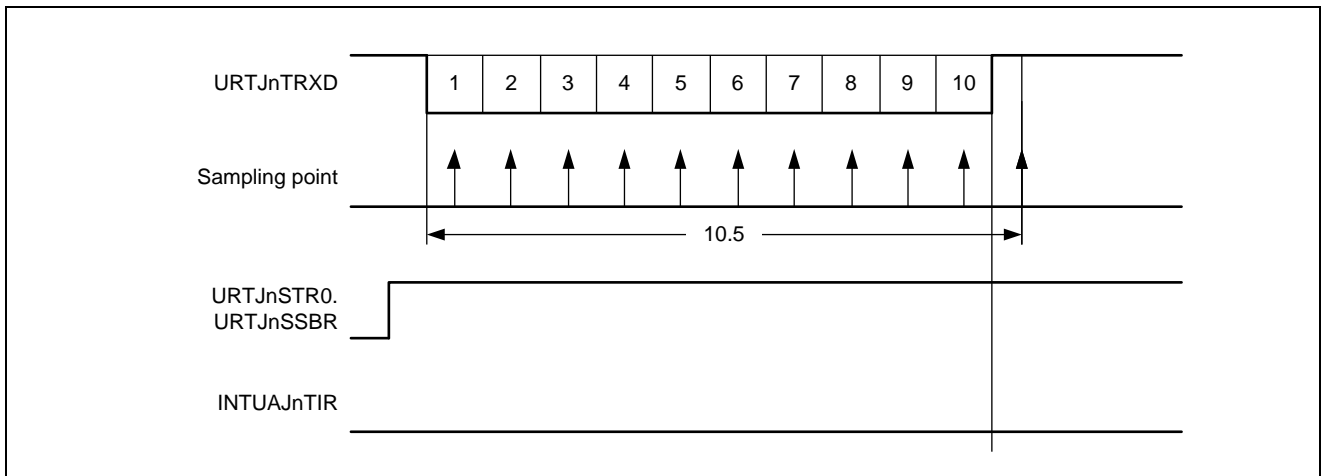


Figure 16.10 BF Reception Error (Stop Bit within 10.5 “L” Bits)

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.

16.6.5 UARTJn Transmission

(1) Transmission FIFO

The Tx FIFO comprises 8 bit x 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.

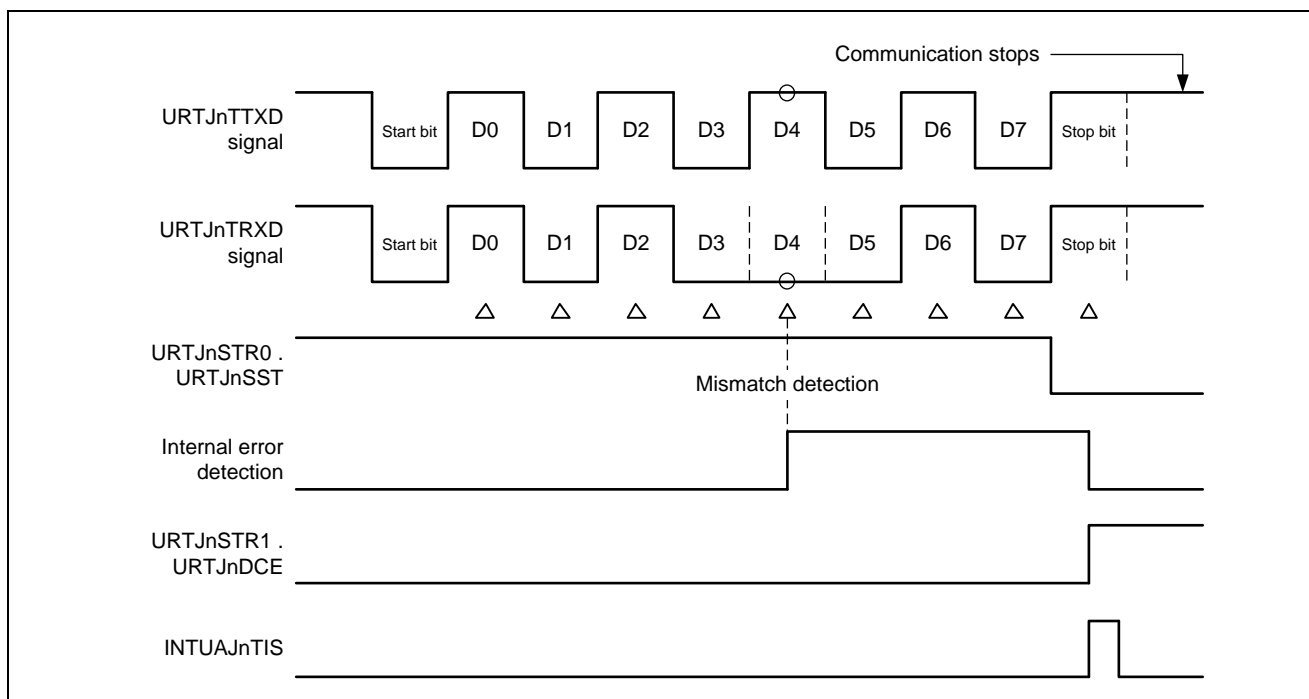


Figure 16.11 Timing Example of Data Consistency Error (No BF Reception Active, i.e. $URTJnSTR0.URTJnSSBR = 0$)

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].

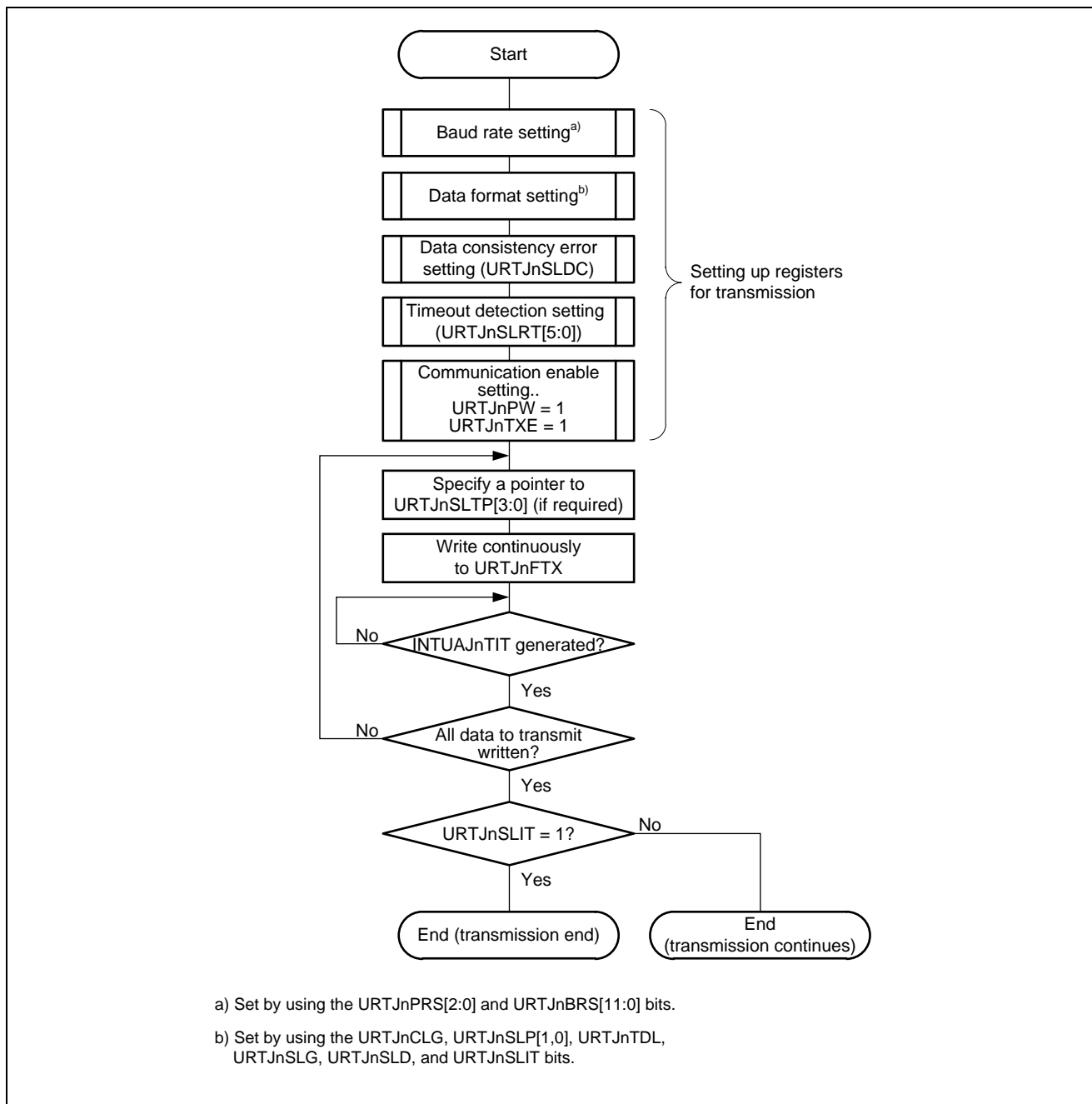


Figure 16.12 Flowchart of Data Transmission

16.6.6 UARTJn Reception

(1) Reception FIFO

The Rx FIFO comprises 10 bit x 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.

An overrun error also occurs if the URTJnFRX register is read while the Rx FIFO is empty.

(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.

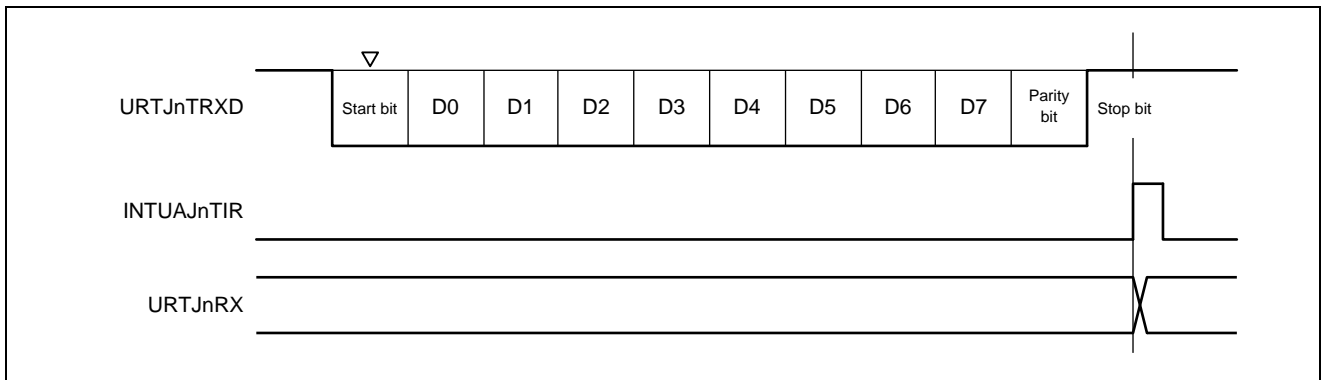


Figure 16.13 UARTJn Reception

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.

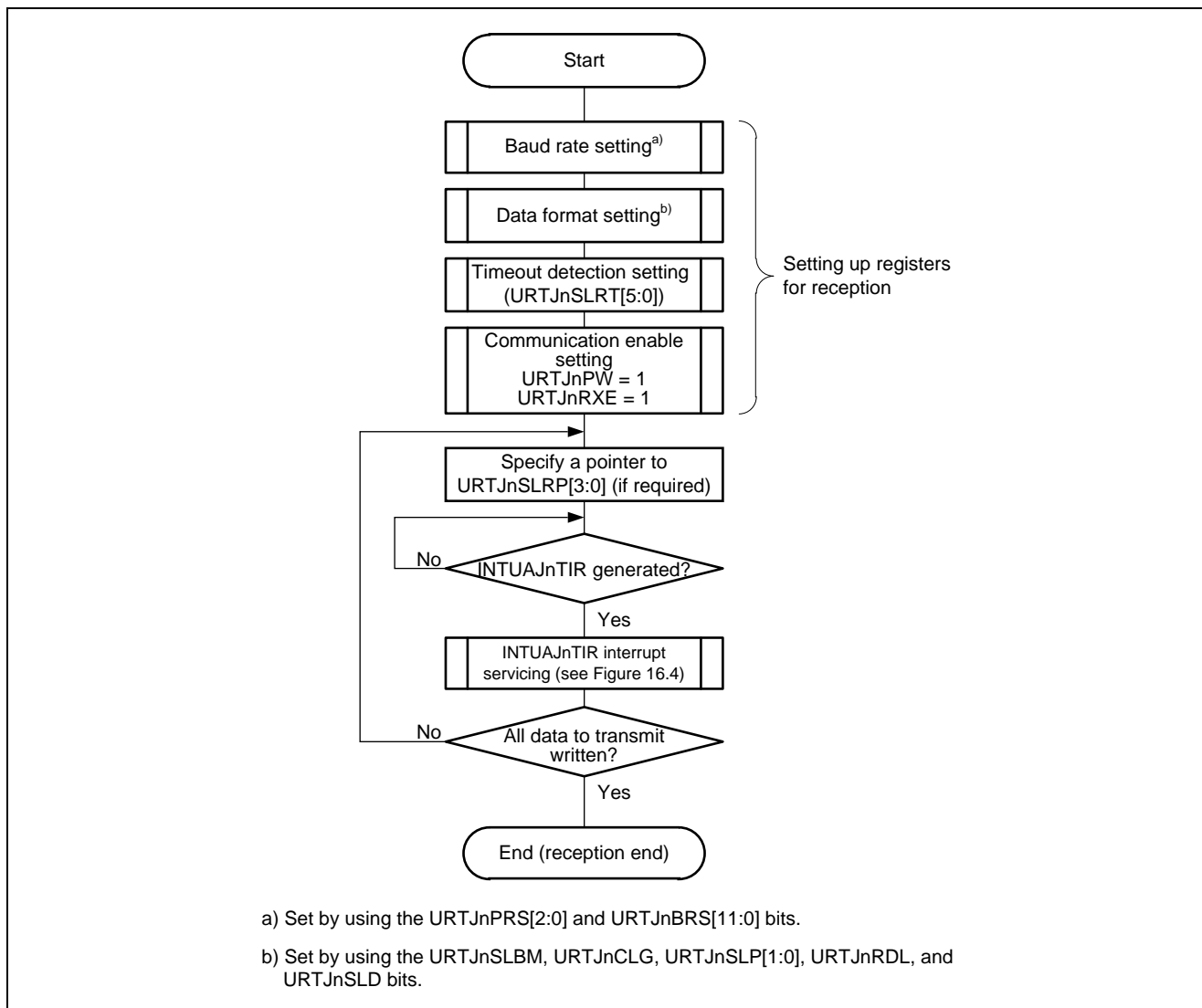


Figure 16.14 Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 0

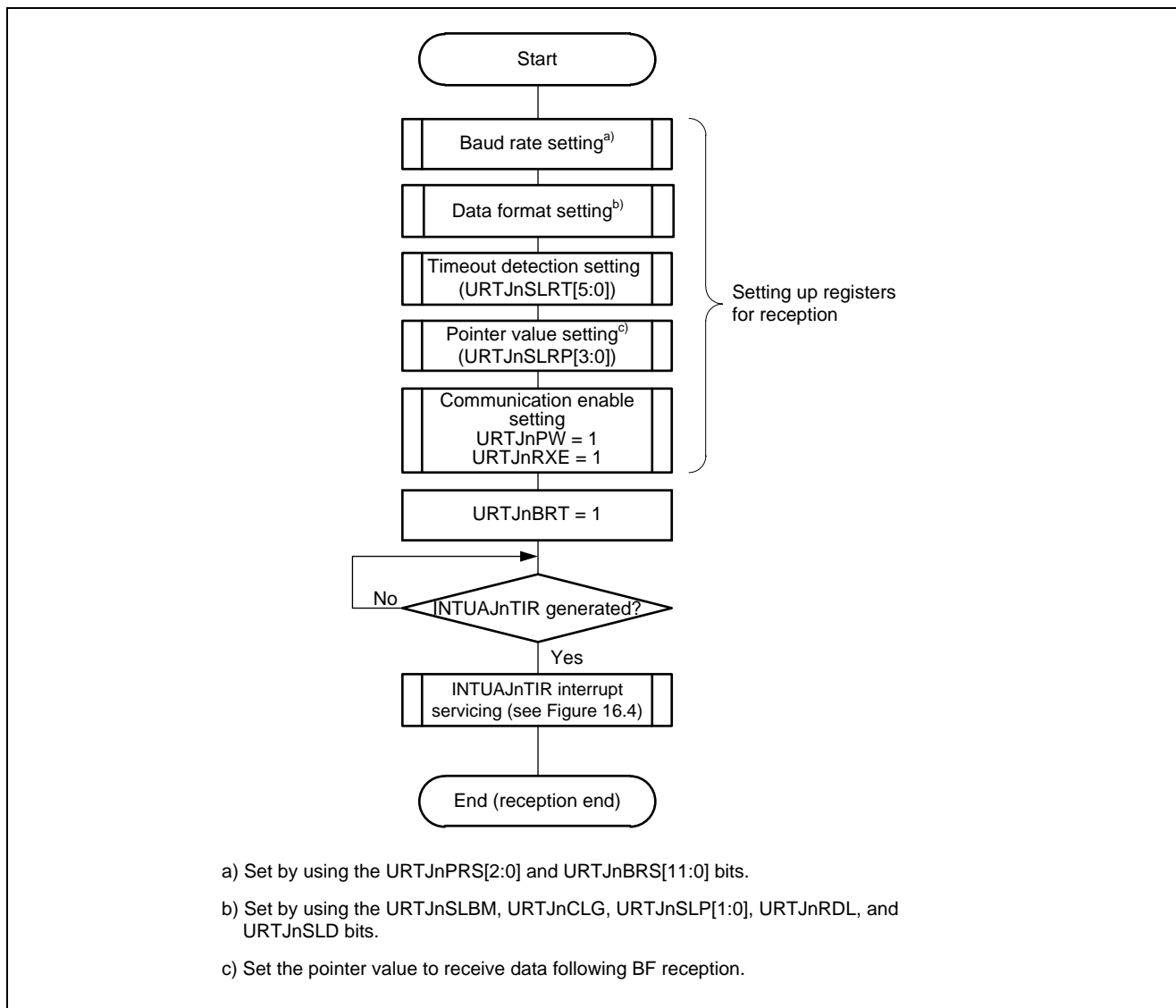


Figure 16.15 Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 1

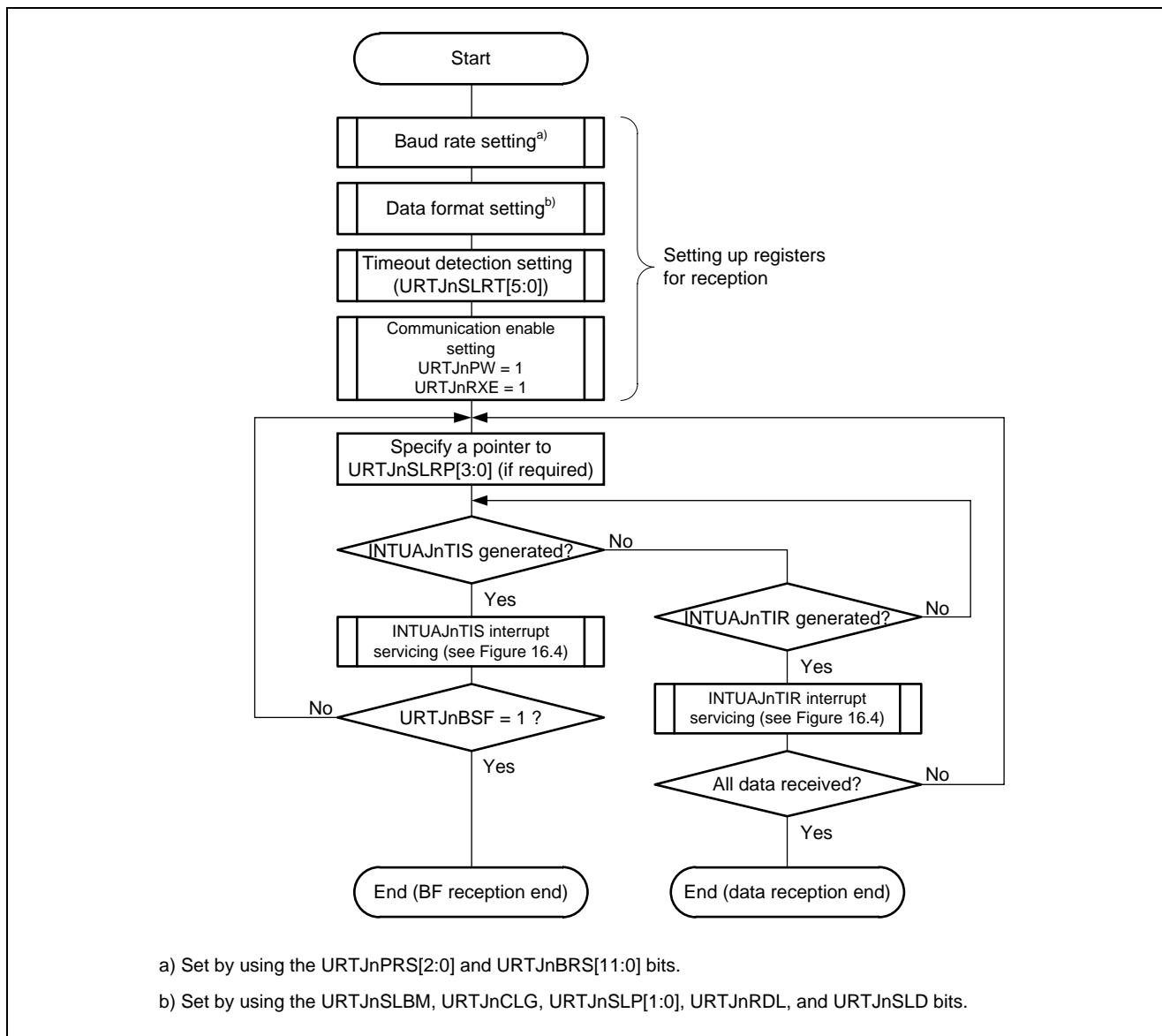


Figure 16.16 Flowchart of Data Reception when URTJnSLBM = 0, URTJnSSBR = 0

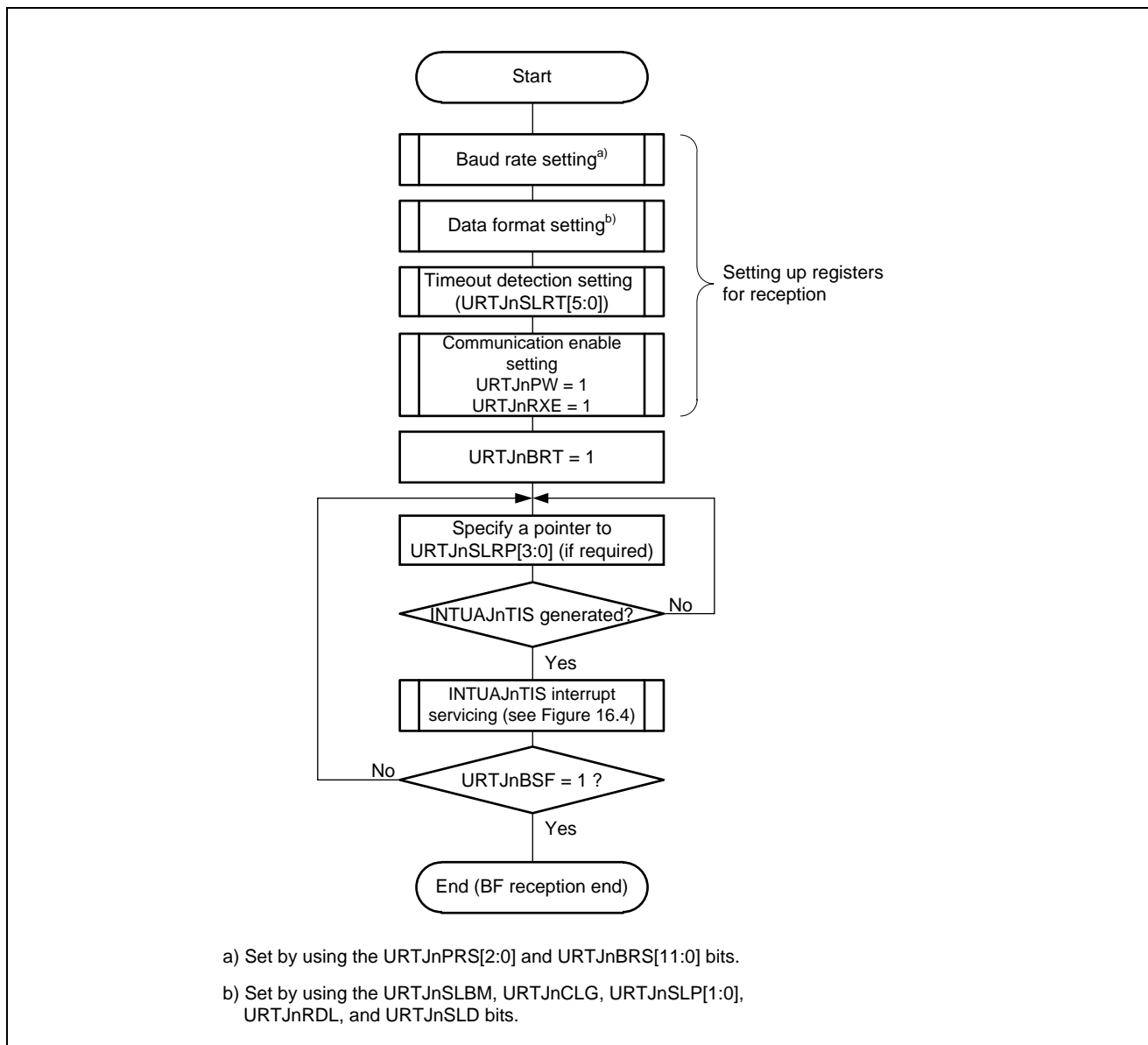


Figure 16.17 Flowchart of Data Reception when URTJnSLBM = 1, URTJnSSBR = 1

16.6.7 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 16.6 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.

16.6.8 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.

16.6.9 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 21.10, Noise Eliminator.

16.7 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.

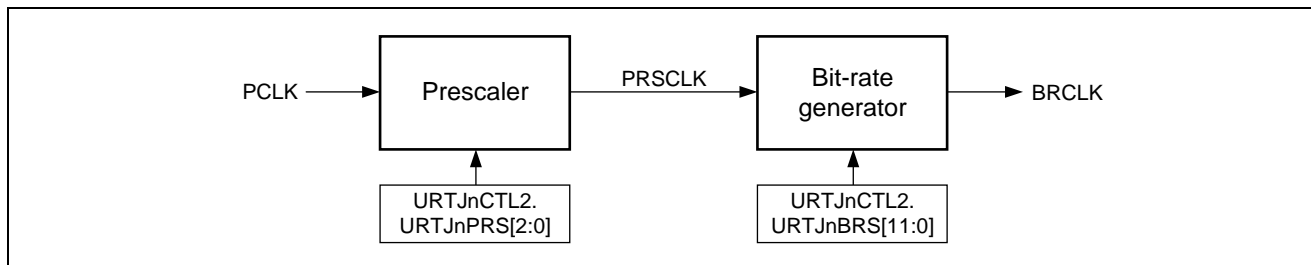


Figure 16.18 Configuration of Bit-Rate Generator

The prescaler output clock PRSCLK is a fraction of PCLK, the divisor is set up the value URTE_nCTL2.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 16.7 Bit-Rate Generator Clocks Output

URTJnCTL2.URTJnBRS[11:0]	Transmit/Receive BRCLK	BF Receive Clock
000H	PRSCLK/(2 x 4)	PRSCLK/4
001H		
002H		
003H		
004H		
005H	PRSCLK/(2 x 5)	PRSCLK/5
...	PRSCLK/(2 x URTJnBRS[11:0])	PRSCLK/URTJnBRS[11:0]
FFEH	PRSCLK/(2 x 4094)	PRSCLK/4094
FFFH	PRSCLK/(2 x 4095)	PRSCLK/4095

The clock setting for the bit rate is calculated from the following formula. For details of the register, see section 16.4(3), UARTJn control register 2 (URTJnCTL2).

$$\text{Bit rate} = \frac{\text{PCLK frequency}}{2 \times (\text{URTJnCTL2.URTJnBRS}11-0) \times 2^{\text{URTJnCTL2.URTJnPRS}2-0}} = \text{BRT}[\text{bps}]$$

$$\text{Error in the bit rate} = \left\{ \frac{\text{Bit rate (BRT)}}{\text{Target bit rate (target BRT)}} - 1 \right\} \times 100 = \text{ERR}[\%]$$

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-IN32M3 and the allowable scope of error in the bit rate.

Table 16.8 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 16.9 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

17. Clocked Serial Interface H (CSIH)

This section explains clocked serial interface H (CSIH).

17.1 Features of CSIH

R-IN32M3 products incorporate two channels of clocked serial interface H (CSIHn) listed below.

Table 17.1 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 17.2 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 17.3 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), and for updating the real-time port pins (RP00-RP37).

Table 17.4 CSIHn Interrupts and Requests to Peripheral Modules

CSIHn Signals	Function	Connected to
CSIH0		
CSIHTIC	Communication status interrupt	<ul style="list-style-type: none"> • Interrupt controller (INTCSIH0IC) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
CSIHTIR	Reception status interrupt	<ul style="list-style-type: none"> • Interrupt controller (INTCSIH0IR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
CSIHTIRE	Reception error interrupt	<ul style="list-style-type: none"> • Interrupt controller (INTCSIH0IRE) • HW-RTOS (Hardware ISR)
CSIHTIJC	Job completion interrupt	<ul style="list-style-type: none"> • Interrupt controller (INTCSIH0IJC) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
CSIH1		
CSIHTIC	Communication status interrupt	<ul style="list-style-type: none"> • Interrupt controller (INTCSIH1IC) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
CSIHTIR	Reception status interrupt	<ul style="list-style-type: none"> • Interrupt controller (INTCSIH1IR) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
CSIHTIRE	Reception error interrupt	<ul style="list-style-type: none"> • Interrupt controller (INTCSIH1IRE) • HW-RTOS (Hardware ISR)
CSIHTIJC	Job completion interrupt	<ul style="list-style-type: none"> • Interrupt controller (INTCSIH1IJC) • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)

- I/O signals: The I/O signals of CSIH are listed in the following table.

Table 17.5 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 70 CSICS00
CSIHTCSS0	Chip select signal 0	Port 71 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 72 CSICS10
CSIHTCSS0	Chip select signal 0	Port 73 CSICS11

17.2 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.

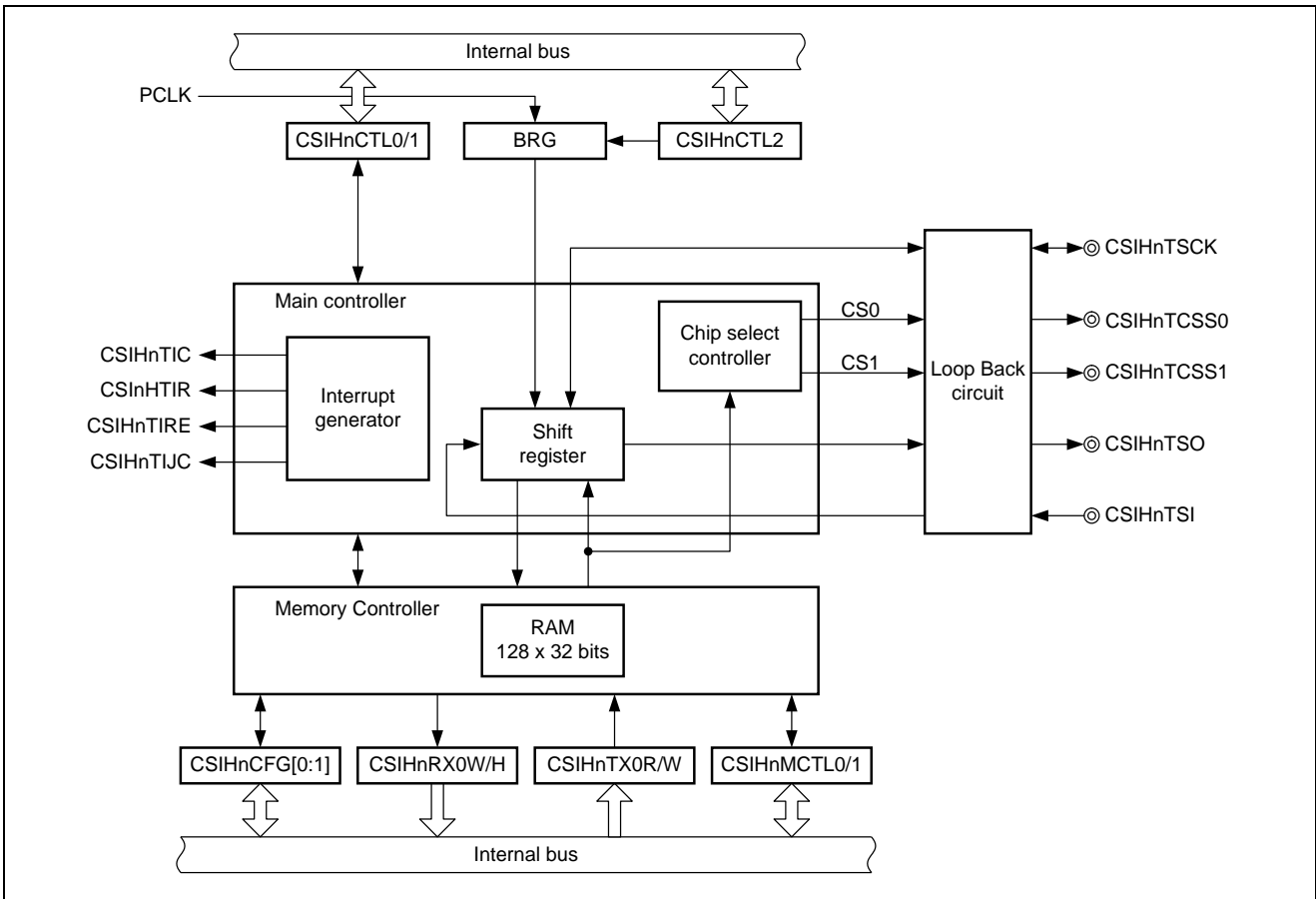


Figure 17.1 CSIH Block Diagram

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 17.4.1, Operating Modes (Master/Slave)).
- “Job mode” is related to the Autosar job concept (for details, see section 17.4.5, Job Concept).
- In the “memory mode”, various settings for the associated buffer memory are available (for details, see section 17.4.7, CSIH Buffer Memory).
- “Data transfer mode” specifies the mode of communications – transmit only, receive only, or both (for details, see section 17.4.8, Data Transfer Modes).

17.3 CSIH Control Registers

CSIHn is controlled and operated by means of the following registers:

Table 17.6 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 17.7 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

17.3.1 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 or 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	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: center;">CSIHnPWR</td> <td style="width: 10%; text-align: center;">CSIHnTXE</td> <td style="width: 10%; text-align: center;">CSIHnRXE</td> <td style="width: 10%; text-align: center;">0</td> <td style="width: 10%; text-align: center;">0</td> <td style="width: 10%; text-align: center;">0</td> <td style="width: 10%; text-align: center;">CSIHnJOBE</td> <td style="width: 10%; text-align: center;">CSIHnMBS</td> </tr> </table>	CSIHnPWR	CSIHnTXE	CSIHnRXE	0	0	0	CSIHnJOBE	CSIHnMBS	4000 0100H +100H × n Initial Value 0000 0000H
CSIHnPWR	CSIHnTXE	CSIHnRXE	0	0	0	CSIHnJOBE	CSIHnMBS				
R/W	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, CSIHnJOB, 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 or written in 32-bit units.

Caution: Changing the contents of this register is only permitted when CSIHnCTL0.CSIHnPWR = 0.

(1/2)

CSIHnCTL1	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	4000 0110H +100H × n
	0 0	Initial Value	0000 0000H
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 17.4.12(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 17.4.3, 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 17.4.9(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 17.4.5, Job Concept. The CSIHnCTL0.CSIHnJOB, 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 17.4.13(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 17.4.14, 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 17.4.12(5), Delay for all interrupts.
1, 0	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.

Remark: n = 0, 1

(4) CSIH status register 0 (CSIHnSTRO)

This register indicates the status of the CSIH.

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

(1/4)

CSIHnSTRO	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 4000 0104H +100H × n							
	CSIHnSRP[7:0]		Initial Value 0000 0010H							
R/W	R	R	R R 0 0 0 0 0 0 R 0 R R R 0 R R							
Bit Position	Bit Name	Function								
31 to 24	CSIHnSRP [7:0]	Indicates the number of received words in FIFO mode.								
		<table border="1"> <thead> <tr> <th>CSIHnSRP[7:0]</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00H</td> <td rowspan="3">Number of received words (0 to 128)</td> </tr> <tr> <td>:</td> </tr> <tr> <td>80H</td> </tr> <tr> <td>Other than the above</td> <td>Setting prohibited</td> </tr> </tbody> </table> <p>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.</p>			CSIHnSRP[7:0]	Description	00H	Number of received words (0 to 128)	:	80H
CSIHnSRP[7:0]	Description									
00H	Number of received words (0 to 128)									
:										
80H										
Other than the above	Setting prohibited									
23 to 16	CSIHnSPF [7:0]	Indicates the number of unsent data in FIFO mode.								
		<table border="1"> <thead> <tr> <th>CSIHnSPF[7:0]</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00H</td> <td rowspan="3">Number of unsent data (0-128)</td> </tr> <tr> <td>:</td> </tr> <tr> <td>80H</td> </tr> <tr> <td>Other than the above</td> <td>Setting prohibited</td> </tr> </tbody> </table> <p>These bits are cleared by CSIHnSTCR0.CSIHnPCT. In the 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.</p>			CSIHnSPF[7:0]	Description	00H	Number of unsent data (0-128)	:	80H
CSIHnSPF[7:0]	Description									
00H	Number of unsent data (0-128)									
:										
80H										
Other than the above	Setting prohibited									
<p>Remark: n = 0, 1</p>										

(2/4)

Bit Position	Bit Name	Function
15	CSIHnTMOE	<p>Timeout error flag in FIFO mode</p> <p>Indicates whether a timeout error was detected in FIFO mode.</p> <p>0: No timeout error was detected in FIFO mode.</p> <p>1: A timeout error was detected in in FIFO mode.</p> <p>For details, see 17.4.13(3), Timeout error.</p> <p>This bit is cleared by CSIHnSTCR0.CSIHnTMOEC.</p> <p>This bit can be written to when CSIHnSTCR0.CSIHnPWR = 0.</p> <p>This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.</p> <p>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.</p>
14	CSIHnOFE	<p>Overflow error flag in FIFO mode</p> <p>Indicates whether an overflow error was detected in FIFO mode.</p> <p>0: No overflow error was detected in FIFO mode.</p> <p>1: An overflow error was detected in FIFO mode.</p> <p>For details, see 17.4.13(4), Overflow error.</p> <p>This bit is cleared by CSIHnSTCR0.CSIHnOFEC.</p> <p>This bit can be written to when CSIHnSTCR0.CSIHnPWR = 0.</p> <p>This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.</p> <p>If 129 transmission data items are written to the CSIHnTX0W or CSIHnTX0H register when CSIHnCTL0.CSIHnPWR = 0, an overflow error occurs.</p> <p>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.</p>

Remark: n = 0, 1

(3/4)

Bit Position	Bit Name	Function																										
7	CSIHnTSF	<p>Transfer status flag</p> <p>0: Idle state 1: Transmission is in progress or being prepared</p> <p>Setting and clearing conditions of this bit are as follows:</p> <table border="1"> <thead> <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> </thead> <tbody> <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> </tbody> </table> <table border="1"> <thead> <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> </thead> <tbody> <tr> <td>Transmission mode</td> <td rowspan="2">Writing to transmit data register</td> <td rowspan="2">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> <td></td> </tr> </tbody> </table>	Master Mode	Setting Condition		Clearing Condition	Direct Access Mode, FIFO Mode	Dual Buffer Mode, Transmit Only Buffer Mode	Transmission mode	Writing to transmit data register	Setting CSIHnMCTL2. CSIHnBTST	Within 0.5 clock cycles from the last CSIHnTSCK edge	Transmission / reception mode	Reception mode	Slave Mode	Setting Condition		Clearing Condition	Direct Access Mode, FIFO Mode	Dual Buffer Mode, Transmit Only Buffer Mode	Transmission mode	Writing to transmit data register	Setting CSIHnMCTL2. CSIHnBTST	Within 0.5 clock cycles from the last CSIHnTSCK edge	Transmission / reception mode	Reception mode	CSIHnTSCK input timing	
Master Mode	Setting Condition			Clearing Condition																								
	Direct Access Mode, FIFO Mode	Dual Buffer Mode, Transmit Only Buffer Mode																										
Transmission mode	Writing to transmit data register	Setting CSIHnMCTL2. CSIHnBTST	Within 0.5 clock cycles from the last CSIHnTSCK edge																									
Transmission / reception mode																												
Reception mode																												
Slave Mode	Setting Condition		Clearing Condition																									
	Direct Access Mode, FIFO Mode	Dual Buffer Mode, Transmit Only Buffer Mode																										
Transmission mode	Writing to transmit data register	Setting CSIHnMCTL2. CSIHnBTST	Within 0.5 clock cycles from the last CSIHnTSCK edge																									
Transmission / reception mode																												
Reception mode	CSIHnTSCK input timing																											
5	CSIHnFLF	<p>Indicates whether the buffer is full while in FIFO mode.</p> <p>0: FIFO buffer is not full 1: FIFO buffer is full</p> <p>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.</p> <p>This bit is cleared by CSIHnSTCR0.CSIHnPCT.</p> <p>The FIFO buffer may be filled up with data that has not been transmitted and reception data.</p>																										

Remark: n = 0, 1

(4/4)

Bit Position	Bit Name	Function
4	CSIHnEMF	<p>Indicates whether the buffer is empty while in FIFO mode.</p> <p>0: FIFO buffer is not empty 1: FIFO buffer is empty</p> <p>This bit is set by CSIHnSTCR0.CSIFnPCT.</p> <p>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.</p> <p>The FIFO buffer may not contain any data that has not been transmitted or reception data.</p>
3	CSIHnDCE	<p>Data consistency error flag</p> <p>0: No data consistency error detected 1: Data consistency error detected</p> <p>This bit is cleared by setting CSIHnSTCR0.CSIHnDCEC.</p> <p>This bit can be written to when CSIHnCTL0.CSIHnPWR = 0.</p> <p>This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.</p> <p>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.</p>
1	CSIHnPE	<p>Parity error flag</p> <p>0: No parity error detected 1: Parity error detected</p> <p>This bit is cleared by setting CSIHnSTCR0.CSIHnPEC.</p> <p>This bit can be written to when CSIHnCTL0.CSIHnPWR = 0.</p> <p>This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.</p> <p>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.</p>
0	CSIHnOVE	<p>Overflow error flag (fixed to 0 in dual buffer mode)</p> <p>0: No overflow error detected 1: Overflow error detected</p> <p>This bit is cleared by setting CSIHnSTCR0.CSIHnOVEC.</p> <p>This bit can be written to when CSIHnCTL0.CSIHnPWR = 0.</p> <p>This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.</p> <p>This bit is fixed to 0 in dual buffer mode.</p> <p>If this bit is set due to an overflow error being detected and cleared by CSIHnSTCR0.CSIHnOVEC at the same time, setting the bit is prioritized.</p>
13 to 8, 6, 2	—	Reserved. These bits are read as 0.

Remark: n = 0, 1

Table 17.8 Memory Mode Operation

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

(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 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 4000 01C0H +100H × n
CSIHn MCTL0	0 0	CSIHnMMS[1:0] 0 0 0	Initial Value 0000 001FH
R/W	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. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 15%;">CSIHnMMS1</th> <th style="width: 15%;">CSIHnMMS0</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>FIFO mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>Dual buffer mode</td> </tr> <tr> <td>1</td> <td>0</td> <td>Transmit-only buffer mode</td> </tr> <tr> <td>1</td> <td>1</td> <td>Prohibited</td> </tr> </tbody> </table> <p style="font-size: small; margin-top: 5px;">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.</p>	CSIHnMMS1	CSIHnMMS0	Description	0	0	FIFO mode	0	1	Dual buffer mode	1	0	Transmit-only buffer mode	1	1	Prohibited
CSIHnMMS1	CSIHnMMS0	Description															
0	0	FIFO mode															
0	1	Dual buffer mode															
1	0	Transmit-only buffer mode															
1	1	Prohibited															
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. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">CSIHnTo[4:0]</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>00000B</td> <td>No timeout detection</td> </tr> <tr> <td>00001B</td> <td>Timeout is (1 × 8 × BRG output clock cycles)</td> </tr> <tr> <td>00010B</td> <td>Timeout is (2 × 8 × BRG output clock cycles)</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111B</td> <td>Timeout is (31 × 8 × BRG output clock cycles)</td> </tr> </tbody> </table> <p style="font-size: small; margin-top: 5px;">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 17.4.13(3), Timeout error.</p>	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)			
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)																

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 or written in 32-bit units.

CSIHnMCTL1	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
	<div style="display: flex; justify-content: space-between;"> <div style="text-align: center;">0 0 0 0 0 0 0 0 0 0</div> <div style="text-align: center;">CSIHnFES[6:0]</div> <div style="text-align: center;">0 0 0 0 0 0 0 0</div> <div style="text-align: center;">CSIHnFFS[6:0]</div> </div>	4000 0180H +100H × n Initial Value 0000 0000H
R/W	0 0 0 0 0 0 0 0 0 0 R/W 0 0 0 0 0 0 0 0 R/W	
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 or 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	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
CSIHnMCTL2	CSIHnBTST								CSIHnND[7:0]										CSIHnSOP[6:0]								4000 0184H +100H × n						
R/W	RW	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	0	0	0	0	0	0	0	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]	<p>Specify the number of data items. When read, these bits indicate the number of remaining communication data items.</p> <table border="1"> <thead> <tr> <th>CSIHnND [7:0]</th> <th>Dual Buffer Mode</th> <th>Transmit-Only Buffer Mode</th> <th>FIFO Mode</th> <th>Direct Access Mode</th> </tr> </thead> <tbody> <tr> <td>00H</td> <td>Transmit 0 data items.</td> <td>Transmit 0 data items.</td> <td>No effect</td> <td>No effect</td> </tr> <tr> <td>01H</td> <td>Transmit 1 data items.</td> <td>Transmit 1 data items.</td> <td>No effect</td> <td>No effect</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>No effect</td> <td>No effect</td> </tr> <tr> <td>3FH</td> <td>Transmit 63 data items.</td> <td>Transmit 63 data items.</td> <td>No effect</td> <td>No effect</td> </tr> <tr> <td>40H</td> <td>Transmit 64 data items.</td> <td>Transmit 64 data items.</td> <td>No effect</td> <td>No effect</td> </tr> <tr> <td>...</td> <td>Prohibited</td> <td>...</td> <td>No effect</td> <td>No effect</td> </tr> <tr> <td>7FH</td> <td>Prohibited</td> <td>Transmit 127 data items.</td> <td>No effect</td> <td>No effect</td> </tr> <tr> <td>80H</td> <td>Prohibited</td> <td>Transmit 128 data items.</td> <td>No effect</td> <td>No effect</td> </tr> <tr> <td>other</td> <td colspan="4">Setting prohibited</td> </tr> </tbody> </table> <p>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.</p>	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			
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																																																			
15 to 7	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.																																																		
6 to 0	CSIHnSOP [6:0]	<p>Select the transmission data pointer.</p> <table border="1"> <thead> <tr> <th>CSIHnSOP [6:0]</th> <th>Dual Buffer Mode</th> <th>Transmit-Only Buffer Mode</th> <th>FIFO Mode</th> <th>Direct Access Mode</th> </tr> </thead> <tbody> <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>...</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> </tbody> </table> <p>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.</p>	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										
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 or 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)

CSIHnMRWP0

R/W: 0 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]	<p>Selects the read pointer of the Rx buffer.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 5px;"> <thead> <tr> <th>CSIHnRRA [6:0]</th> <th>Dual Buffer Mode</th> <th>Transmit-Only Buffer Mode</th> <th>FIFO Mode</th> <th>Direct Access Mode</th> </tr> </thead> <tbody> <tr> <td>00H</td> <td>0000H</td> <td>No effect</td> <td>0000H</td> <td>No effect</td> </tr> <tr> <td>01H</td> <td>0004H</td> <td>No effect</td> <td>0004H</td> <td>No effect</td> </tr> <tr> <td>...</td> <td>...</td> <td>No effect</td> <td>...</td> <td>No effect</td> </tr> <tr> <td>3FH</td> <td>00FCH</td> <td>No effect</td> <td>00FCH</td> <td>No effect</td> </tr> <tr> <td>40H</td> <td>Prohibited</td> <td>No effect</td> <td>0100H</td> <td>No effect</td> </tr> <tr> <td>...</td> <td>Prohibited</td> <td>No effect</td> <td>...</td> <td>No effect</td> </tr> <tr> <td>7FH</td> <td>Prohibited</td> <td>No effect</td> <td>01FCH</td> <td>No effect</td> </tr> </tbody> </table> <p>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.</p>	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
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																																						

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function																																								
15 to 7	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.																																								
6 to 0	CSIHnTRWA [6:0]	<p>Selects the read/write pointer of the Tx buffer.</p> <table border="1"> <thead> <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> </thead> <tbody> <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> </tbody> </table> <p>When transmission data is read or written from the CPU, these bits are automatically incremented.</p> <p>These bits are cleared when CSIHnSTCR0.CSIHnPCT is set.</p> <p>In direct access mode, these bits are not incremented.</p> <p>In FIFO mode, these bits indicate the read/write address of the transmission data.</p>	CSIHnTRWA [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	No effect	3FH	00FCH	00FCH	00FCH	No effect	40H	Prohibited	0100H	0100H	No effect	...	Prohibited	No effect	7FH	Prohibited	01FCH	01FCH	No effect
CSIHnTRWA [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																																						
...	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

(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)

	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	
CSIHnCFGx	CSIHnPSCLx[1:0]		CSIHnPSx[1:0]		CSIHnDLSx[3:0]			0	0	0	0	CSIHnRCBx	CSIHnDIRx	CSIHnCKPx	CSIHnDAPx	CSIHnDLx	CSIHnIDx[2:0]	CSIHnHDx[3:0]	CSIHnINx[3:0]	CSIHnSPx[3:0]							refer to above							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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, 30	CSIHnPSCLx [1:0]	Selects the prescaler for chip select x. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">CSIHnPSCLx1</th> <th style="width: 15%;">CSIHnPSCLx0</th> <th style="width: 70%;">Prescaler Output</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>CSIHnBCLK</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>CSIHnBCLK / 2</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>CSIHnBCLK / 4</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>CSIHnBCLK / 8</td> </tr> </tbody> </table> <p style="margin-top: 10px; font-size: small;">These bits are only available in master mode. For details about CSIHnBCLK, see section 17.4.6, Serial Clock Selection.</p>	CSIHnPSCLx1	CSIHnPSCLx0	Prescaler Output	0	0	CSIHnBCLK	0	1	CSIHnBCLK / 2	1	0	CSIHnBCLK / 4	1	1	CSIHnBCLK / 8
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]	<p>Selects the parity for chip select x for transmission and reception.</p> <table border="1"> <thead> <tr> <th>CSIHnPSx1</th> <th>CSIHnPSx0</th> <th>Transmission</th> <th>Reception</th> </tr> </thead> <tbody> <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> </tbody> </table>	CSIHnPSx1	CSIHnPSx0	Transmission	Reception	0	0	No parity transmitted	Parity reception is not expected.	0	1	Add parity bit fixed at 0	Parity bit reception is expected, but parity judgment is not performed.	1	0	Add odd parity	Odd parity bit reception is expected.	1	1	Add odd parity	Even parity bit reception is expected.
CSIHnPSx1	CSIHnPSx0	Transmission	Reception																			
0	0	No parity transmitted	Parity reception is not expected.																			
0	1	Add parity bit fixed at 0	Parity bit reception is expected, but parity judgment is not performed.																			
1	0	Add odd parity	Odd parity bit reception is expected.																			
1	1	Add odd parity	Even parity bit reception is expected.																			
27 to 24	CSIHnDLSx [3:0]	<p>Selects the data length for chip select x.</p> <table border="1"> <thead> <tr> <th>CSIHnDLSx[3:0]</th> <th>Data Length</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Note: For details about the CSIHnDLSx[3:0] bit setting, see section 17.4.9, 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.</p>	CSIHnDLSx[3:0]	Data Length	0000B	16 bits	0001B	1 bit	0010B	2 bits	1111B	15 bits								
CSIHnDLSx[3:0]	Data Length																					
0000B	16 bits																					
0001B	1 bit																					
0010B	2 bits																					
...	...																					
1111B	15 bits																					
23 to 20	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.																				
19	CSIHnRCBx	<p>Selects the recessive configuration for broadcasting for chip select x.</p> <p>0: Dominant (higher priority) 1: Recessive (lower priority)</p> <p>For details, see 17.4.3(1), Configuration registers.</p>																				
18	CSIHnDIRx	<p>Selects the serial data direction for chip select x.</p> <p>0: Data is sent/received with MSB first 1: Data is sent/received with LSB first</p> <p>For details, see section 17.4.10, Serial Data Direction Selection.</p>																				

Note: n = 0, 1; x = 0, 1

Bit Position	Bit Name	Function																											
17, 16	CSIHnCKPx, CSIHnDAPx	<p>CSIHnCKPx: Clock phase select bit CSIHnDAPx: Data phase select bit</p> <p>CSIHnCTL1.CSIHnCKR = 0</p> <table border="1"> <thead> <tr> <th>CSIHnCKPx</th> <th>CSIHnDAPx</th> <th>Specifying the Timing of Transmission or Reception for CSIHnTSCk</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>CSIHnCTL1.CSIHnCKR = 1</p> <table border="1"> <thead> <tr> <th>CSIHnCKPx</th> <th>CSIHnDAPx</th> <th>Specifying the Timing of Transmission or Reception for CSIHnTSCk</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>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.</p>	CSIHnCKPx	CSIHnDAPx	Specifying the Timing of Transmission or Reception for CSIHnTSCk	0	0		0	1		1	0		1	1		CSIHnCKPx	CSIHnDAPx	Specifying the Timing of Transmission or Reception for CSIHnTSCk	0	0		0	1		1	X	Setting prohibited
CSIHnCKPx	CSIHnDAPx	Specifying the Timing of Transmission or Reception for CSIHnTSCk																											
0	0																												
0	1																												
1	0																												
1	1																												
CSIHnCKPx	CSIHnDAPx	Specifying the Timing of Transmission or Reception for CSIHnTSCk																											
0	0																												
0	1																												
1	X	Setting prohibited																											

Remark: n = 0, 1; x = 0, 1

(4/5)

Bit Position	Bit Name	Function																																																			
15	CSIHnIDLx	<p>Selects the setting of the forced idle state for chip select x.</p> <p>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.</p> <p>1: An idle state is inserted after every transfer to chip select x.</p> <p>This bit is only available in master mode.</p> <p>If CSIHnCTL1.CSIHnJE = 1 and CSIHnTX0W.CSIHnEOJ = 1, chip select signal x definitely becomes idle even if CSIHnCFG0-1.CSIHnIDLn is cleared to 0.</p> <p>For details about the idle state, see section 17.4.3, Chip Selection (CS) Features.</p>																																																			
14 to 12	CSIHnIDx [2:0]	<p>Selects the idle time for chip select x.</p> <table border="1"> <thead> <tr> <th>CSIHnIDx[2:0]</th> <th>Idle Timing</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>These bits are only available in master mode.</p>	CSIHnIDx[2:0]	Idle Timing	000B	0.5 serial clock cycles	001B	1.0 serial clock cycles	010B	1.5 serial clock cycles	011B	2.5 serial clock cycles	100B	3.5 serial clock cycles	101B	4.5 serial clock cycles	110B	6.5 serial clock cycles	111B	8.5 serial clock cycles																																	
CSIHnIDx[2:0]	Idle Timing																																																				
000B	0.5 serial clock cycles																																																				
001B	1.0 serial clock cycles																																																				
010B	1.5 serial clock cycles																																																				
011B	2.5 serial clock cycles																																																				
100B	3.5 serial clock cycles																																																				
101B	4.5 serial clock cycles																																																				
110B	6.5 serial clock cycles																																																				
111B	8.5 serial clock cycles																																																				
11 to 8	CSIHnHDx [3:0]	<p>Selects the hold time for chip select x in transmission clock cycles.</p> <table border="1"> <thead> <tr> <th>CSIHnHDx [3:0]</th> <th>Hold Timing with CSIHnCTL1.CSIHnSIT = 0</th> <th>Hold Timing with CSIHnCTL1.CSIHnSIT = 1</th> </tr> </thead> <tbody> <tr><td>0000B</td><td>0.5 serial clock cycles</td><td>1.0 serial clock cycles</td></tr> <tr><td>0001B</td><td>1.0 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> </tbody> </table> <p>These bits are only available in master mode.</p>	CSIHnHDx [3:0]	Hold Timing with CSIHnCTL1.CSIHnSIT = 0	Hold Timing with CSIHnCTL1.CSIHnSIT = 1	0000B	0.5 serial clock cycles	1.0 serial clock cycles	0001B	1.0 serial clock cycles	1.5 serial clock cycles	0010B	1.5 serial clock cycles	2.0 serial clock cycles	0011B	2.5 serial clock cycles	3.0 serial clock cycles	0100B	3.5 serial clock cycles	4.0 serial clock cycles	0101B	4.5 serial clock cycles	5.0 serial clock cycles	0110B	6.5 serial clock cycles	7.0 serial clock cycles	0111B	8.5 serial clock cycles	9.0 serial clock cycles	1000B	9.5 serial clock cycles	10.0 serial clock cycles	1001B	10.5 serial clock cycles	11.0 serial clock cycles	1010B	11.5 serial clock cycles	12.0 serial clock cycles	1011B	12.5 serial clock cycles	13.0 serial clock cycles	1100B	14.5 serial clock cycles	15.0 serial clock cycles	1101B	16.5 serial clock cycles	17.0 serial clock cycles	1110B	18.5 serial clock cycles	19.0 serial clock cycles	1111B	20.5 serial clock cycles	21.0 serial clock cycles
CSIHnHDx [3:0]	Hold Timing with CSIHnCTL1.CSIHnSIT = 0	Hold Timing with CSIHnCTL1.CSIHnSIT = 1																																																			
0000B	0.5 serial clock cycles	1.0 serial clock cycles																																																			
0001B	1.0 serial clock cycles	1.5 serial clock cycles																																																			
0010B	1.5 serial clock cycles	2.0 serial clock cycles																																																			
0011B	2.5 serial clock cycles	3.0 serial clock cycles																																																			
0100B	3.5 serial clock cycles	4.0 serial clock cycles																																																			
0101B	4.5 serial clock cycles	5.0 serial clock cycles																																																			
0110B	6.5 serial clock cycles	7.0 serial clock cycles																																																			
0111B	8.5 serial clock cycles	9.0 serial clock cycles																																																			
1000B	9.5 serial clock cycles	10.0 serial clock cycles																																																			
1001B	10.5 serial clock cycles	11.0 serial clock cycles																																																			
1010B	11.5 serial clock cycles	12.0 serial clock cycles																																																			
1011B	12.5 serial clock cycles	13.0 serial clock cycles																																																			
1100B	14.5 serial clock cycles	15.0 serial clock cycles																																																			
1101B	16.5 serial clock cycles	17.0 serial clock cycles																																																			
1110B	18.5 serial clock cycles	19.0 serial clock cycles																																																			
1111B	20.5 serial clock cycles	21.0 serial clock cycles																																																			

Remark: n = 0, 1; x = 0, 1

(5/5)

Bit Position	Bit Name	Function																																																			
7 to 4	CSIHnINx [3:0]	<p>Selects the inter-data time for chip select x in transmission clock cycles.</p> <table border="1"> <thead> <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> </thead> <tbody> <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> </tbody> </table> <p>These bits are only available in master mode.</p>	CSIHnINx [3:0]	Inter-Data Time when CSIHnCTL1.CSIHnSIT = 0	Inter-Data Time when CSIHnCTL1.CSIHnSIT = 1	0000B	0.0 serial clock cycles	0.5 serial clock cycles	0001B	0.5 serial clock cycles	1.0 serial clock cycles	0010B	1.0 serial clock cycles	1.5 serial clock cycles	0011B	2.0 serial clock cycles	2.5 serial clock cycles	0100B	3.0 serial clock cycles	3.5 serial clock cycles	0101B	4.0 serial clock cycles	4.5 serial clock cycles	0110B	6.0 serial clock cycles	6.5 serial clock cycles	0111B	8.0 serial clock cycles	8.5 serial clock cycles	1000B	9.0 serial clock cycles	9.5 serial clock cycles	1001B	10.0 serial clock cycles	10.5 serial clock cycles	1010B	11.0 serial clock cycles	11.5 serial clock cycles	1011B	12.0 serial clock cycles	12.5 serial clock cycles	1100B	14.0 serial clock cycles	14.5 serial clock cycles	1101B	16.0 serial clock cycles	16.5 serial clock cycles	1110B	18.0 serial clock cycles	18.5 serial clock cycles	1111B	20.0 serial clock cycles	20.5 serial clock cycles
CSIHnINx [3:0]	Inter-Data Time when CSIHnCTL1.CSIHnSIT = 0	Inter-Data Time when CSIHnCTL1.CSIHnSIT = 1																																																			
0000B	0.0 serial clock cycles	0.5 serial clock cycles																																																			
0001B	0.5 serial clock cycles	1.0 serial clock cycles																																																			
0010B	1.0 serial clock cycles	1.5 serial clock cycles																																																			
0011B	2.0 serial clock cycles	2.5 serial clock cycles																																																			
0100B	3.0 serial clock cycles	3.5 serial clock cycles																																																			
0101B	4.0 serial clock cycles	4.5 serial clock cycles																																																			
0110B	6.0 serial clock cycles	6.5 serial clock cycles																																																			
0111B	8.0 serial clock cycles	8.5 serial clock cycles																																																			
1000B	9.0 serial clock cycles	9.5 serial clock cycles																																																			
1001B	10.0 serial clock cycles	10.5 serial clock cycles																																																			
1010B	11.0 serial clock cycles	11.5 serial clock cycles																																																			
1011B	12.0 serial clock cycles	12.5 serial clock cycles																																																			
1100B	14.0 serial clock cycles	14.5 serial clock cycles																																																			
1101B	16.0 serial clock cycles	16.5 serial clock cycles																																																			
1110B	18.0 serial clock cycles	18.5 serial clock cycles																																																			
1111B	20.0 serial clock cycles	20.5 serial clock cycles																																																			
3 to 0	CSIHnSPx [3:0]	<p>Selects the setup time for chip select x in transmission clock cycles.</p> <table border="1"> <thead> <tr> <th>CSIHnSPx[3:0]</th> <th>Setup Delay</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>These bits are only available in master mode.</p>	CSIHnSPx[3:0]	Setup Delay	0000B	0.5 serial clock cycles	0001B	1.0 serial clock cycles	0010B	1.5 serial clock cycles	0011B	2.5 serial clock cycles	0100B	3.5 serial clock cycles	0101B	4.5 serial clock cycles	0110B	6.5 serial clock cycles	0111B	8.5 serial clock cycles	1000B	9.5 serial clock cycles	1001B	10.5 serial clock cycles	1010B	11.5 serial clock cycles	1011B	12.5 serial clock cycles	1100B	14.5 serial clock cycles	1101B	16.5 serial clock cycles	1110B	18.5 serial clock cycles	1111B	20.5 serial clock cycles																	
CSIHnSPx[3:0]	Setup Delay																																																				
0000B	0.5 serial clock cycles																																																				
0001B	1.0 serial clock cycles																																																				
0010B	1.5 serial clock cycles																																																				
0011B	2.5 serial clock cycles																																																				
0100B	3.5 serial clock cycles																																																				
0101B	4.5 serial clock cycles																																																				
0110B	6.5 serial clock cycles																																																				
0111B	8.5 serial clock cycles																																																				
1000B	9.5 serial clock cycles																																																				
1001B	10.5 serial clock cycles																																																				
1010B	11.5 serial clock cycles																																																				
1011B	12.5 serial clock cycles																																																				
1100B	14.5 serial clock cycles																																																				
1101B	16.5 serial clock cycles																																																				
1110B	18.5 serial clock cycles																																																				
1111B	20.5 serial clock cycles																																																				

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 or 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)

	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 4000 0188H +100H × n Initial Value Undefined																										
CSIHnTX0W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">CSIHnCIRE</td> <td style="width: 20px; text-align: center;">CSIHnEOJ</td> <td style="width: 20px; text-align: center;">CSIHnEDL</td> <td style="width: 20px; text-align: center;">0</td> <td style="width: 20px; text-align: center;">0</td> <td style="width: 20px; text-align: center;">0</td> <td style="width: 20px; text-align: center;">0</td> <td style="width: 20px; text-align: center;">0</td> <td style="width: 20px; text-align: center;">0</td> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">CSIHnCS[1:0]</td> <td colspan="10" style="text-align: center; vertical-align: middle;">CSIHnTX[15:0]</td> </tr> </table>	CSIHnCIRE	CSIHnEOJ	CSIHnEDL	0	0	0	0	0	0	1	1	1	1	1	1	CSIHnCS[1:0]	CSIHnTX[15:0]											
CSIHnCIRE	CSIHnEOJ	CSIHnEDL	0	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																										

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 17.4.12(1), CSIHnTIC (communication interrupt), and 17.4.12(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	<p>Specifies whether the associated data requires the extended data length (EDL) option.</p> <p>0: Normal operation 1: Extended data length activated</p> <p>The associated data is transmitted as of 16 bits. The inter-data delay time and idle time are not inserted after data transmission.</p> <p>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.</p> <p>Caution: This bit can only be used when CSIHnCTL1.CSIHnEDLE = 1.</p>
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]	<p>Activates one or several chip select signals.</p> <p>0: Chip select x is activated for the associated transmission 1: Chip select x is deactivated for the associated transmission</p> <p>Setting CSIHnTX0W.CSIHnCS[1:0] to 3H is prohibited.</p> <p>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.</p> <p>For use in slave mode, set the CSIHnCS[1:0] bits to 2H.</p>
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

Remark: n = 0, 1; x = 0, 1

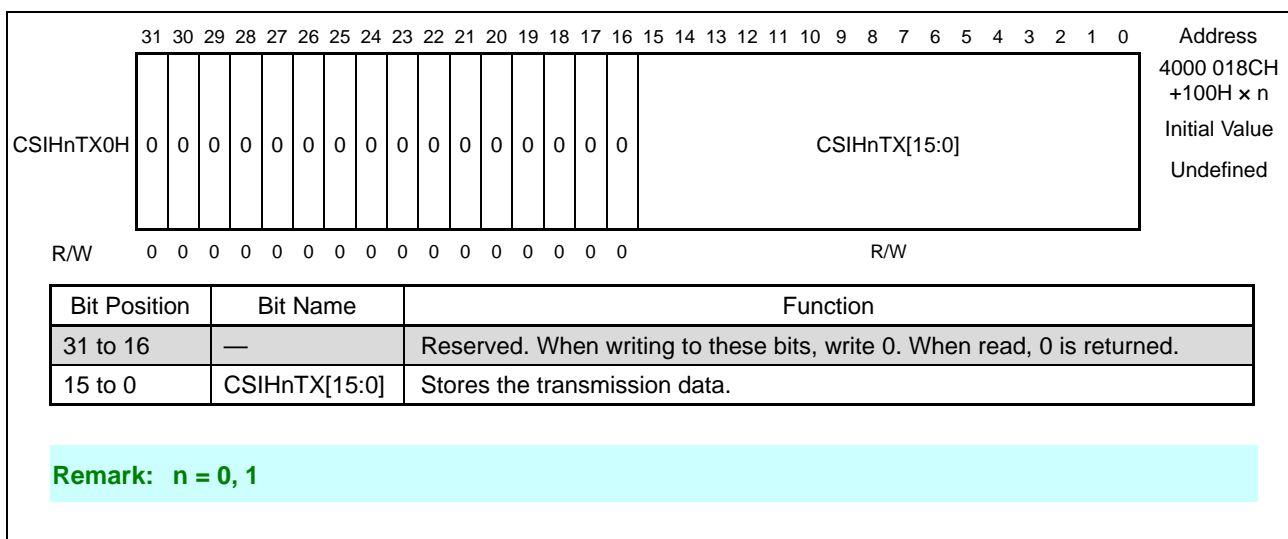
(12) CSIH transmit data register 0 for half word access (CSIHnTX0H)

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 or 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.



(14) CSIH receive data register 0 for half word access (CSIHnRX0H)

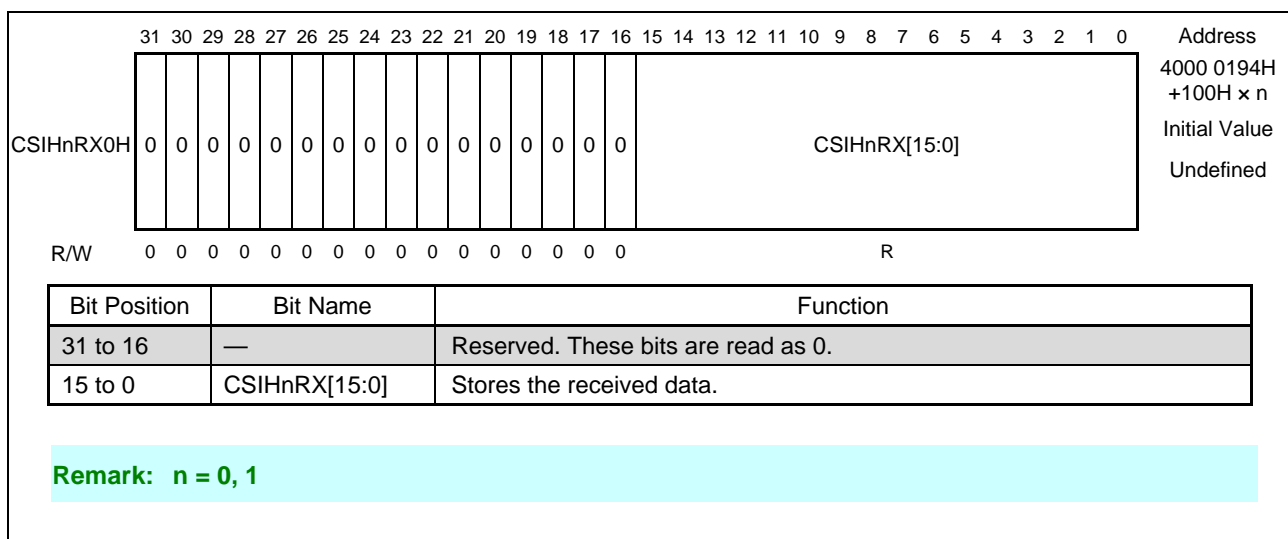
This register stores the reception data. This register is the same as bits 15 to 0 of register CSIHnRX0W.

- 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.



17.4 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

17.4.1 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 17.4.3, 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:

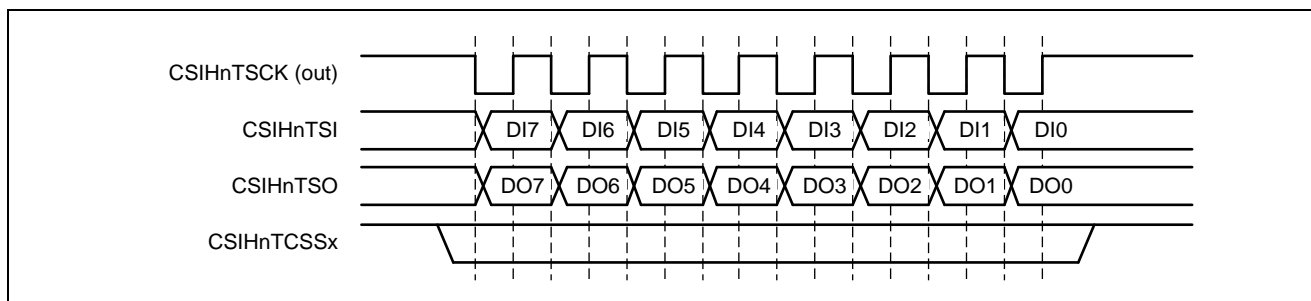


Figure 17.2 Transmission/Reception in Master Mode

(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 register 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.

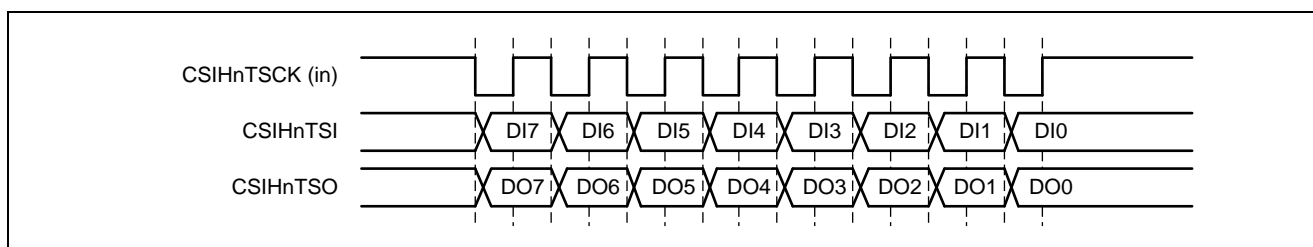


Figure 17.3 Transmission/Reception in Slave Mode

17.4.2 Master/Slave Connections

(1) One master and one slave

The following figure illustrates the connections between one master and one slave.

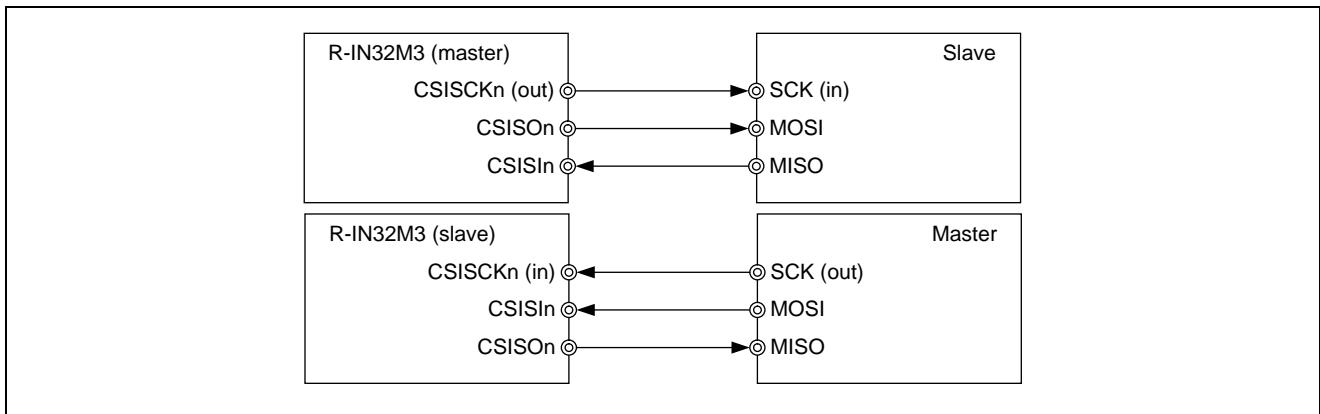


Figure 17.4 Direct Master/Slave Connection

(2) One master and two slaves

The following figure illustrates the connections between an R-IN32M3 as a master and two slaves. In this example, an R-IN32M3 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.

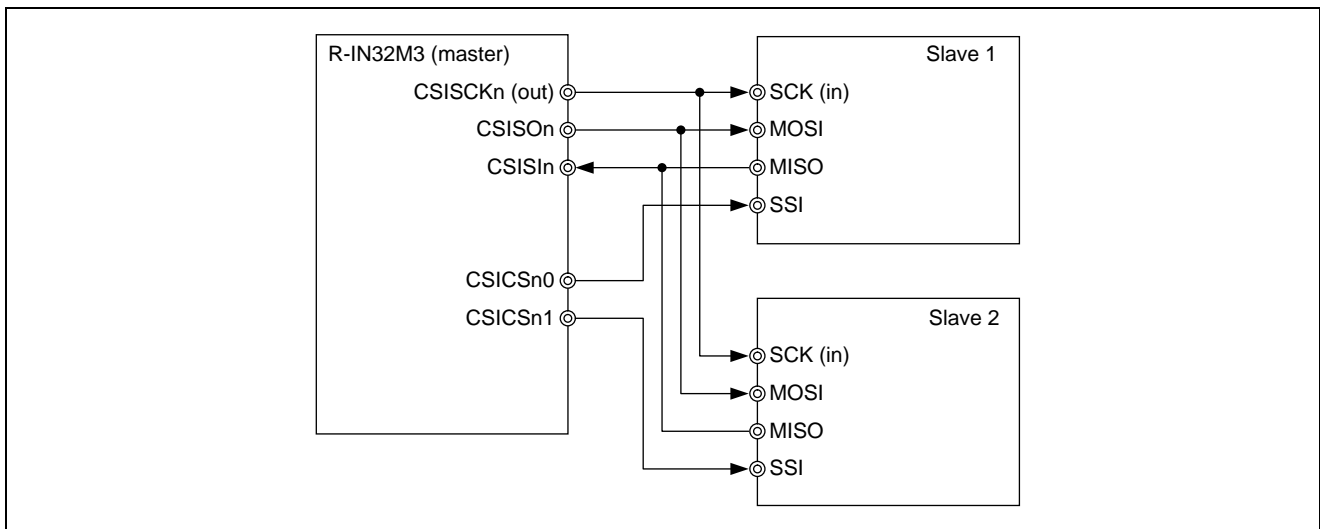


Figure 17.5 Connection between One Master and Two Slaves

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.

17.4.3 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 17.6 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.

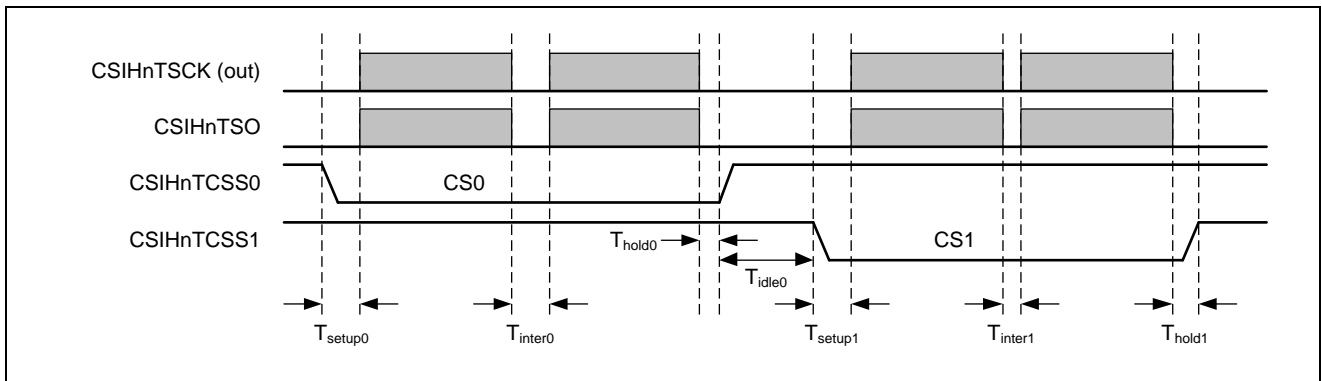


Figure 17.6 Chip Select Timings

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”.

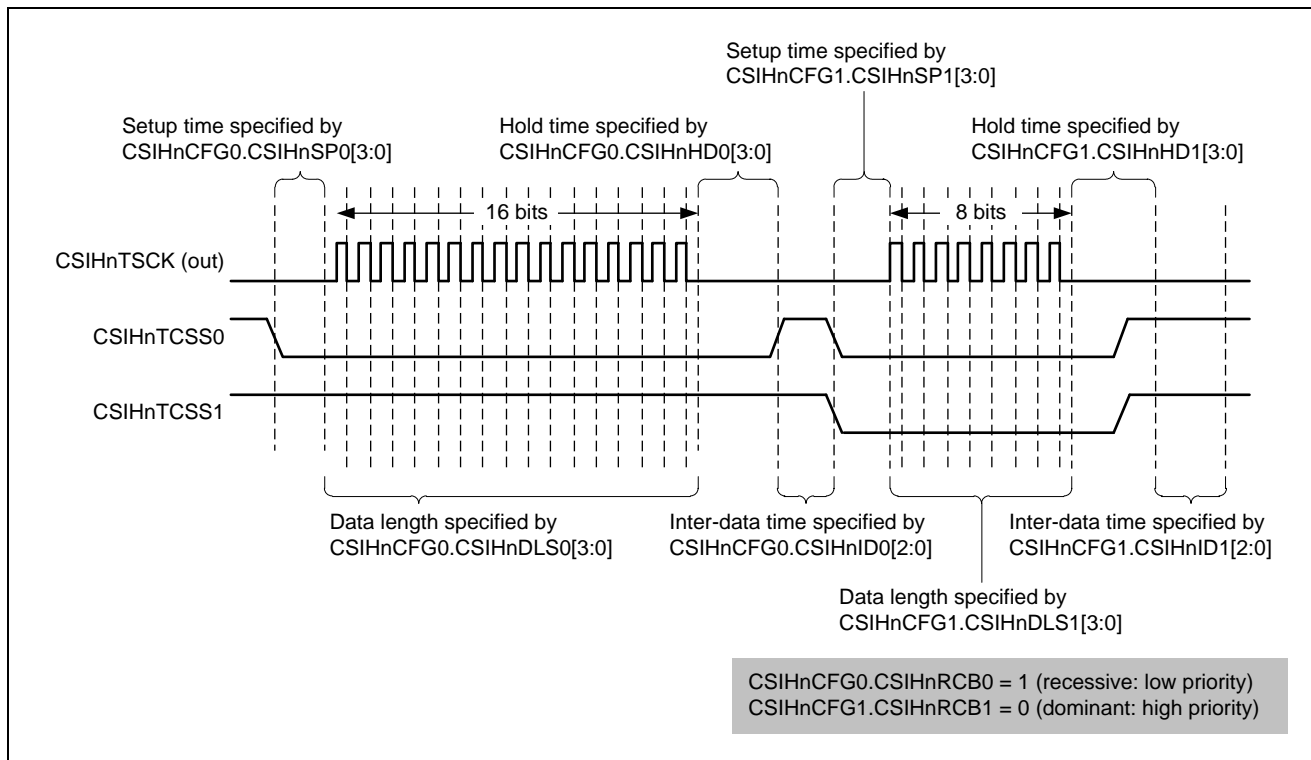


Figure 17.7 Chip Select and RCB Example

17.4.4 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 $CSIHnTSCK$.

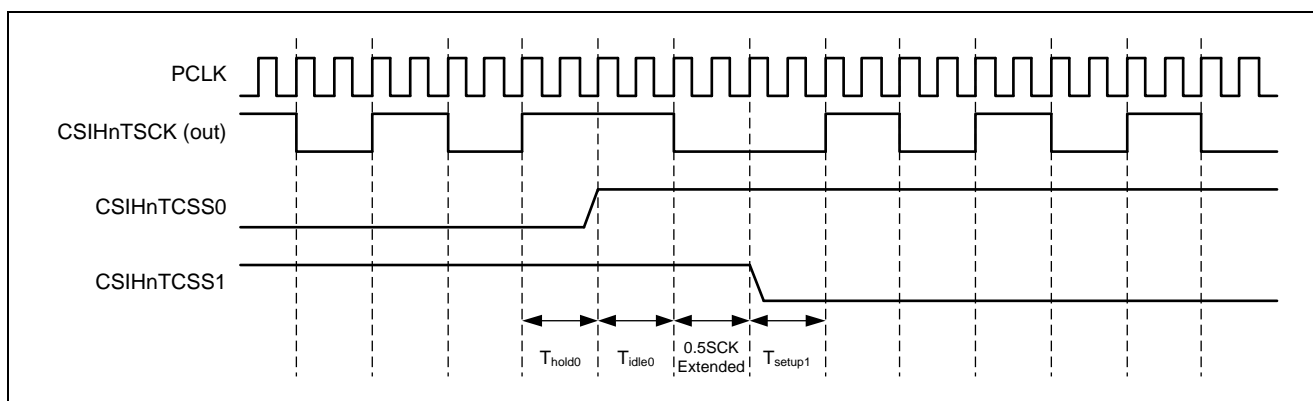


Figure 17.8 Clock Phase Timing (in the case of $PCLK/4$, $T_{hold0} = T_{setup1} = 0.5SCK$, $T_{idle0} = 0.5SCK$, $CKP0 = 0$ ($CSIHnTCSS0$) \rightarrow $CKP1 = 1$ ($CSIHnTCSS1$))

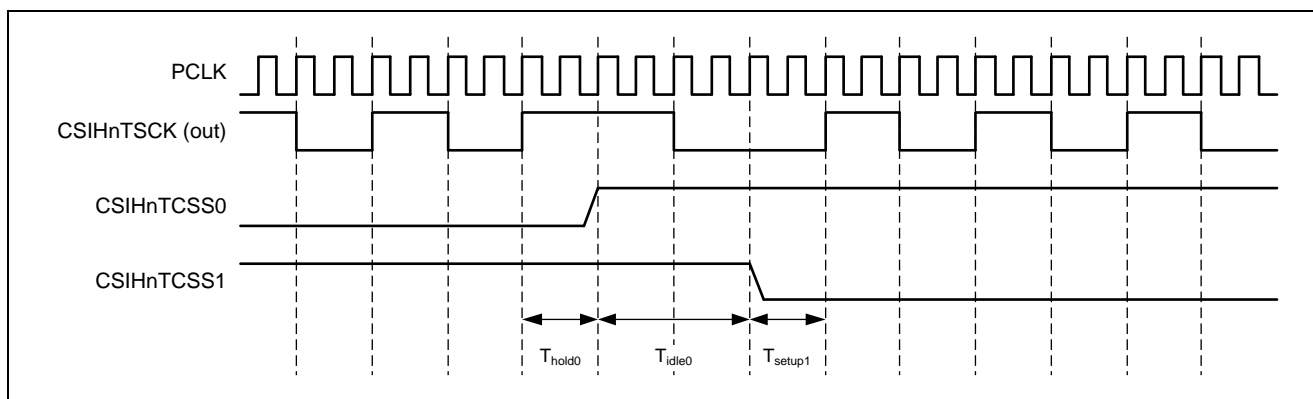


Figure 17.9 Clock Phase Timing (in the case of $PCLK/4$, $T_{hold0} = T_{setup1} = 0.5SCK$, $T_{idle0} = 1.0SCK$, $CKP0 = 0$ ($CSIHnTCSS0$) \rightarrow $CKP1 = 1$ ($CSIHnTCSS1$))

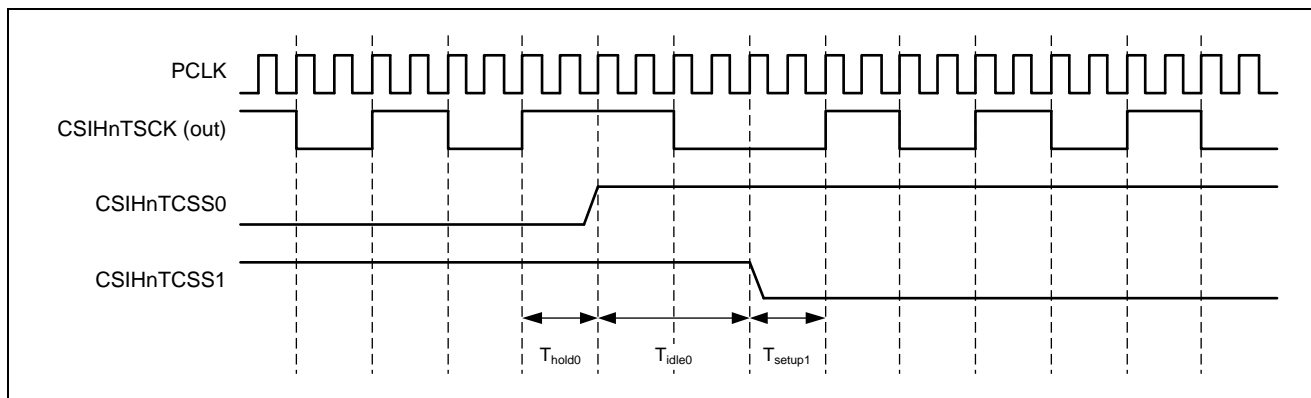


Figure 17.10 Clock Phase Timing (in the case of $PCLK/4$, $T_{hold0} = T_{setup1} = 0.5SCK$, $T_{idle0} = 0.5SCK$, $CKP0 = 0$ (CSIHnTCSS0) → $CKP1 = 0$ (CSIHnTCSS1))

(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.

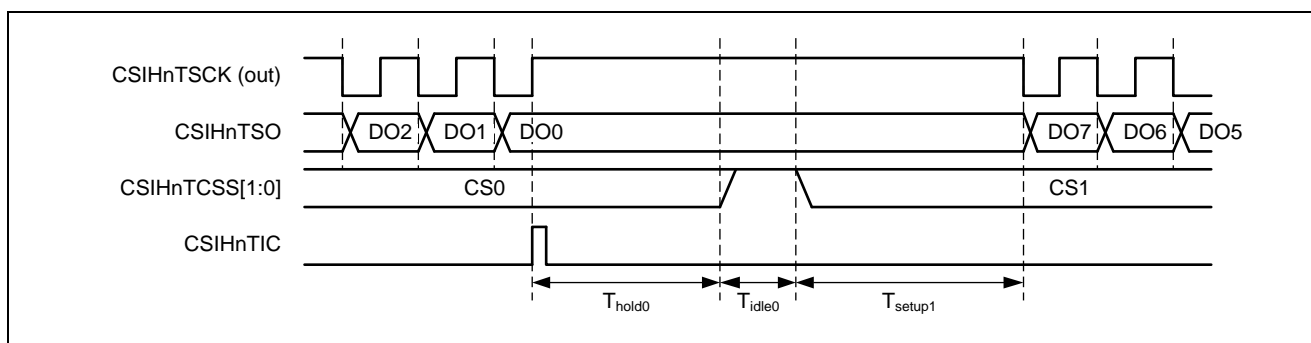


Figure 17.11 Data Phase Timing with $CSIHnCFG0.CSIHnCKP0 = 0$, $CSIHnCFG0.CSIHnDAP0 = 0$ and $CSIHnCFG1.CSIHnCKP1 = 0$, $CSIHnCFG1.CSIHnDAP1 = 0$

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:

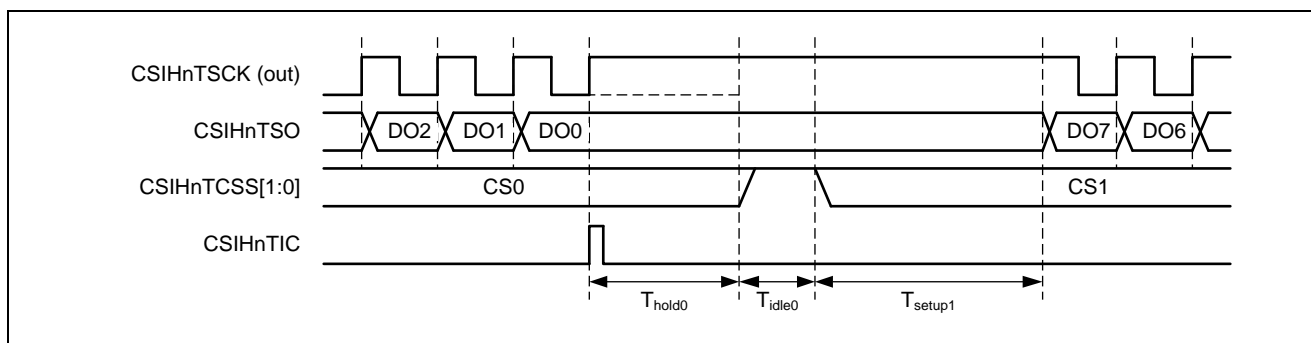


Figure 17.12 Data Phase Timing with $CSIHnCFG0.CSIHnCKP0 = 0$, $CSIHnCFG0.CSIHnDAP0 = 1$ and $CSIHnCFG1.CSIHnCKP1 = 0$, $CSIHnCFG1.CSIHnDAP1 = 1$

Note that the minimum idle time of one CSIHnTSCK cycle is automatically inserted, if CSIHnCFGx.CSIHnIDx[2:0] = 0 (Tidle1 = 0.5 CSIHnTSCK cycles).

17.4.5 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.

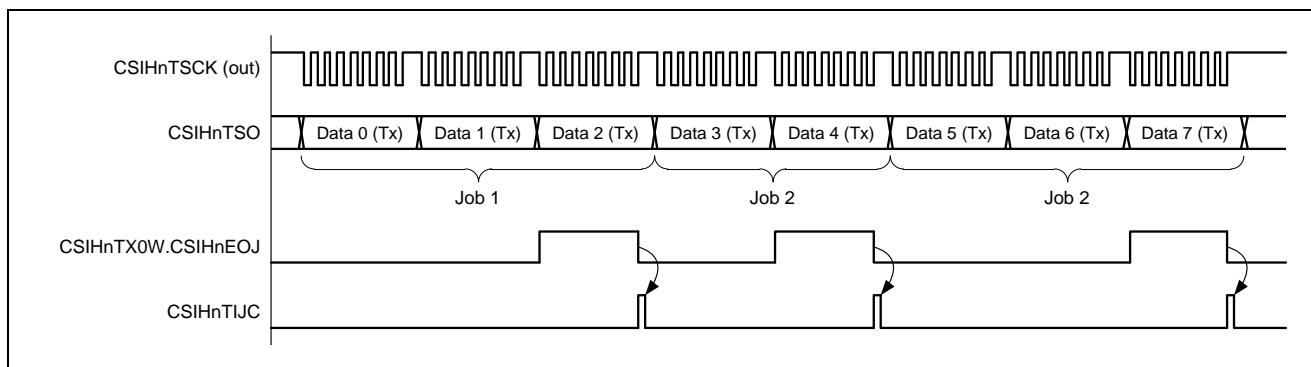


Figure 17.13 Job Examples

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.

17.4.6 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.

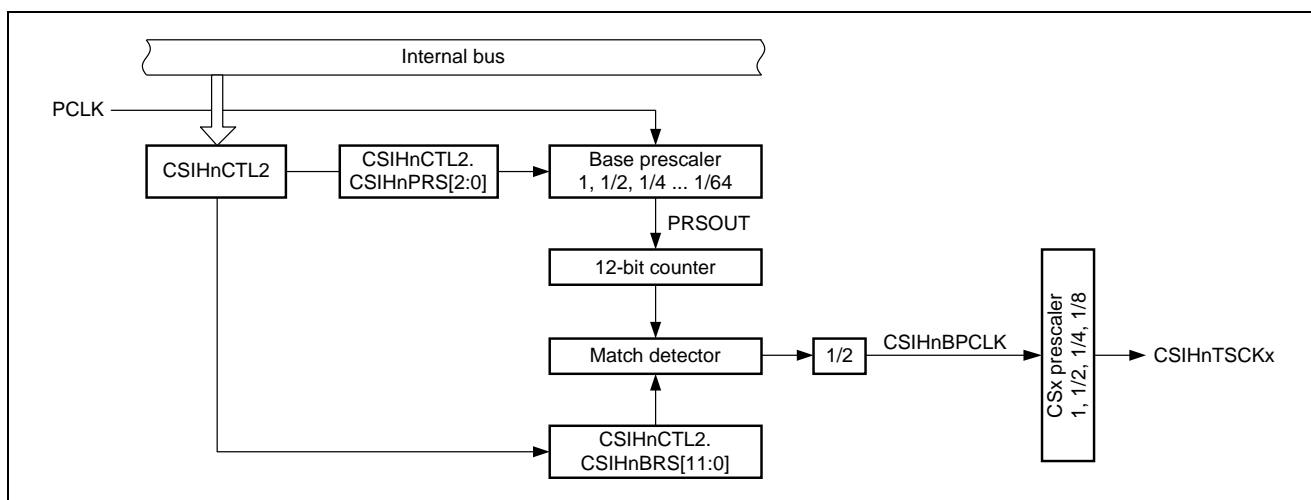


Figure 17.14 Baud Rate Generator Block Diagram

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:

$$\text{CSIHnTSCkx} = \text{PCLK} / (2^m \times k \times 2 \times 2^j)$$

where

$$m = \text{CSIHnCTL2.CSIHnPRS}[2:0] = 0-6$$

$$k = \text{CSIHnCTL2.CSIHnBRS}[11:0] = 1-4095$$

$$j = \text{CSIHnCFGx.CSIHnPSCLx}[1:0] = 0-3$$

17.4.6.1 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 ($PCLK/524160$).

17.4.7 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.

17.4.8 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	Master, Slave	When transmission data is written to the $CSIHnTX0W$ or $CSIHnTX0H$ register
	Transmission/reception mode		
	Reception mode	Master	When dummy data is written to the $CSIHnTX0W$ or $CSIHnTX0H$ register
Transmit-only buffer mode, dual buffer mode	Reception mode	Slave	When the serial clock $CSIHnTSCK$ is received from the master
	Transmission mode	Master, Slave	When 1 is written to $CSIHnMCTL2.CSIHnBTST$
	Transmission/reception mode		
	Reception mode		

17.4.9 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):

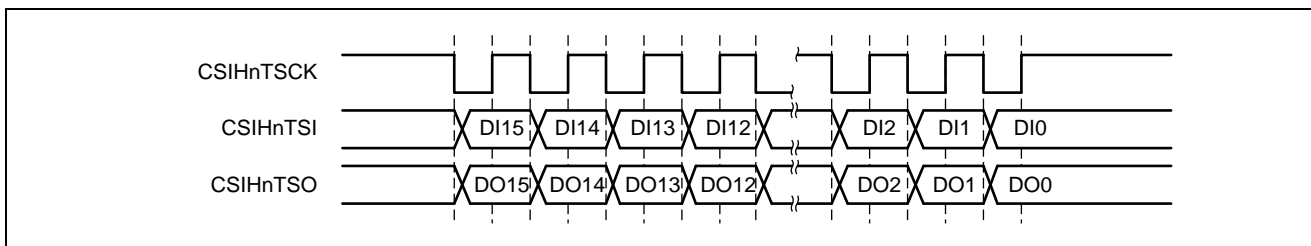


Figure 17.15 16-Bit Data, MSB First

- Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110B):

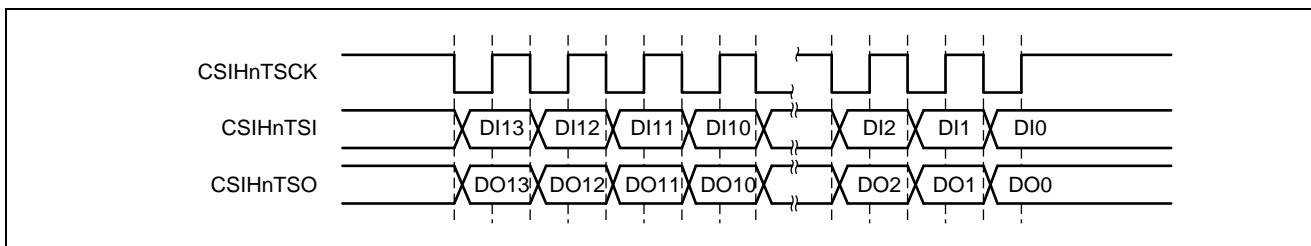


Figure 17.16 14 Bit-Data, MSB First

(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.

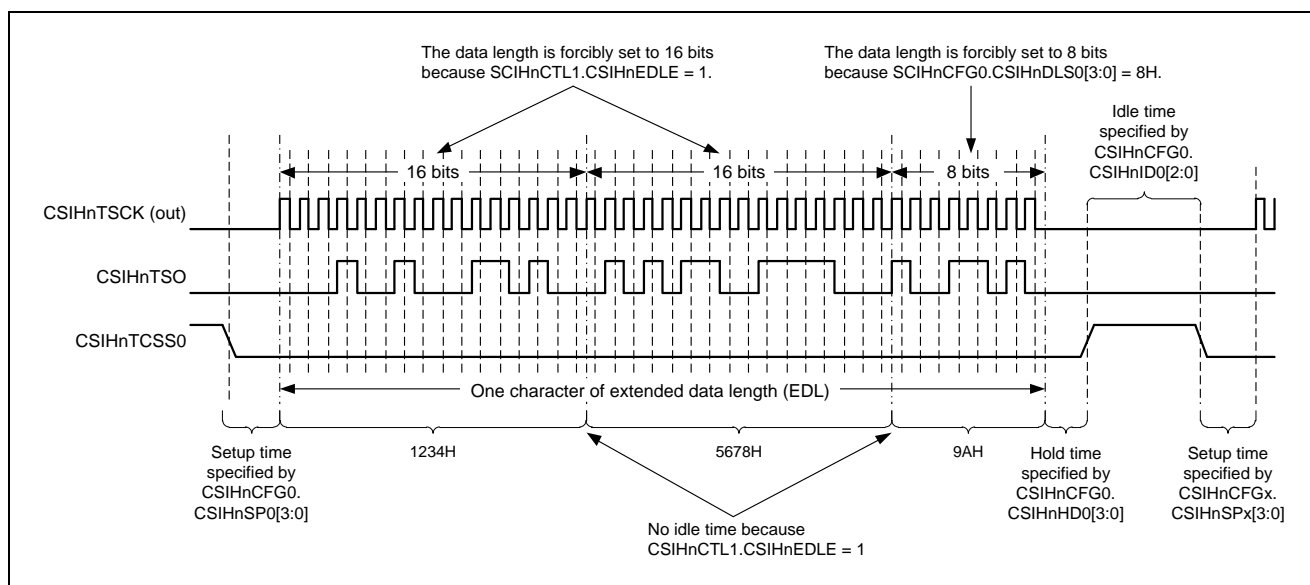


Figure 17.17 EDL Timing Chart

- 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).

17.4.10 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):

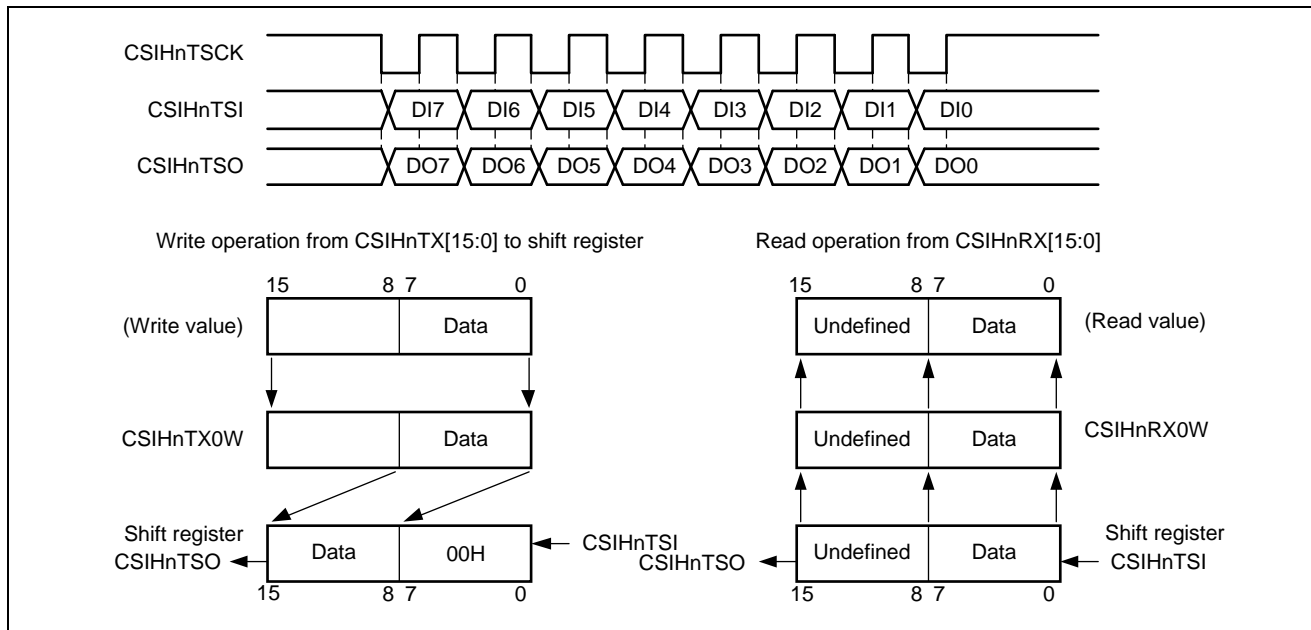


Figure 17.18 Serial Data Direction Select Function – MSB First (CSIHnDIR = 0)

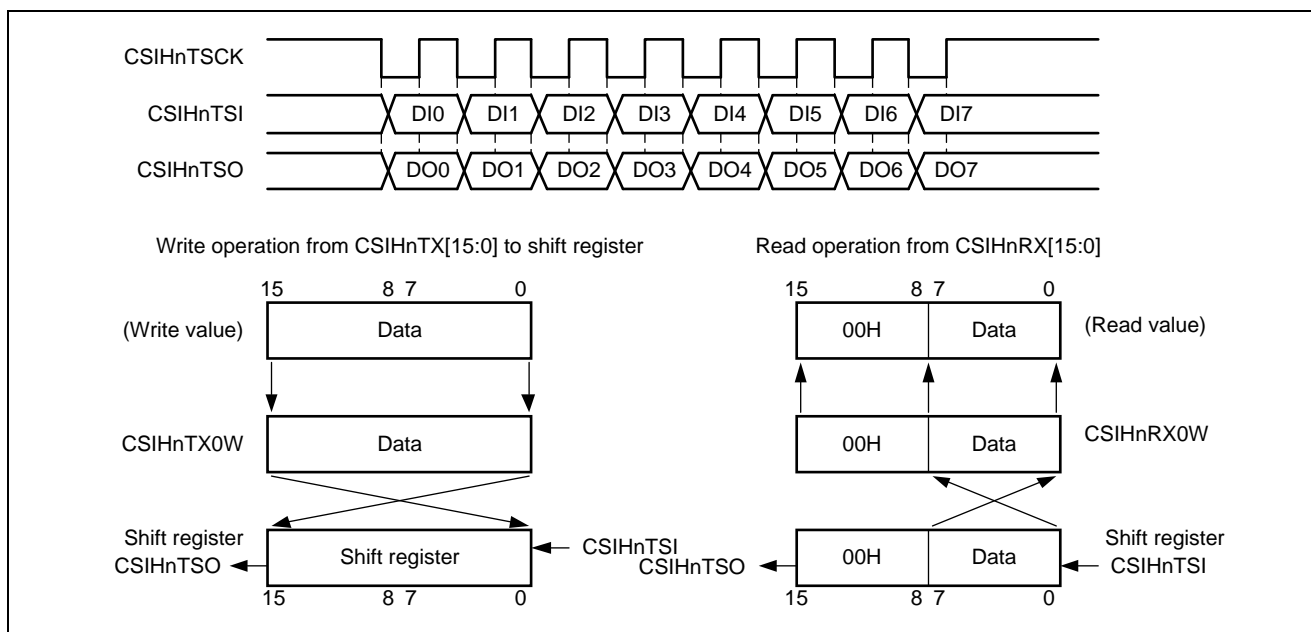


Figure 17.19 Serial Data Direction Select Function – LSB First (CSIHnDIR = 1)

17.4.11 Communication in Slave Mode

The following figure illustrates the communication signals and timings in slave mode.

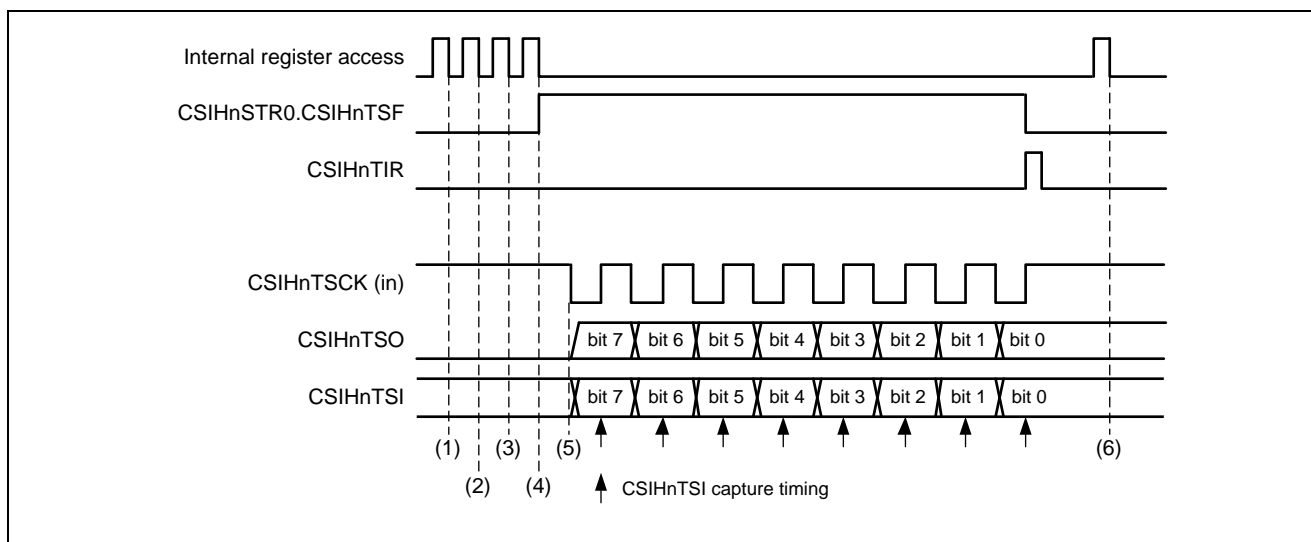


Figure 17.20 Transmit/Receive Communication Timing in Slave Mode

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 17.5, Operating Procedures.

17.4.12 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.

Memory Modes	Interrupt Source	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0
FIFO mode	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].	
	However, CSIHnTIC is not generated if the job is interrupted ^{Note} .	—
Transmit-only buffer mode, dual buffer mode	CSIHnTIC is generated when communication ends (as specified by the CSIHnMCTL2.CSIHnND[7:0] bits).	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.
Direct access mode	CSIHnTIC is generated each time a data transfer is performed. However, CSIHnTIC is not generated if the job is interrupted ^{Note} .	Except when communication is interrupted, CSIHnTIC is generated each time a data transfer is performed. However, when the data and a job interruption request ^{Note} are transmitted while CSIHnTX0W.CSIHnCIRE is "1", CSIHnTIJC is generated instead of CSIHnTIC.

Note: Condition for job interruption: 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)

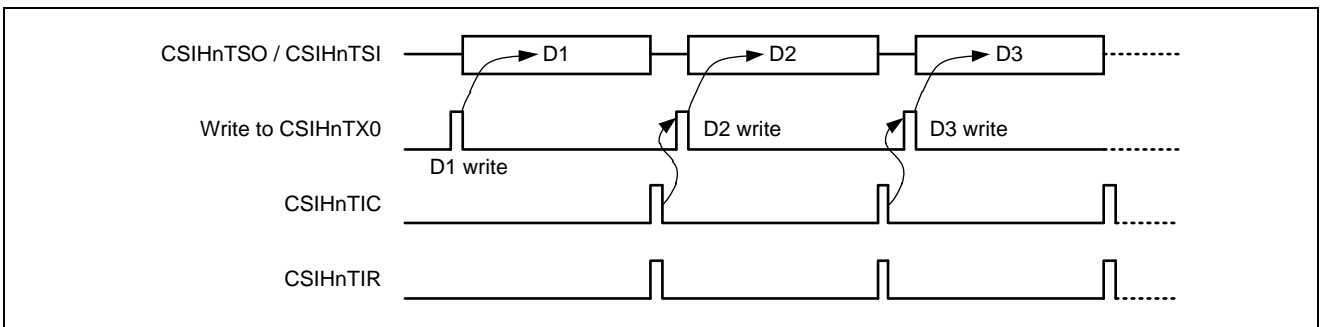


Figure 17.21 Generation of CSIHnTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

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.CSIHnJOB = 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.

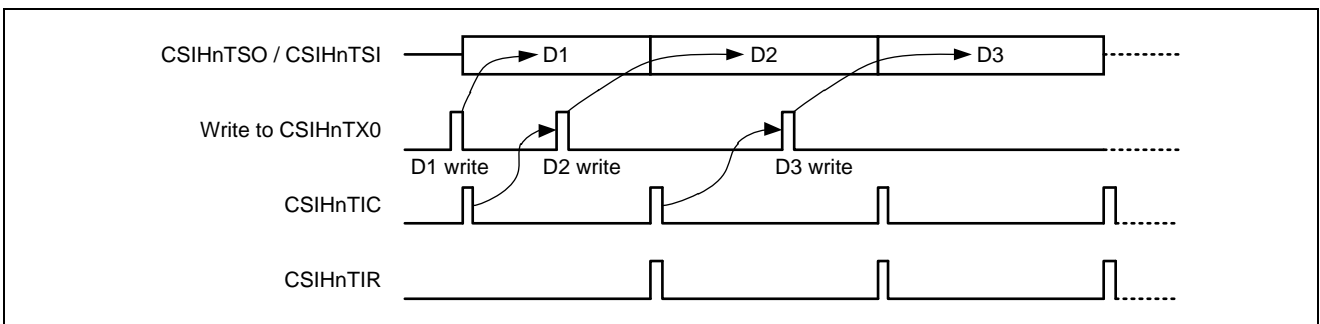


Figure 17.22 Immediate Generation of CSIHnTIC (CSIHnCTL1.CSIHnSLIT = 1)

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)

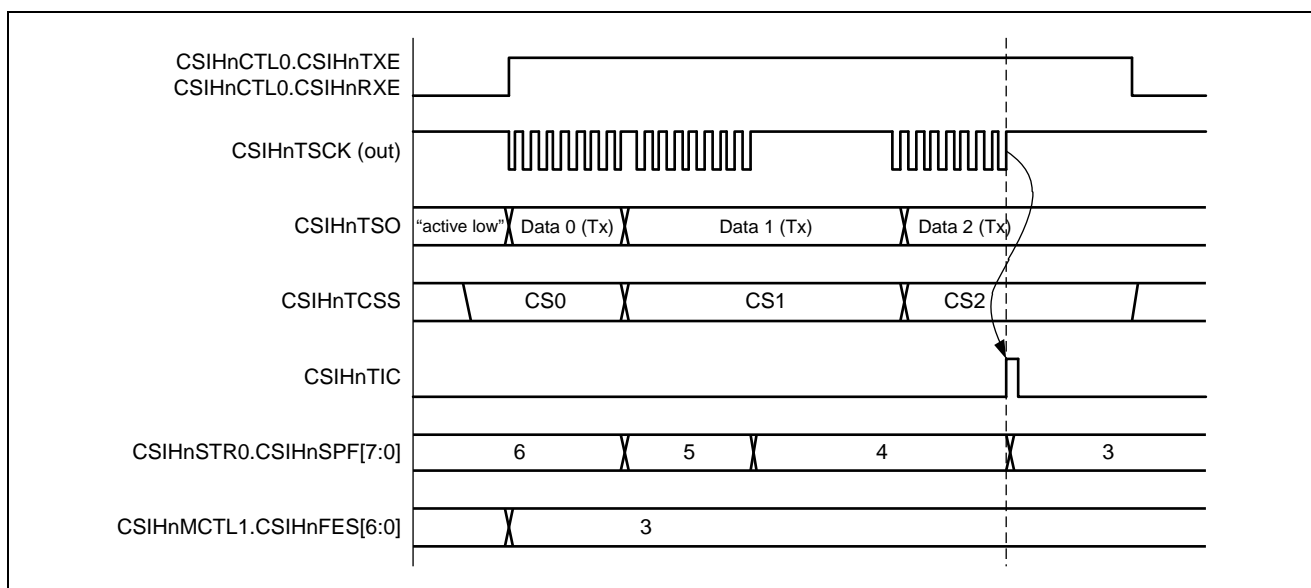


Figure 17.23 Generation of CSIHnTIC in FIFO Memory Mode

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)

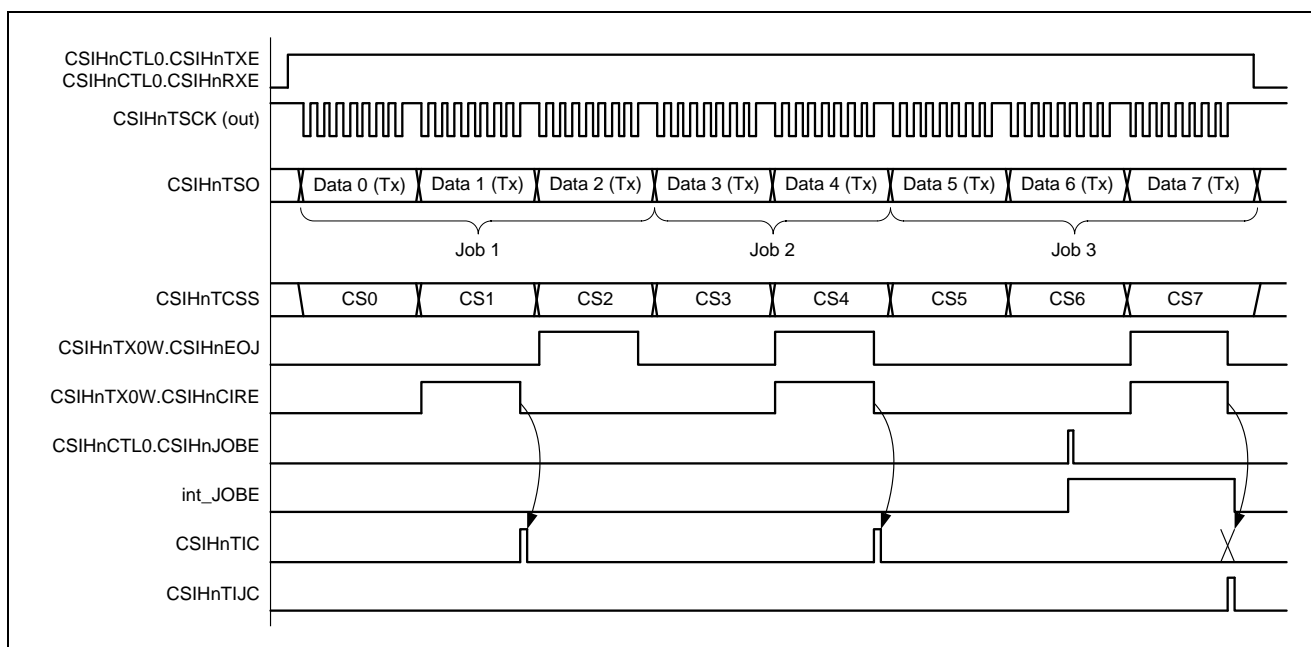


Figure 17.24 Generation of CSIHnTIC in Job Mode

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 17.9 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 17.10 CSIHnTIR Interrupt Generation

Memory Mode	Master and Slave	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0
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)

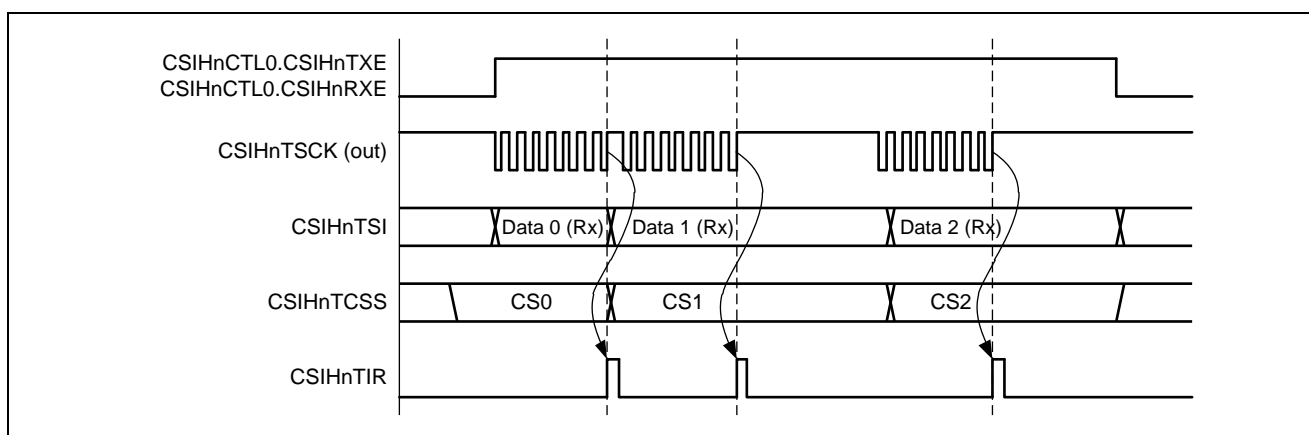


Figure 17.25 Generation of CSIHnTIR in Direct Access Memory Mode

(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)

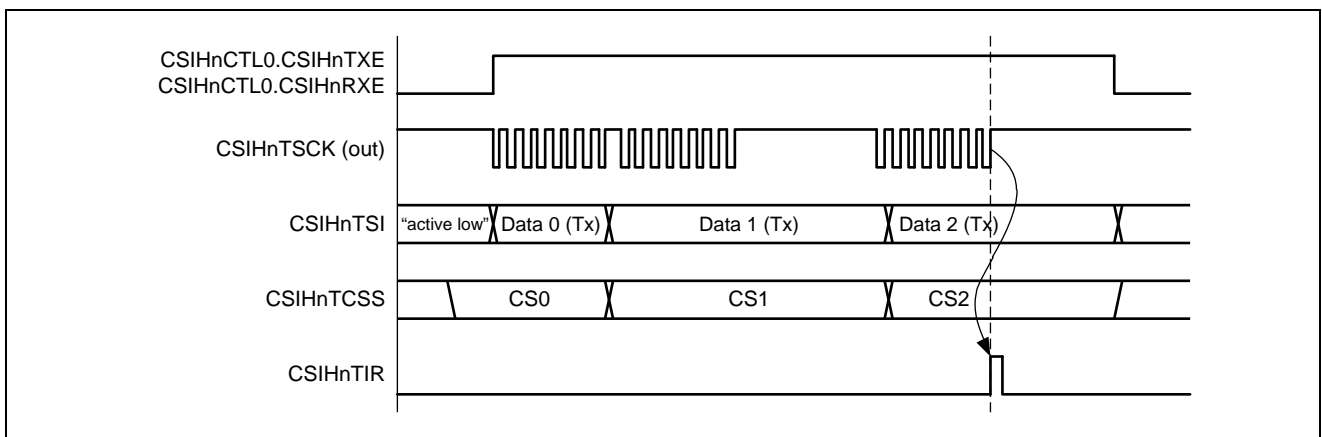


Figure 17.26 CSIHnTIR Generation in Dual Buffer Mode

(3) CSIHnTIRE reception error interrupt

This interrupt is generated whenever an error is detected.

Table 17.11 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 the 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 17.4.13, Error Detection.

(4) CSIHnTIJC job completion interrupt

This interrupt supports the handling of jobs – refer to section 17.4.5, 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 17.12 CSIHnTIJC Interrupt Generation

Memory Mode	Interrupt Source	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0
FIFO mode	Not applicable	After job interruption ^{Note} is triggered, communication stops on job completion.
Transmit-only buffer mode		
Dual buffer mode		
Direct access mode		

Note: Condition for job interruption: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOB = 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).

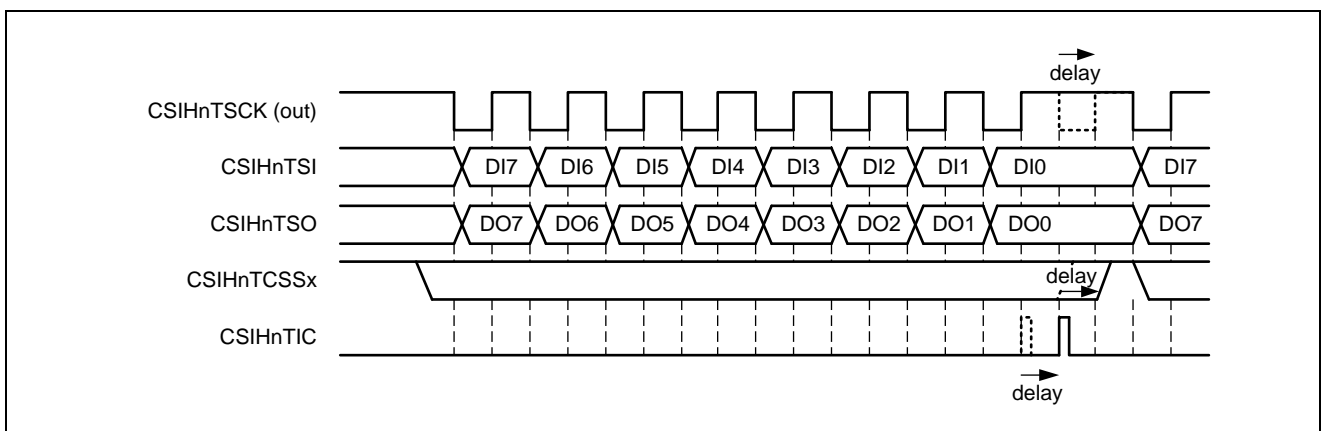


Figure 17.27 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

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).

17.4.13 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, CSIHnRX0W.CSIHnTDCE is set with the corresponding data.

The figure below is a block diagram of data consistency checking.

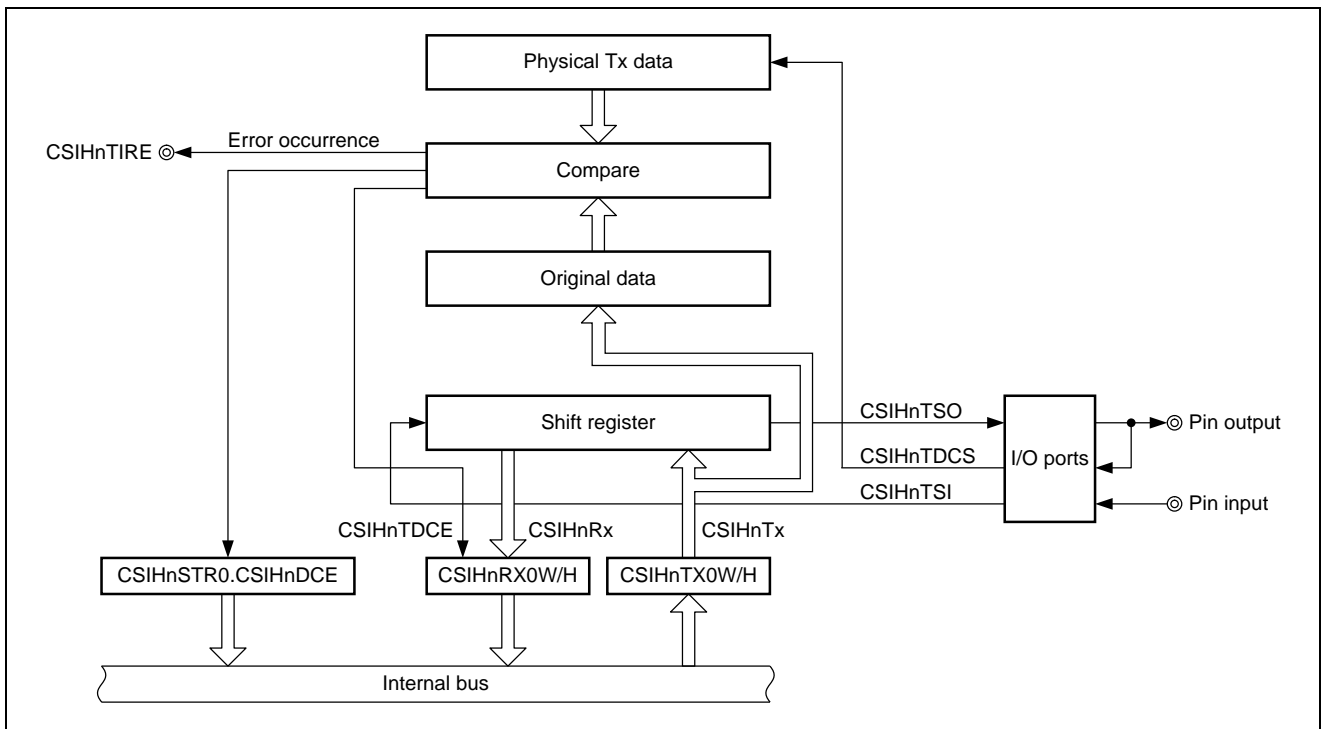


Figure 17.28 Block Diagram of Data Consistency Checking

(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.

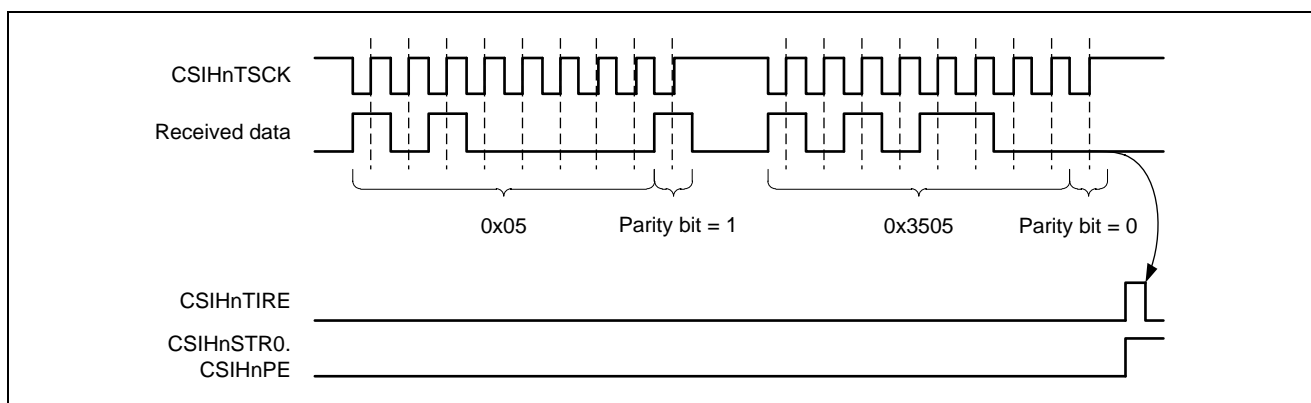


Figure 17.29 Parity Check Example

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.

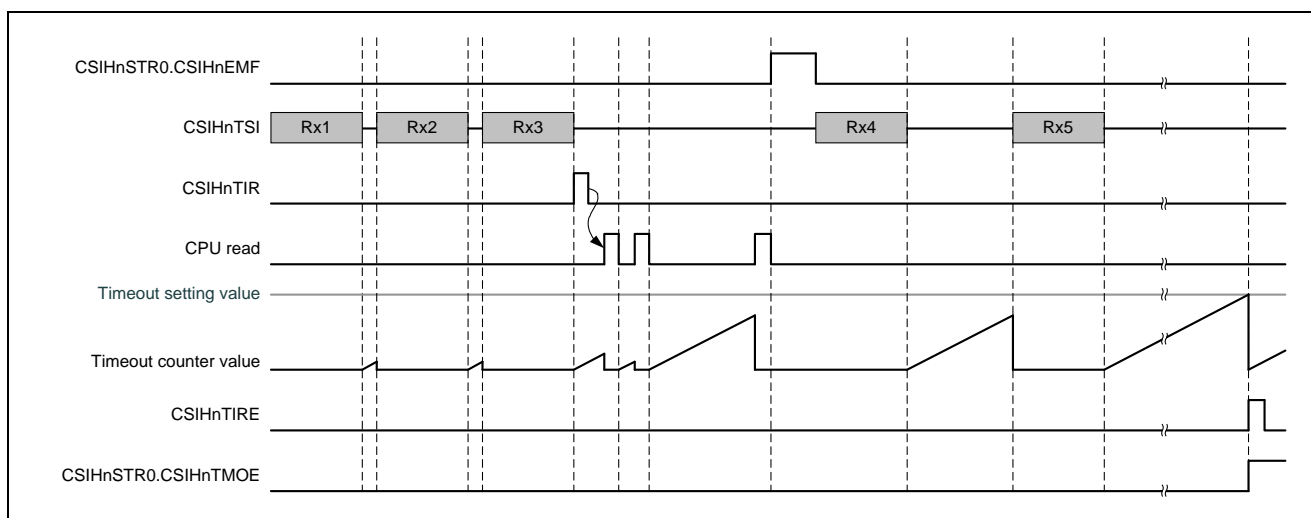


Figure 17.30 Timeout Error Check Functional Timing Chart

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:

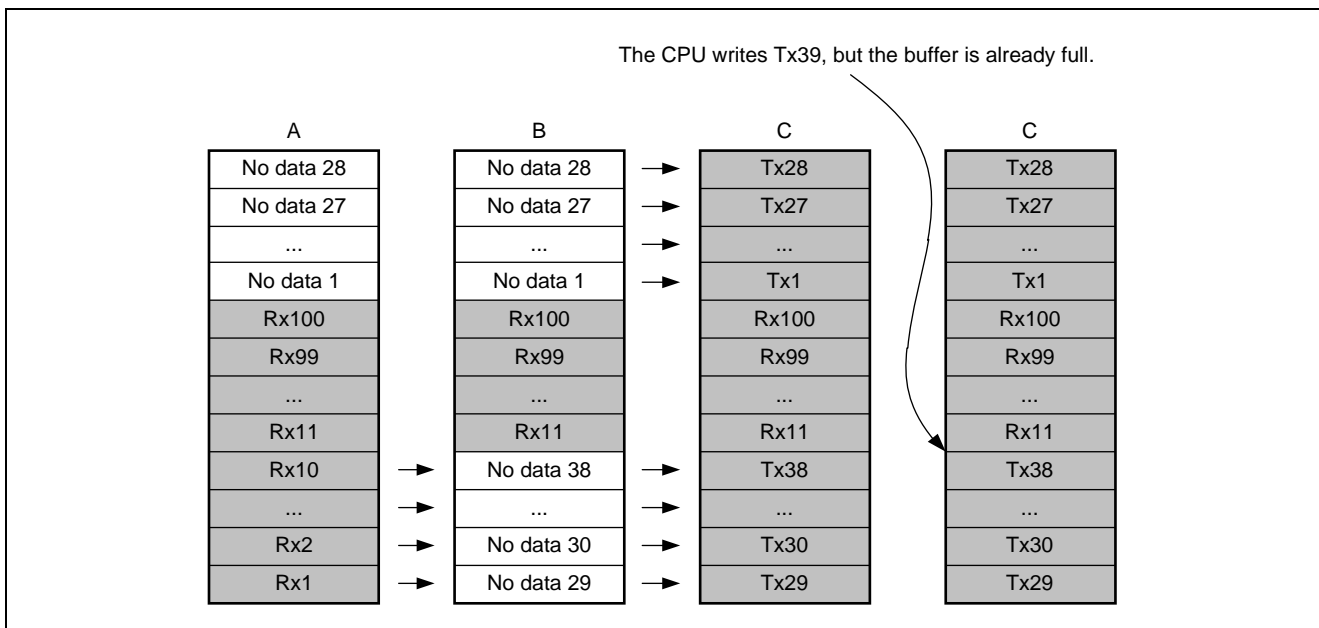


Figure 17.31 FIFO Overflow

The data after 39 are discarded. The following figure shows the associated timing.

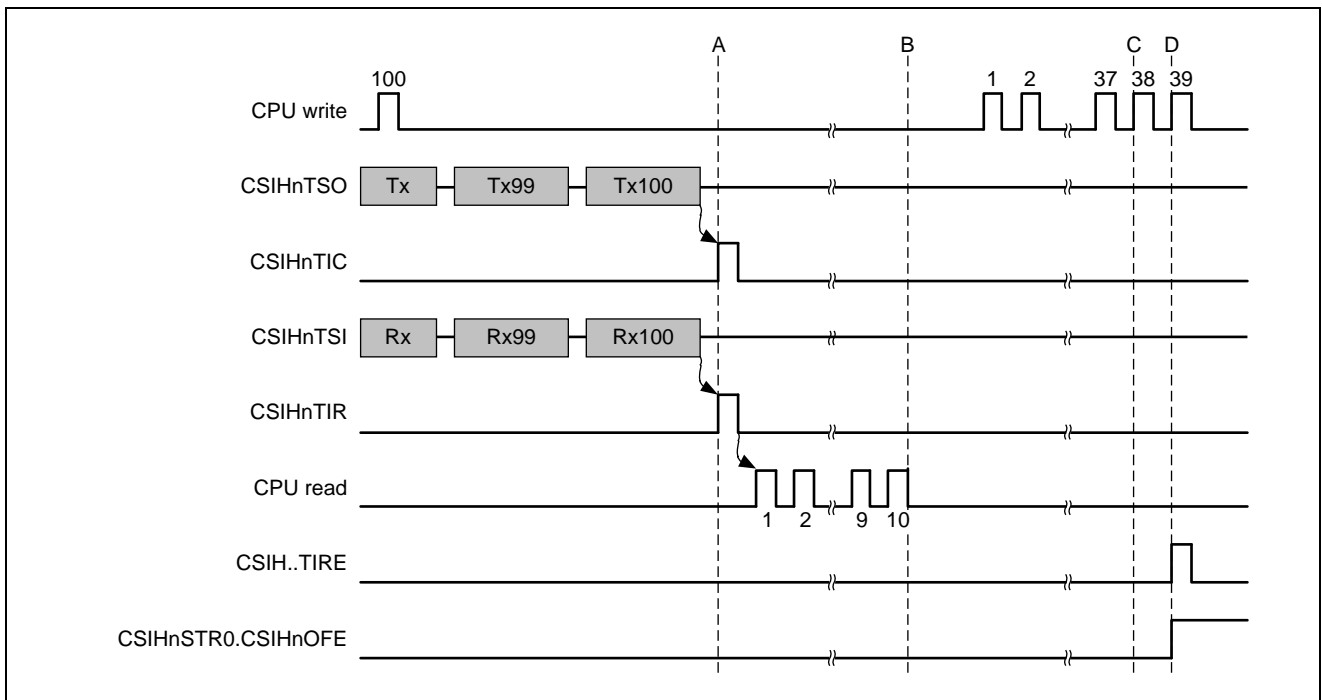


Figure 17.32 FIFO Overflow Timing

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 the 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.

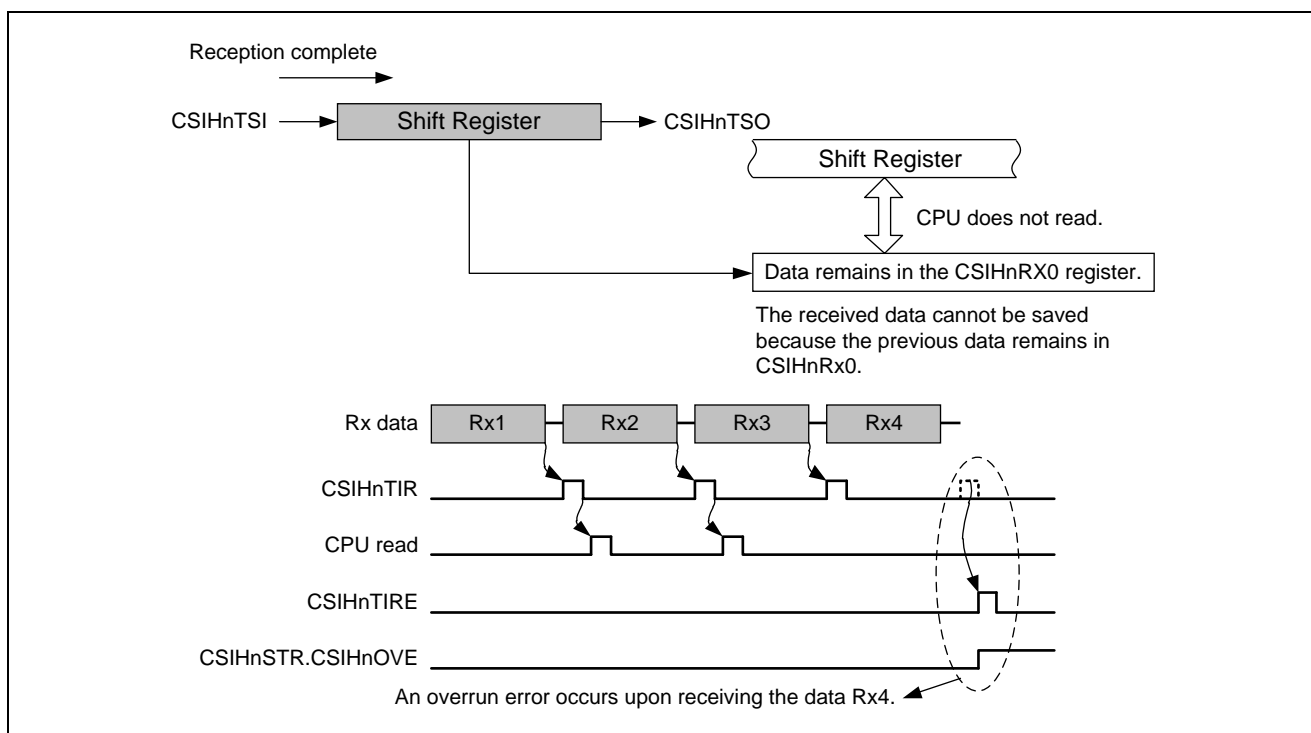


Figure 17.33 Overrun Error Detection in Direct Access and Transmit-Only Buffer Mode

(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).

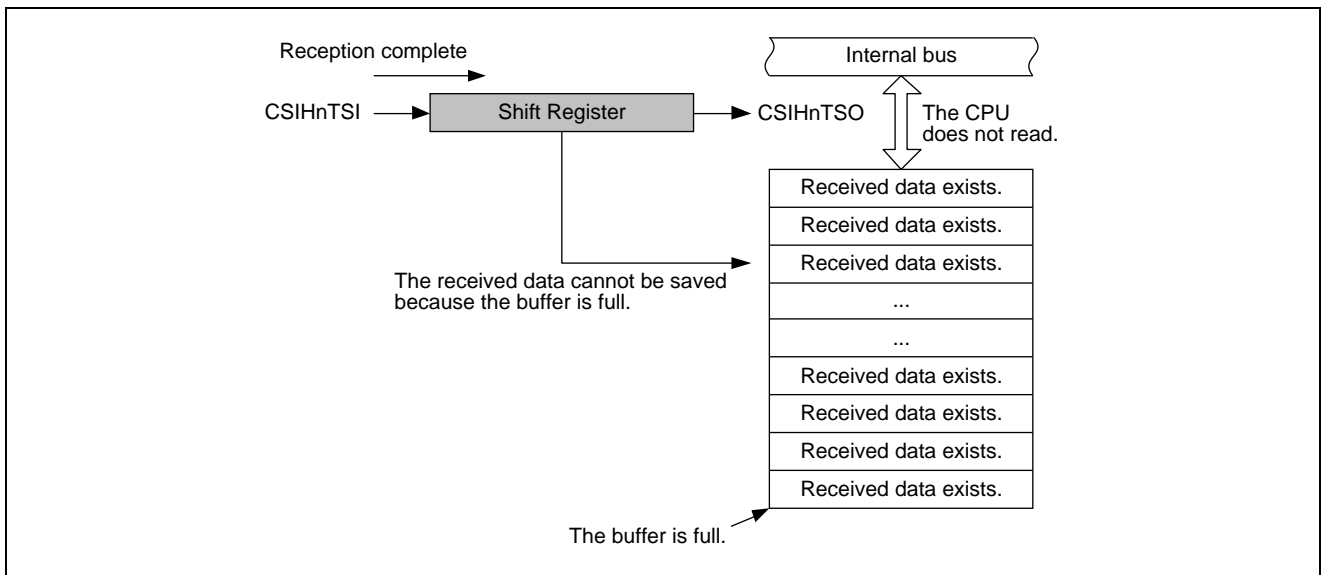


Figure 17.34 Overrun Error Detection in FIFO Mode (FIFO Full)

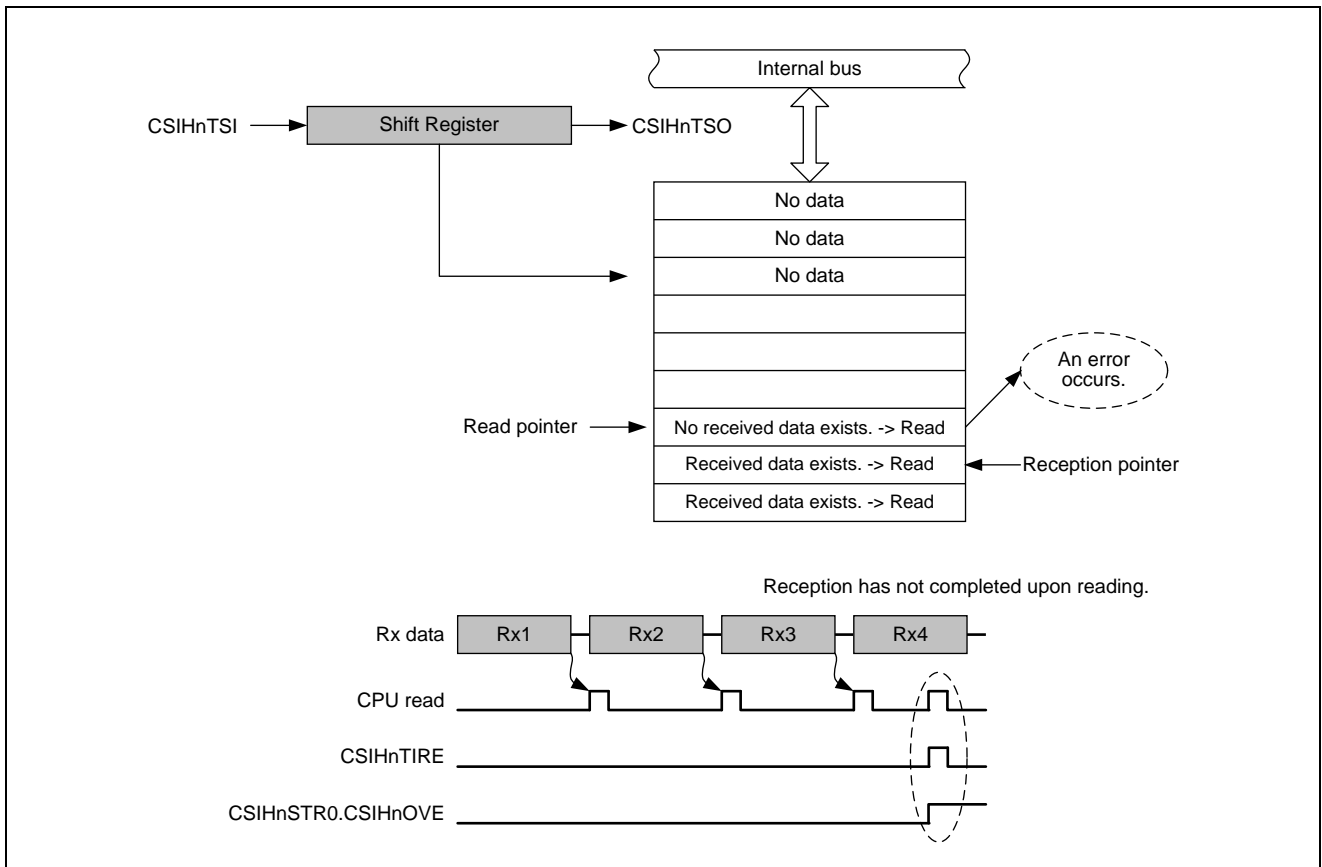


Figure 17.35 Overrun Error Detection in FIFO Mode (No Data)

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).

17.4.14 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.

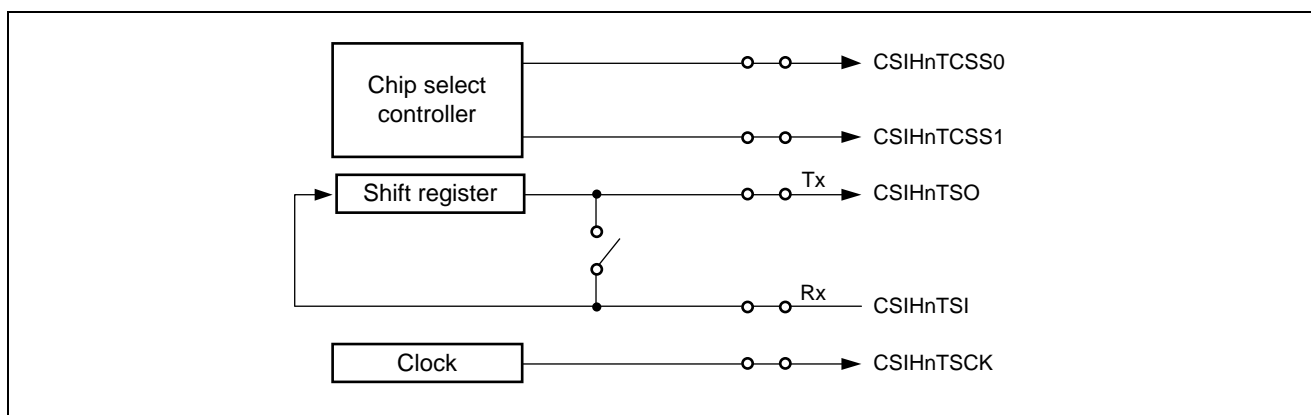


Figure 17.36 Normal Operation (CSIHnCTL1.CSIHnLBM = 0)

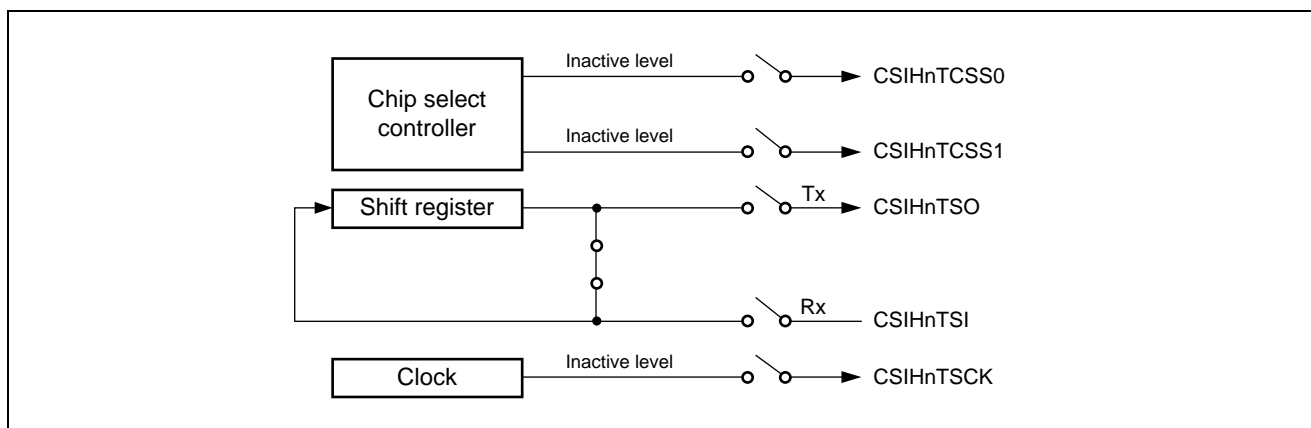


Figure 17.37 Loop-Back Operation (CSIHnCTL1.CSIHnLBM = 1)

17.5 Operating Procedures

The following examples and instructions are sorted according to the memory mode:

- Direct access
- Transmit-only buffer
- Dual buffer
- FIFO

17.5.1 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)

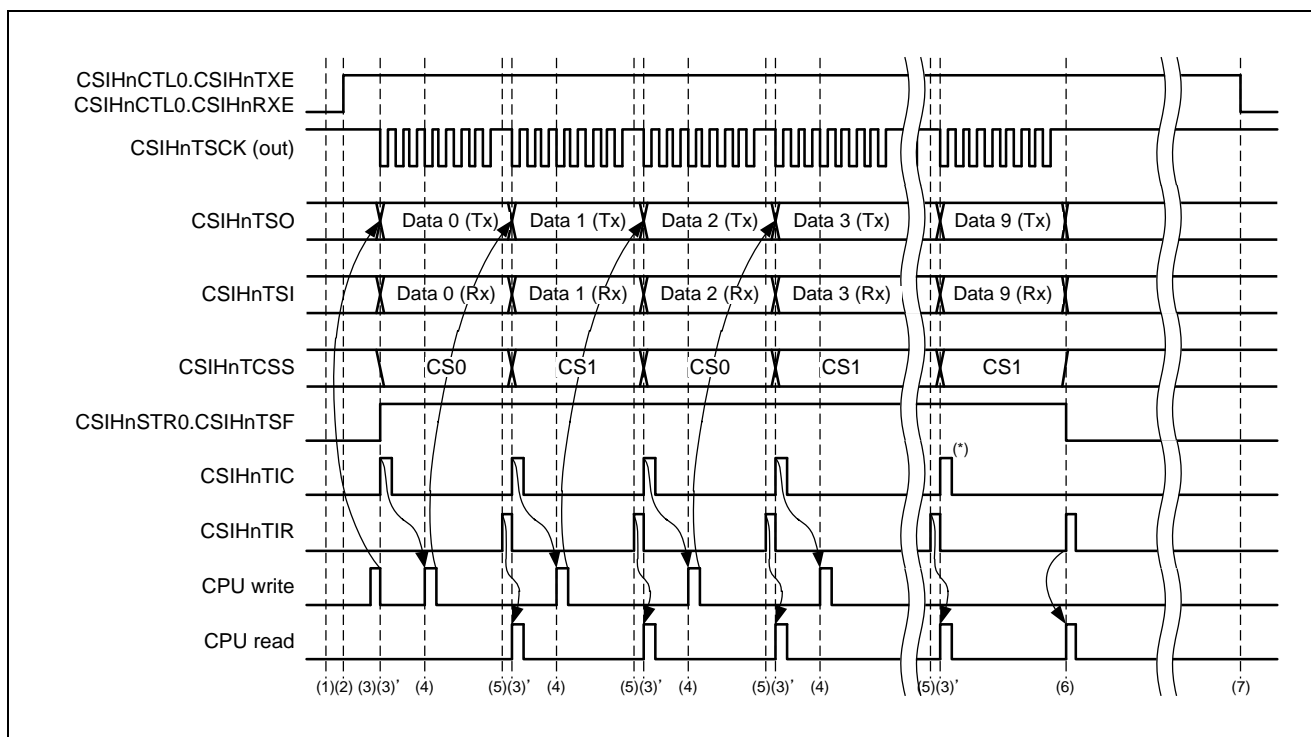


Figure 17.38 Direct Access Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)

- 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)

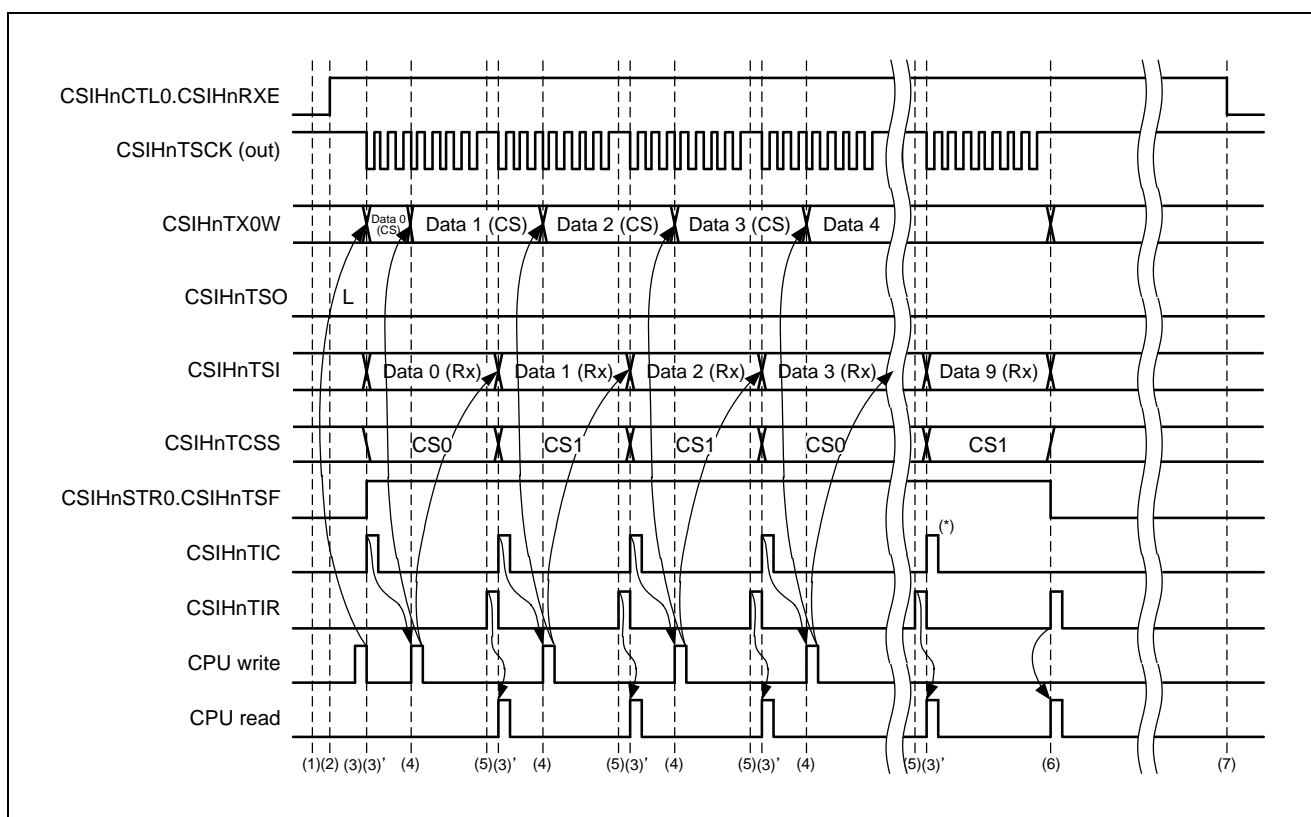


Figure 17.39 Direct Access Mode (for Reception in Master Mode, and when Job Mode is Disabled)

- 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)

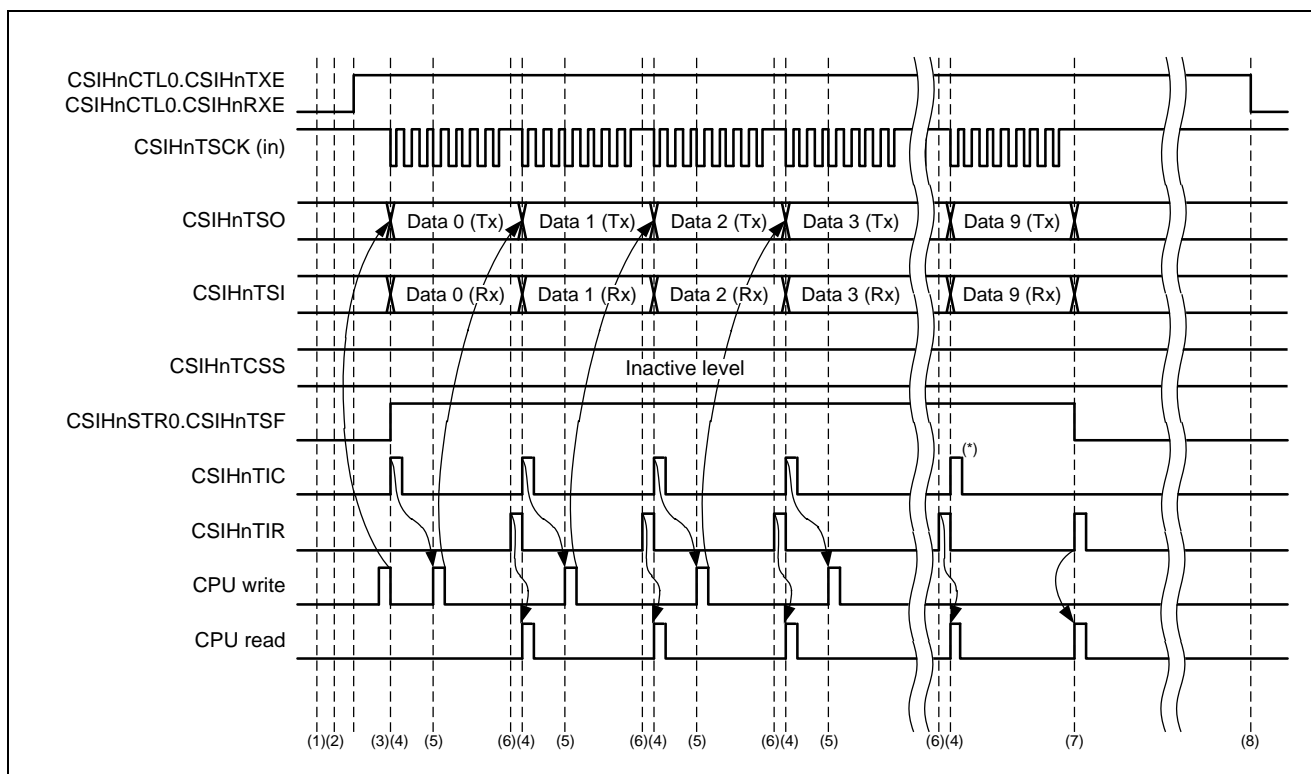


Figure 17.40 Direct Access Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)

- 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)

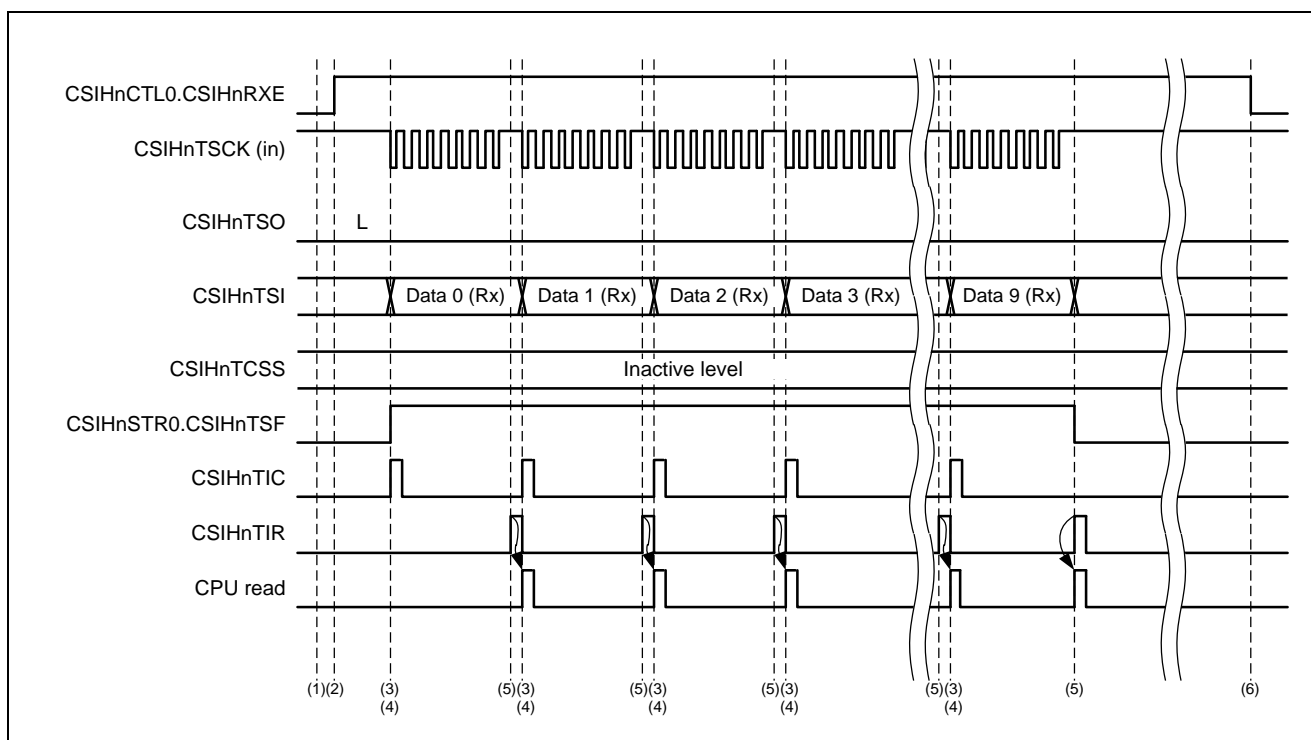


Figure 17.41 Direct Access Mode (for Reception in Slave Mode, and when Job Mode is Disabled)

- 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

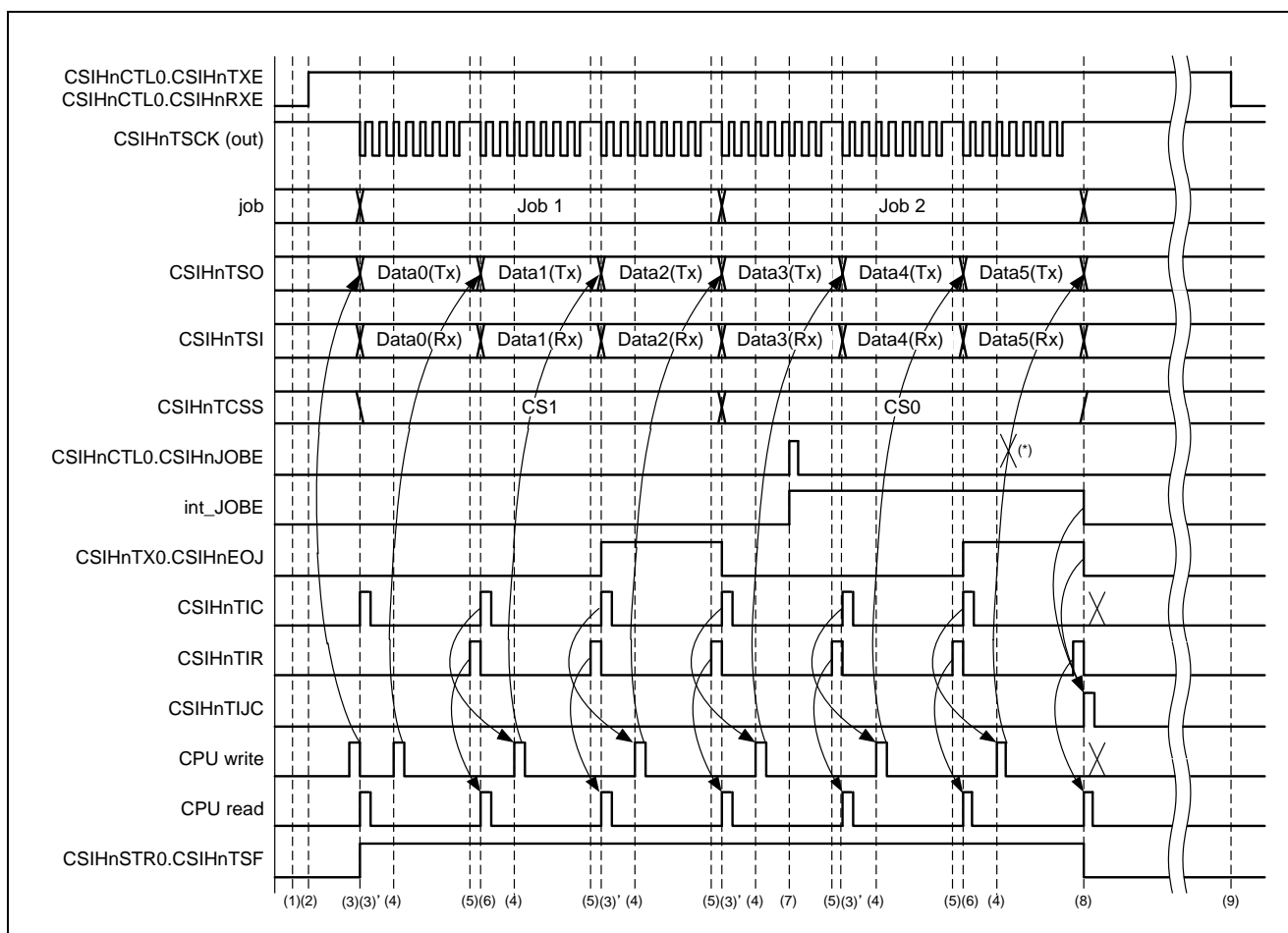


Figure 17.42 Direct Access Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)

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.CSIHnJOB 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

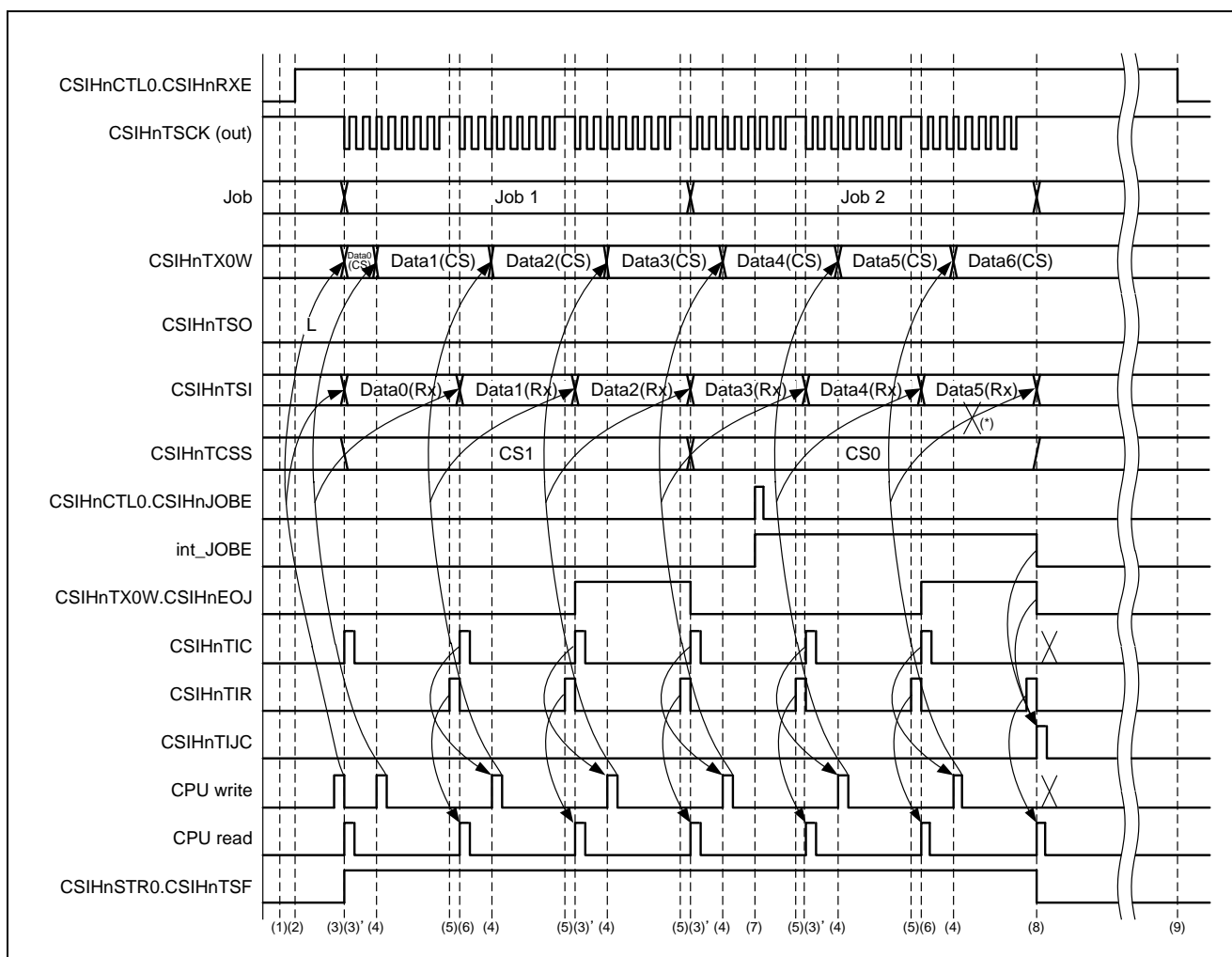


Figure 17.43 Direct Access Mode (for Reception in Master Mode, and when Job Mode is Enabled)

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.CSIHnJOB2 to 1, communication is forcibly stopped when the current job (job 2) ends.
 8. When int_JOB2 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.

17.5.2 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)

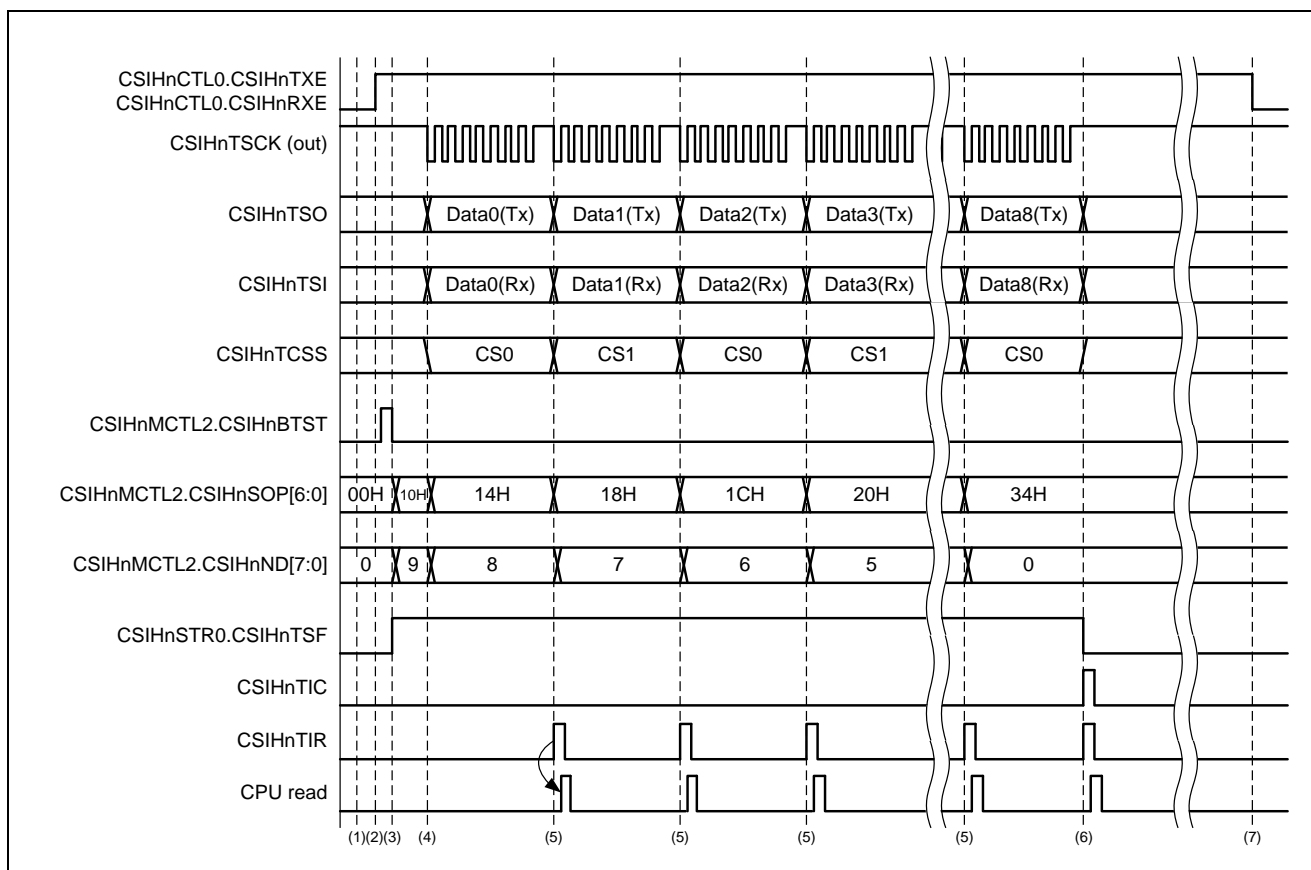


Figure 17.44 Transmit-Only Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)

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)

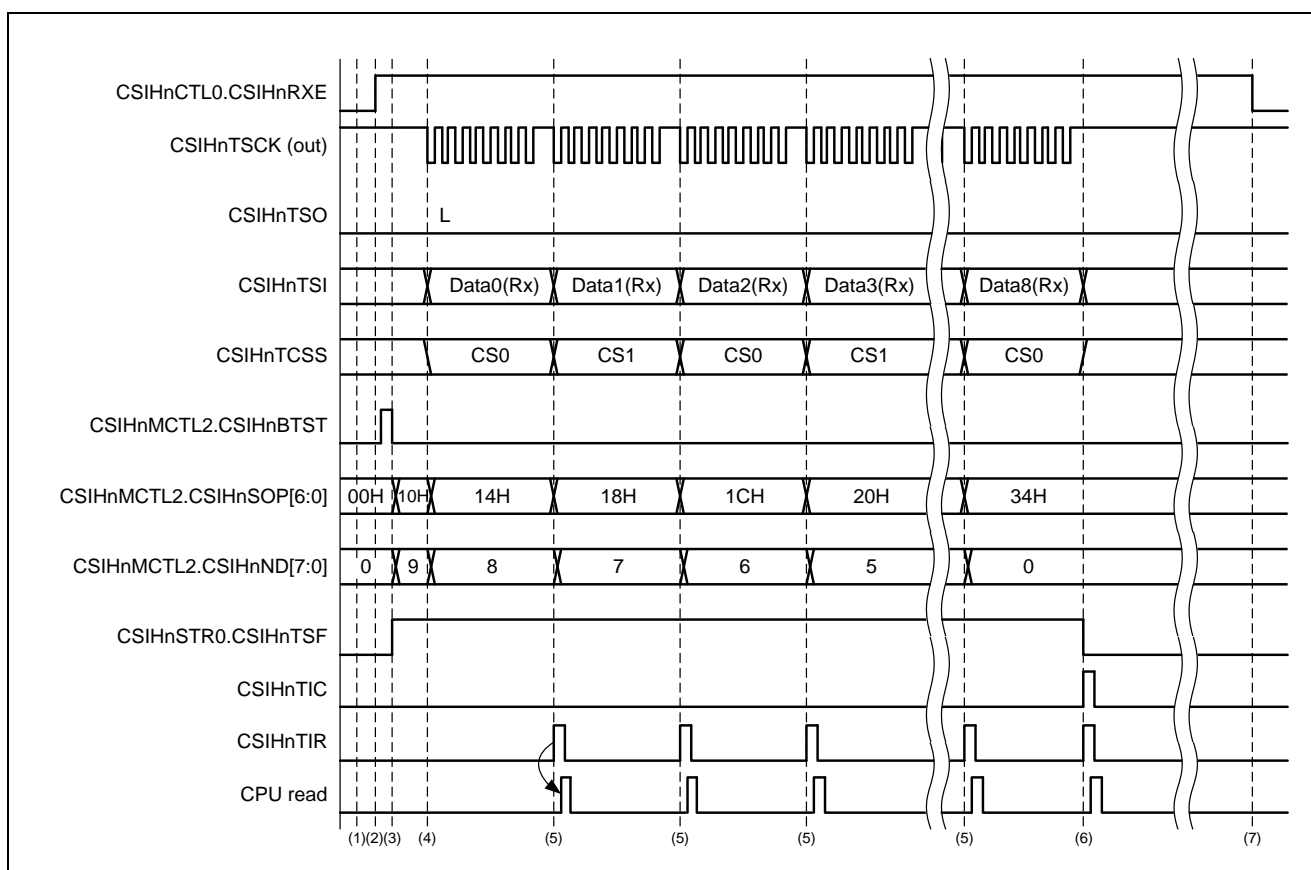


Figure 17.45 Transmit-Only Buffer Mode (for Reception in Master Mode, and when Job Mode is Disabled)

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)

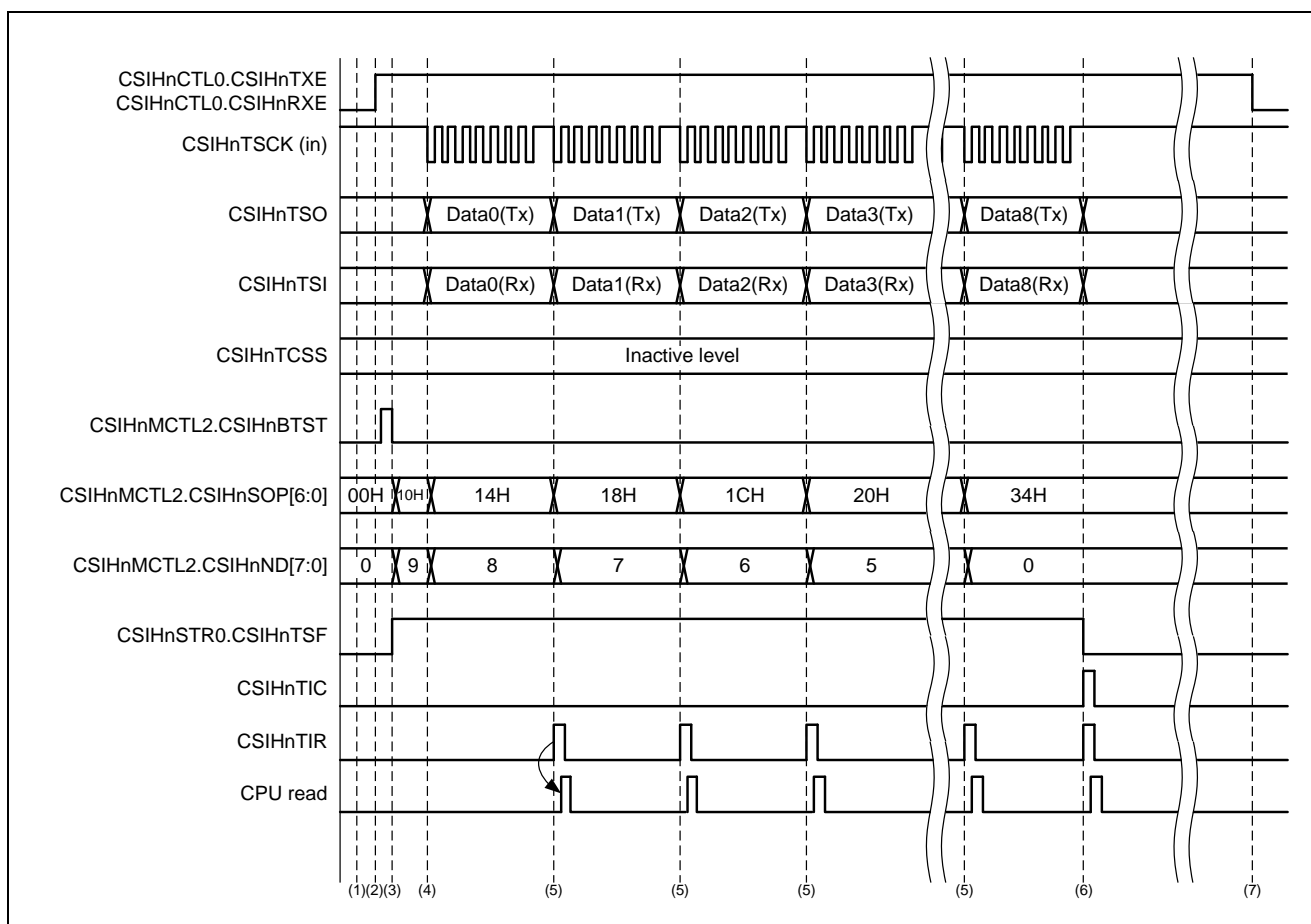


Figure 17.46 Transmit-Only Buffer Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)

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)

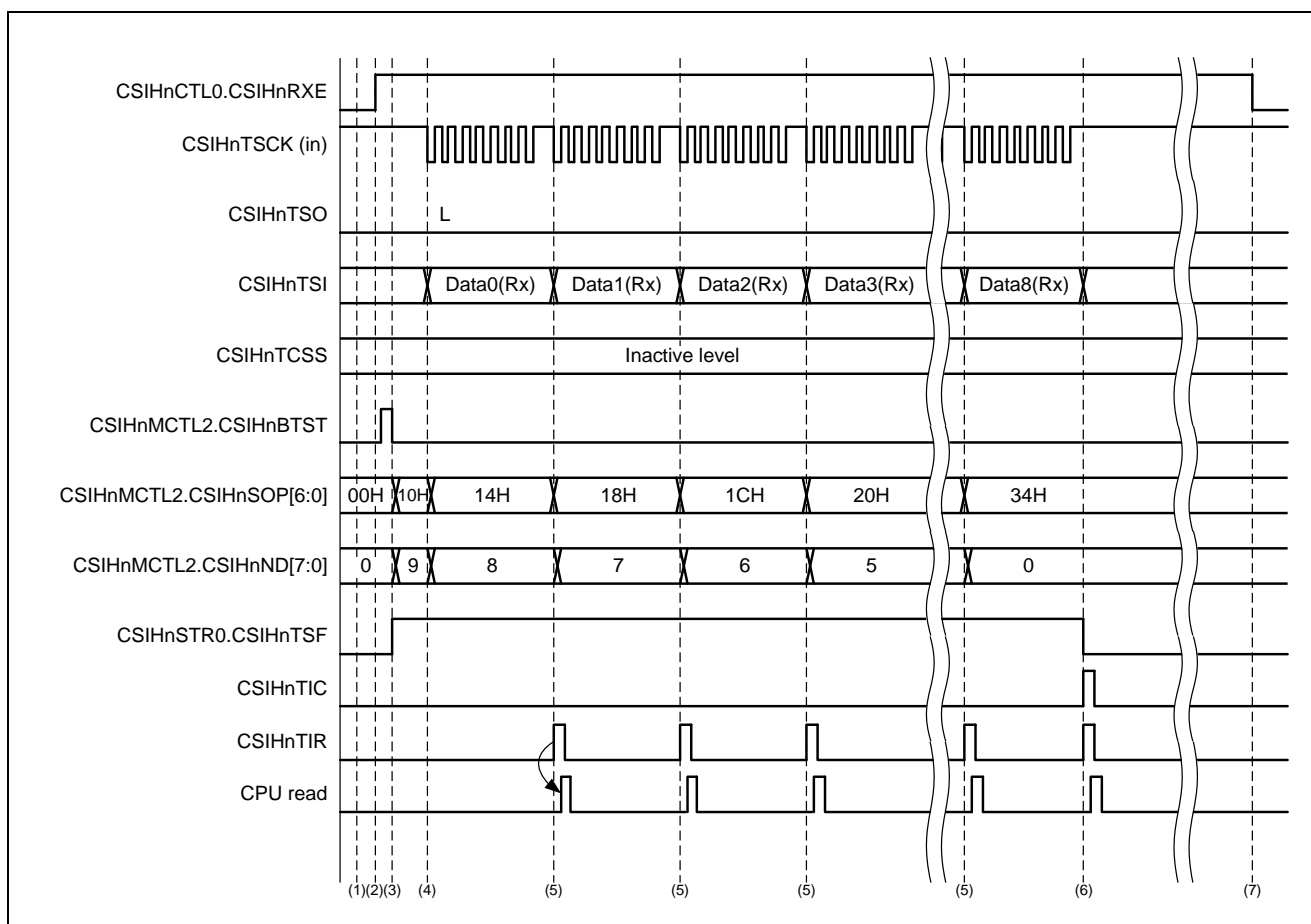


Figure 17.47 Transmit-Only Buffer Mode (for Reception in Slave Mode, and when Job Mode is Disabled)

- 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)

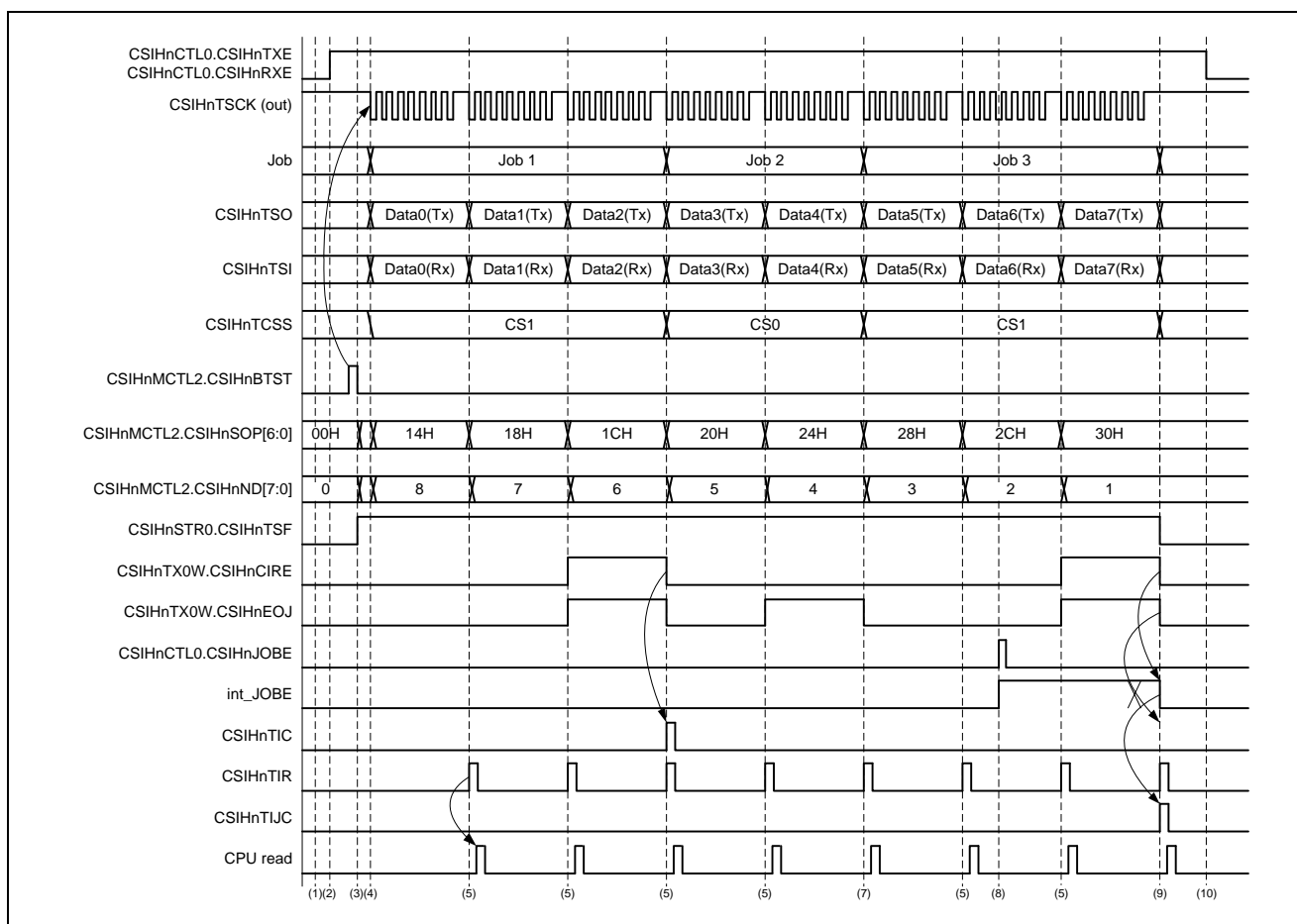


Figure 17.48 Transmit-Only Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)

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.CSIHnJOB3, 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)

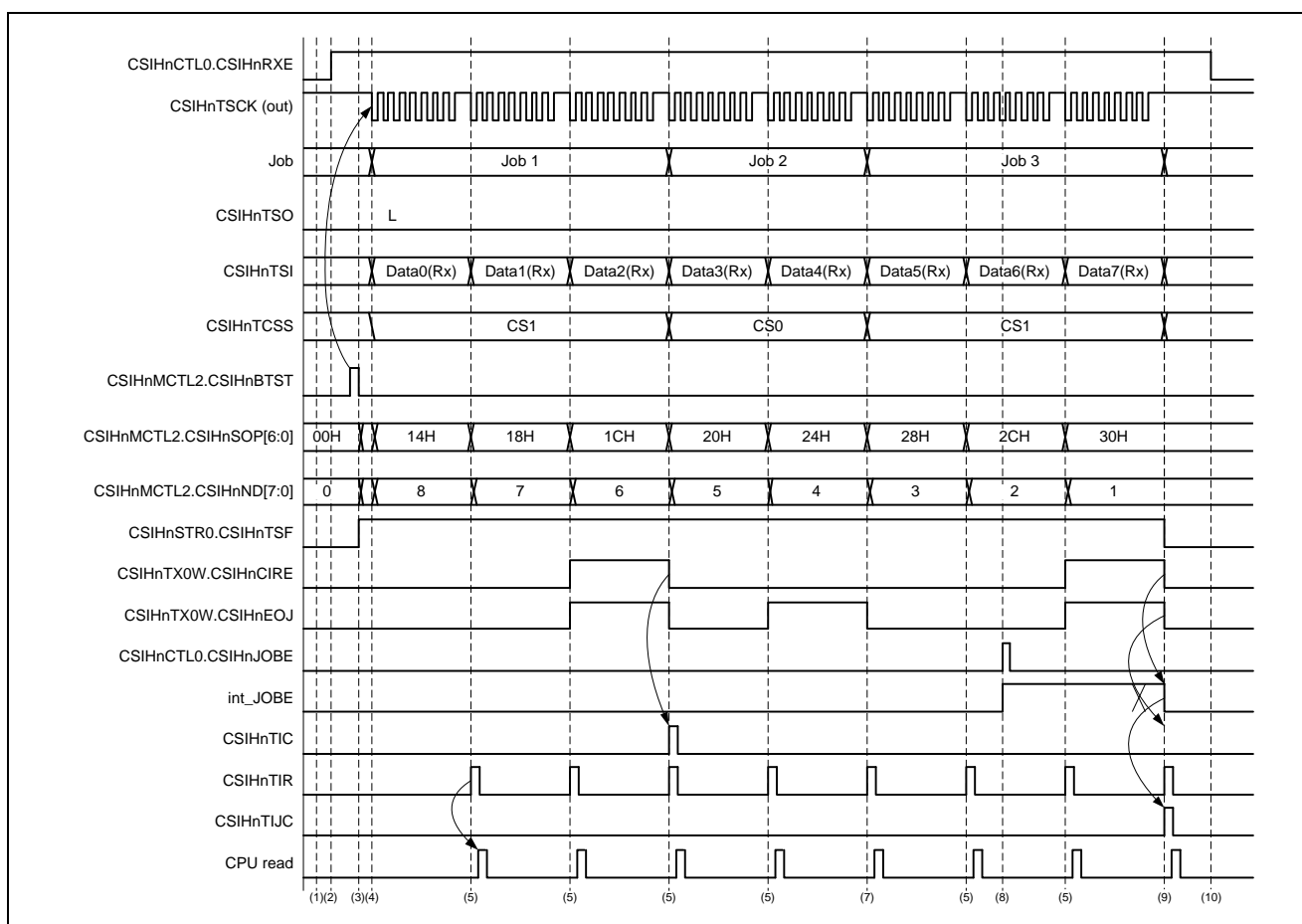


Figure 17.49 Transmit-Only Buffer Mode (for Reception in Master Mode, and when Job Mode is Enabled)

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.

17.5.3 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)

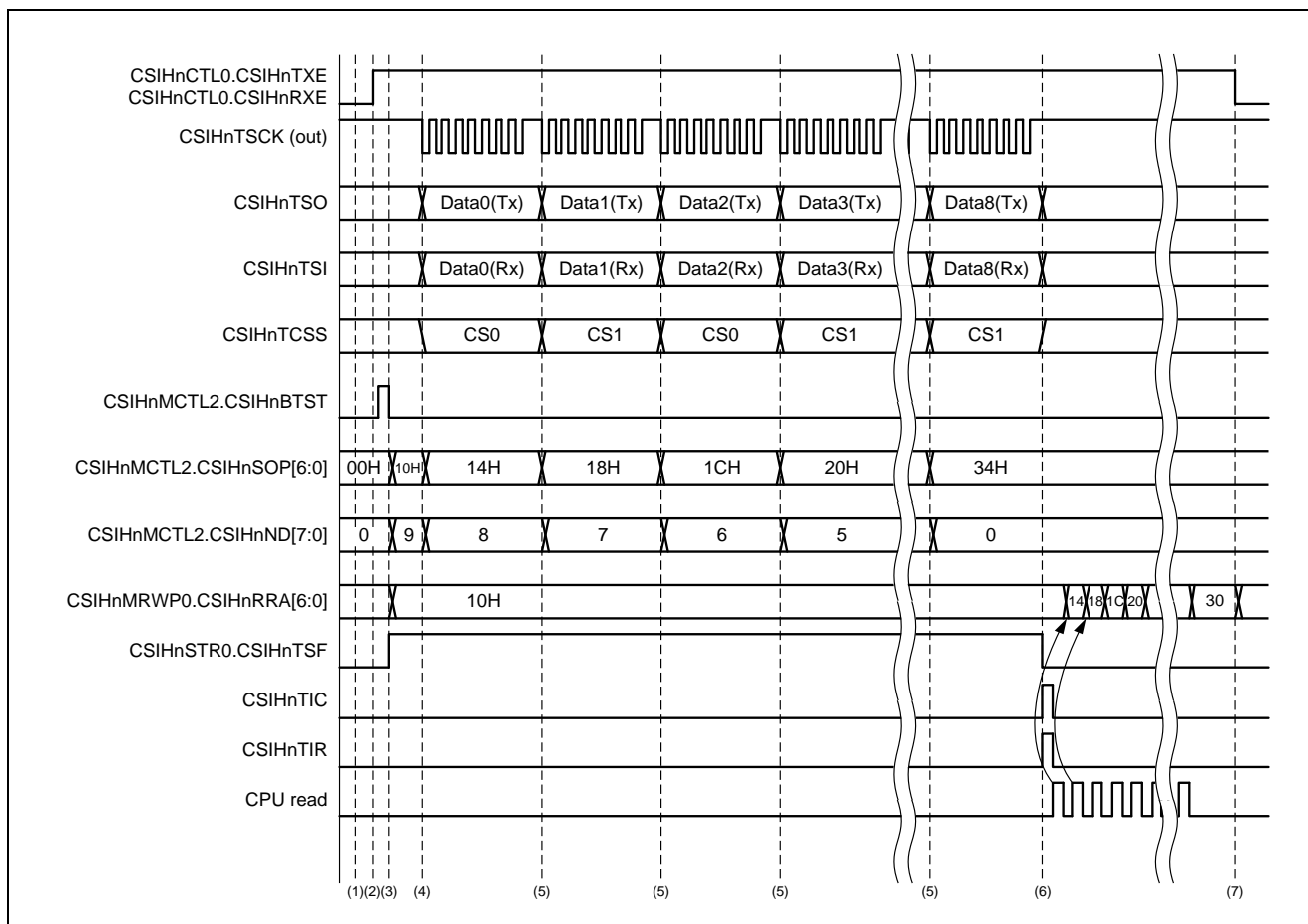


Figure 17.50 Dual Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)

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)

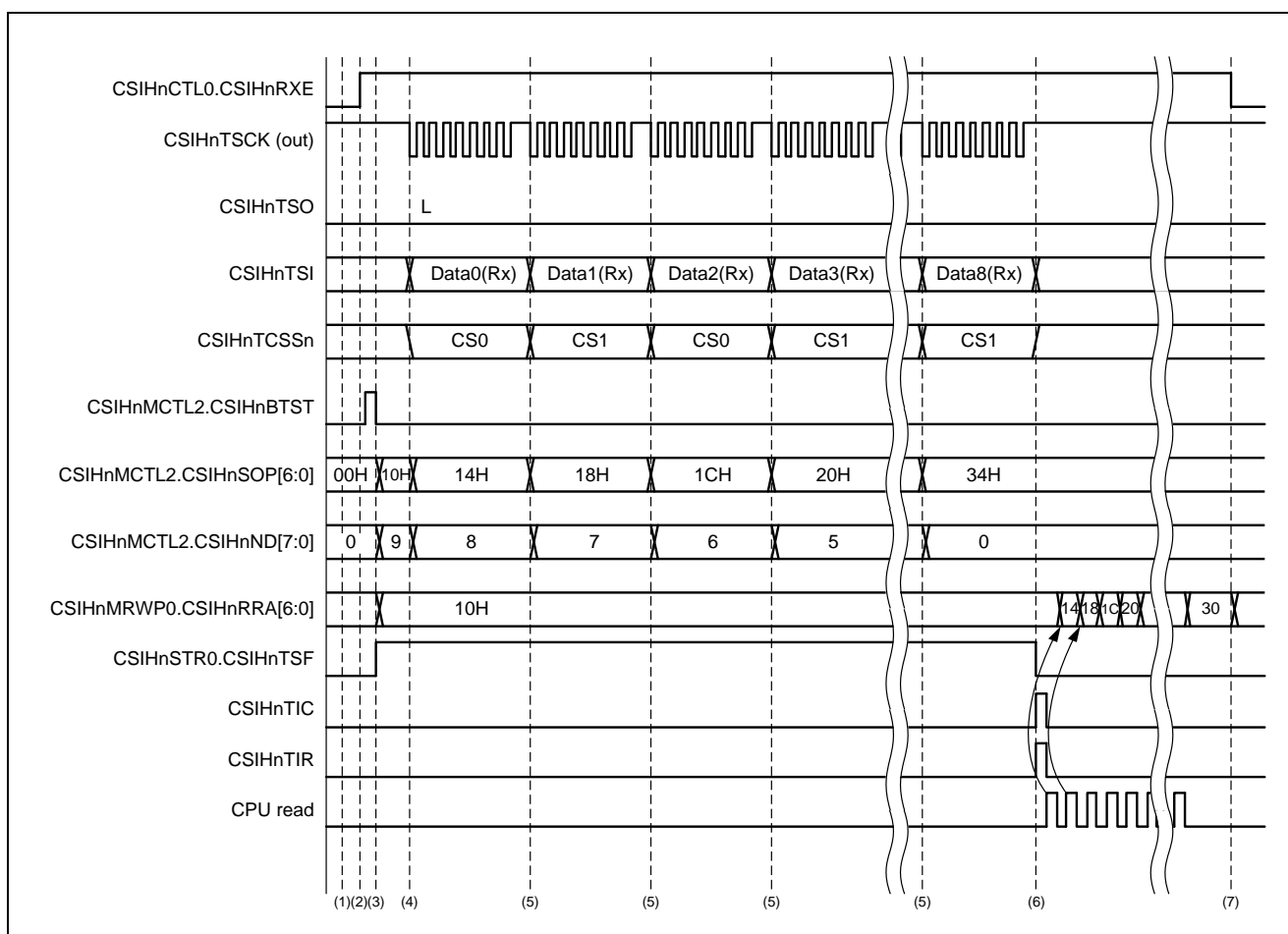


Figure 17.51 Dual Buffer Mode (for Reception in Master Mode, and when Job Mode is Disabled)

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)

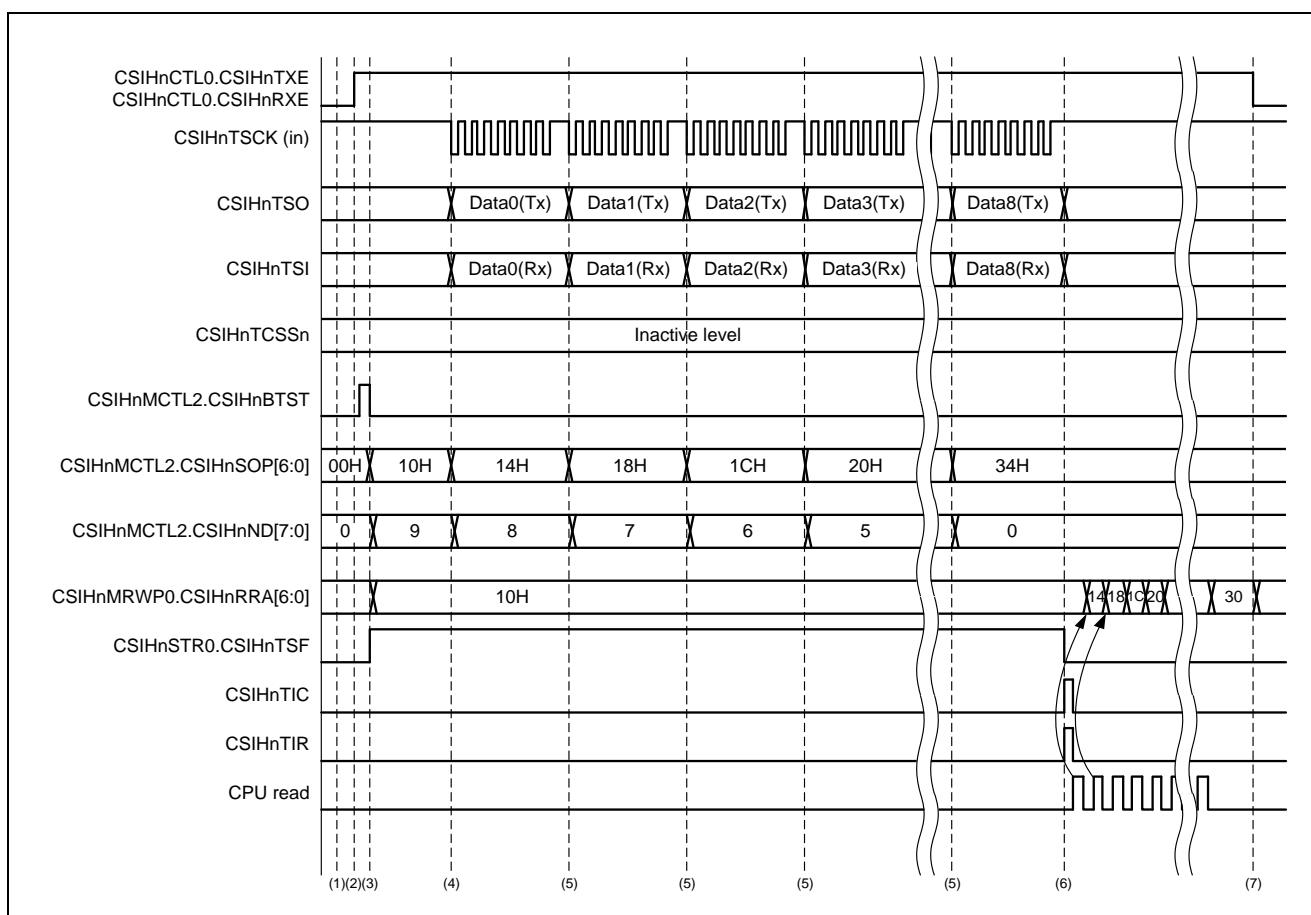


Figure 17.52 Dual Buffer Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)

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)

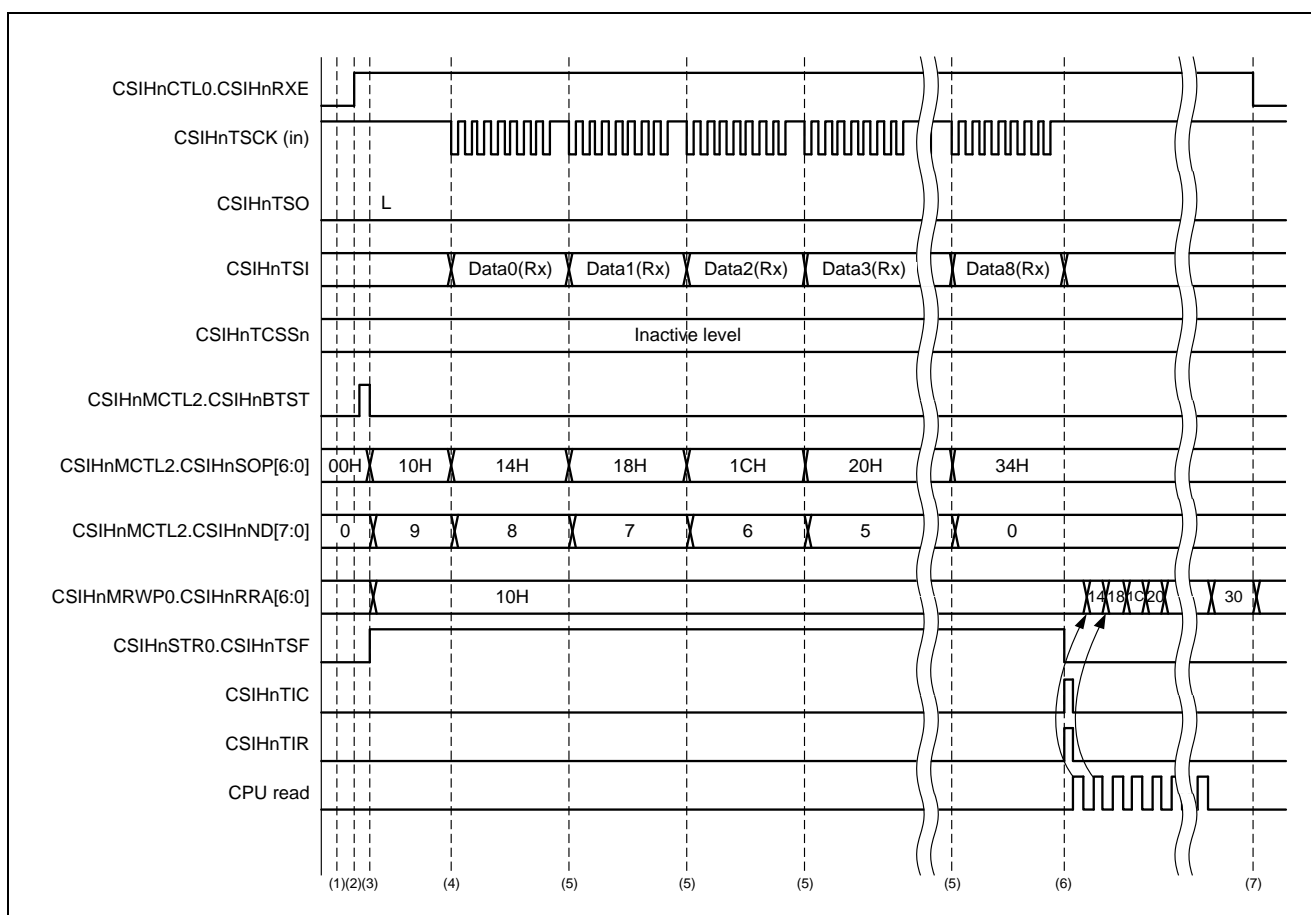


Figure 17.53 Dual Buffer Mode (for Reception in Slave Mode, and when Job Mode is Disabled)

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)

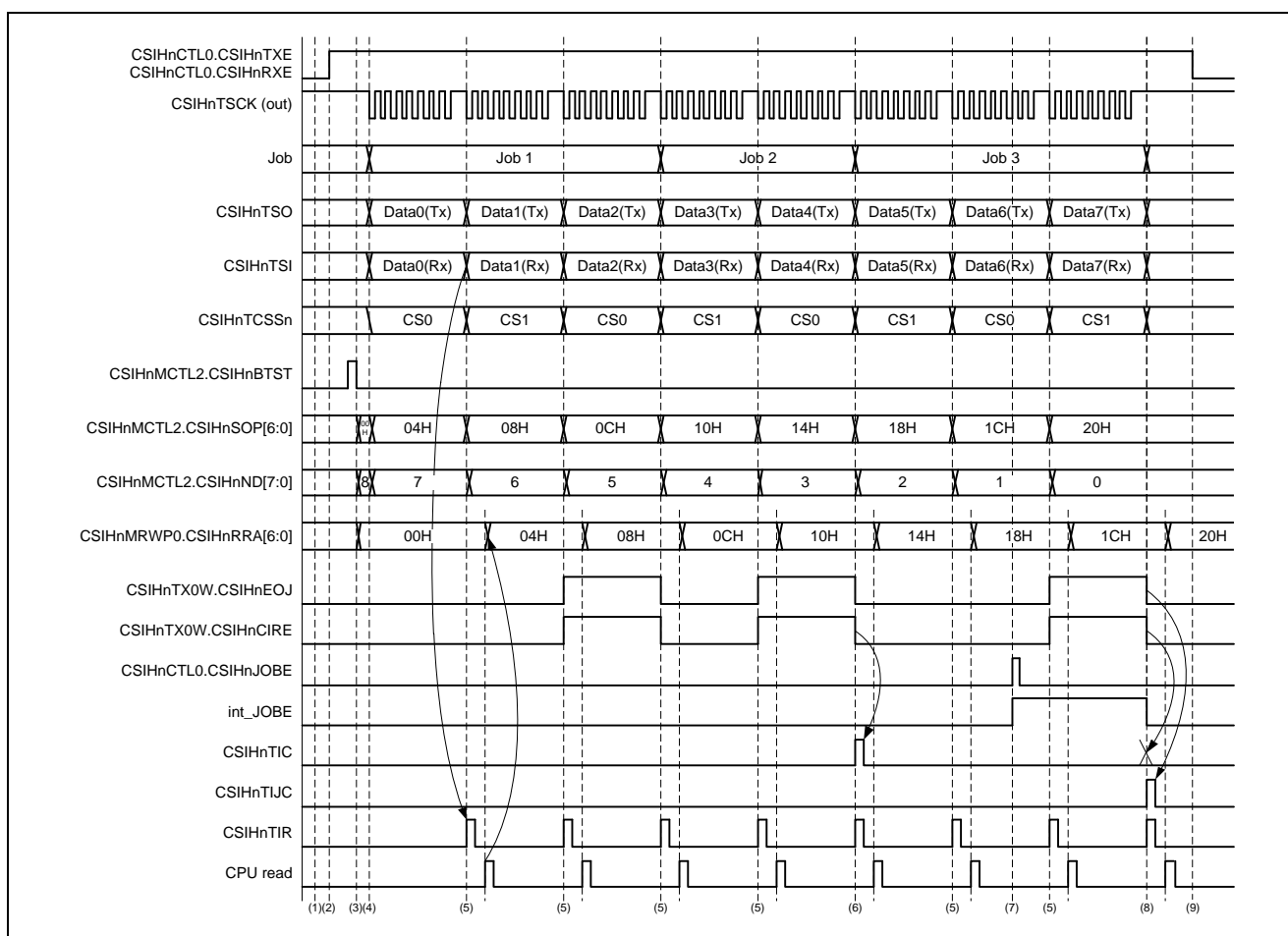


Figure 17.54 Dual Buffer Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)

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_JOB 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)

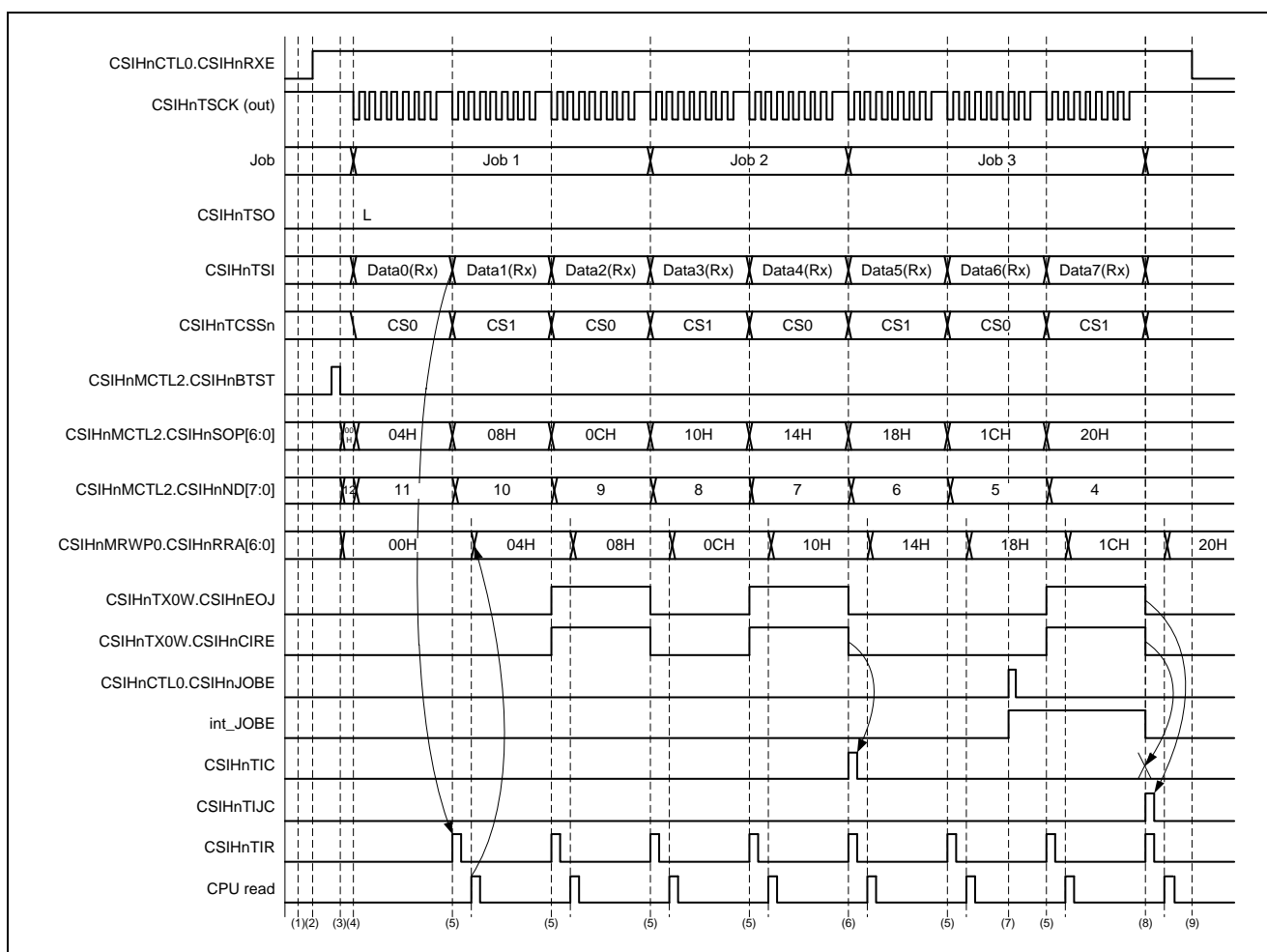


Figure 17.55 Dual Buffer Mode (for Reception in Master Mode, and when Job Mode is Enabled)

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_JOB signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOB 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.CSIHnJOB 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.

17.5.4 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)

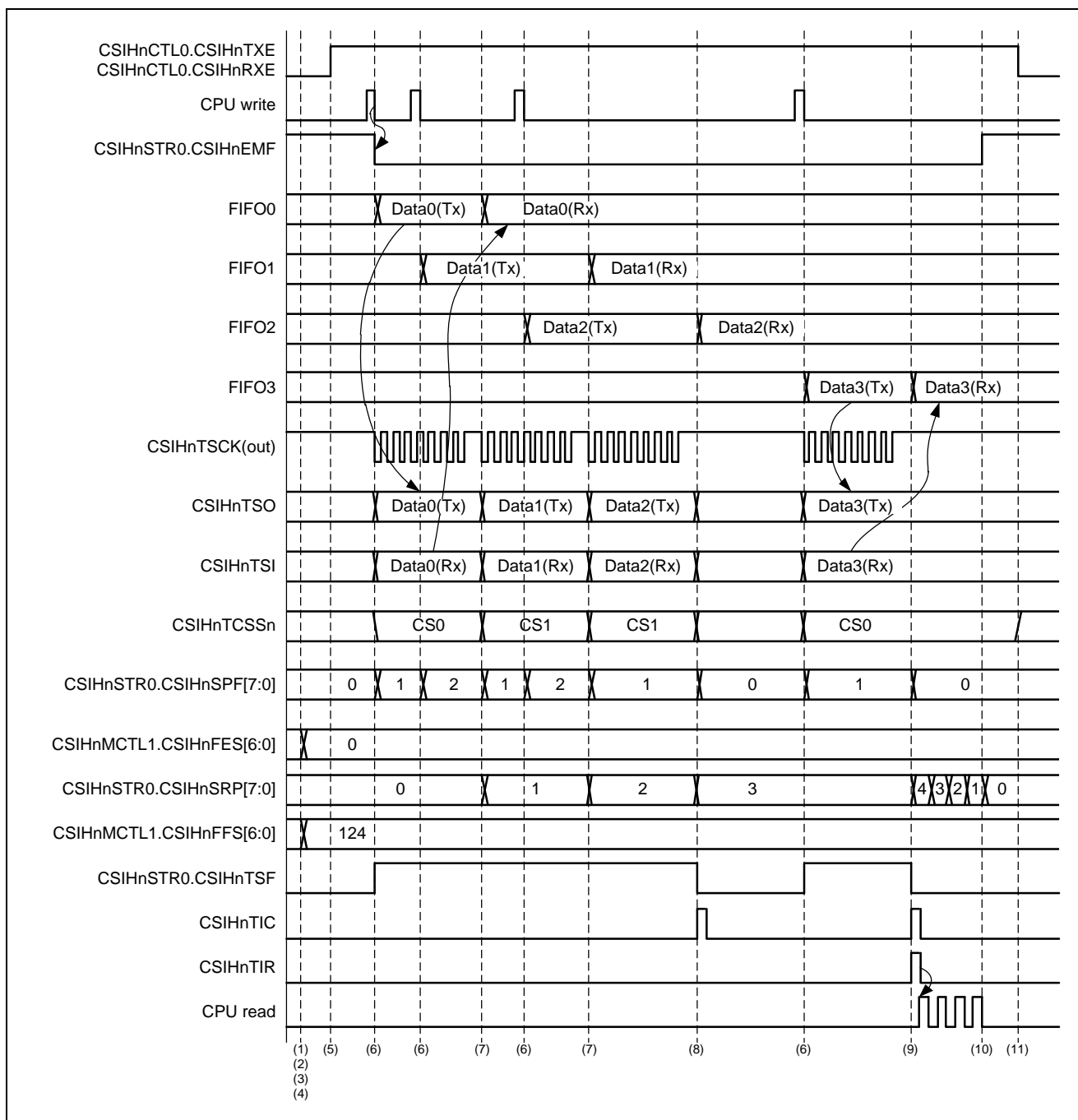


Figure 17.56 FIFO Mode (for Transmission/Reception in Master Mode, and when Job Mode is Disabled)

- 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)

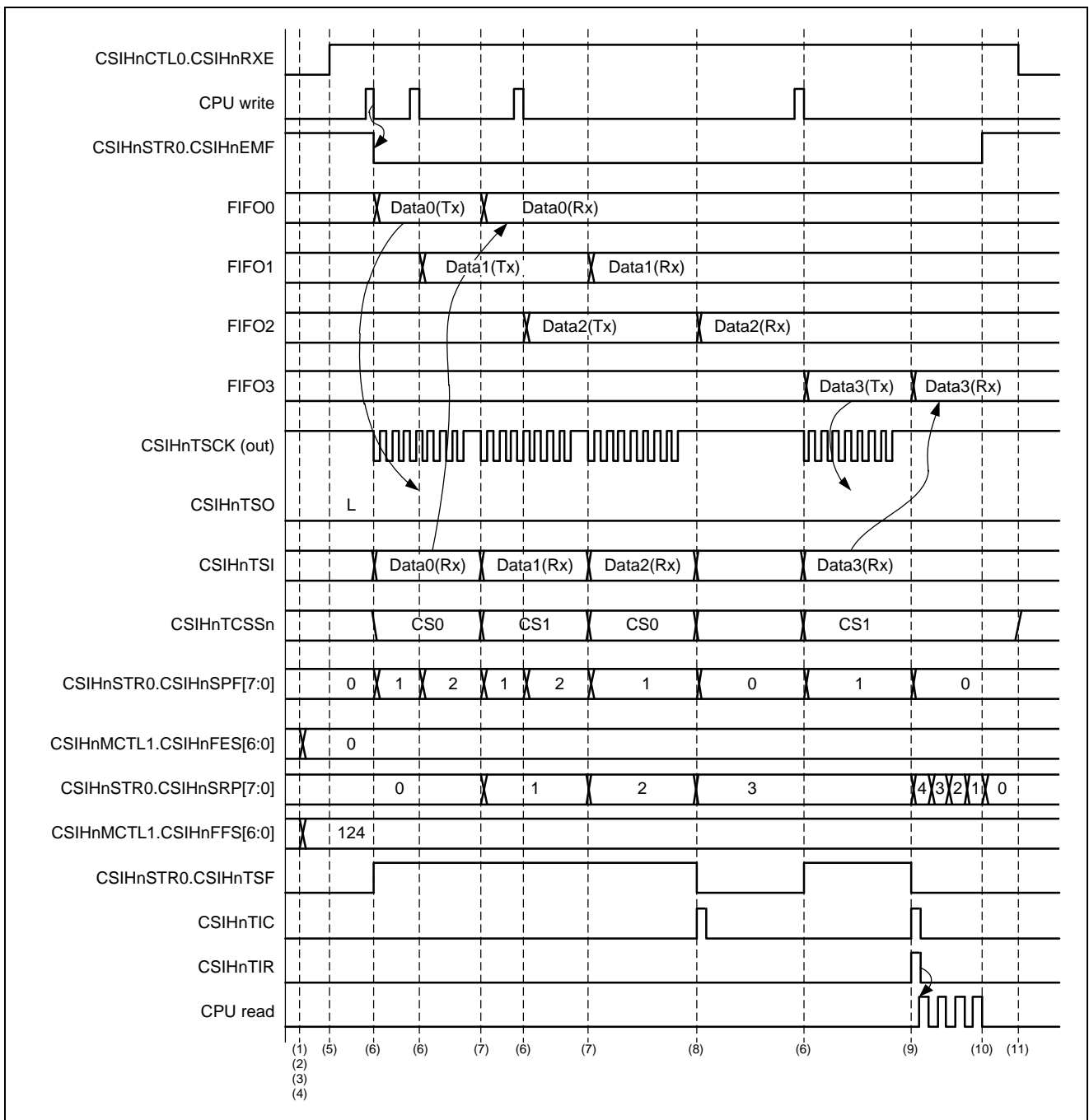


Figure 17.57 FIFO Mode (for Reception in Master Mode, and when Job Mode is Disabled)

- 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)

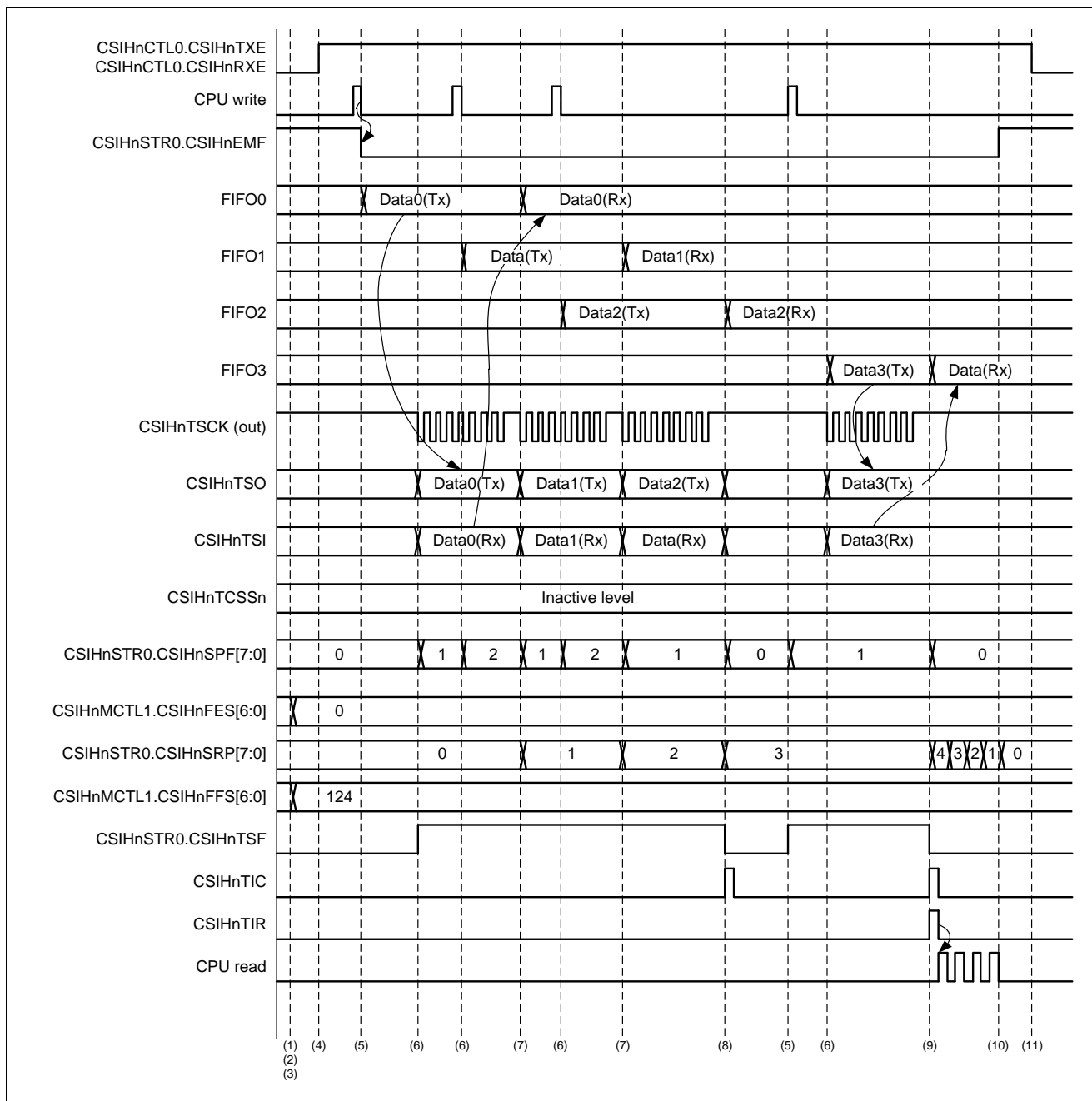


Figure 17.58 FIFO Mode (for Transmission/Reception in Slave Mode, and when Job Mode is Disabled)

- 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)

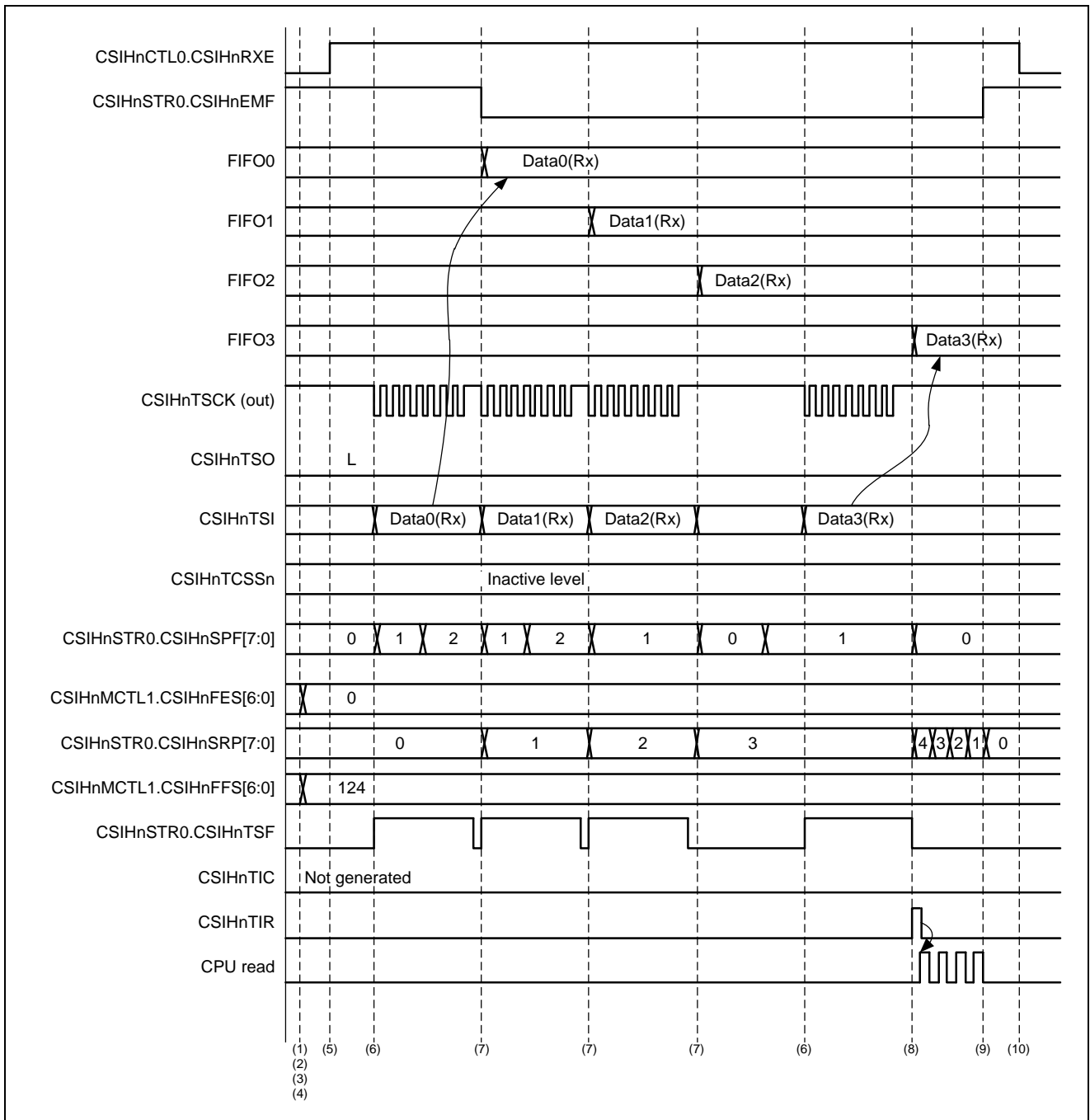


Figure 17.59 FIFO Mode (for Reception in Slave Mode, and when Job Mode is Disabled)

- 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

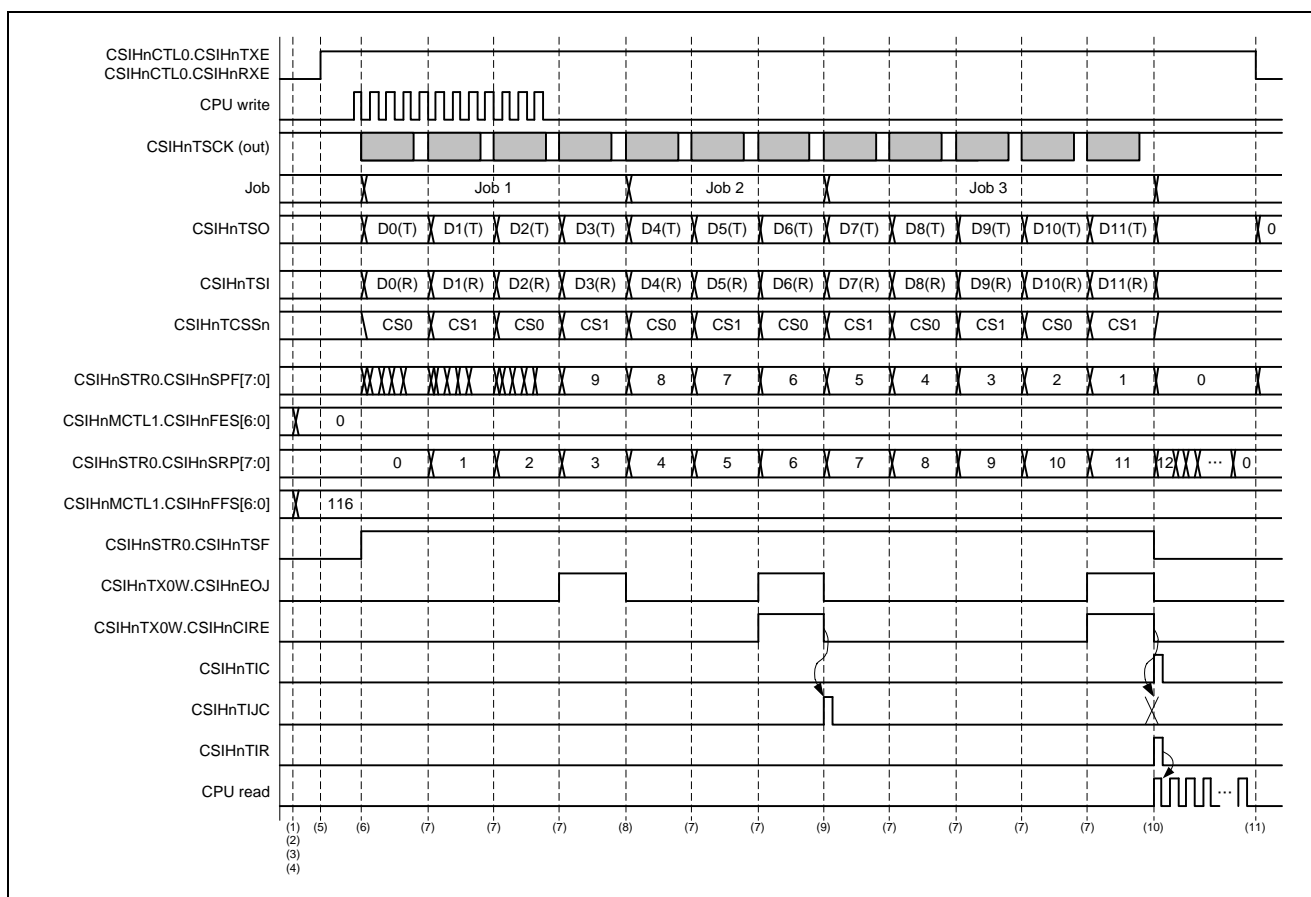


Figure 17.60 FIFO Mode (for Transmission/Reception in Master Mode, and when Job Mode is Enabled)

Remark: The int_JOB 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

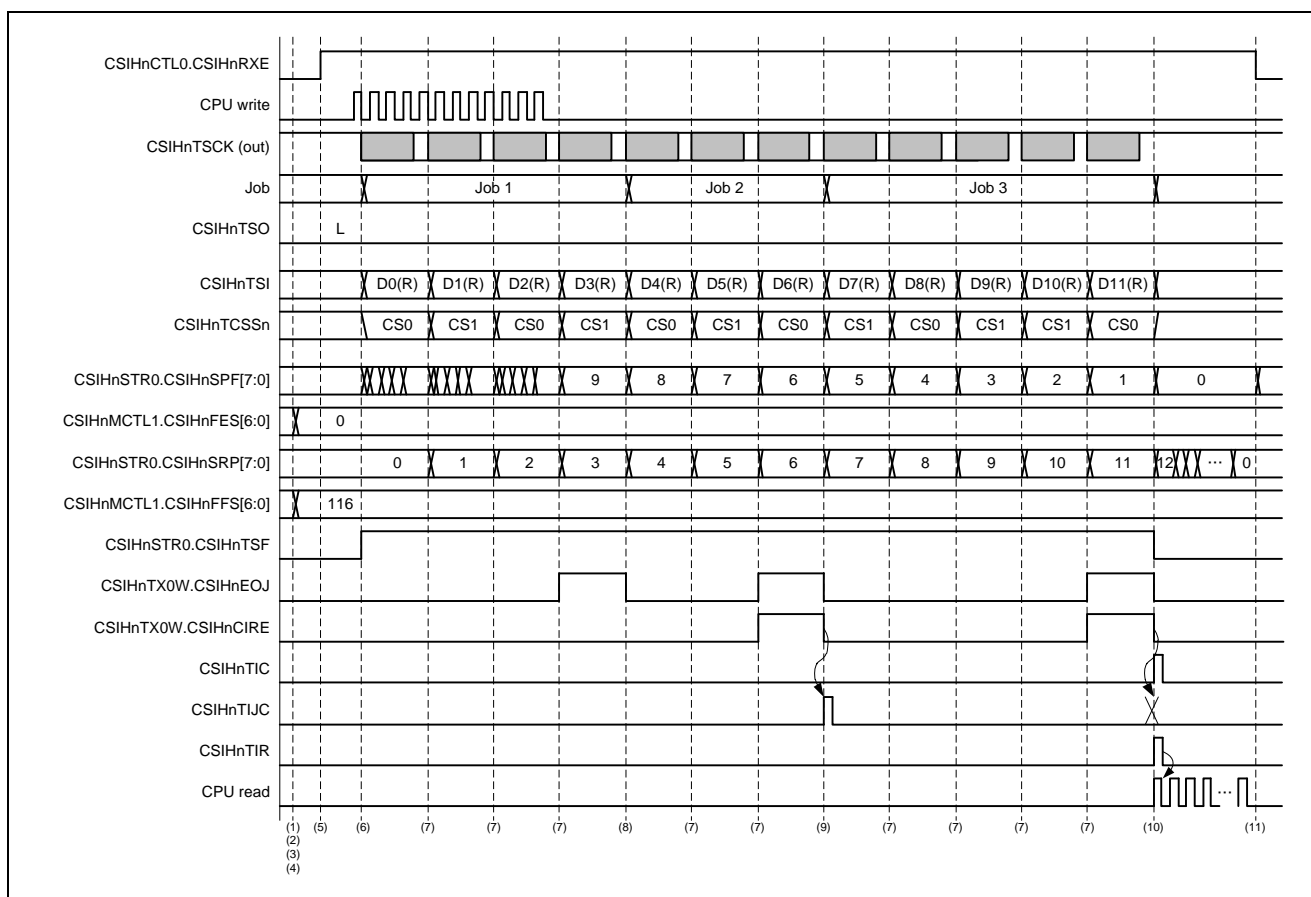


Figure 17.61 FIFO Mode (for Reception in Master Mode, and when Job Mode is Enabled)

Remark: The int_JOB 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.

18. I²C BUS (IICB)

This section describes the I²C bus (IICB).

18.1 Features of IICB

- Number of channels: R-IN32M3 products incorporate two channels of I²CB (IICBn).

Table 18.1 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), and for updating the real-time port pins (RP00-RP37).

Table 18.2 IICBn Interrupts and Requests for Peripheral Modules

Interrupt Request Signal	Function	Connected to:
IICB0		
INTIICB0TIA	Data transmit/receive interrupt request signal	<ul style="list-style-type: none"> • Interrupt controller INTIICB0TIA • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTIICB0TIS	Status interrupt request signal	<ul style="list-style-type: none"> • Interrupt controller INTIICB0TIS • HW-RTOS (Hardware ISR)
IICB1		
INTIICB1TIA	Data transmit/receive interrupt request signal	<ul style="list-style-type: none"> • Interrupt controller INTIICB1TIA • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTIICB1TIS	Status interrupt request signal	<ul style="list-style-type: none"> • Interrupt controller INTIICB1TIS • HW-RTOS (Hardware ISR)

18.2 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 width at high and low levels of the serial clock signal (SCLn) can be changed.
- Automatic detection: The start and stop conditions can be detected automatically.

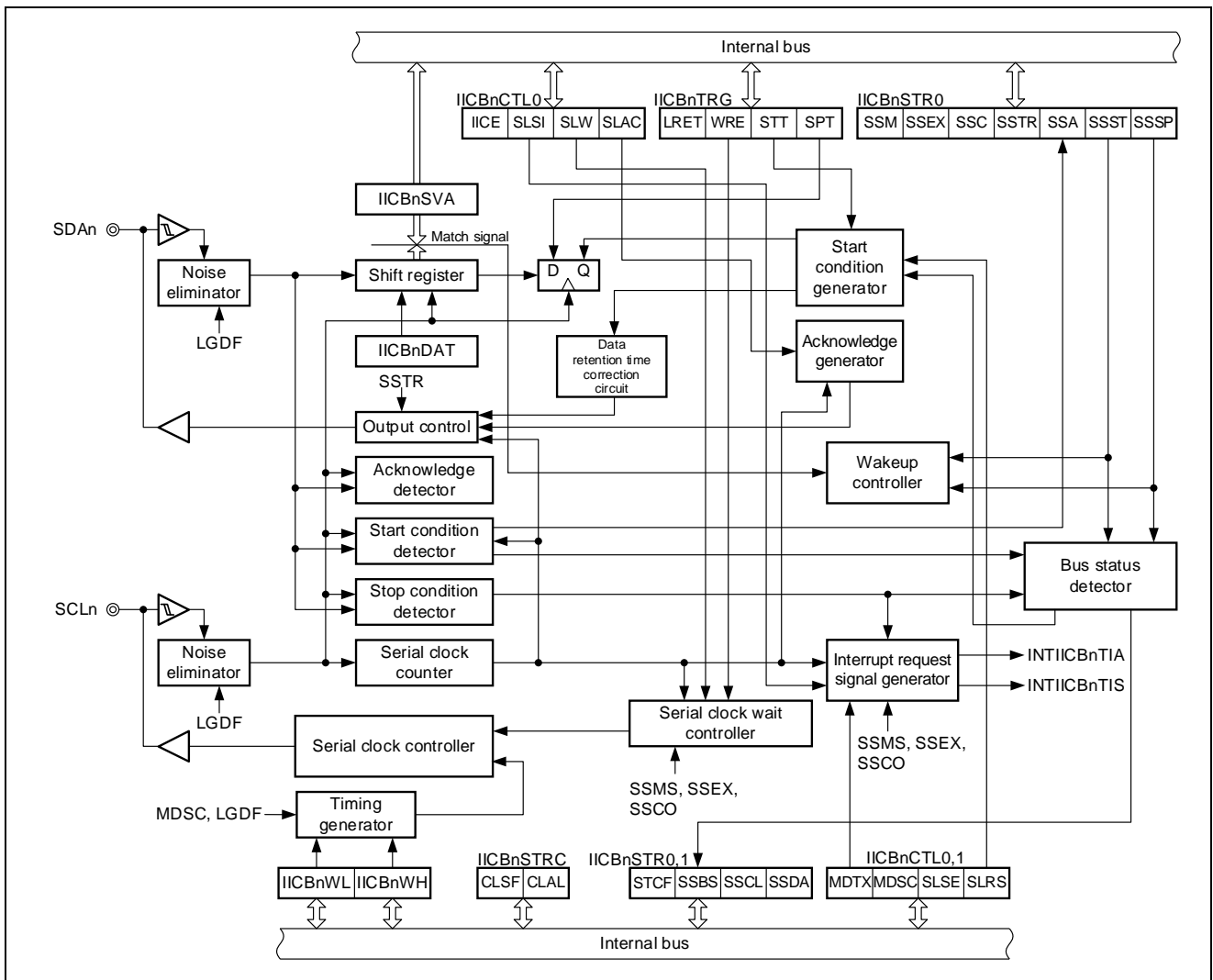


Figure 18.1 Block Diagram of IICBn

18.3 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 18.6.5, Extension Code.

I2Cn is controlled and operated by means of the following registers.

Table 18.3 I²C Register

Register	Shortcut	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 or 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 x 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 18.9, Setting Procedure. The IICBn exits the wait state by performing access to the IICBnDAT register. <ul style="list-style-type: none"> • In single transfer mode <ul style="list-style-type: none"> - When write access to the IICBnDAT register is performed • In continuous transfer mode <ul style="list-style-type: none"> - 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 or 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 x 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 or written in 8- or 1-bit units.

(1/3)

									Address	Initial Value								
IICBnCTL0	7	IICBn IICE	6	0	5	0	4	IICBn MDTX1	3	IICBn MDTX0	2	IICBn SLSI	1	IICBn SLWT	0	IICBn SLAC	4000 0508H +100H × n	00H
	R/W	R/W	0	0	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
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. <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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 18.6, Operation.

Remark: n = 0, 1

(2/4)

Bit Position	Bit Name	Function
3	IICBnMDTX0	<p>Specifies the transfer mode when the address matches between the master and slave.</p> <p>0: Single transfer mode 1: Continuous transfer mode</p> <ul style="list-style-type: none"> • 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). <p>For the operation in each mode, refer to section 18.6, Operation.</p> <p>Caution: Changing the value of this bit is only allowed while IICBnCTL0.IICBnIICE is 0.</p>
2	IICBnSLSI	<p>Enables/disables status interrupt request signal (INTIICBnTIS) output when a stop condition is detected.</p> <p>0: Disables INTIICBnTIS signal output when a stop condition is detected. 1: Enables INTIICBnTIS signal output when a stop condition is detected.</p> <p>Set this bit to 1 when performing the following types of communication.</p> <ul style="list-style-type: none"> - 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	<p>Controls a wait and interrupt request output timing.</p> <p>0: The IICBn enters the wait state and an interrupt request is output at the falling edge of the 8th clock during single transfer.</p> <p>1: The IICBn enters the wait state and an interrupt request is output at the falling edge of the 9th clock during single transfer.</p> <p>The IICBnCTL0.IICBnSLWT bit controls wait state transition and interrupt request output at the following timing.</p> <ul style="list-style-type: none"> - 8th and 9th clocks during data transfer <p>For the conditions for transition to the wait state, refer to section 18.6.4, Entering and Exiting Wait State.</p> <p>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.</p> <ul style="list-style-type: none"> • In single transfer mode <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> 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. <p>Caution: In single transfer mode, changing the value of this bit is only allowed while IICBnCTL0.IICBnIICE is 0 or during the wait period.</p>

Remark: n = 0, 1

(4/4)

Bit Position	Bit Name	Function
0	IICBnSLAC	<p>Controls acknowledge signal output.</p> <p>0: Disables acknowledge signal output.</p> <p>Master: The acknowledge signal is not output during data reception (SDAn = "H").</p> <p>Slave: The acknowledge signal is not output during data transfer when an address match occurs (SDAn = "H").</p> <p>1: Enables acknowledge signal output.</p> <p>Master: The acknowledge signal is output during data reception (SDAn = "L").</p> <p>Slave: The acknowledge signal is output during data transfer when an address match occurs (SDAn = "L").</p> <p>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").</p> <p>Also, no acknowledge signal is output (SDAn = "H") while the IICBn is transmitting data or when it does not participate in communications.</p>

Remark: n = 0, 1

(4) IICBn control register 1 (IICBnCTL1)

This register controls operation of IICBn.

- Access This register can be read or 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 x 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

Remark: n = 0, 1

(2/2)

Bit Position	Bit Name	Function
3	IICBnMDLB	<p>Specifies the loop back mode.</p> <p>0: Do not loop back. 1: Loop back.</p> <p>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).</p> <p>The output SCLn and SDAn immediately before output will be looped back.</p> <p>Note that both SCLn and SDAn are at the high level if the IICBnCTL1.IICBnMDLB bit is "1".</p>
2	—	Reserved. When writing to this bit, write 0. When read, 0 is returned.
1	IICBnSLSE	<p>Enables/disables start condition output in the initial communication state.</p> <p>0: Disables start condition output in the initial communication state. 1: Enables start condition output in the initial communication state.</p> <p>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).</p> <p>The IICBnCTL1.IICBnSLSE bit is automatically cleared to 0 upon detection of a start condition (even without a 0 write operation).</p> <p>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.</p>
0	IICBnSLRS	<p>Enables/disables the communication reserve function.</p> <p>0: Enables communication reserve function. 1: Disables communication reserve function.</p> <p>Communication reserve function enabled state:</p> <p>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.</p> <p>Whether the communication reserve state is set can be confirmed by checking the IICBnSTR0.IICBnSSRS bit.</p> <p>Communication reserve function disabled state:</p> <p>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.</p>

Remark: n = 0, 1

(5) IICBn low level width setting register (IICBnWL)

This register is used to set the width at low level of the serial clock signal (SCLn).

- Access This register can be read or 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	IICBnWL9	IICBnWL8	IICBnWL7	IICBnWL6	IICBnWL5	IICBnWL4	IICBnWL3	IICBnWL2	IICBnWL1	IICBnWL0	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 (the width at low level the SCLn clock signal) of the I ² C bus specification. The value of the IICBnWL register is used to determine the serial output timing of other I ² C bus specifications. For the serial output timing setting conditions, see Table 18.4, Conditions for Generating Serial Output Timing.

(6) IICBn high-level width setting register (IICBnWH)

This register is used to set the width at high level of the serial clock signal (SCLn).

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

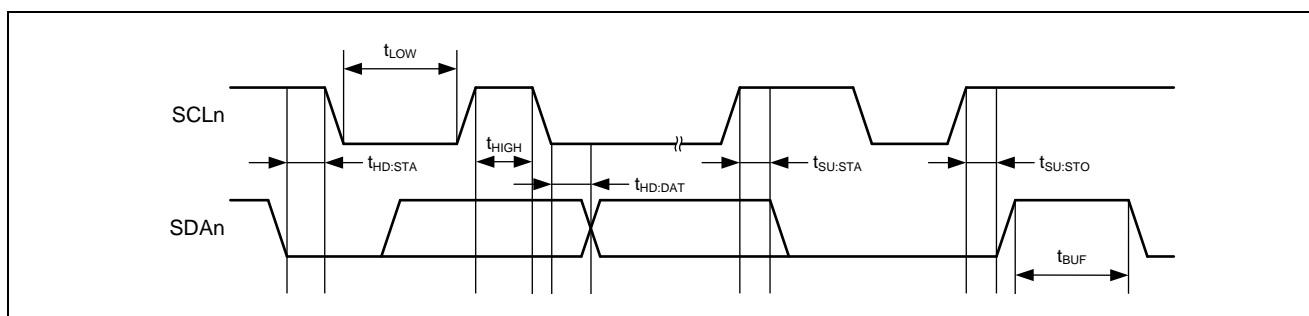
Caution: Write access to the IICBnWH register is prohibited when the value of the IICBnCTL0.IICBnIICE bit is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
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	Address 4000 0528H +100H x n	Initial Value 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 tHIGH period (the width at high level of the SCLn clock signal) of the I ² C bus specification. The value of the IICBnWH register is used to determine the serial output timing of other I ² C bus specifications. For the serial output timing setting conditions, refer to Table 18.4, Conditions for Generating Serial Output Timing.

Table 18.4 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 I²C bus specifications can be set by setting the IICBnWL register and IICBnWH register.

[Setting transfer clock on master side]

$$\text{Transfer clock (Hz)} = \frac{\text{PCLK}}{(\text{IICBnWL} + \text{IICBnWH}) + \text{PCLK}(t_R + t_F)}$$

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 IICBWL 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 or written in 8- or 1-bit units.

(1/5)

IICBnTRG	7	6	5	4	3	2	1	0	Address 4000 050CH +100H × n	Initial Value 00H
	0	0	0	0	IICBn LRET	IICBn WRET	IICBn STT	IICBn SPT		
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	<p>Communication exit trigger bit</p> <p>0: The read value is always 0, and writing 0 is ignored.</p> <p>1: The IICBn exits the current communication and enters the wait state. This bit is automatically cleared to 0 following execution.</p> <p>The following occurs when IICBnTRG.IICBnLRET is 1.</p> <ul style="list-style-type: none"> - 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. <p>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.</p> <p>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.</p>

Remark: n = 0, 1

(2/5)

Bit Position	Bit Name	Function
2	IICBnWRET	<p>This is the trigger bit for exiting the wait state.</p> <p>0: Does not exit the wait state.</p> <p>1: Exits the wait state and resumes communication. This bit is automatically cleared following execution.</p> <p>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.)</p> <p>If the IICBn is not in the wait state (IICBnSTR0.IICBnSSWT = 0), setting this bit to 1 has no meaning.</p> <p>There are other conditions for exiting the wait state in addition to the setting of this bit. For details, refer to section 18.6.4, Entering and Exiting Wait State.</p>
1	IICBnSTT	<p>Start condition trigger bit</p> <p>0: Does not output a start condition.</p> <p>1: Outputs a start condition (This bit is automatically cleared to 0 after it has been set to 1.</p> <p>The IICBnTRG.IICBnSTT bit can be set to 1 under the following conditions:</p> <p>[1] IICBnSTR0.IICBnSSMS bit = Master state (1)</p> <ul style="list-style-type: none"> • Single transfer mode <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> - 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 <p>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.</p> <p>Next, SCLn is set to the low level after the start condition hold time has elapsed.</p> <p>For the individual time settings, see Table 18.4, Conditions for Generating Serial Output Timing.</p>

Remark: n = 0, 1

(3/5)

Bit Position	Bit Name	Function
1	IICBnSTT	<p>[2] Slave state or communication wait state (IICBnSTR0.IICBnSSMS = 0)</p> <ul style="list-style-type: none"> 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 18.4, 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. <ul style="list-style-type: none"> - 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 18.4, 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. <p>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.</p>

Remark: n = 0, 1

(4/5)

Bit Position	Bit Name	Function
1	IICBnSTT	<p>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.</p> <ul style="list-style-type: none"> - 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.) <p>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.</p> <p>2. Setting the IICBnTRG.IICBnSTT bit at the same time as the IICBnTRG.IICBnSPT bit is prohibited.</p>

Remark: n = 0, 1

(5/5)

Bit Position	Bit Name	Function
0	IICBnSPT	<p>Stop condition trigger bit</p> <p>0: Does not output a stop condition. 1: Outputs a stop condition (This bit is automatically cleared after it has been set to 1).</p> <p>The IICBnTRG.IICBnSPT bit can be set to 1 under the following conditions while the IICBn is performing communication as a master.</p> <ul style="list-style-type: none"> • Single transfer mode <ul style="list-style-type: none"> - 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 <p>The IICBnTRG.IICBnSPT bit can be set to 1 in the following states.</p> <ul style="list-style-type: none"> - 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 <p>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 tSU: STO time, a stop condition is output by setting SDAn to high level.</p> <p>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.</p> <ul style="list-style-type: none"> - 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 <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;"> <p>Cautions 1. Setting the IICBnTRG.IICBnSPT bit to 1 is prohibited during slave operation (IICBnSTR0.IICBnSSMS bit = 0)</p> <p>2. Setting the IICBnTRG.IICBnSPT bit to 1 at the same time as the IICBnTRG.IICBnSTT bit is prohibited.</p> </div>

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	IICBnSSMS	0	IICBnSSDR	IICBnSSWT	IICBnSSEX	IICBnSSCO	IICBnSSTR	IICBnSSAC	IICBnSSRS	IICBnSSBS	IICBnSSST	IICBnSSSP	0	0	IICBnSTCF	IICBnALDF	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	<p>IICBnDAT register status flag</p> <p>1: Indicates that data in the IICBnDAT register remains unprocessed.</p> <p>During reception operation: Received data remains unread in the IICBnDAT register.</p> <p>During transmission operation: Data written to the IICBnDAT register has not been transferred to the shift register.</p> <p>Setting conditions:</p> <ul style="list-style-type: none"> - 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 <p>Clearing conditions:</p> <ul style="list-style-type: none"> • Clearing conditions given priority over setting conditions <ul style="list-style-type: none"> - 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)) <ul style="list-style-type: none"> - 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	<p>Wait state flag</p> <p>1: Indicates that the IICBn is in the wait state.</p> <p>Setting conditions:</p> <p>[In single transfer mode]</p> <p><Common to master/slave></p> <ul style="list-style-type: none"> - 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 <p><Master></p> <ul style="list-style-type: none"> - 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 <p><Slave></p> <ul style="list-style-type: none"> - Upon detection of the falling edge of the 9th clock during address transfer when an address match occurred <p>[In continuous transfer mode]</p> <p><During data transfer period, common to master/slave></p> <ul style="list-style-type: none"> • During data transmission, when the data to be transmitted next has not been written <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> - 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 (continued)	<p><During address transfer period, operating as master></p> <ul style="list-style-type: none"> - 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) <p><During address transfer period, operating as slave></p> <ul style="list-style-type: none"> - Upon detection of the falling edge of the 9th clock while IICBnSTR0.IICBnSSSTR bit is 0 during address transfer when an address match occurred - Upon NACK detection <p>Clearing conditions:</p> <ul style="list-style-type: none"> • Clearing conditions given priority over setting conditions <ul style="list-style-type: none"> - 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 (ACK) has been received • Clearing conditions for which setting conditions are given priority <ul style="list-style-type: none"> - 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 <p>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.IICBnSSSTR bit is cleared to 0 and the bus is released (both SCLn and SDAn go into high impedance).</p>

Remark: n = 0, 1

(5/8)

Bit Position	Bit Name	Function
11	IICBnSSEX	<p>Expansion code reception detection flag</p> <p>1: Indicates that an expansion code has been received.</p> <p>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</p> <p>Clearing conditions:</p> <ul style="list-style-type: none"> - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - Upon detection of a start condition <p>Caution: When the expansion codes match, the processing after the interrupt differs according to the ensuing data, and therefore is dependent on software processing.</p>
10	IICBnSSCO	<p>Address match detection flag</p> <p>1: Indicates that an address that matches the IICBnSVA register has been detected.</p> <p>Setting condition: Upon detection of the falling edge of the 8th clock while transferring a received address that matches the IICBnSVA register</p> <p>Clearing conditions:</p> <ul style="list-style-type: none"> - 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	<p>Transmission status detection flag</p> <p>1: Indicates that data is being transmitted to the serial data bus.</p> <p>Setting conditions:</p> <p><Master></p> <ul style="list-style-type: none"> - Upon detection of a start condition after 1 is written to the IICBnTRG.IICBnSTT bit <p><Slave></p> <ul style="list-style-type: none"> - Upon detection of the falling edge of the 8th clock following reception of 1 to R/W bit during address transfer when an address match occurred <p>Clearing conditions:</p> <p><Common to master/slave></p> <ul style="list-style-type: none"> - 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 <p><Master></p> <ul style="list-style-type: none"> - Upon detection of the falling edge of the 8th clock following reception of 1 to R/W bit during address transfer - Upon detection of arbitration loss <p><Slave></p> <ul style="list-style-type: none"> - Upon detection of a start (restart) condition
8	IICBnSSAC	<p>Acknowledge (ACK) detection flag</p> <p>1: Indicates that an acknowledge signal has been detected.</p> <p>Setting condition:</p> <p>Upon detection of the falling edge of SCLn when a low level has been received at the ACK bit during participation in communications</p> <p>Clearing conditions:</p> <ul style="list-style-type: none"> - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of the rising edge of SCLn <p>Caution: The value of the IICBnSTR0.IICBnSSAC bit changes regardless of whether or not an interrupt has occurred.</p>
7	IICBnSSRS	<p>Communication reserve state flag</p> <p>0: Not communication reserve state</p> <p>1: Communication reserve state</p> <p>Setting condition:</p> <p>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)</p> <p>Clearing conditions:</p> <ul style="list-style-type: none"> - 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	<p>IICBn bus status flag</p> <p>0: Bus released state (initial communication state when IICBnCTL1.IICBnSLSE = 1)</p> <p>1: Bus communication state (initial communication state when IICBnCTL1.IICBnSLSE = 0)</p> <p>Setting condition:</p> <ul style="list-style-type: none"> - Upon detection of a start condition - When 1 is written to the IICBnCTL0.IICBnIICE bit when IICBnCTL1.IICBnSLSE = 0 <p>Clearing conditions:</p> <p>Upon detection of a stop condition</p> <p>Remark: The IICBnSTR0.IICBnSSBS bit operates whether or not the IICBn is participating in communications.</p>
5	IICBnSSST	<p>Start condition detection flag</p> <p>1: Indicates that a start condition has been detected.</p> <p>Setting condition:</p> <p>Upon detection of a start condition</p> <p>Clearing conditions:</p> <ul style="list-style-type: none"> - 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 <p>Remark: The IICBnSTR0.IICBnSSST bit operates whether or not the IICBn is participating in communications.</p>
4	IICBnSSSP	<p>Stop condition detection flag</p> <p>1: Indicates that a stop condition has been detected.</p> <p>Setting condition:</p> <p>Upon detection of a stop condition</p> <p>Clearing conditions:</p> <p>Upon detection of the falling edge of the first SCLn following start condition detection</p> <p>Remark: The IICBnSTR0.IICBnSSSP bit operates whether or not the IICBn is participating in communications.</p>
3, 2	—	Reserved. These bits are read as 0.

Remark: n = 0, 1

(8/8)

Bit Position	Bit Name	Function
1	IICBnSTCF	<p>IICBnTRG.IICBnSTT bit clear flag</p> <p>1: Indicates that the IICBnTRG.IICBnSTT bit has been cleared because start condition output failed.</p> <p>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)</p> <p>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).</p> <p>Clearing condition: When 1 is written to the IICBnSTRC.IICBnCLSFB bit</p>
0	IICBnALDF	<p>Arbitration loss detection flag</p> <p>1: Indicates that an arbitration loss has been detected.</p> <p>Setting condition: Upon detection of arbitration loss</p> <p>Clearing condition: When 1 is written to the IICBnSTRC.IICBnCLAF bit</p> <p>If a setting condition coincides with a clearing condition, the setting condition takes priority.</p> <p>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.)</p> <p>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.</p>

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 x 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 or written in 8-bit units.

								Address	Initial Value	
	7	6	5	4	3	2	1	0		
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

18.4 IIC Bus Mode Functions

18.4.1 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-ch 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-ch open-drain output for both master and slave devices.

Because the outputs of the serial clock line and serial data bus line are N-ch open-drain outputs, an external pull-up resistor must be connected to these lines.

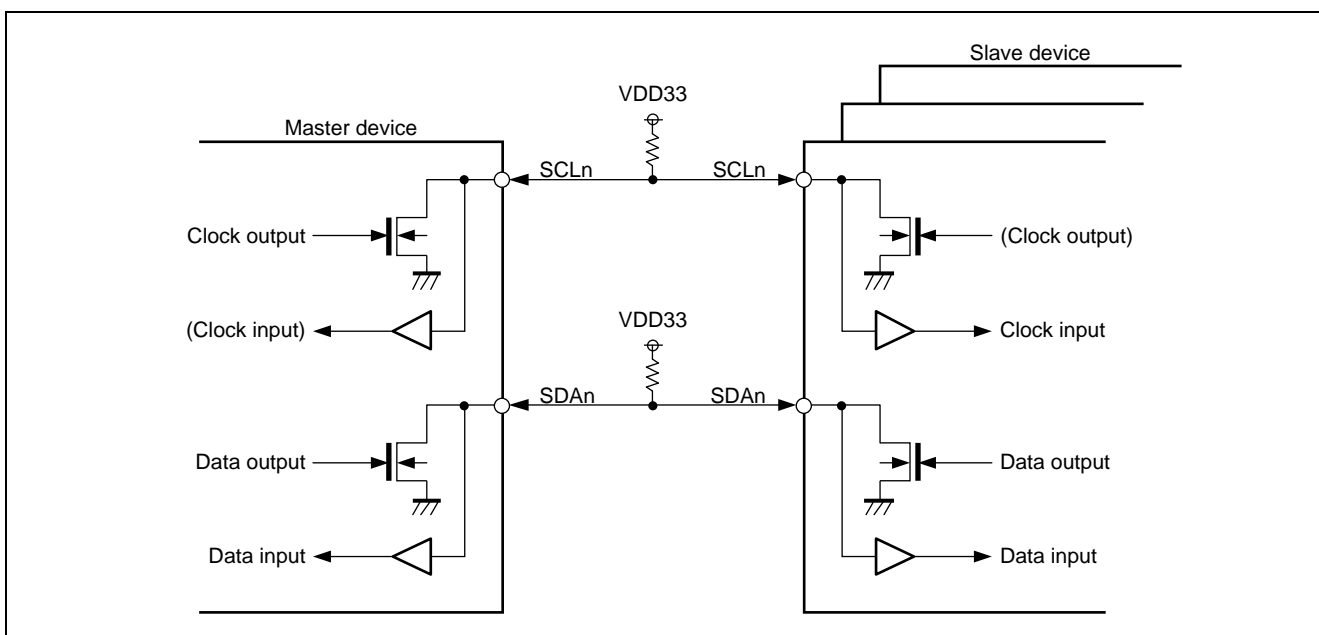


Figure 18.2 Pin Configuration Diagram

18.5 IIC Bus Definition

This section describes the IIC bus's serial data communication format and the signals used by the IIC bus.

Figure 18.3 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.

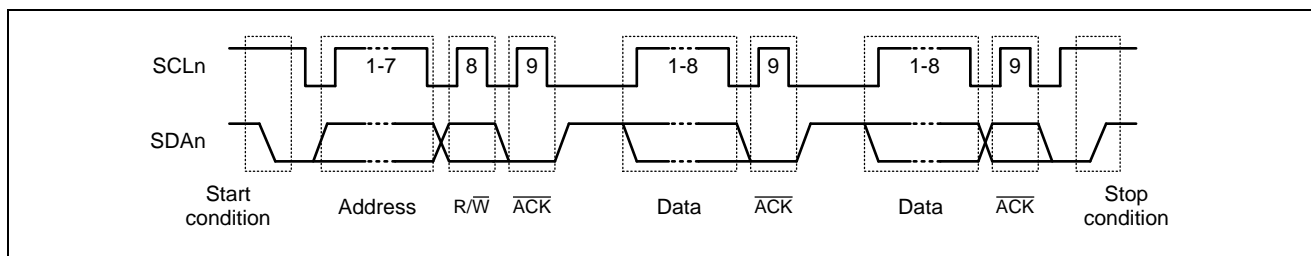


Figure 18.3 IIC Bus Serial Data Transfer Timing

The start condition, slave address, and stop condition are output by the master device.

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.

18.5.1 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.

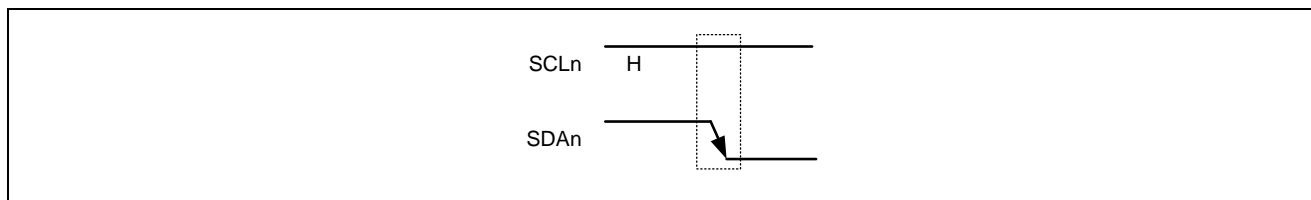


Figure 18.4 Start Condition

18.5.2 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.

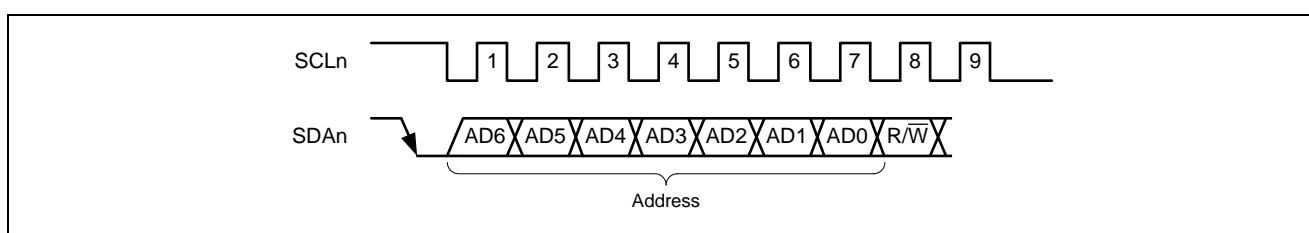


Figure 18.5 Address

18.5.3 Extension Code

When the higher-order 4 bits of the address are 0000 or 1111, these bits are called extension code. Table 18.5 lists the bit definitions of extension code.

Table 18.5 Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	x	CBUS address
0000 010	x	Address reserved for different bus format
0000 011	x	Reserved for future use
0000 1xx	x	HS mode master code ^{Note}
1111 0xx	x	10-bit slave address specification
1111 1xx	x	Reserved for future use

Note: The HS mode cannot be used for IICB.

18.5.4 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.

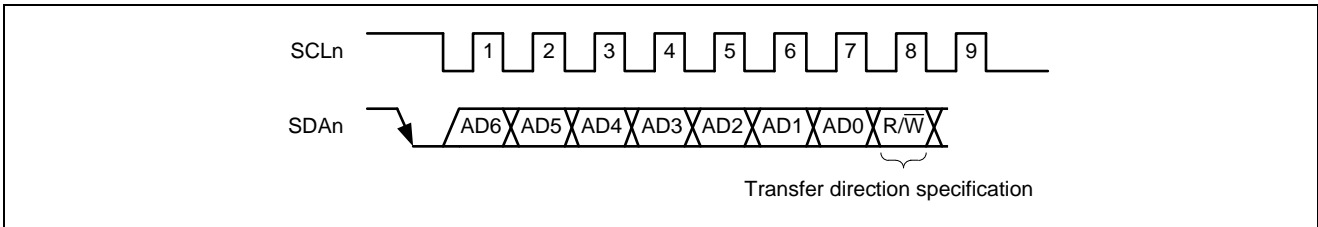


Figure 18.6 Transfer Direction Specification

18.5.5 Acknowledge (ACK)

The 1-bit data following the transfer direction (R/W) and the 1-bit data following the 8-bit data during address transfer are defined as an acknowledge signal (ACK). ACK is used to check the serial data status of the transmitting and receiving devices.

The receiving device returns ACK after receiving 8-bit data.

The transmitting device normally receives ACK after transmitting 8-bit data. If the transmitting device receives 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 ACK and outputs a stop condition. If the slave device is the receiving device and does not return ACK, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return 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.

ACK is output when the SDAn line of the receiving device changes to low level at the 9th clock (normal reception).

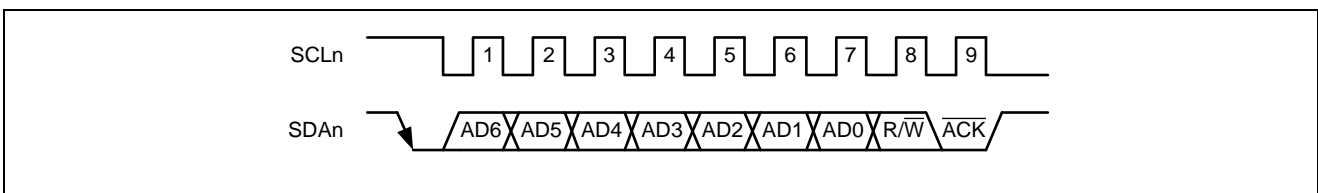


Figure 18.7 Acknowledge (ACK)

18.5.6 Data

The bits other than the nine bits following the start condition (seven address bits, an $\overline{R/\overline{W}}$ 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.

18.5.7 Stop Condition

A stop condition is met if the \overline{SDAn} signal level changes from low to high while the \overline{SCLn} signal is high.

The stop condition is output when serial data transfer from the master device to the slave device has been completed.

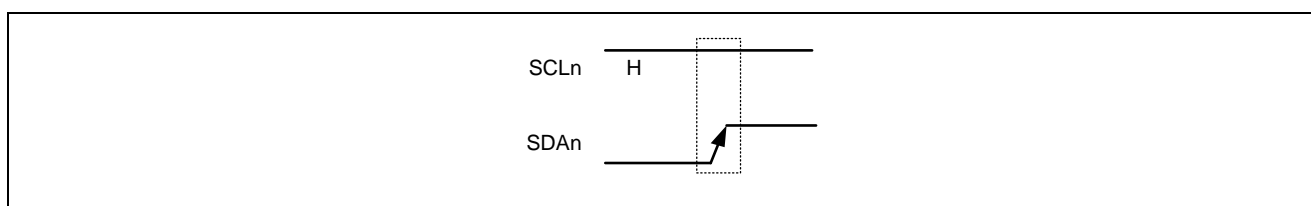


Figure 18.8 Stop Condition

18.5.8 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.

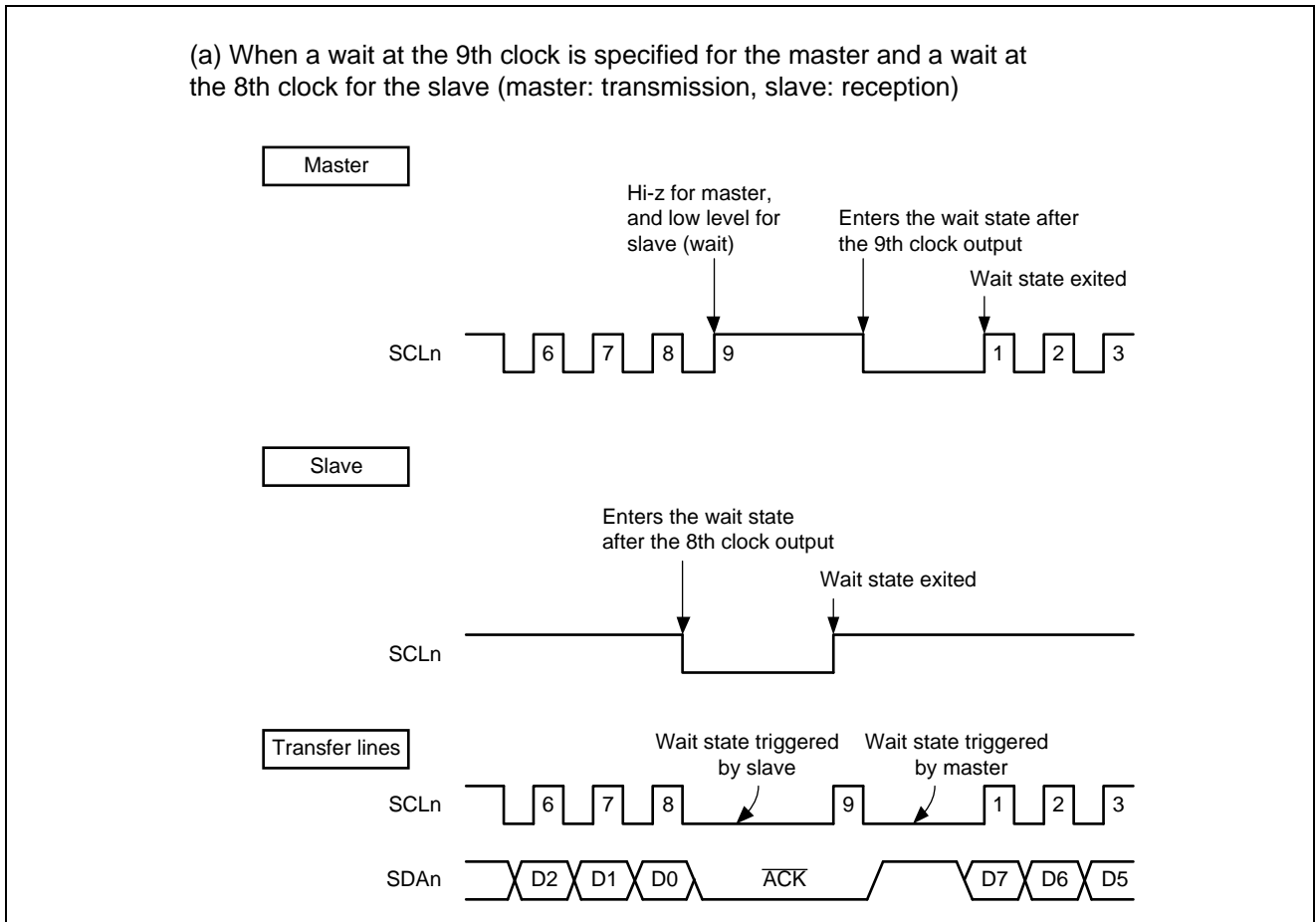


Figure 18.9 Wait State (1/2)

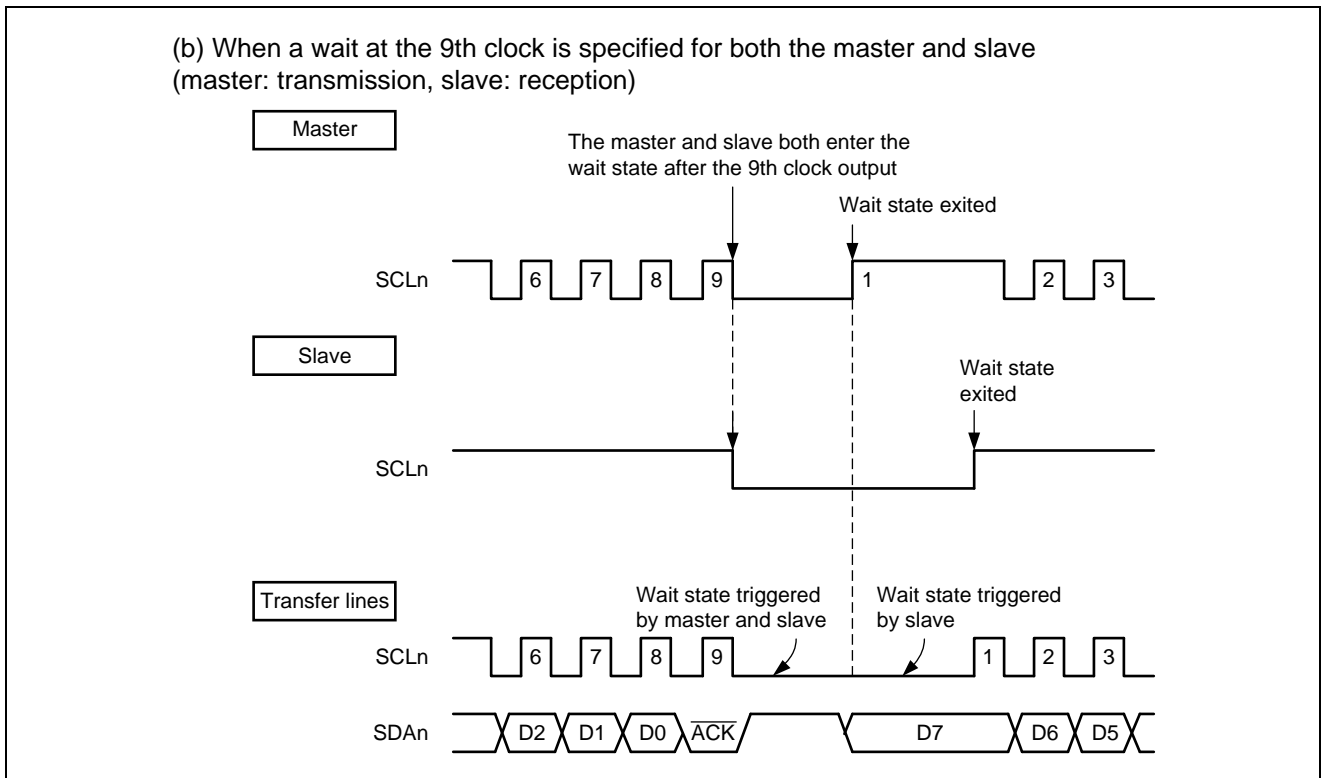


Figure 18.9 Wait State (2/2)

18.5.9 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 SDA_n line high (master 1) and the other master outputs the SDA_n line low (master 2) while the SCL_n 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 SCL_n and SDA_n line to high impedance.

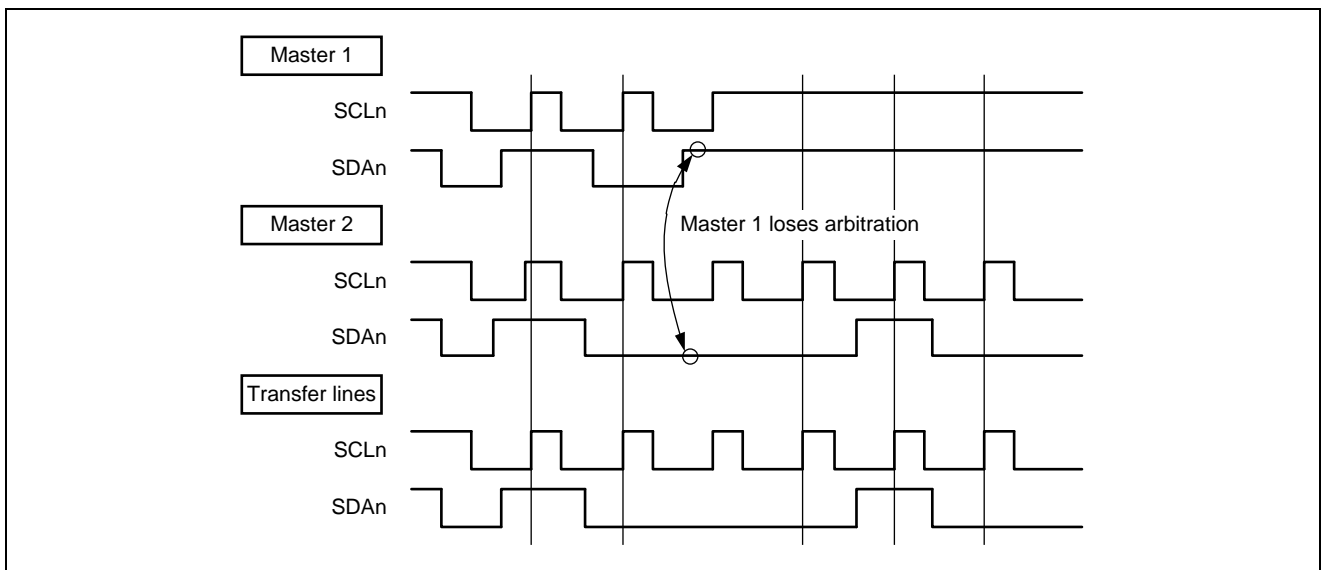


Figure 18.10 Arbitration Timing Example

18.6 Operation

The IICBn supports two transfer modes, single transfer mode and continuous transfer mode.

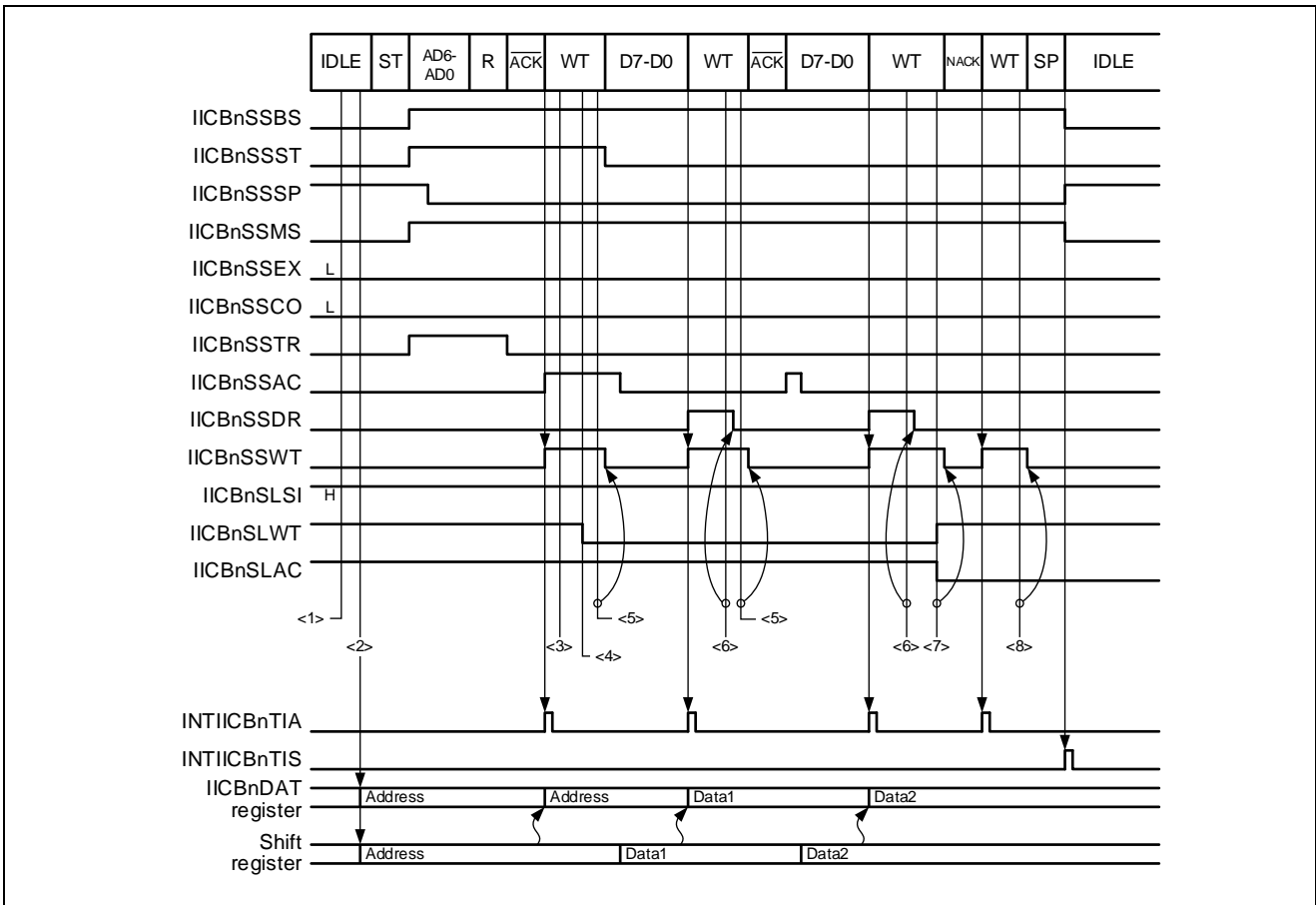
The transfer mode when the addresses of the master device and the slave device match is selected with the IICBnCTL0.IICBnMDTX0 bit, and the transfer mode when the extension code is detected by the slave device is selected with the IICBnCTL0.IICBnMDTX1 bit.

18.6.1 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.

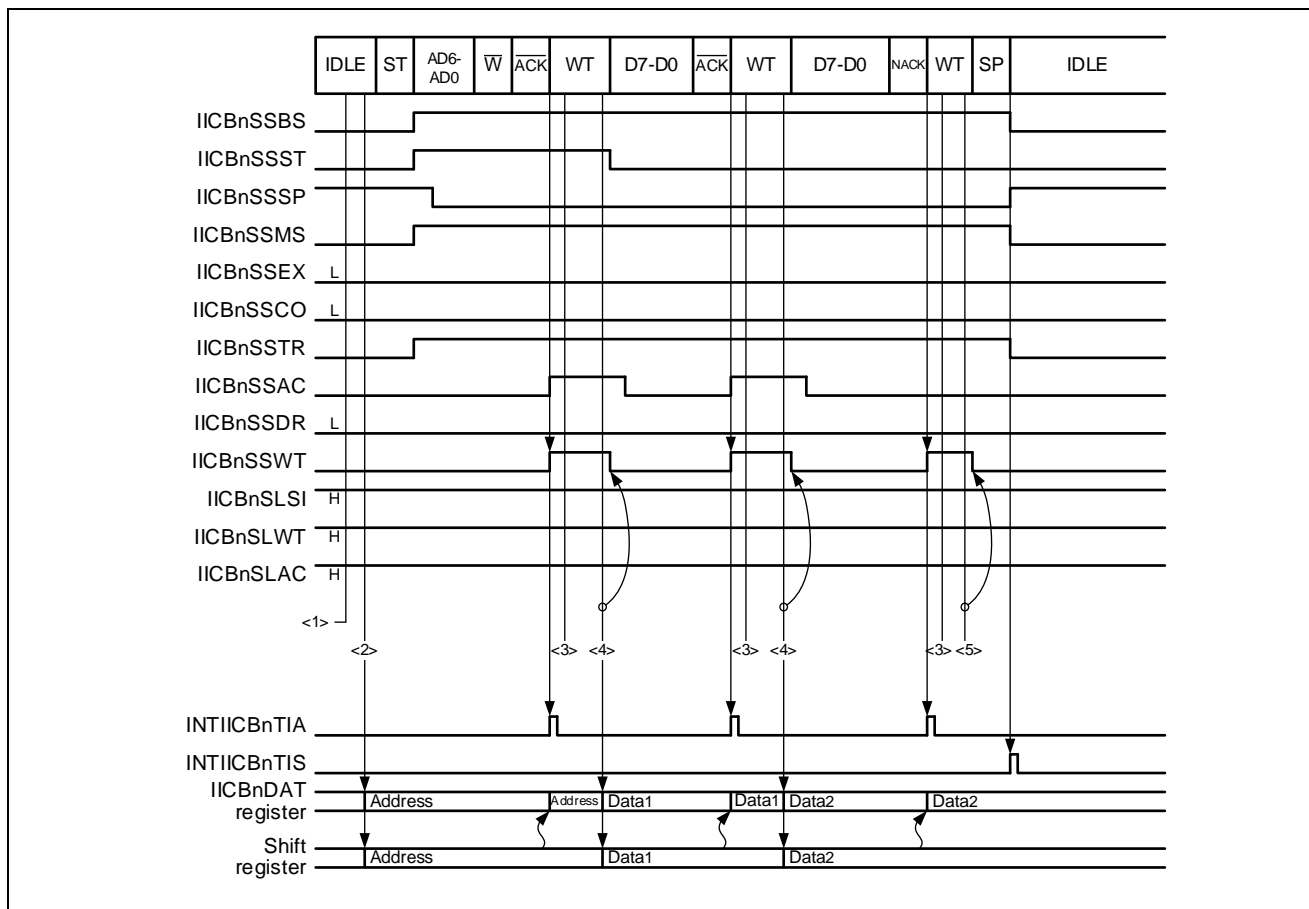
The various processing operations are described below.

(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).

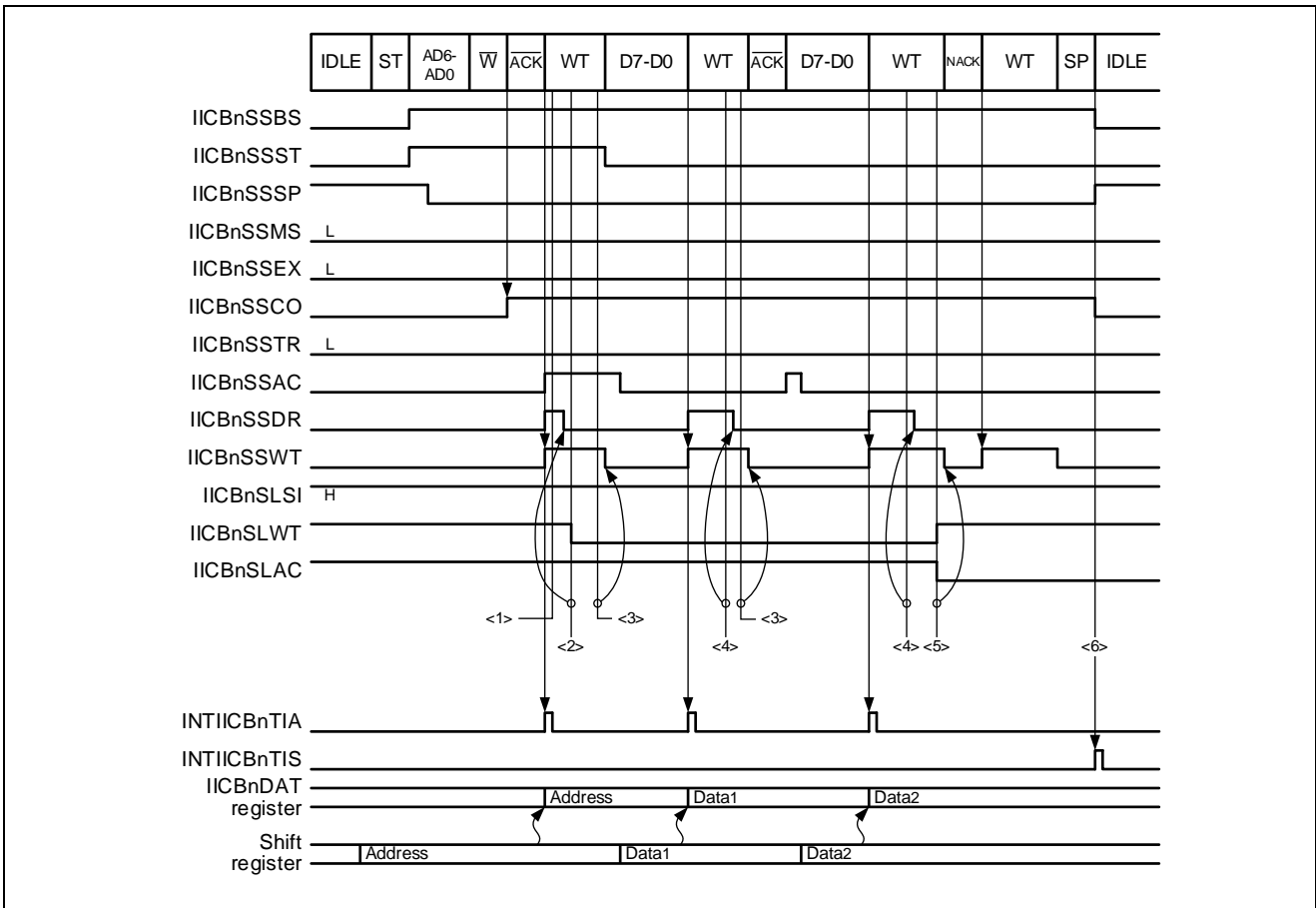
(2) Example of communications in single transfer mode (master transmission)



- <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

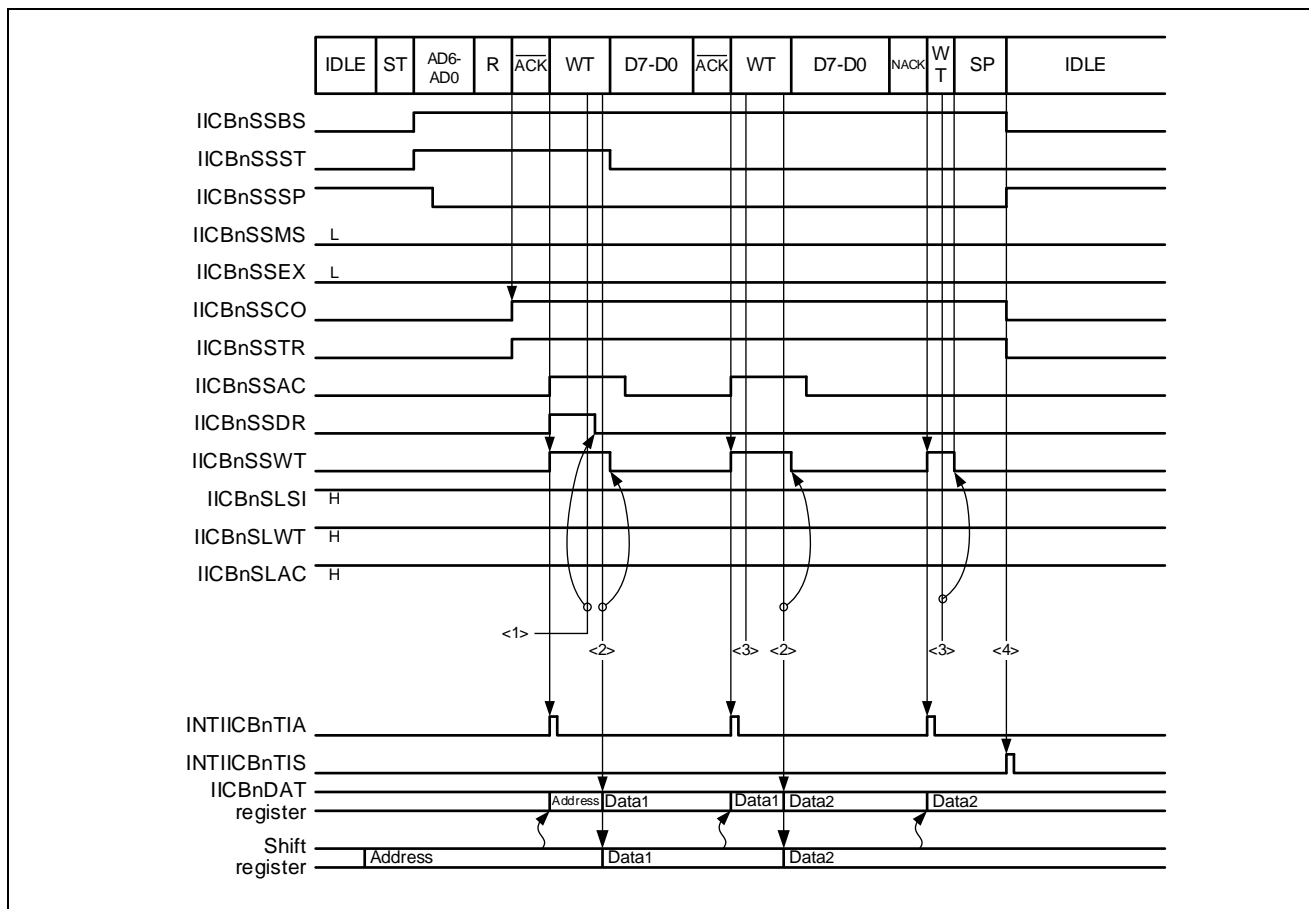
During data reception, clear the IICBnCTL0.IICBnSLWT bit (to 0) so that the IICBn enters the wait state at the falling edge of the 8th clock.
- <3> Data reception

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 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.

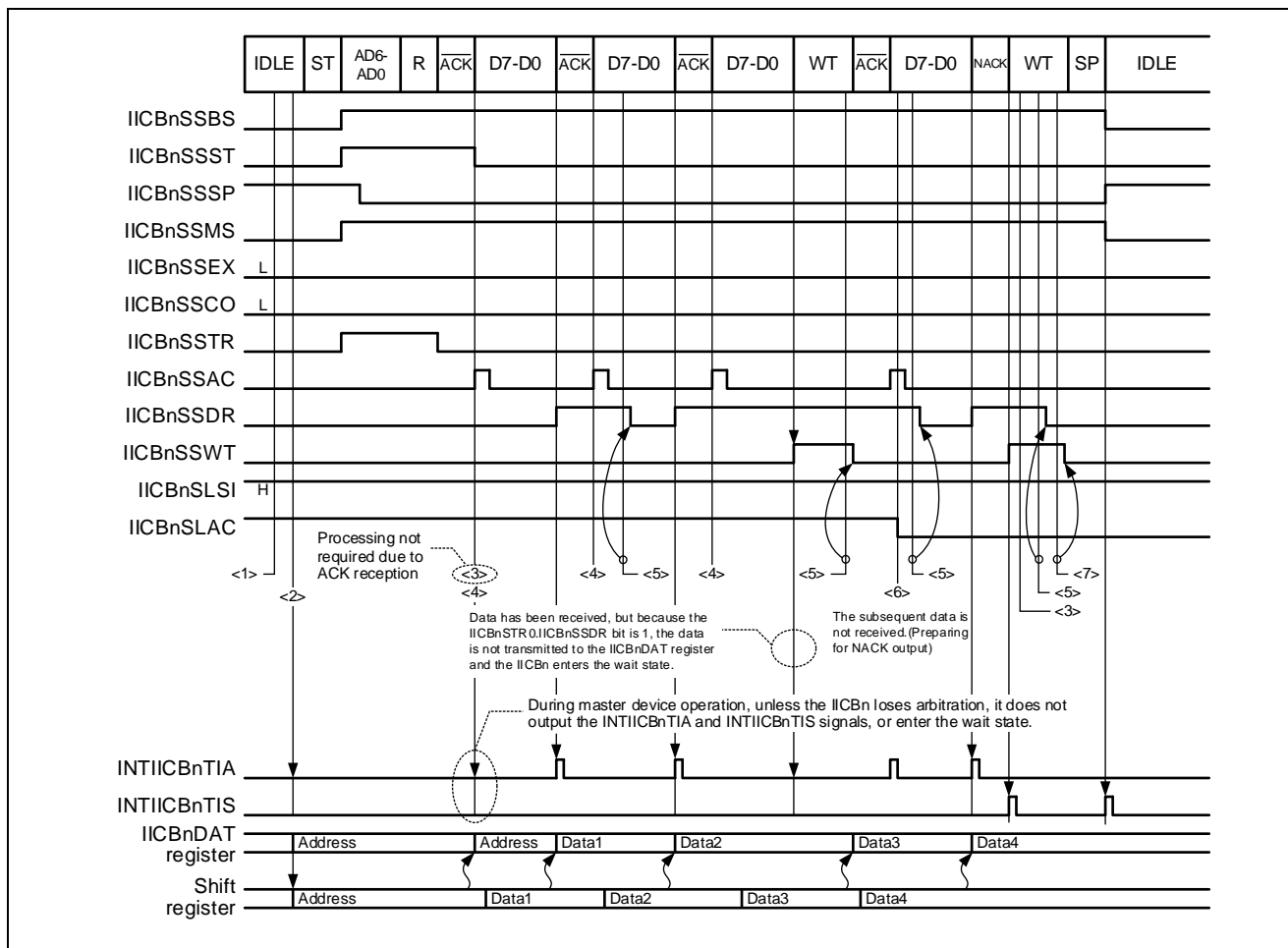
Remark: During data transmission, set the IICBnCTL0.IICBnSLWT bit (to 1) so that the IICBn enters the wait state at the falling edge of the 9th clock.

18.6.2 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.

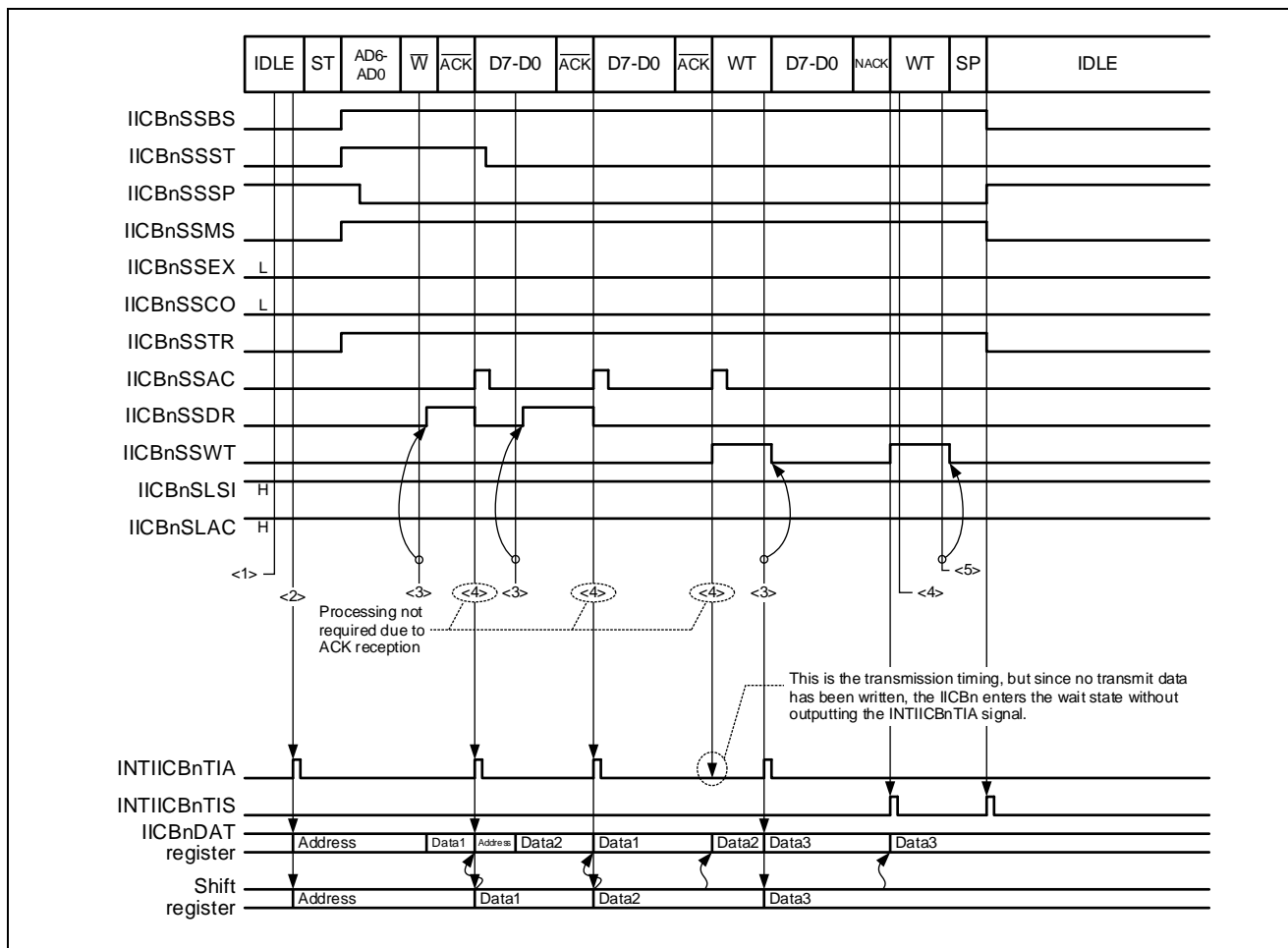
The processing operations are described below.

(1) Example of communications in continuous transfer mode (master reception)



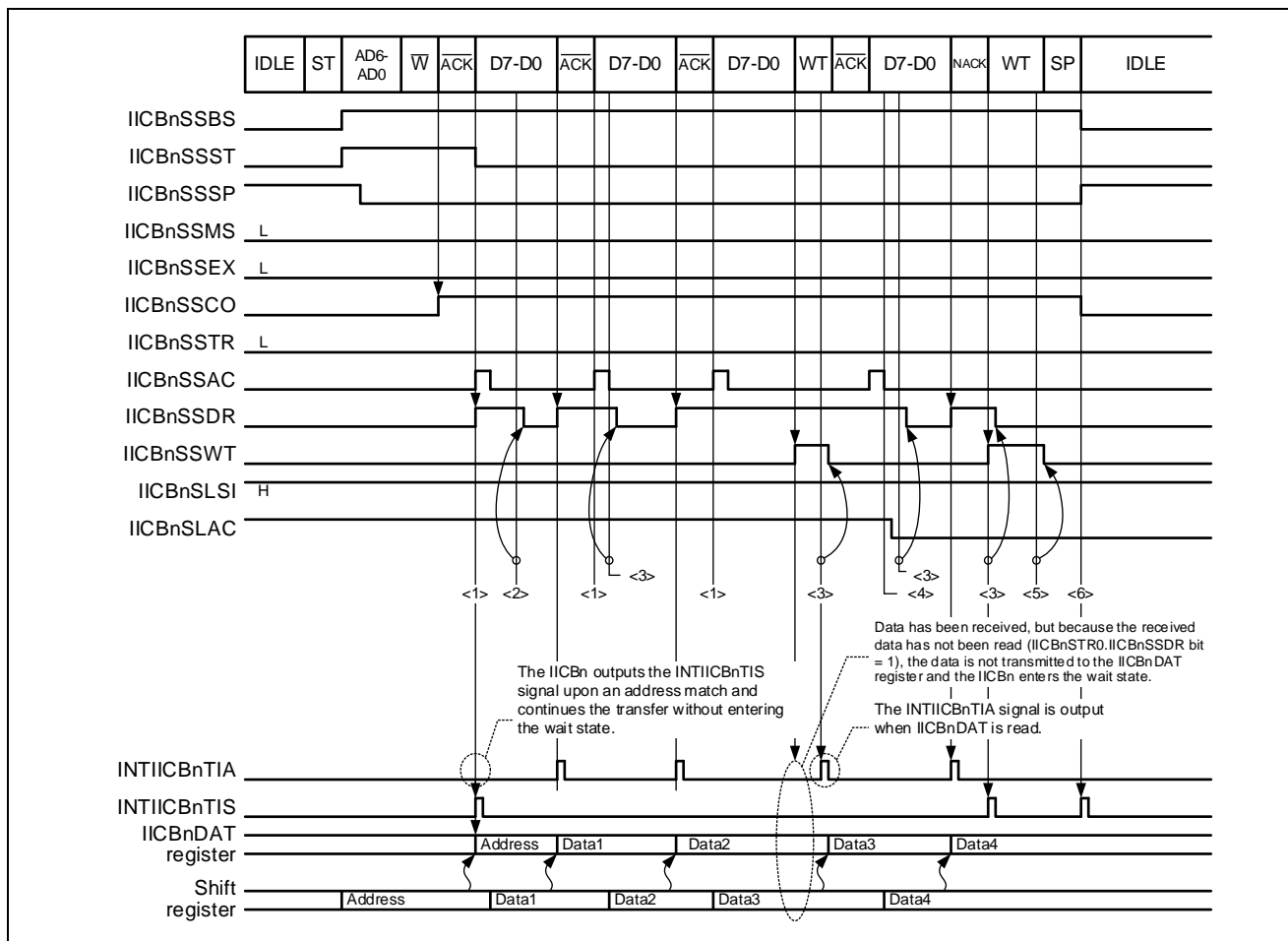
- <1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).
- <2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.
- <3> Acknowledge result check
The INTIICBnTIS interrupt occurs only if the 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)



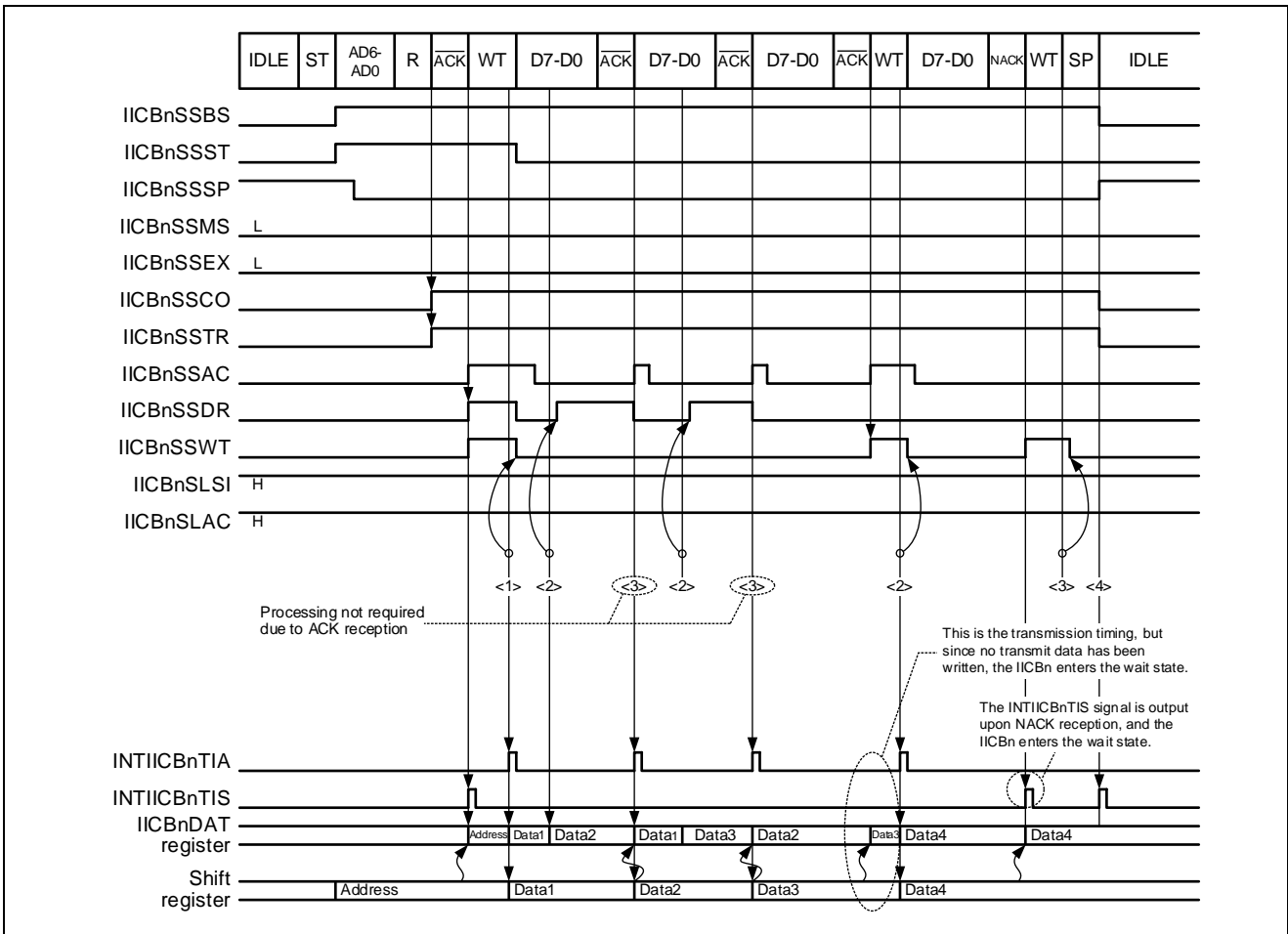
- <1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).
- <2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.
- <3> Data transmission
Set the transmit data to the IICBnDAT register using the INTIICBnTIA interrupt.
- <4> Acknowledge result check
The INTIICBnTIS interrupt occurs only if the slave device does not return $\overline{\text{ACK}}$. 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 $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$.
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) Slave transmission in continuous transfer mode



- <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 $\overline{\text{ACK}}$. Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit.
 - If $\overline{\text{ACK}}$ 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.

18.6.3 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

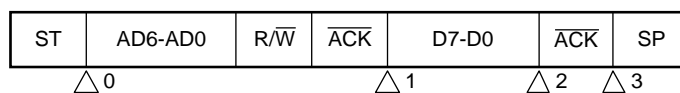
The statuses upon occurrence of arbitration during master device operation (IICBnSTR0.IICBnSSMS bit = 1) are listed below.

- (1) Address transmission
- (2) R/\bar{W} bit transmission of address transfer
- (3) Extension code transmission
- (4) R/\bar{W} bit transmission of extension code transfer
- (5) Data transmission
- (6) \overline{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

18.6.4 Entering and Exiting Wait State

The IICBn enters the wait state at the following timings.

Table 18.6 Wait State Transit Timings



Timing	Description	Refer to:
$\Delta 0$	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"
$\Delta 1$	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"
$\Delta 2$	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"
$\Delta 3$	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/W : Transfer direction specification
 $\overline{\text{ACK}}$: 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

$\Delta 0$ 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 $\Delta 0$ 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 18.11.

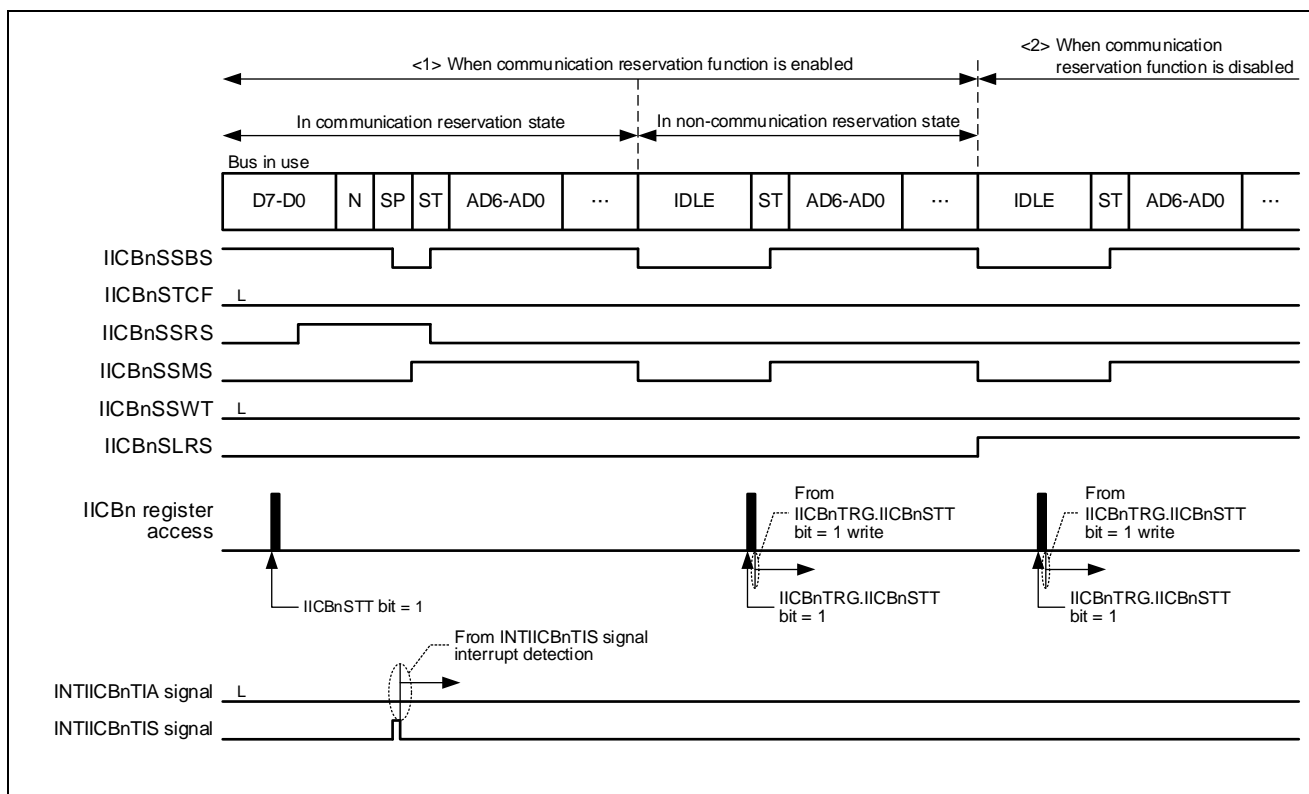


Figure 18.11 Valid Times to Write to IICBnDAT Register

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

$\Delta 2$ 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.IICBnSSSTR 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 and IICBnTRG.IICBnSPT bits 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{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 18.7.

Table 18.7 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

18.6.5 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.

18.7 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 18.6.5, 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.

18.7.1 Single Transfer Mode

The interrupt request signal timing in single transfer mode is described in Table 18.8 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 18.8 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	18.7.1 (1)
Δ2	Upon detection of the falling edge of the 8th SCLn during data transfer	18.7.1 (2)
Δ3	Upon detection of the falling edge of the 9th SCLn during data transfer	18.7.1 (2)
Δ4	Upon detection of a stop condition	18.7.1 (3)

Remark: ST : Start condition
 AD6-AD0 : Address
 R/W : Transfer direction specification
 ACK : Acknowledge
 D7-D0 : Data
 SP : Stop condition

(1) Interrupt request signal output conditions and output interrupt request signals during address transfer

$\Delta 1$ in Table 18.8 indicates the interrupt request signal output timing during an address transfer.

Table 18.9 indicates the interrupt request signal output condition and the interrupt request signal that is output (INTIICBnTIA or INTIICBnTIS) at the timing of $\Delta 1$.

Table 18.9 Interrupt Request Signal Output Conditions and Interrupt Request Signals Output during Address Transfer (Single Transfer Mode)

IICBn SSMS	IICBn ALDF	IICBn SLWT	IICBn SSCO	$\Delta 1$		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

$\Delta 2$ and $\Delta 3$ in Table 18.8 indicate the interrupt request signal output timing during a data transfer. The interrupt request signal output timing of $\Delta 2$ or $\Delta 3$ is determined according to the setting of the IICBnCTL0.IICBnSLWT bit. Table 18.10 indicates the interrupt request signal output condition and the interrupt request signal that is output (INTIICBnTIA or INTIICBnTIS) at the timing of $\Delta 2$ and $\Delta 3$.

Table 18.10 Interrupt Request Signal Output Conditions and Interrupt Request Signals Output during Data Transfer (Single Transfer Mode)

IICBn SSMS	IICBn ALDF	IICBn SLWT	IICBn SSCO	$\Delta 2$		$\Delta 3$		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

$\Delta 4$ in Table 18.8 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.

18.7.2 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 18.12)

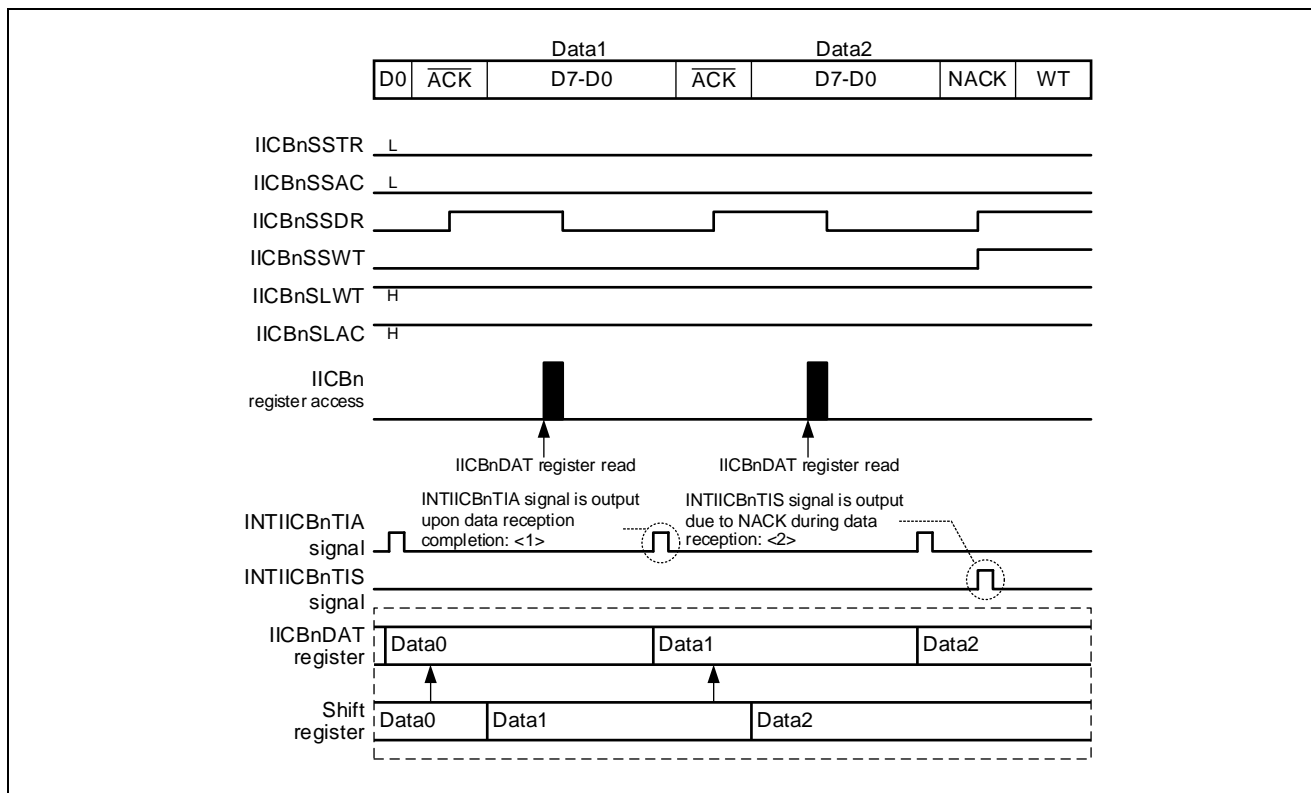


Figure 18.12 INTIICBnTIA Signal Output Timing (Reception in Continuous Transfer Mode)

• 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 18.13).

When data is saved from the IICBnDAT register to the shift register (timing <1> in Figure 18.13).

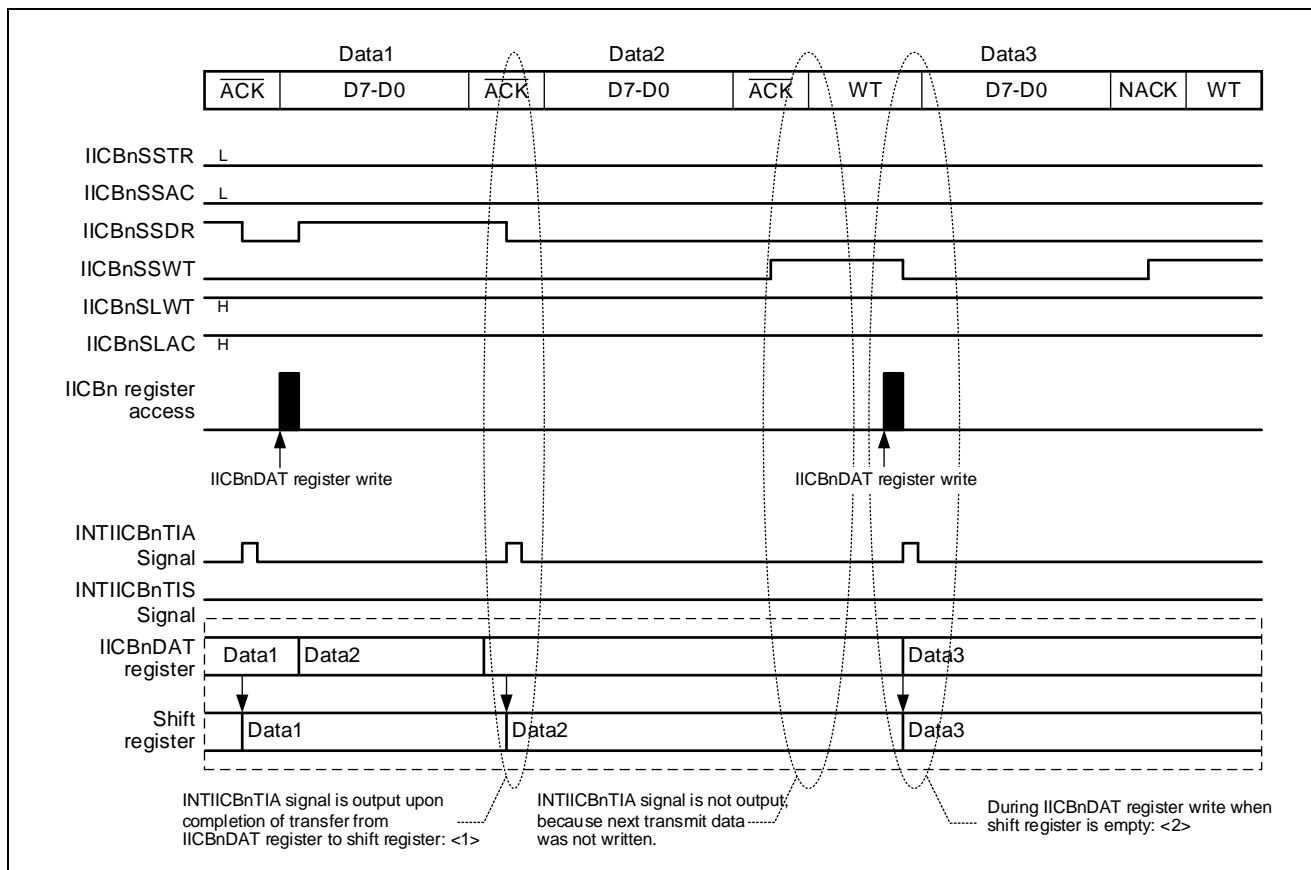
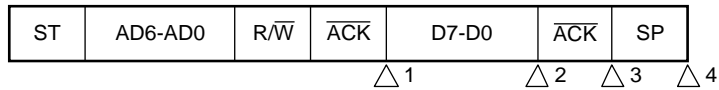


Figure 18.13 INTIICBnTIA Signal Output Timing (Transmission in Continuous Transfer Mode)

(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 18.11 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	18.7.2 (a)
Δ2	Upon detection of the falling edge of the 8th SCLn during data transfer	18.7.2 (b)
Δ3	Upon detection of the falling edge of the 9th SCLn during data transfer	18.7.2 (b)
Δ4	Upon detection of a stop condition	18.7.2 (c)

Remark: **ST** : Start condition
AD6-AD0 : Address
R/W : Transfer direction specification
ACK : Acknowledge
D7-D0 : Data
SP : Stop condition

(a) INTIICBnTIS signal output conditions during address transfer

$\Delta 1$ in Table 18.11 indicates the INTIICBnTIS signal output timing during address transfer. Table 18.12 indicates the INTIICBnTIS signal output conditions at the $\Delta 1$ timing.

Table 18.12 INTIICBnTIS Signal Output Conditions during Address Transfer (Continuous Transfer Mode)

IICBn SSMS	IICBn SSCO	IICBn ALDF	Transfer direction	IICBn SSDR	IICBn SSAC	$\Delta 1$	
						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 $\Delta 1$, the IICBnSTR0.IICBnSSAC bit is always 0.

Remark: X: don't care

(b) INTIICBnTIS signal output conditions during data transfer

$\Delta 2$ and $\Delta 3$ in Table 18.11 indicate the INTIICBnTIS signal output timings during data transfer. Table 18.13 indicates the INTIICBnTIS signal output conditions at the $\Delta 2$ and $\Delta 3$ timings.

Table 18.13 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	$\Delta 2$		$\Delta 3$	
								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	IICBTIS	—	—	—
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

$\Delta 4$ in Table 18.11 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.

18.8 Interrupt Outputs and States

This section describes the states of the IICBnSTR0 register during interrupt output by communication flow.

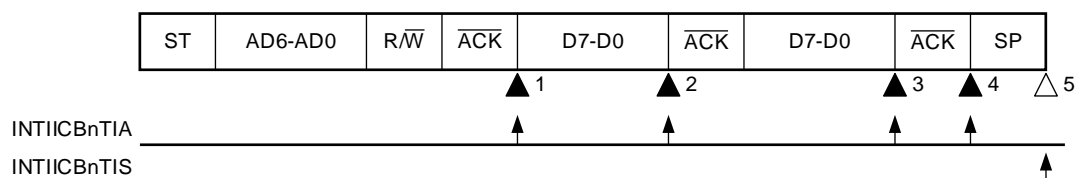
The meanings of the symbols used in the figures are as follows.

ST	: Start condition
AD6-AD0	: Address
R, \overline{W} , R/ \overline{W}	: Transfer direction specification
\overline{ACK}	: Acknowledge
NACK	: Not acknowledge
D7-D0	: Data
SP	: Stop condition

18.8.1 Single Transfer Mode (Master Device Operation)

(1) Start – Address – Data – Data – Stop (normal transmission/reception)

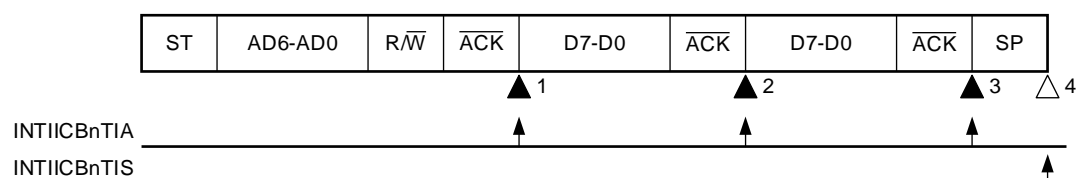
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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

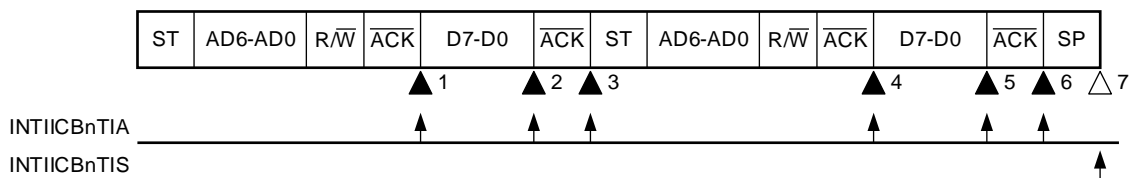


- ▲1: IICBnSTR0 register = 1-0100X1 0110--00B
- ▲2: IICBnSTR0 register = 1-0100X1 0100--00B
- ▲3: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit= 1)
- △4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

(2) Start – Address – Data – Start – Address – Data – Stop (restart)

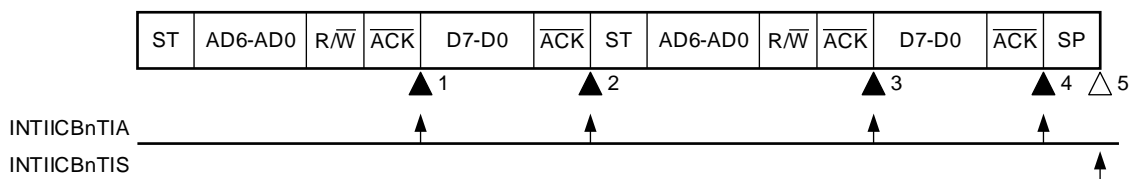
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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

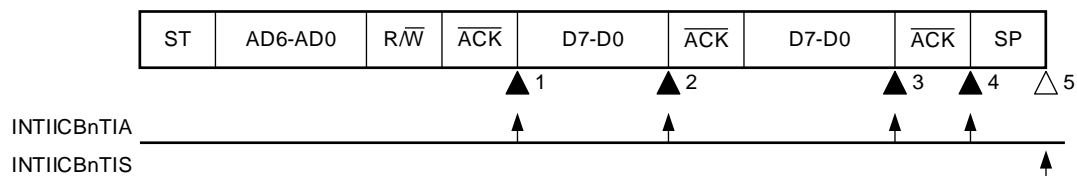


- ▲1: IICBnSTR0 register = 1-0100X1 0110--00B
- ▲2: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSTT bit = 1)
- ▲3: IICBnSTR0 register = 1-0100X1 0110--00B
- ▲4: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)
- △5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

(3) Start – Code – Data – Data – Stop (extension code transmission)

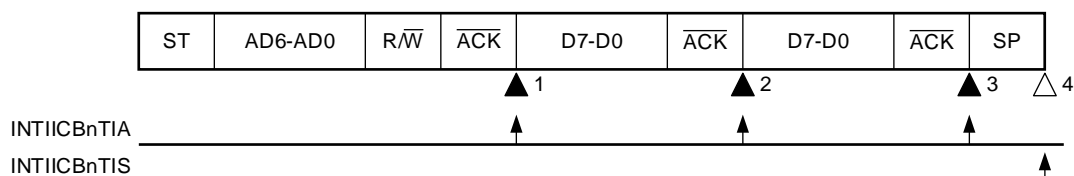
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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



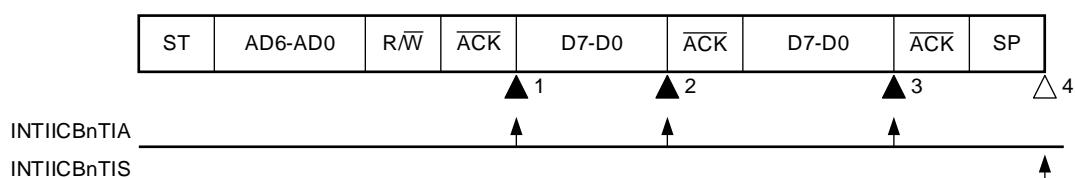
- ▲1: IICBnSTR0 register = 1-0110X1 0110--00B
- ▲2: IICBnSTR0 register = 1-0110X1 0100--00B
- ▲3: IICBnSTR0 register = 1-0110XX 0100--00B (IICBnTRG.IICBnSPT bit= 1)
- △4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.2 Single Transfer Mode (Slave Device Operation: during Slave Address Reception (IICBnSTR0.IICBnSSC0 bit = 1))

(1) Start – Address – Data – Data – Stop

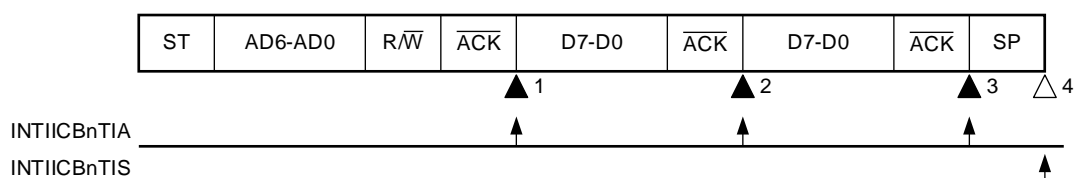
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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

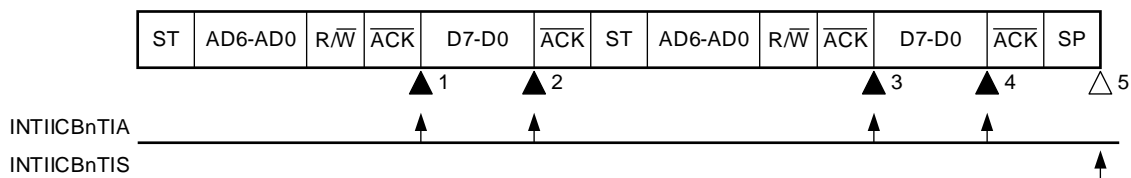


- ▲1: IICBnSTR0 register = 0-0101X1 0110--00B
- ▲2: IICBnSTR0 register = 0-0101X1 0100--00B
- ▲3: IICBnSTR0 register = 0-0101XX 0100--00B
- △4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

(2) Start – Address – Data – Start – Address – Data – Stop

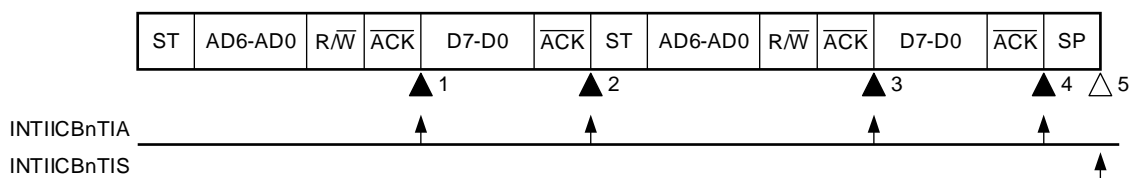
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)



- ▲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)

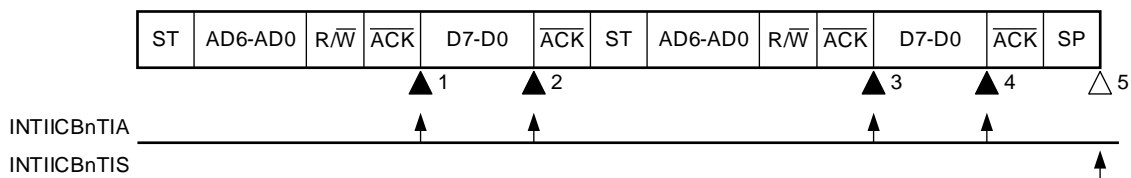


- ▲1: IICBnSTR0 register = 0-0101X1 0110--00B
- ▲2: IICBnSTR0 register = 0-0101XX 0100--00B
- ▲3: IICBnSTR0 register = 0-0101X1 0110--00B
- ▲4: IICBnSTR0 register = 0-0101XX 0100--00B
- △5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

(3) Start – Address – Data – Start – Code – Data – Stop

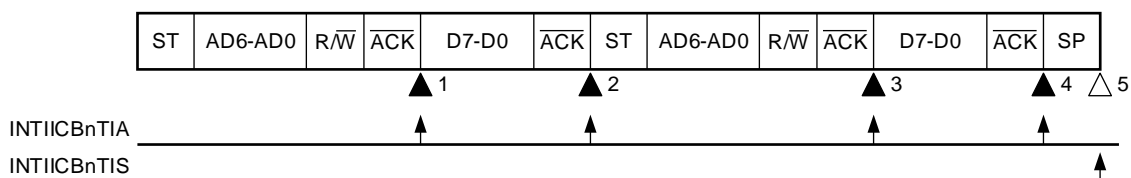
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)



- ▲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)

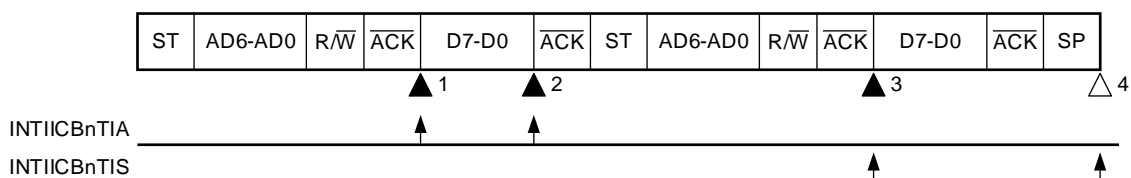


- ▲1: IICBnSTR0 register = 0-0101X1 0110--00B
- ▲2: IICBnSTR0 register = 0-0101XX 0100--00B
- ▲3: IICBnSTR0 register = 0-0110X1 0110--00B
- ▲4: IICBnSTR0 register = 0-0110XX 0100--00B
- △5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

(4) Start – Address – Data – Start – Address – Data – Stop

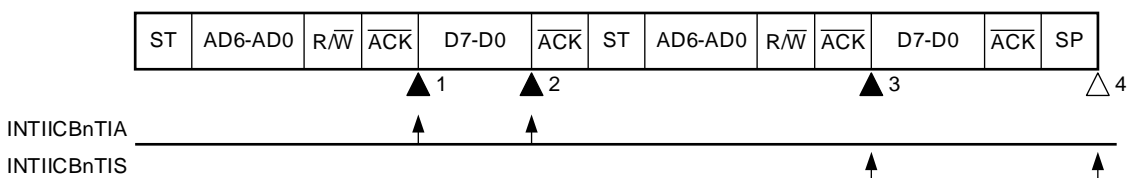
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



- ▲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

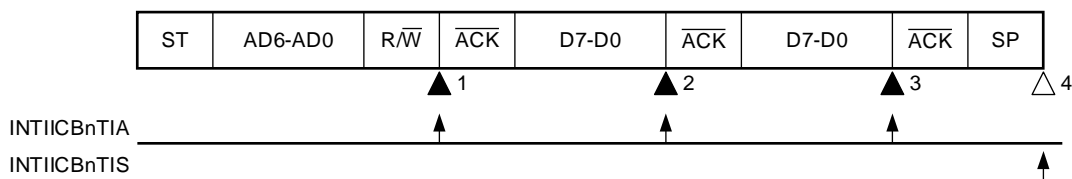
Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.3 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

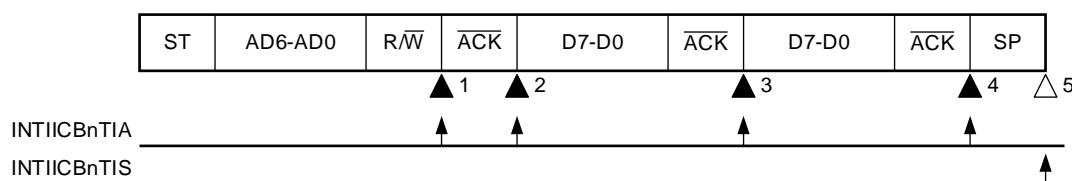
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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

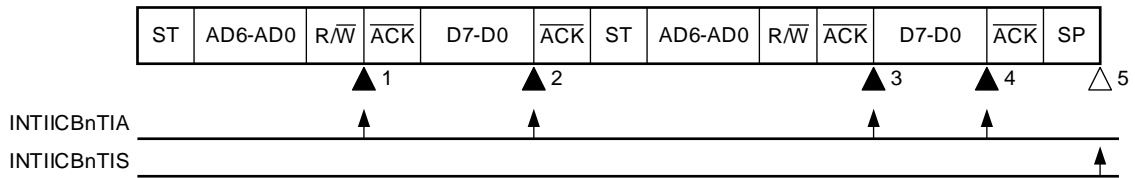


- ▲1: IICBnSTR0 register = 0-0110X0 0110--00B
- ▲2: IICBnSTR0 register = 0-0110X1 0110--00B
- ▲3: IICBnSTR0 register = 0-0110X0 0100--00B
- ▲4: IICBnSTR0 register = 0-0110XX 0100--00B
- △5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

(2) Start – Code – Data – Start – Address – Data – Stop

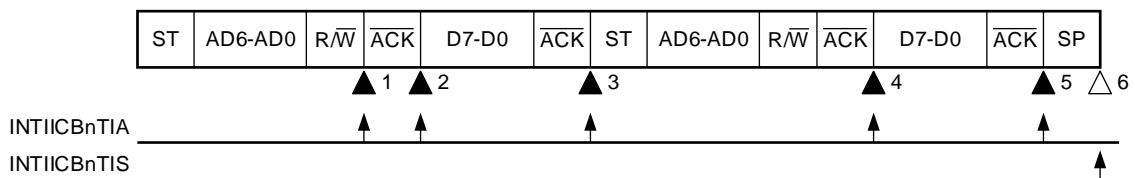
(a) <1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)



- ▲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)

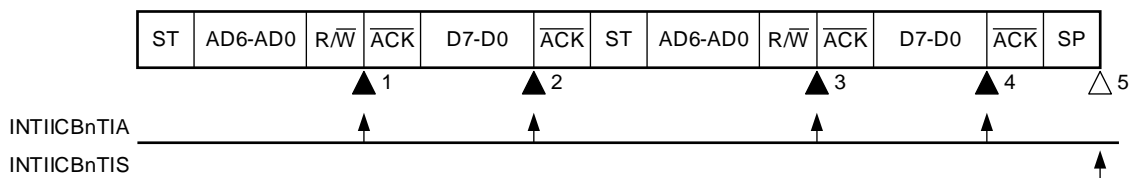


- ▲1: IICBnSTR0 register = 0-0110X0 0110--00B
- ▲2: IICBnSTR0 register = 0-0110X1 0110--00B
- ▲3: IICBnSTR0 register = 0-0110X0 0100--00B
- ▲4: IICBnSTR0 register = 0-0101X1 0110--00B
- ▲5: IICBnSTR0 register = 0-0101XX 0100--00B
- △6: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

(3) Start – Code – Data – Start – Code – Data – Stop

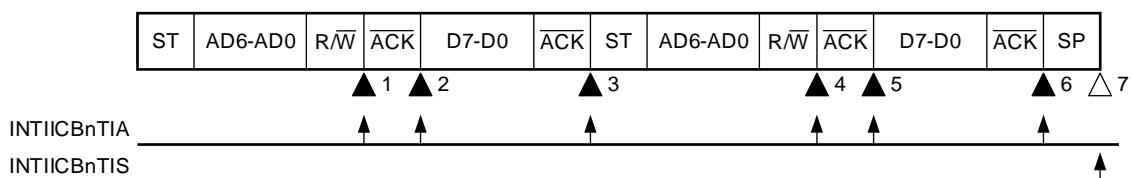
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)



- ▲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)

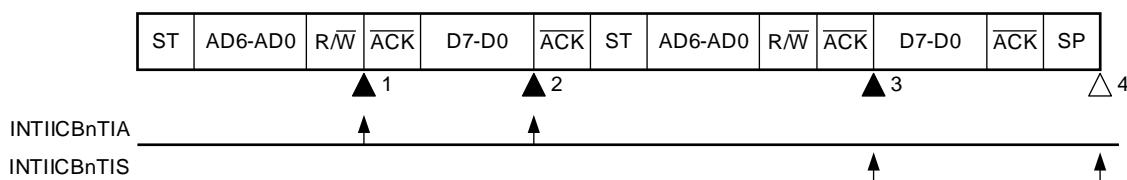


- ▲1: IICBnSTR0 register = 0-0110X0 0110--00B
- ▲2: IICBnSTR0 register = 0-0110X1 0110--00B
- ▲3: IICBnSTR0 register = 0-0110XX 0100--00B
- ▲4: IICBnSTR0 register = 0-0110X0 0110--00B
- ▲5: IICBnSTR0 register = 0-0110X1 0110--00B
- ▲6: IICBnSTR0 register = 0-0110XX 0100--00B
- △7: IICBnSTR0 register = 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

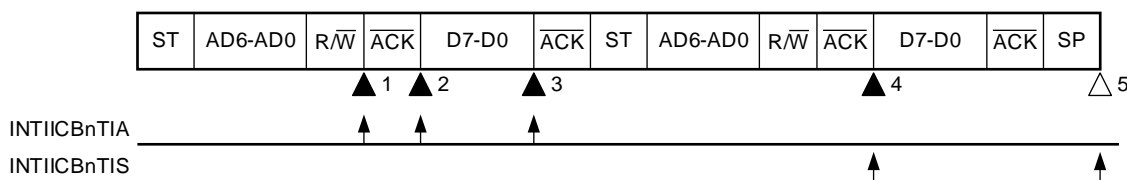
(a) When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



- ▲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))

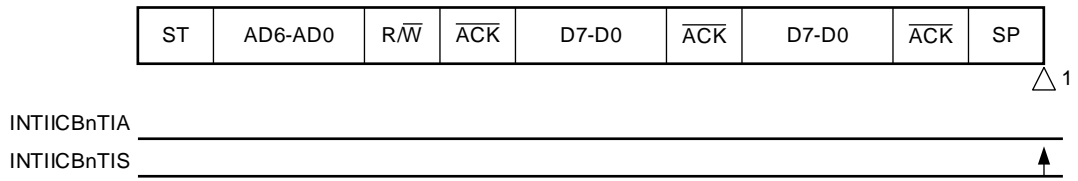


- ▲1: IICBnSTR0 register = 0-0110X0 0110--00B
- ▲2: IICBnSTR0 register = 0-0110X1 0110--00B
- ▲3: IICBnSTR0 register = 0-0000X0 0100--00B
- ▲4: IICBnSTR0 register = 0-0000X0 0110--00B
- △5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.4 Single Transfer Mode (Non-Participation in Communications)

(1) Start – Code – Data – Data – Stop



△1: IICBnSTR0 register = 0-000000 0001--00B

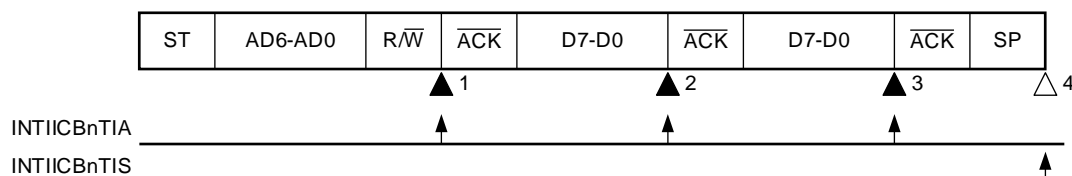
Remark: △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined

18.8.5 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

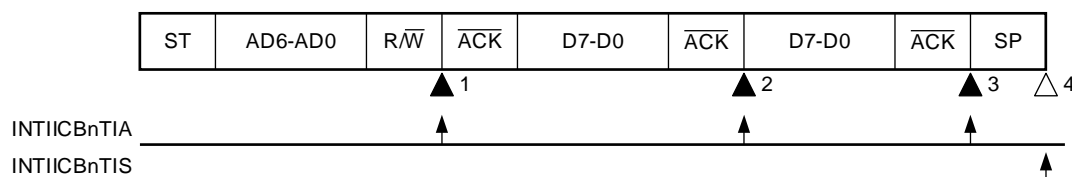
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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

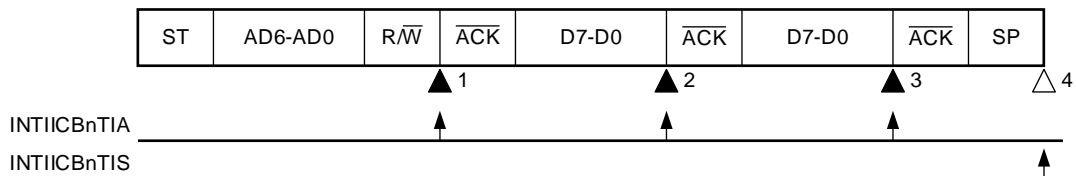


- ▲1: IICBnSTR0 register = 0-0101X1 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)
- ▲2: IICBnSTR0 register = 0-0101X1 0100--00B
- ▲3: IICBnSTR0 register = 0-0101XX 0100--00B
- △4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

(2) Upon extension code detection after arbitration loss

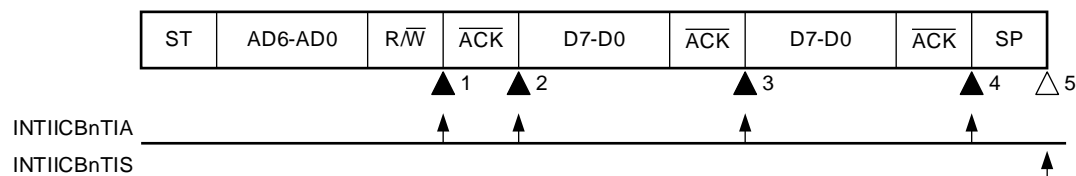
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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



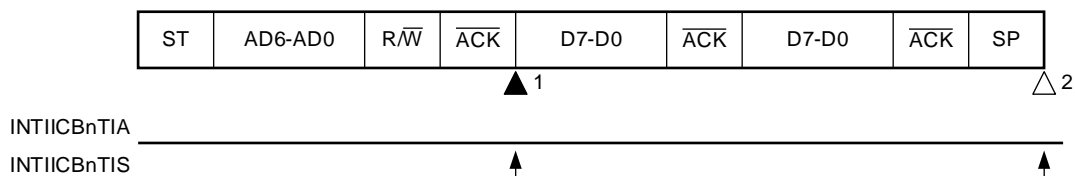
- ▲1: IICBnSTR0 register = 0-0110X0 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)
- ▲2: IICBnSTR0 register = 0-0110X1 0110--00B
- ▲3: IICBnSTR0 register = 0-0110X0 0100--00B
- ▲4: IICBnSTR0 register = 0-0110XX 0100--00B
- △5: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.6 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



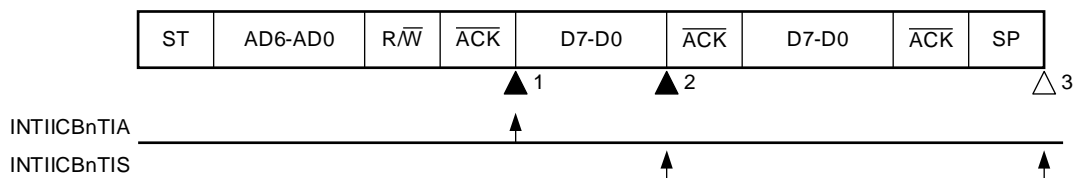
▲ 1: IICBnSTR0 register = 0-0000X1 0110--01B (IICBnSTRC.IICBnCLAF bit= 1)

△ 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

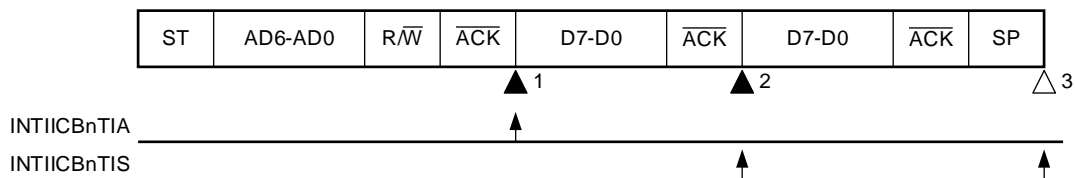
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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

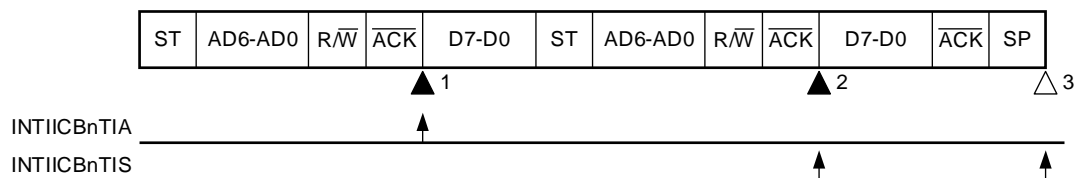


- ▲1: IICBnSTR0 register = 1-1000X1 0110--00B
- ▲2: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
- △3: IICBnSTR0 register = 0-000000 0001--00B

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

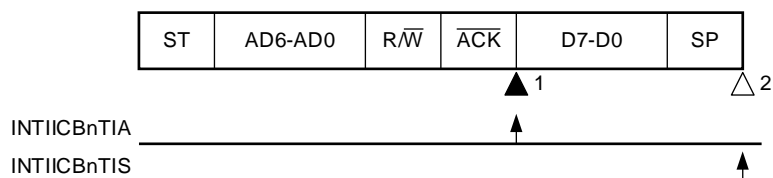
Remark: ▲ Always output

△ 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

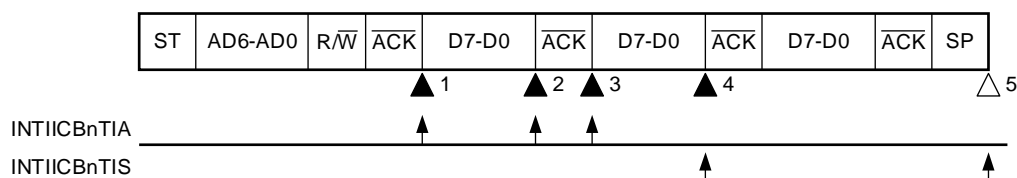
△ Output regardless of the setting of IICBnCTL0.IICBnSLSI bit

- Undefined

X don't care

(5) Arbitration loss because the SDA_n signal is low level when attempting to output restart condition

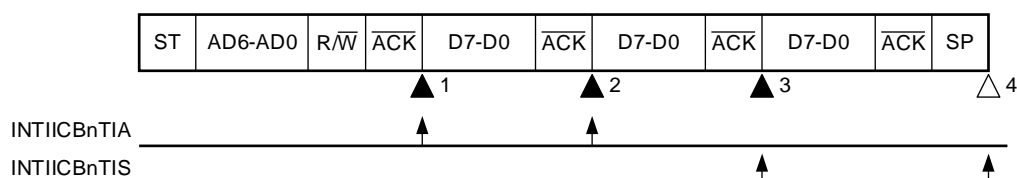
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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

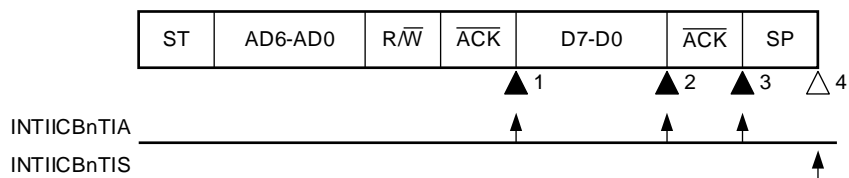


- ▲1: IICBnSTR0 register = 1-1000X1 0110--00B
- ▲2: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnCTL0.IICBnSLWT bit = 0, IICBnTRG.IICBnSTT bit = 1)
- ▲3: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
- △4: IICBnSTR0 register = 0-000000 0001--00B

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

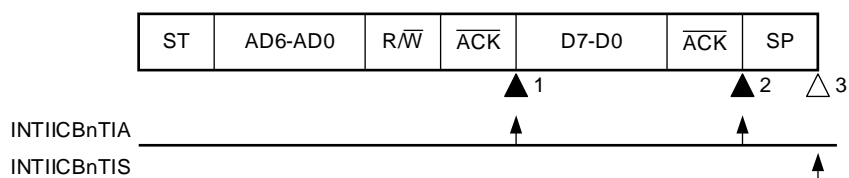
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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

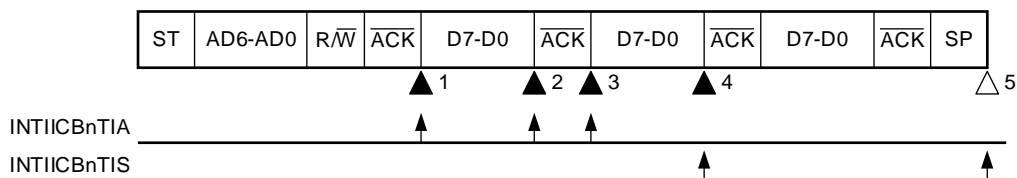


- ▲1: IICBnSTR0 register = 1-1000X1 0110--00B
- ▲2: IICBnSTR0 register = 1-0000XX 0100--00B (IICBnTRG.IICBnSTT bit = 1)
- △3: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
 △ Output regardless of the setting of IICBnCTL0.IICBnSLSI bit
 - Undefined
 X don't care

(7) Arbitration loss because the SDA_n signal is low level when attempting to output stop condition

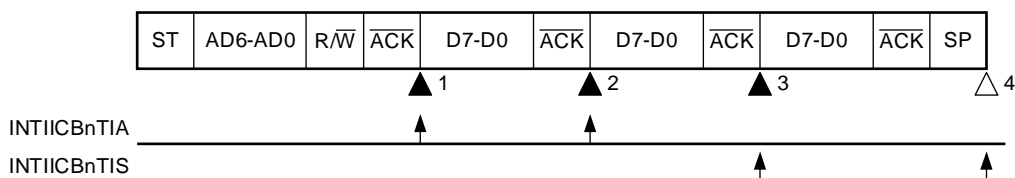
(a) When IICBnCTL0.IICBnSLWT bit is 0



- ▲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, IICBnTRG.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



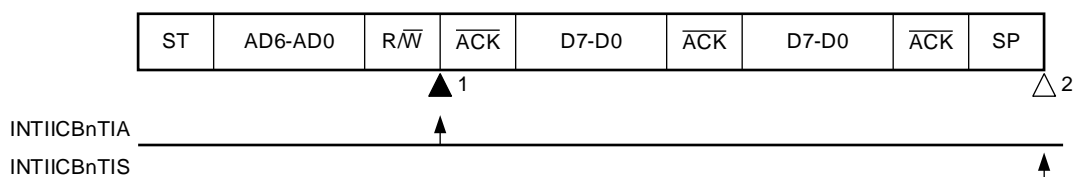
- ▲1: IICBnSTR0 register = 1-1000X1 0110--00B
- ▲2: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnTRG.IICBnSPT bit= 1)
- ▲3: IICBnSTR0 register = 0-0000XX 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)
- △4: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.7 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.

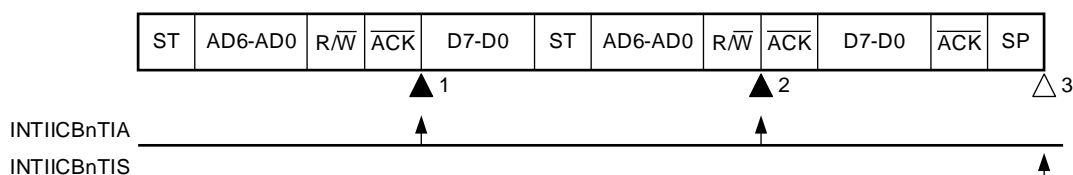
(1) Arbitration loss during extension code transfer



- ▲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

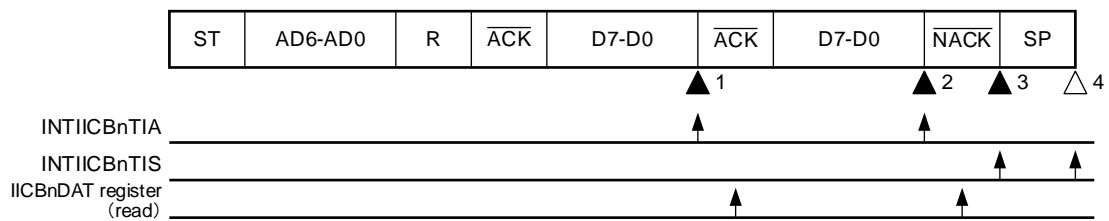
Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.8 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

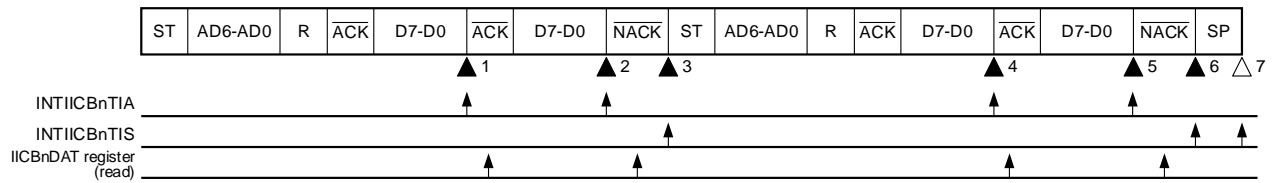
→ IICBnTRG.IICBnSPT bit = 1

△4: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined

(2) Start – Address – Data x 2 – Start – Address – Data x 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

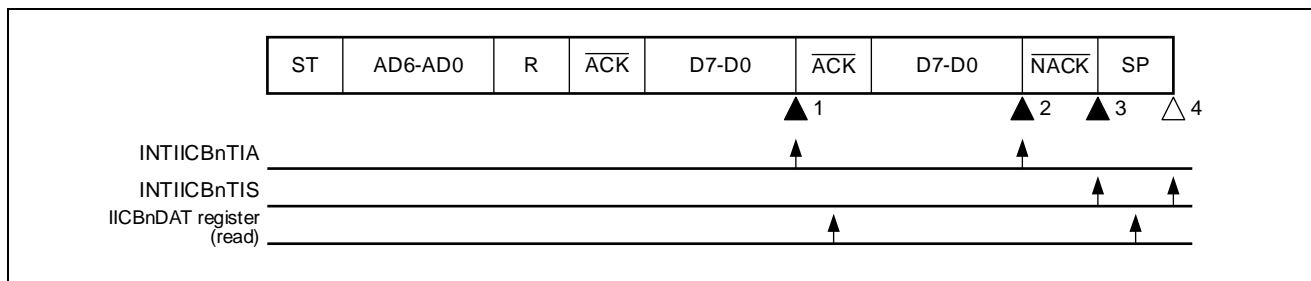
→ IICBnTRG.IICBnSTT bit = 1

△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

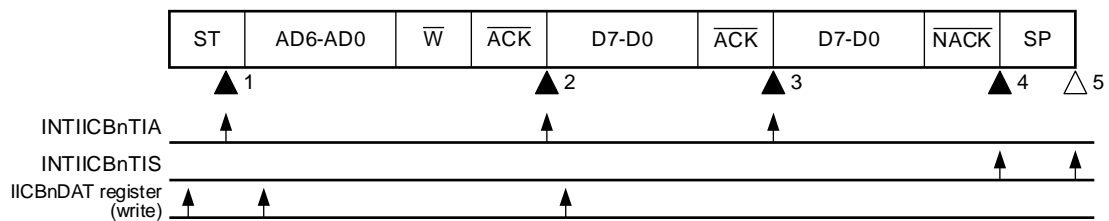
Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined

18.8.9 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

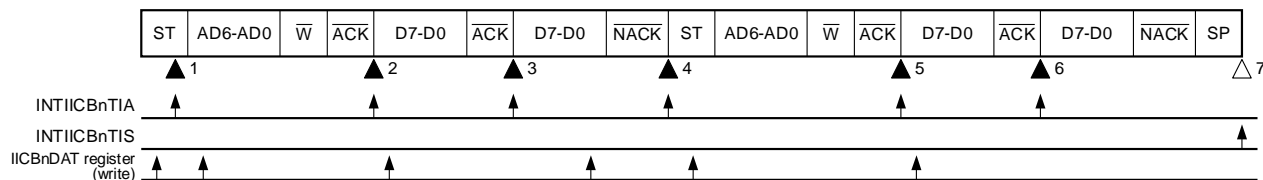
IICBnTRG.IICBnSPT bit= 1

△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 x 2 – Start – Address – Data x 2 – Stop

(a) When IICBnCTL0.IICBnSLWT bit is 1

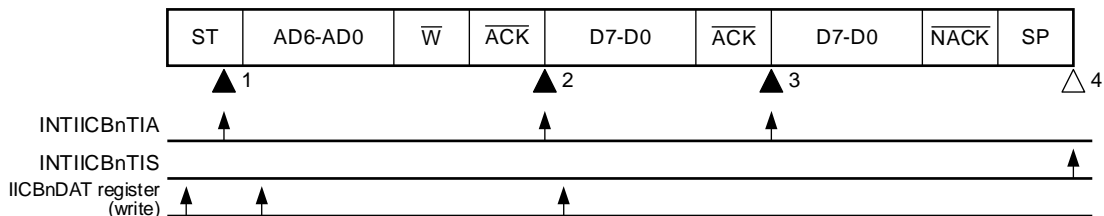


- IICBnDAT register write (address)
- [▲1: IICBnSTR0 register = X-0000X0 0X0X--00B]
- IICBnDAT register write
- [▲2: IICBnSTR0 register = 1-000011 0110--00B]
- IICBnDAT register write
- [▲3: IICBnSTR0 register = 1-000011 0100--00B]
- IICBnTRG.IICBnSTT bit= 1
- IICBnDAT register write (address)
- [▲4: 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

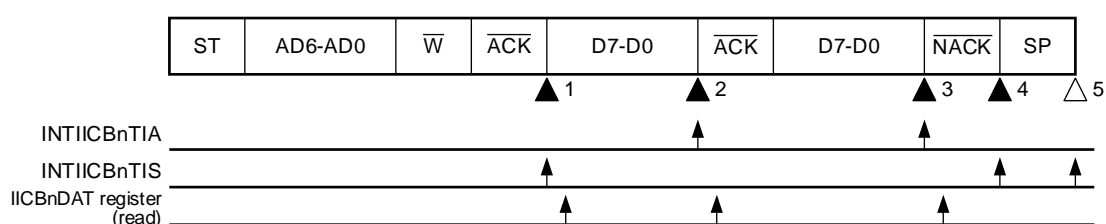
Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.10 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

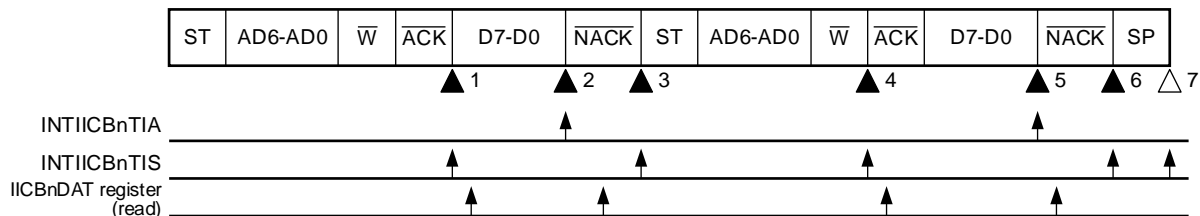
IICBnTRG.IICBnWRET bit = 1

△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

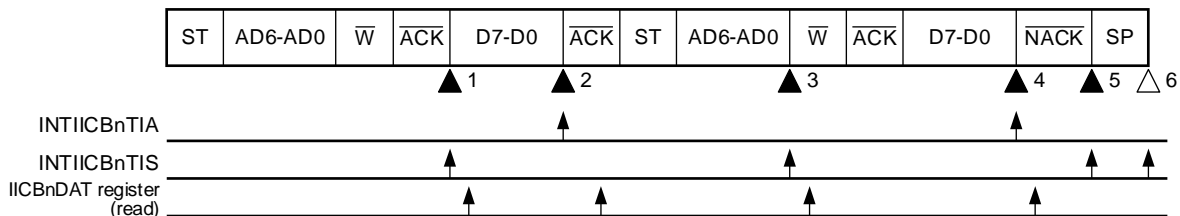
IICBnTRG.IICBnWRET bit = 1

△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)

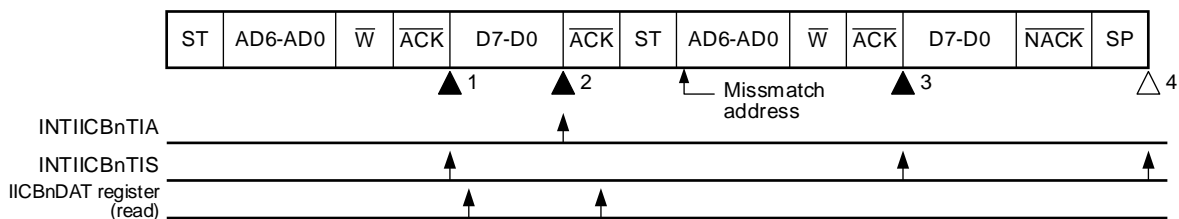


- [▲1: IICBnSTR0 register = 0-100101 0110--00B]
IICBnDAT register read
- [▲2: IICBnSTR0 register = 0-100100 0100--00B]
IICBnDAT register read
- [▲3: IICBnSTR0 register = 0-100100 0110--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
- [▲4: IICBnSTR0 register = 0-100100 0110--00B]
IICBnDAT register 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

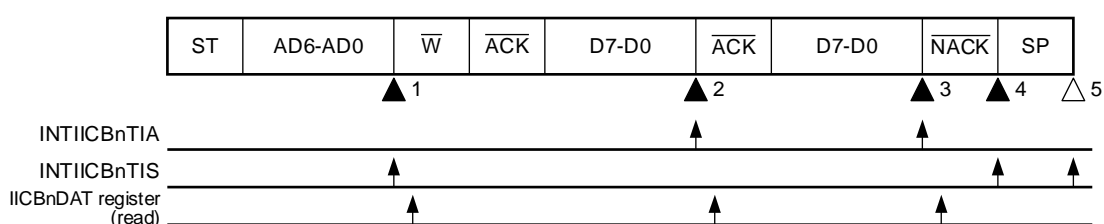
Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.11 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

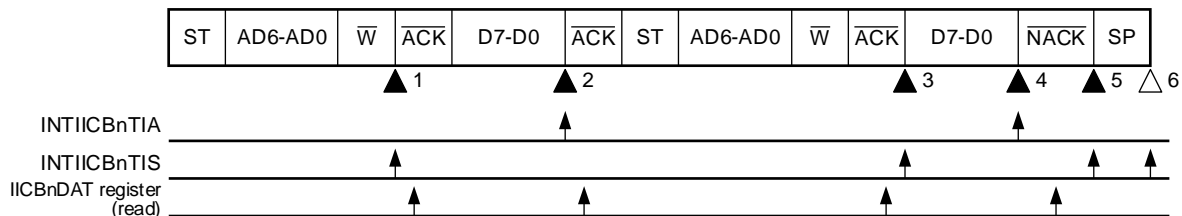
IICBnTRG.IICBnWRET bit = 1

△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)

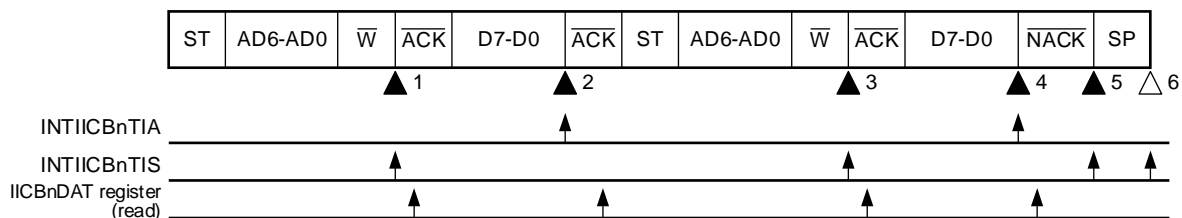


- [▲1: IICBnSTR0 register = 0-101000 0110--00B]
IICBnDAT register read
- [▲2: IICBnSTR0 register = 0-011000 0110--00B]
IICBnDAT register read
- [▲3: IICBnSTR0 register = 0-111001 0100--00B]
IICBnCTL0.IICBnSLAC bit = 0
IICBnDAT register read
- [▲4: IICBnSTR0 register = 0-010100 0110--00B]
IICBnDAT register 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

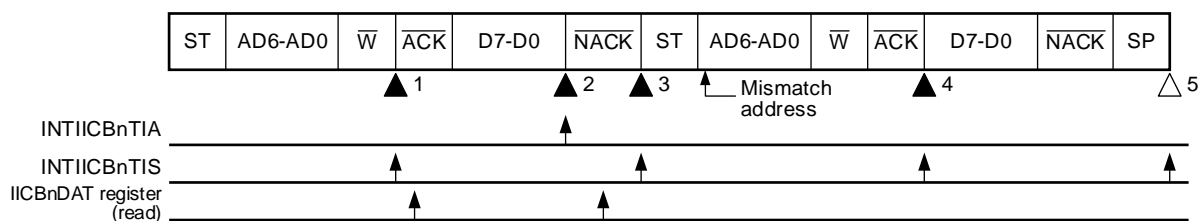
IICBnTRG.IICBnWRET bit = 1

△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

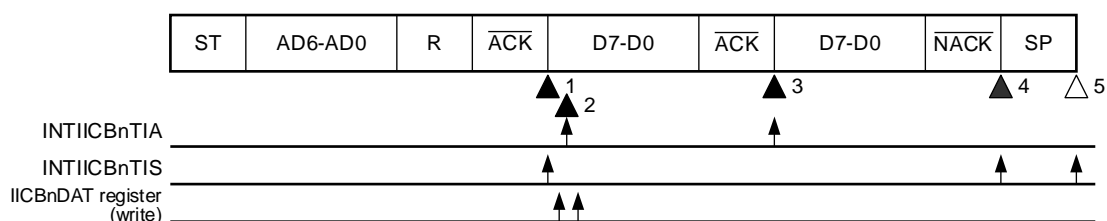
Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.12 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

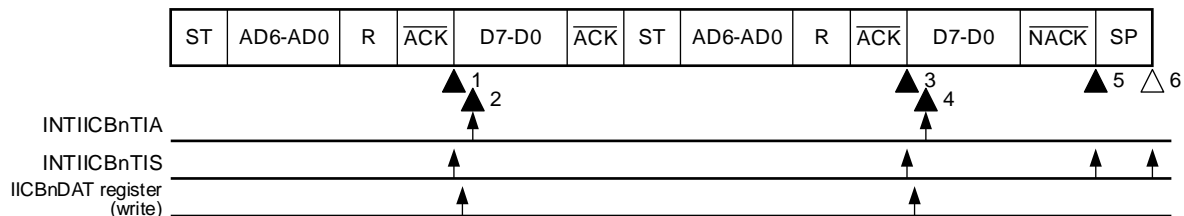


- ▲ 1: IICBnSTR0 register = 0-110111 0110--00B
IICBnDAT register write
- [▲ 2: IICBnSTR0 register = 0-00011X 0100--00B]
IICBnDAT register write
- IICBnSTR0 register = 0-100011X 0100--00B
- ▲ 3: IICBnSTR0 register = 0-000111 0100--00B
- ▲ 4: IICBnSTR0 register = 0-010110 0100--00B
- △ 5: IICBnSTR0 register = 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)

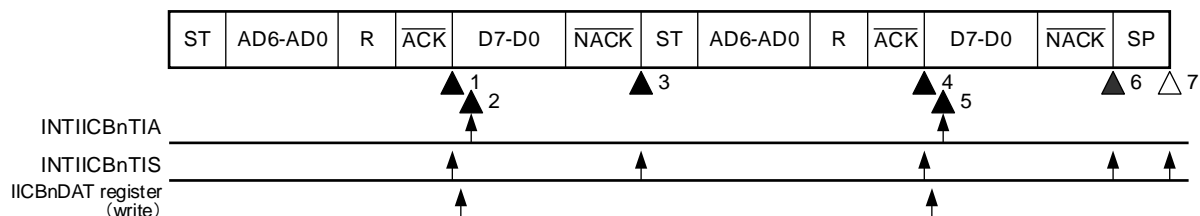


- ▲1: IICBnSTR0 register = 0-010111 0110--00B
IICBnDAT register write
- [▲2: IICBnSTR0 register = 0-00111X 01X0--00B]
- ▲3: IICBnSTR0 register = 0-010111 0110--00B
IICBnDAT register write
- [▲4: IICBnSTR0 register = 0-100101 01X0--00B]
- ▲5: IICBnSTR0 register = 0-110100 0100--00B
- △6: IICBnSTR0 register = 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)

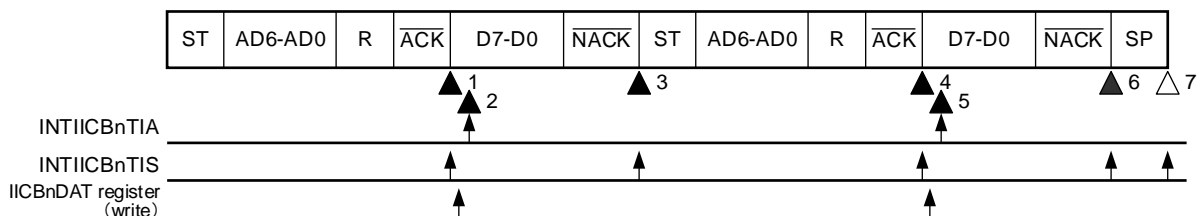


- ▲1: IICBnSTR0 register = 0-110111 0110--00B
IICBnDAT register write
- [▲2: IICBnSTR0 register = 0-100111 0100--00B]
- ▲3: IICBnSTR0 register = 0-111010 0110--00B
- ▲4: IICBnSTR0 register = 0-111010 0110--00B
IICBnDAT register write
- [▲5: IICBnSTR0 register = 0-111011 0110--00B]
- ▲6: IICBnSTR0 register = 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

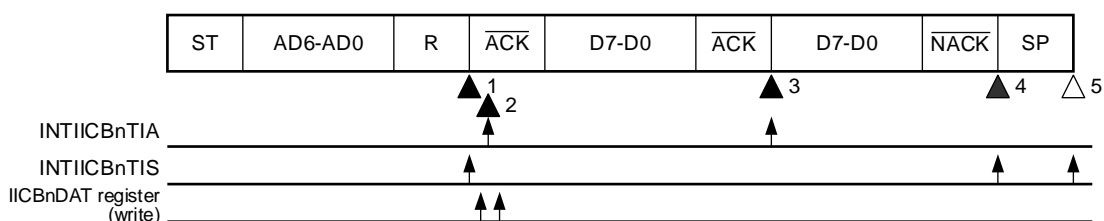
Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.13 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

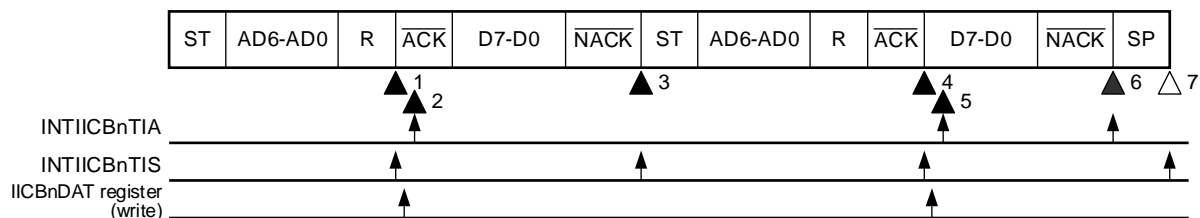


- ▲ 1: IICBnSTR0 register = 0-011010 0110--00B
IICBnDAT register write
- [▲ 2: IICBnSTR0 register = 0-011011 0110--00B]
IICBnDAT register write
- [▲ 3: IICBnSTR0 register = 0-011011 0100--00B]
- ▲ 4: IICBnSTR0 register = 0-111010 0100--00B
- △ 5: IICBnSTR0 register = 0-000010 0001--00B

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)

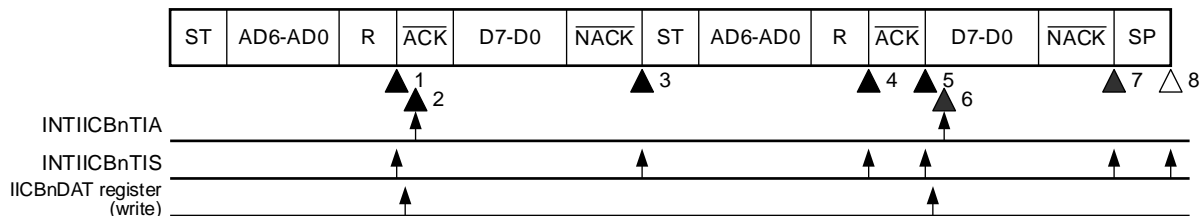


- ▲ 1: IICBnSTR0 register = 0-011000 0110--00B
IICBnDAT register write
- [▲ 2: IICBnSTR0 register = 0-011001 0110--00B]
- ▲ 3: IICBnSTR0 register = 0-011000 0100--00B
- ▲ 4: IICBnSTR0 register = 0-010101 0110--00B
IICBnDAT register write
- [▲ 5: IICBnSTR0 register = 0-010101 0110--00B]
- ▲ 6: IICBnSTR0 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)

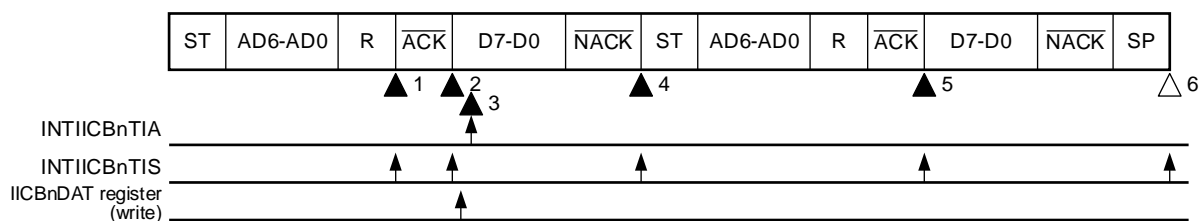


- ▲ 1: IICBnSTR0 register = 0-011000 0110--00B
IICBnDAT register write
- [▲ 2: IICBnSTR0 register = 0-011001 0110--00B]
- ▲ 3: IICBnSTR0 register = 0-011000 0100--00B
- ▲ 4: IICBnSTR0 register = 0-011000 0110--00B
- ▲ 5: IICBnSTR0 register = 0-011001 0110--00B
IICBnDAT register write
- [▲ 6: IICBnSTR0 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))

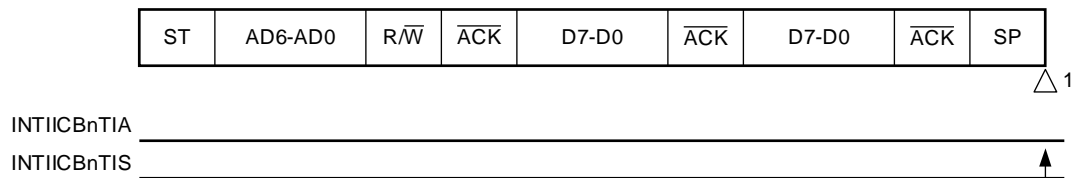


- ▲ 1: IICBnSTR0 register = 0-011000 0110--00B
- ▲ 2: IICBnSTR0 register = 0-011001 0110--00B
- IICBnDAT register write
- [▲ 3: IICBnSTR0 register = 0-011010 0100--00B]
- ▲ 4: IICBnSTR0 register = 0-000000 0100--00B
- ▲ 5: IICBnSTR0 register = 0-000000 0110--00B
- △ 6: IICBnSTR0 register = 0-000000 0001--00B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined

18.8.14 Continuous Transfer Mode (Non-Participation in Communications)

(1) Start – Code – Data – Data – Stop



△1: IICBnSTR0 register = 0-0000X0 0001--00B

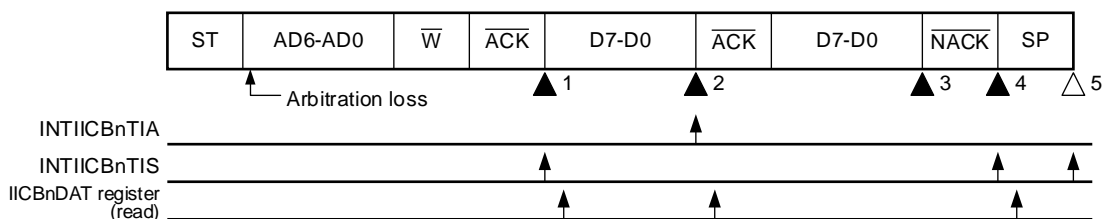
Remark: △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined

18.8.15 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

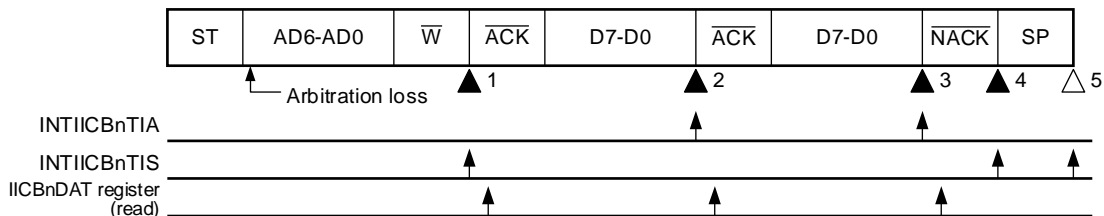
IICBnTRG.IICBnWRET bit = 1

△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

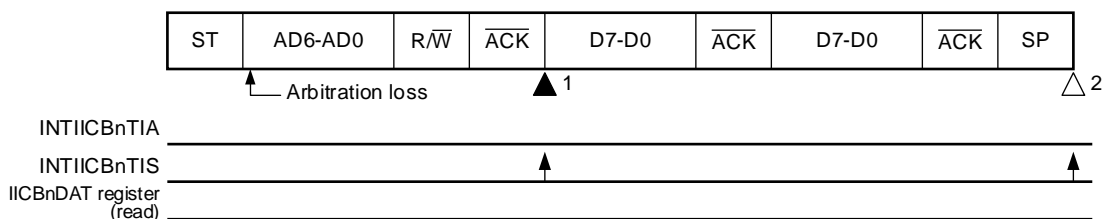
Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined

18.8.16 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



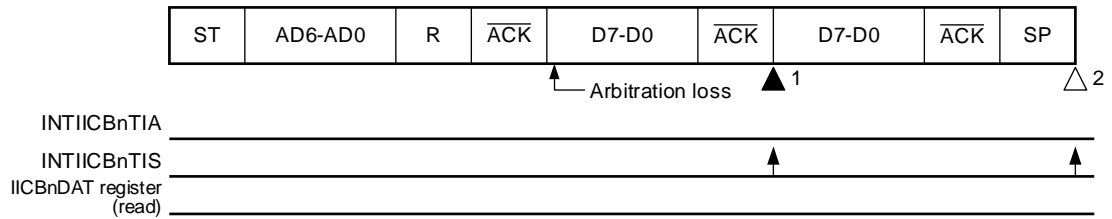
▲ 1: IICBnSTR0 register = 0-000001 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)

△ 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]

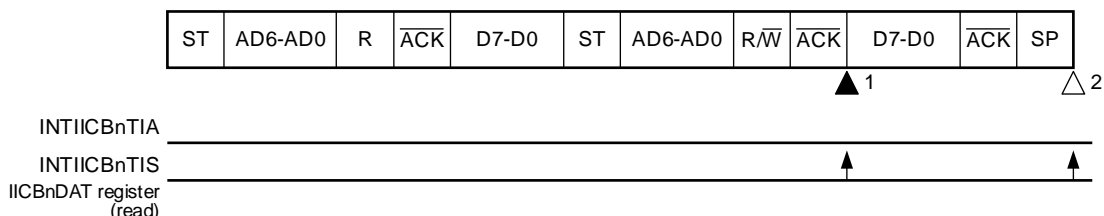
IICBnSTRC.IICBnCLAF bit = 1

△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)

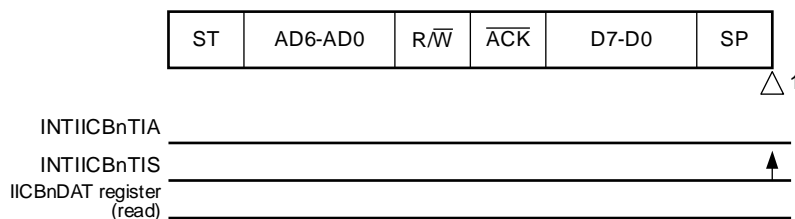


[▲1: IICBnSTR0 register = 0-000001 0100--01B]
 IICBnSTRC.IICBnCLAF bit = 1
 △2: IICBnSTR0 register = 0-000000 0001--00B

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

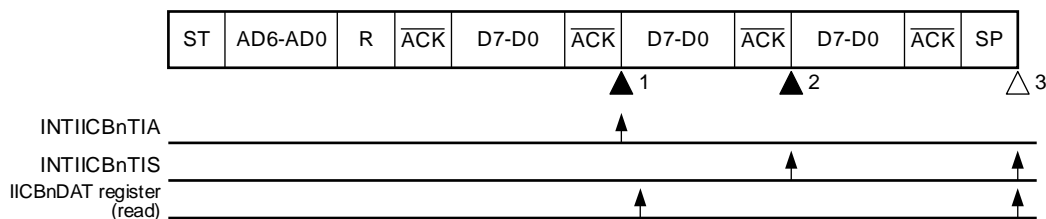


△1: IICBnSTR0 register = 0-000000 0001--01B
 IICBnSTRC.IICBnCLAF bit = 1

Remark: △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined

(5) Arbitration loss because the SDA_n 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

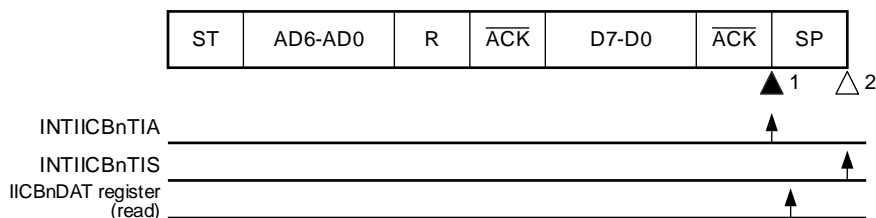
IICBnSTRC.IICBnCLAF bit = 1

△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

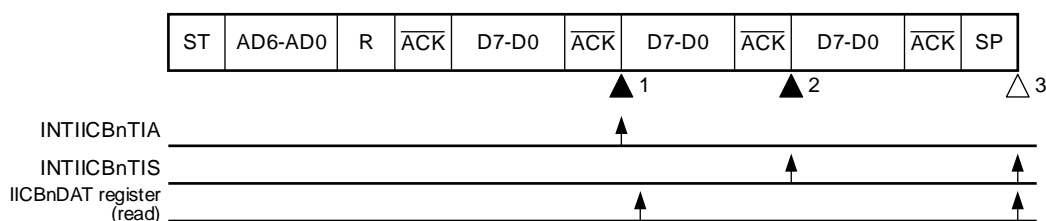
IICBnTRG.IICBnSTT bit = 1

△2: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined

(7) Arbitration loss because the SDA_n 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)]

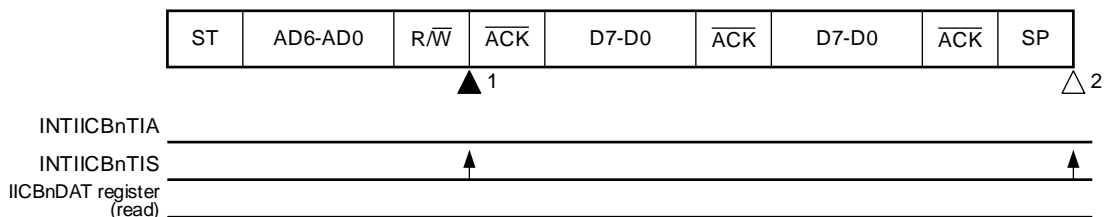
▲3: IICBnSTR0 register = 0-000000 0001--01B

Remark: ▲ Always output
 △ Output only when IICBnCTL0.IICBnSLSI = 1
 - Undefined
 X don't care

18.8.17 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]

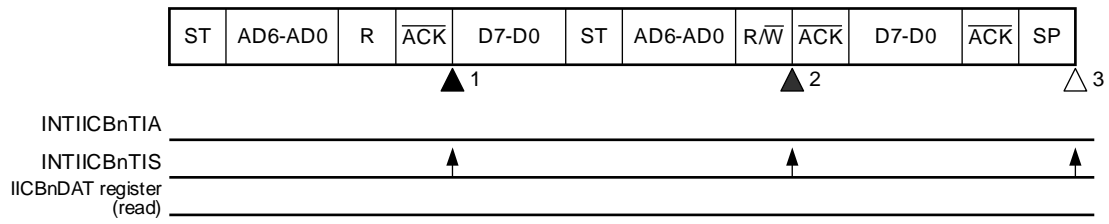
IICBnSTRC.IICBnCLAF bit = 1

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

18.9 Setting Procedure

18.9.1 Single Master Environment

(1) Master operation setting procedure during single transfer mode

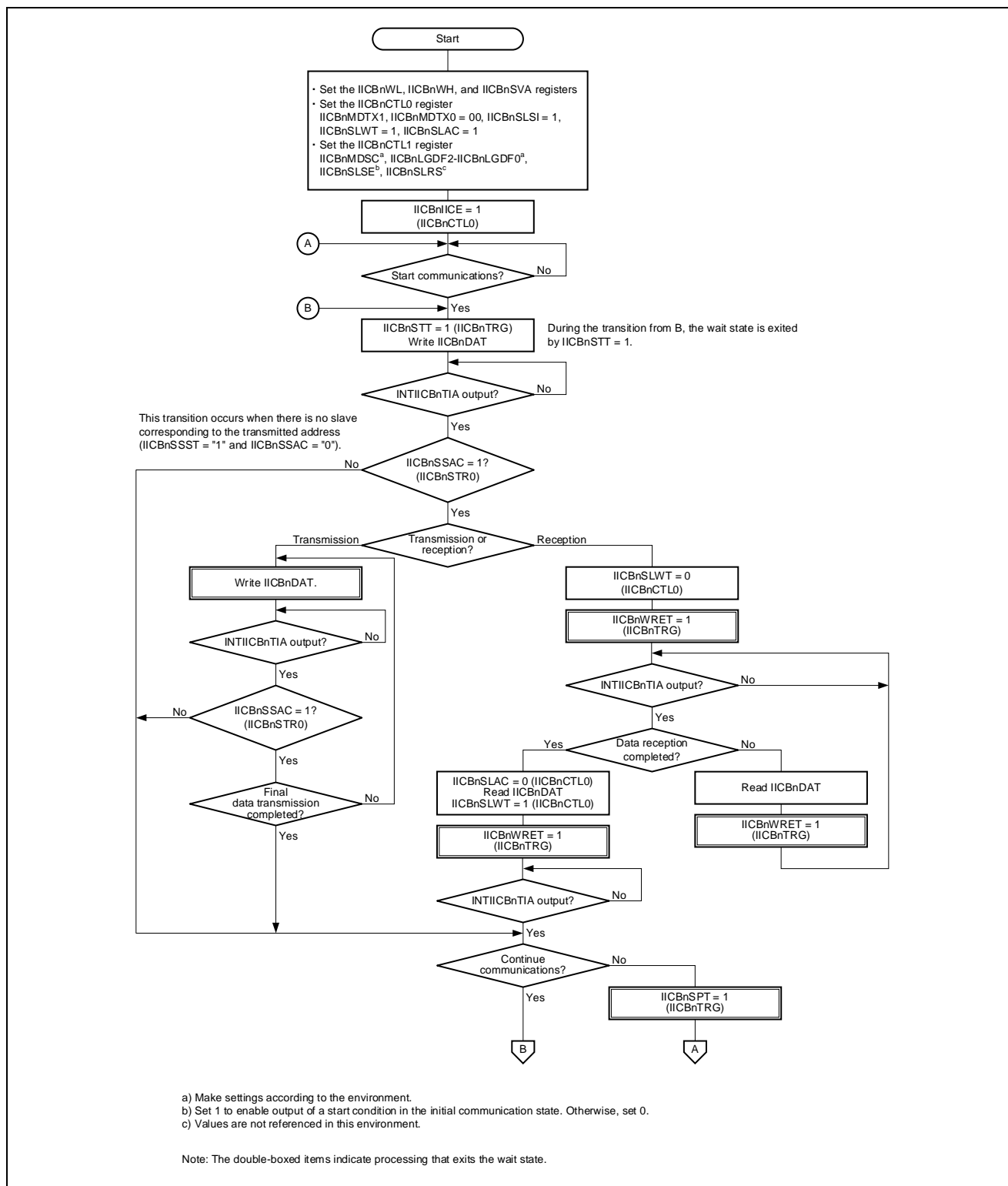


Figure 18.14 Master Operation Setting Procedure during Single Transfer Mode (Single Master Environment)

(2) Slave operation setting procedure during single transfer mode

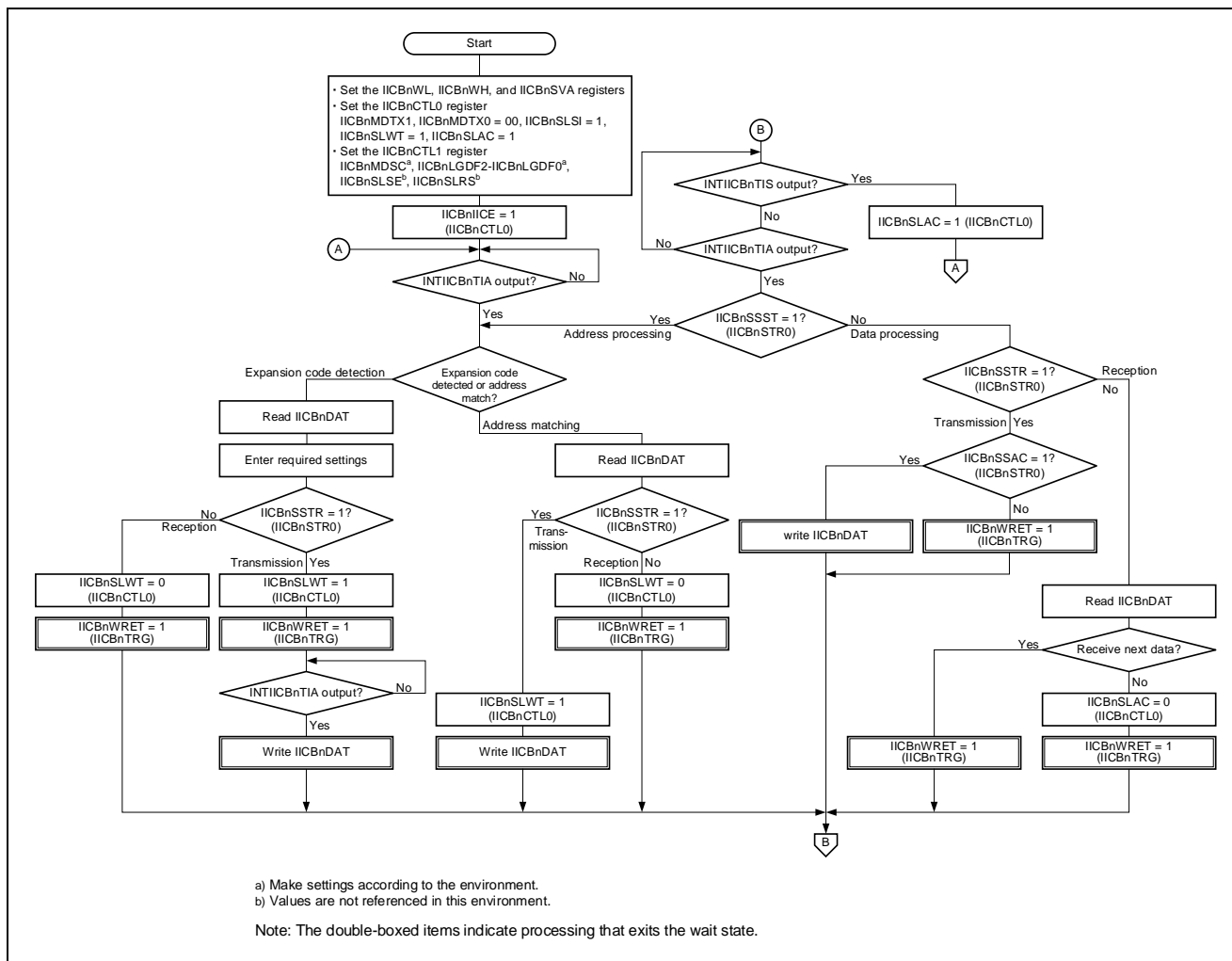


Figure 18.15 Slave Operation Setting Procedure during Single Transfer Mode (Single Master Environment)

(3) Master operation setting procedure during continuous transfer mode

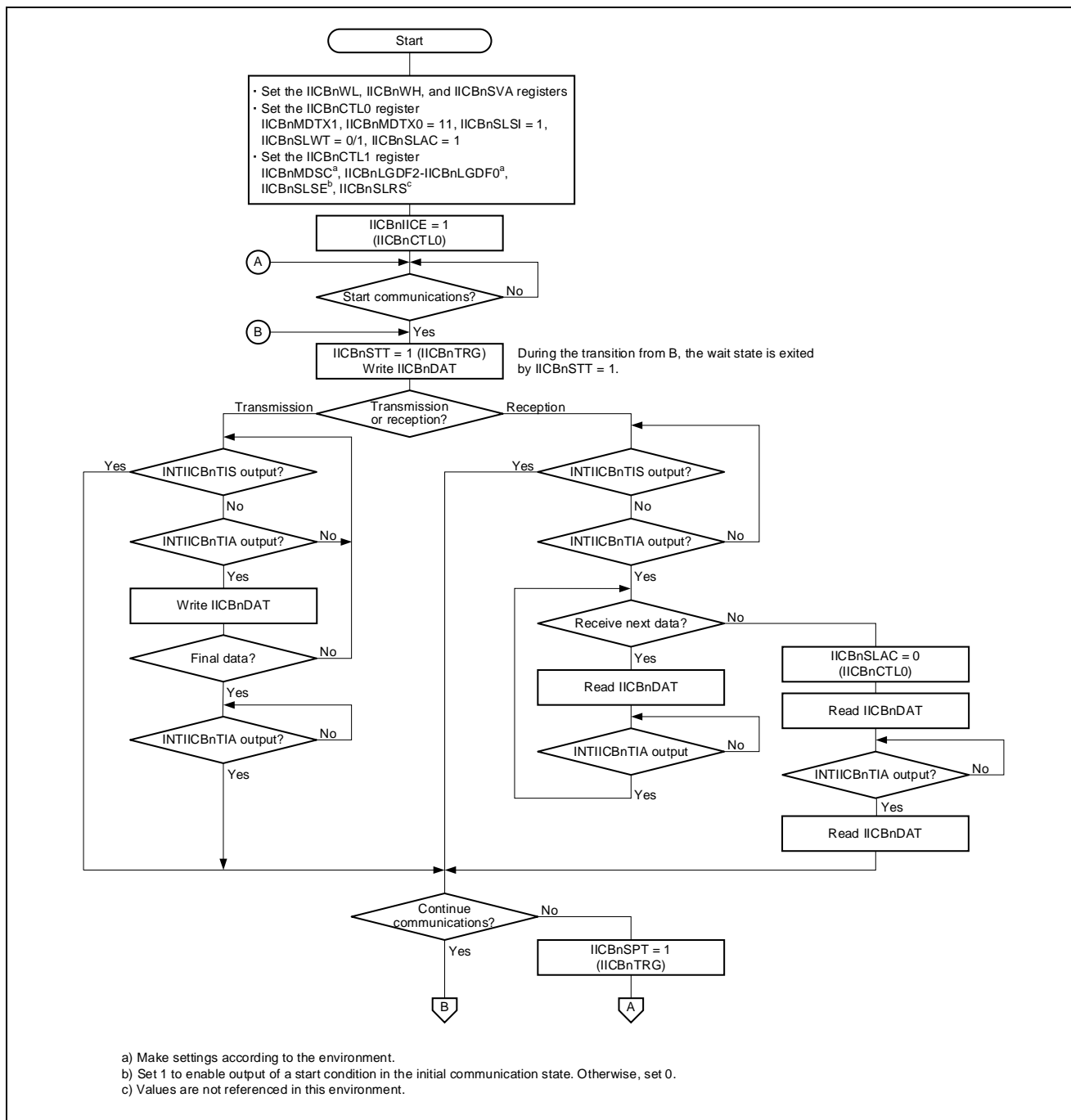


Figure 18.16 Master Operation Setting Procedure during Continuous Transfer Mode (Single Master Environment)

(4) Slave operation setting procedure during continuous transfer mode

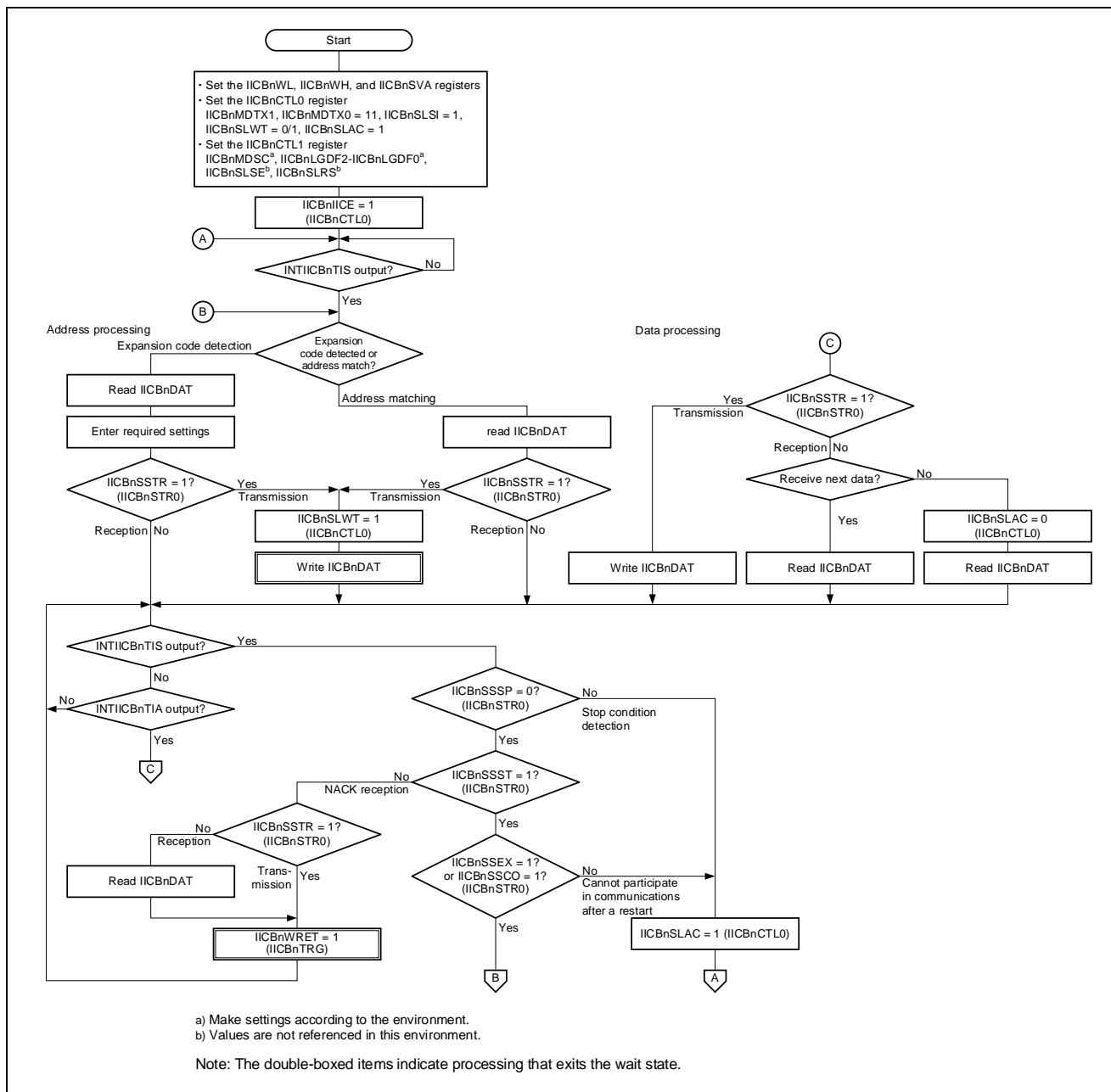


Figure 18.17 Slave Operation Setting Procedure during Continuous Transfer Mode (Single Master Environment)

18.9.2 Multi-Master Environment

(1) Single transfer mode setting procedure when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0)

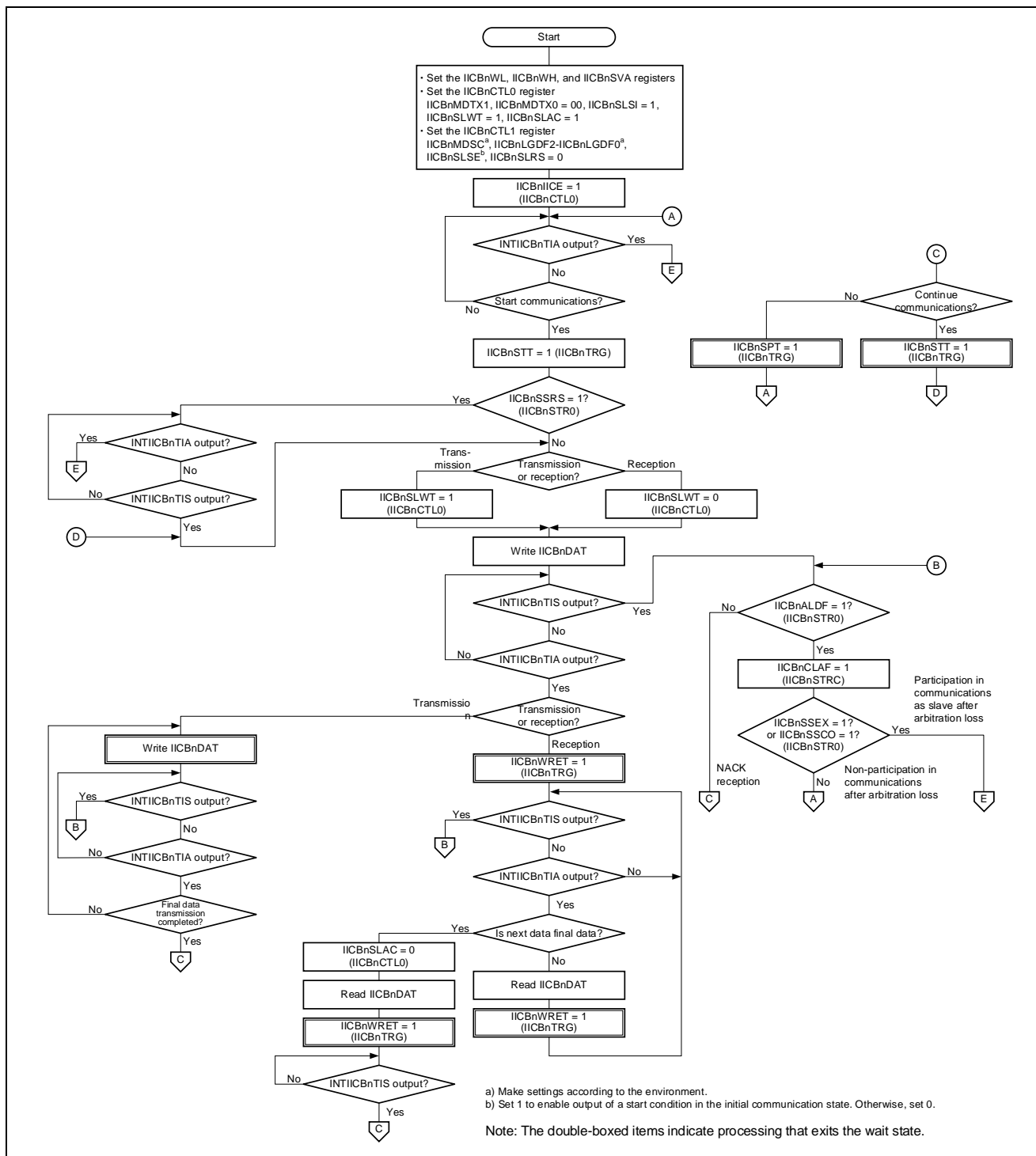
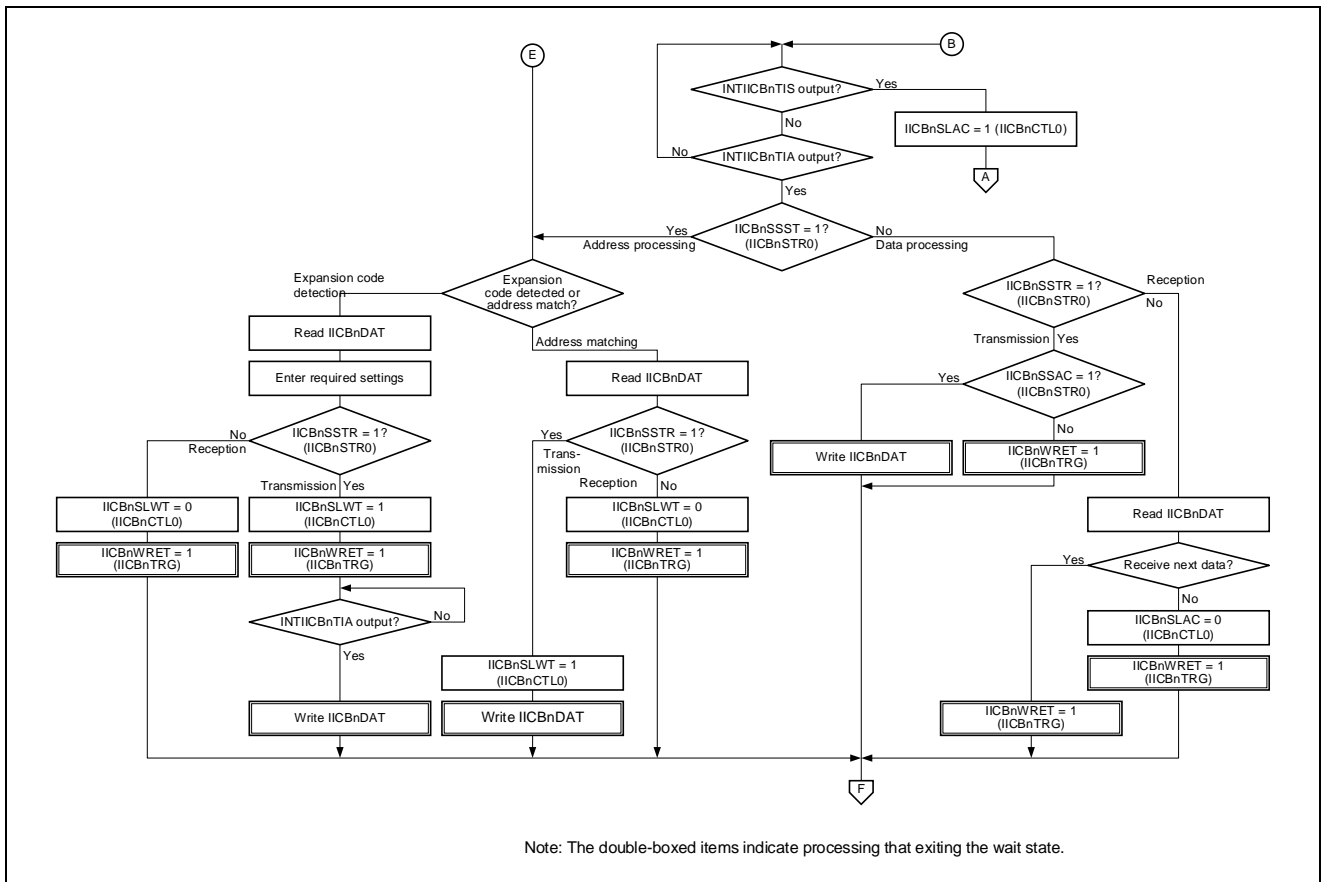


Figure 18.18 Single Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (IICBnCTL1.IICBnSLRS bit = 0) (Multi-Master Environment) (1/2)



Note: The double-boxed items indicate processing that exits the wait state.

Figure 18.18 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)

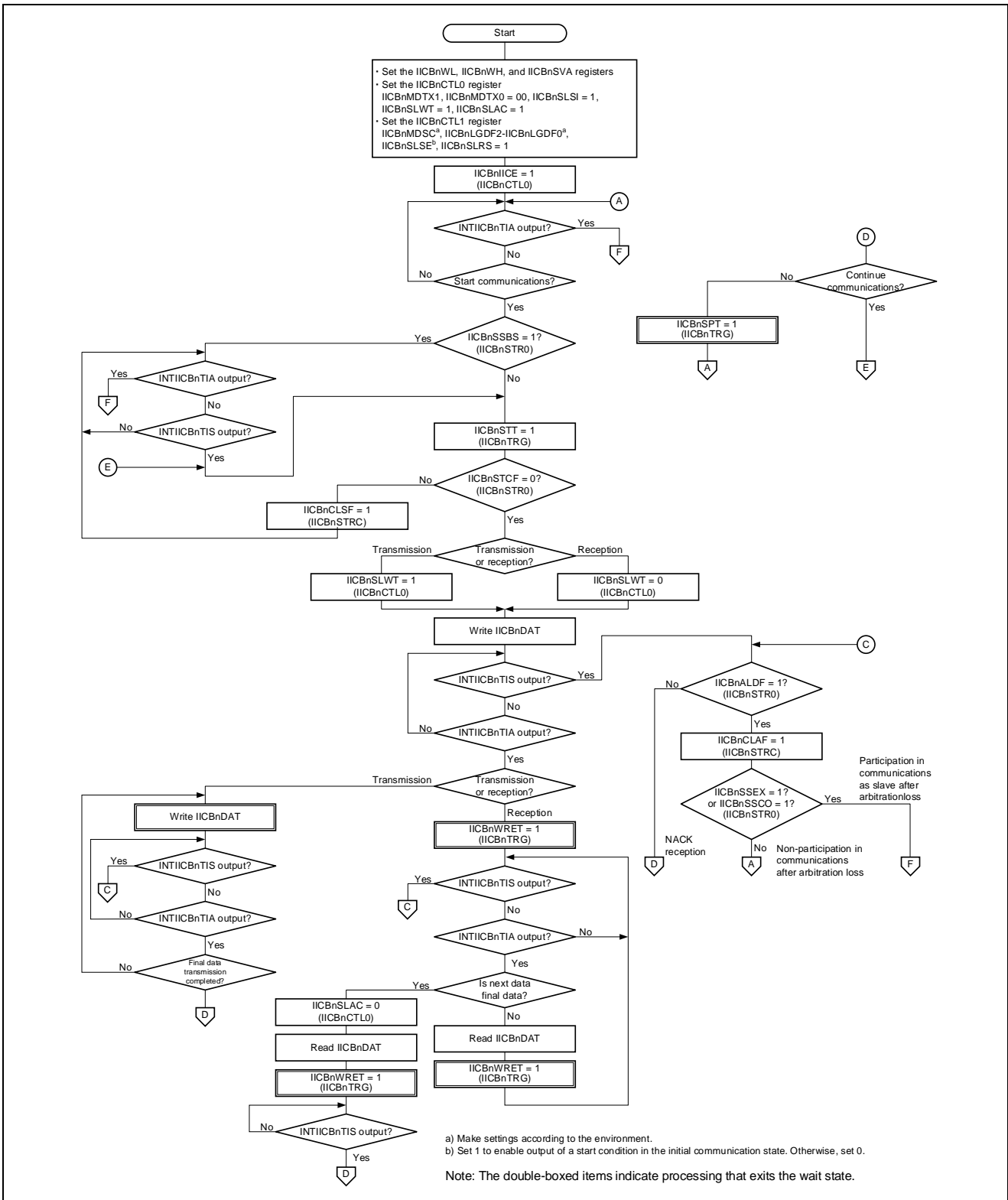


Figure 18.19 Single Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (1/2)

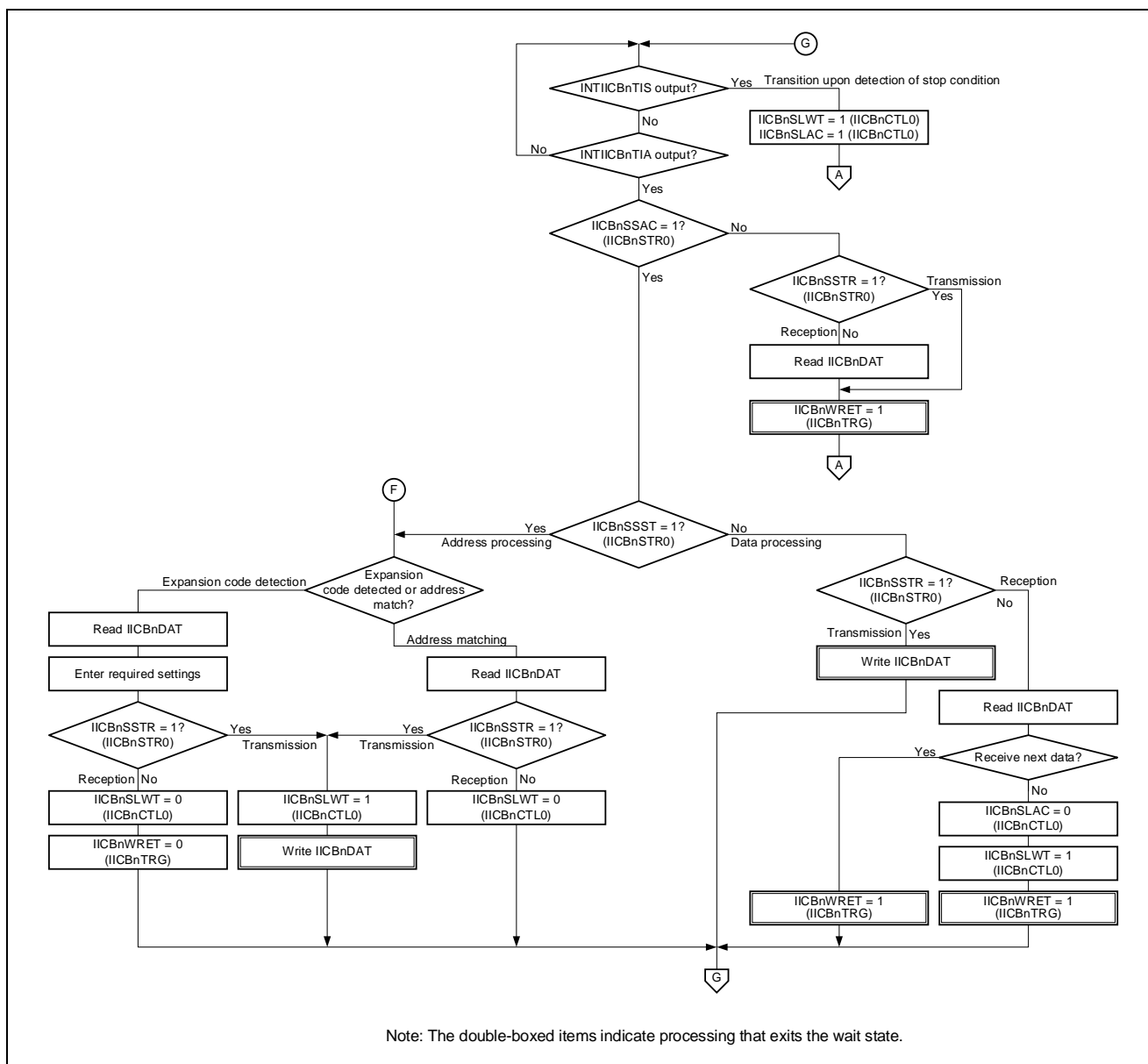


Figure 18.19 Single Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (2/2)

(3) Continuous transfer mode setting procedure when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0)

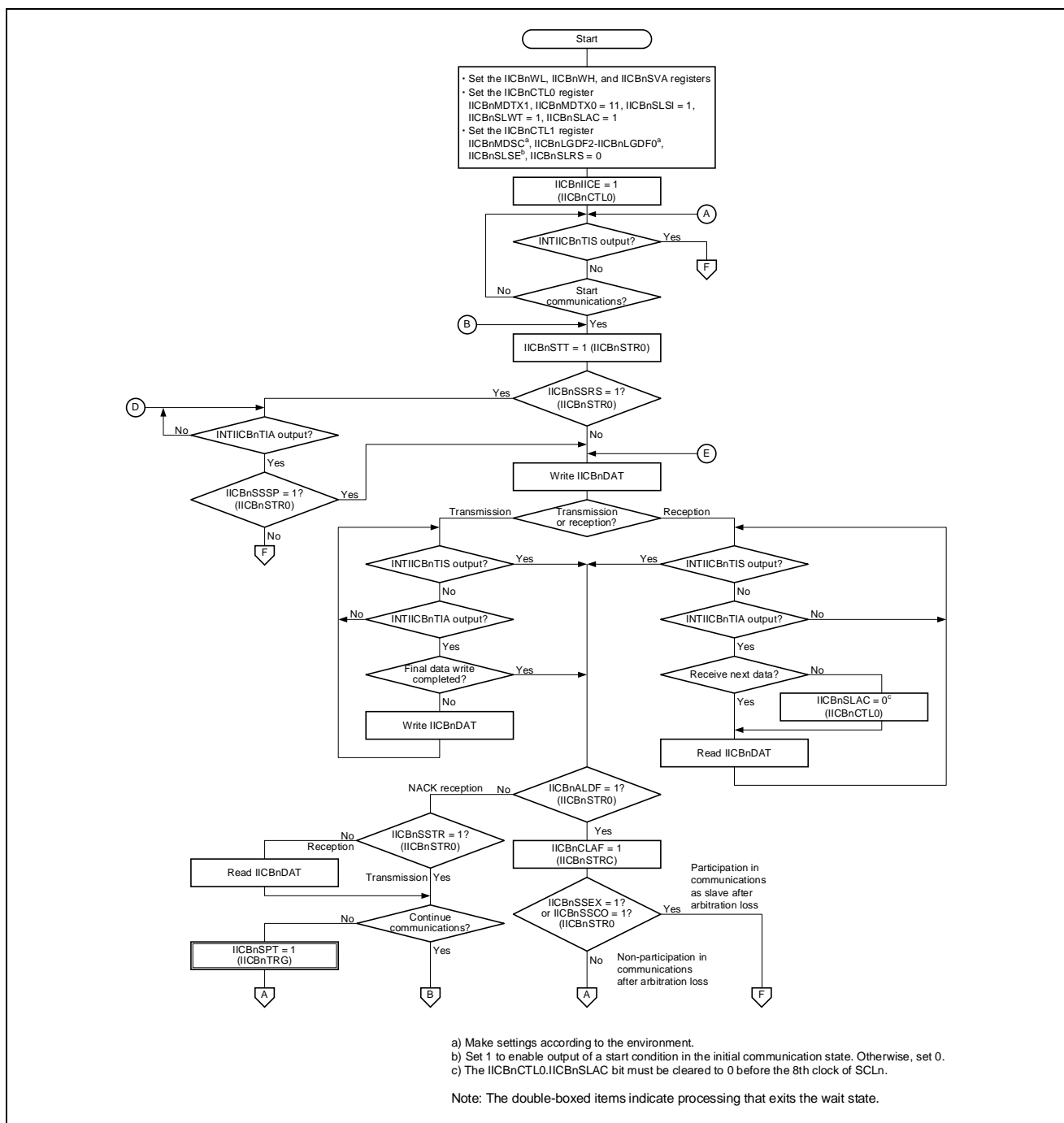


Figure 18.20 Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (IICBnCTL1.IICBnSLRS bit = 0) (Multi-Master Environment) (1/2)

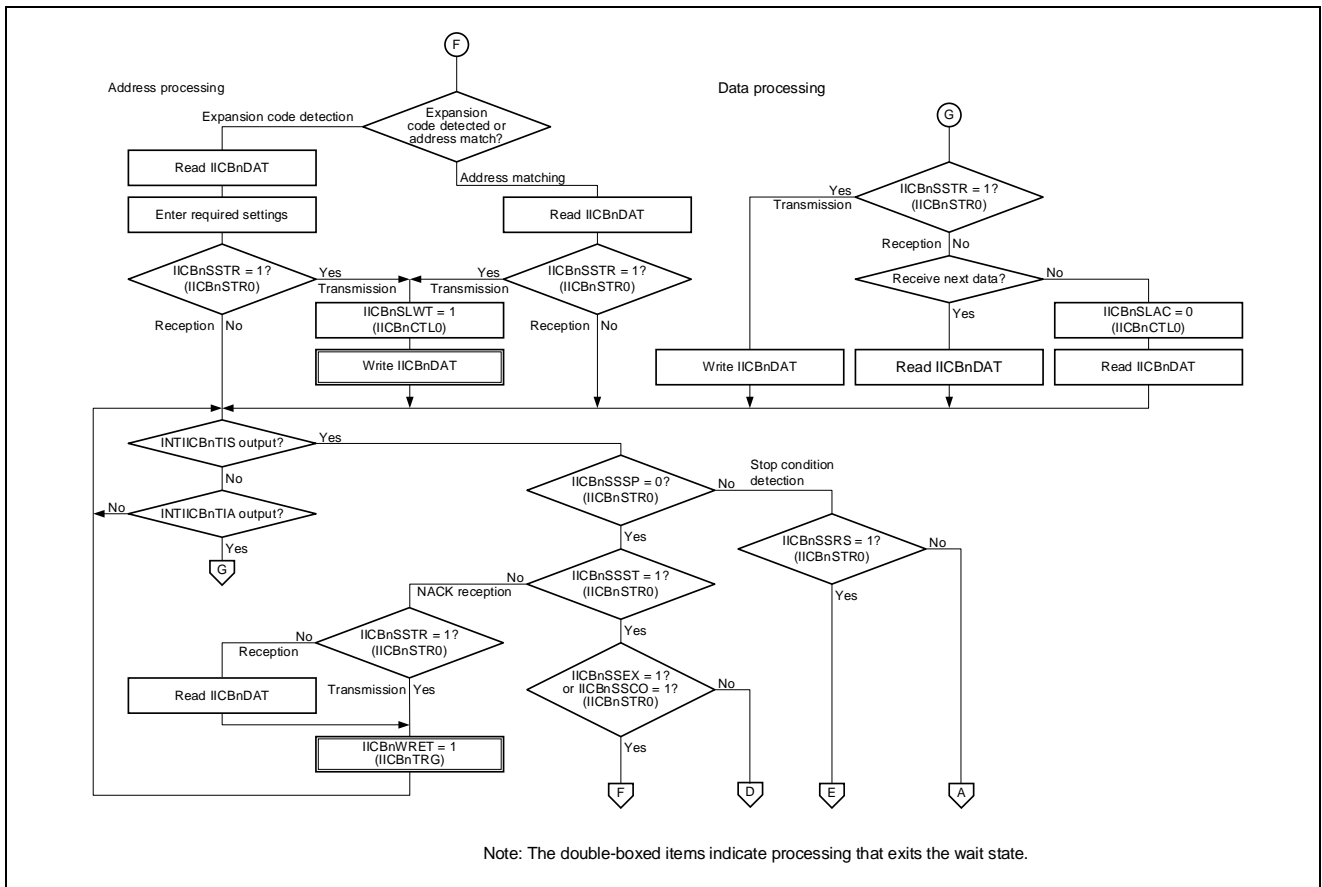
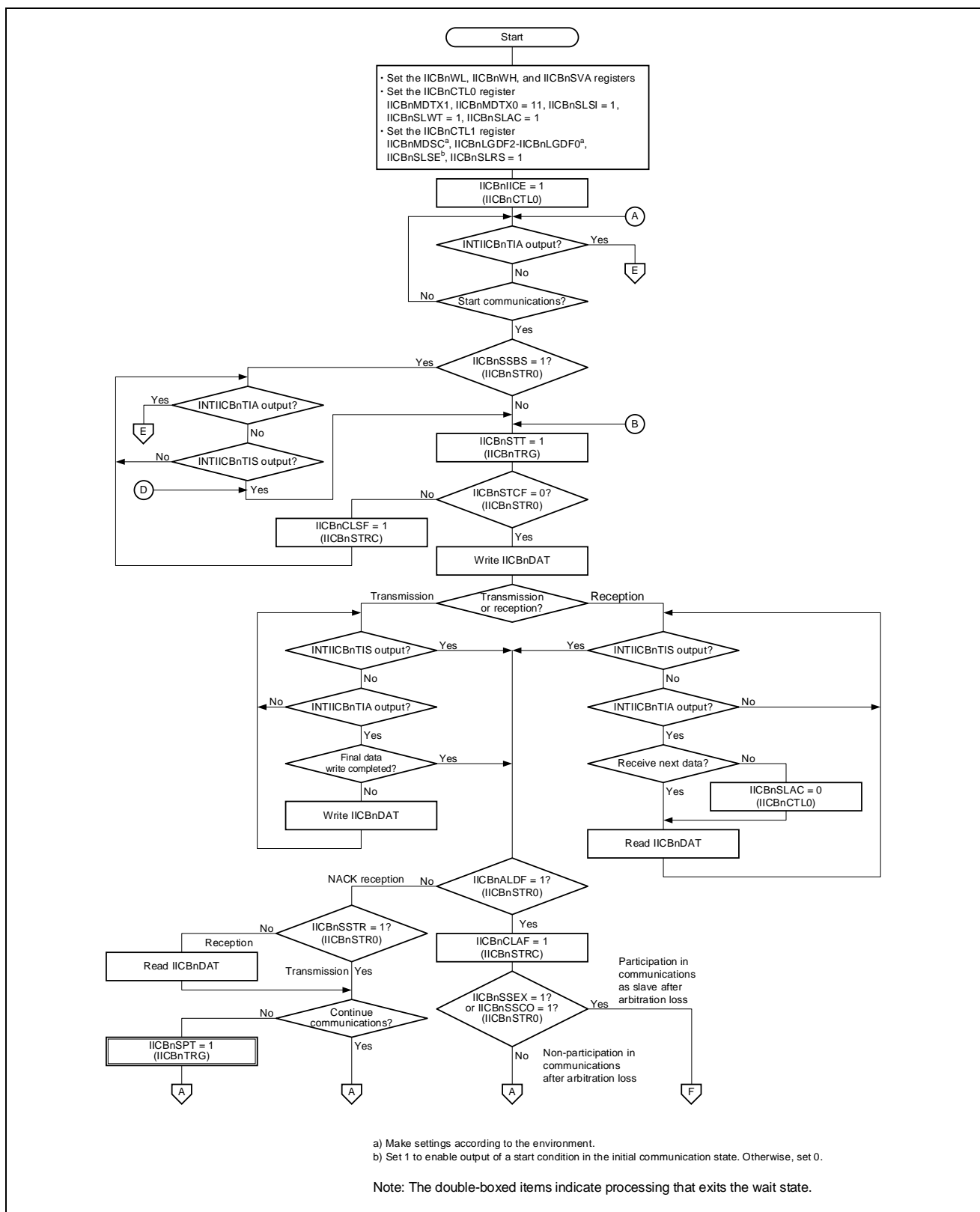


Figure 18.20 Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Enabled (IICBnCTL1.IICBnSLRS bit = 0) (Multi-Master Environment) (2/2)

(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 18.21 Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (1/2)

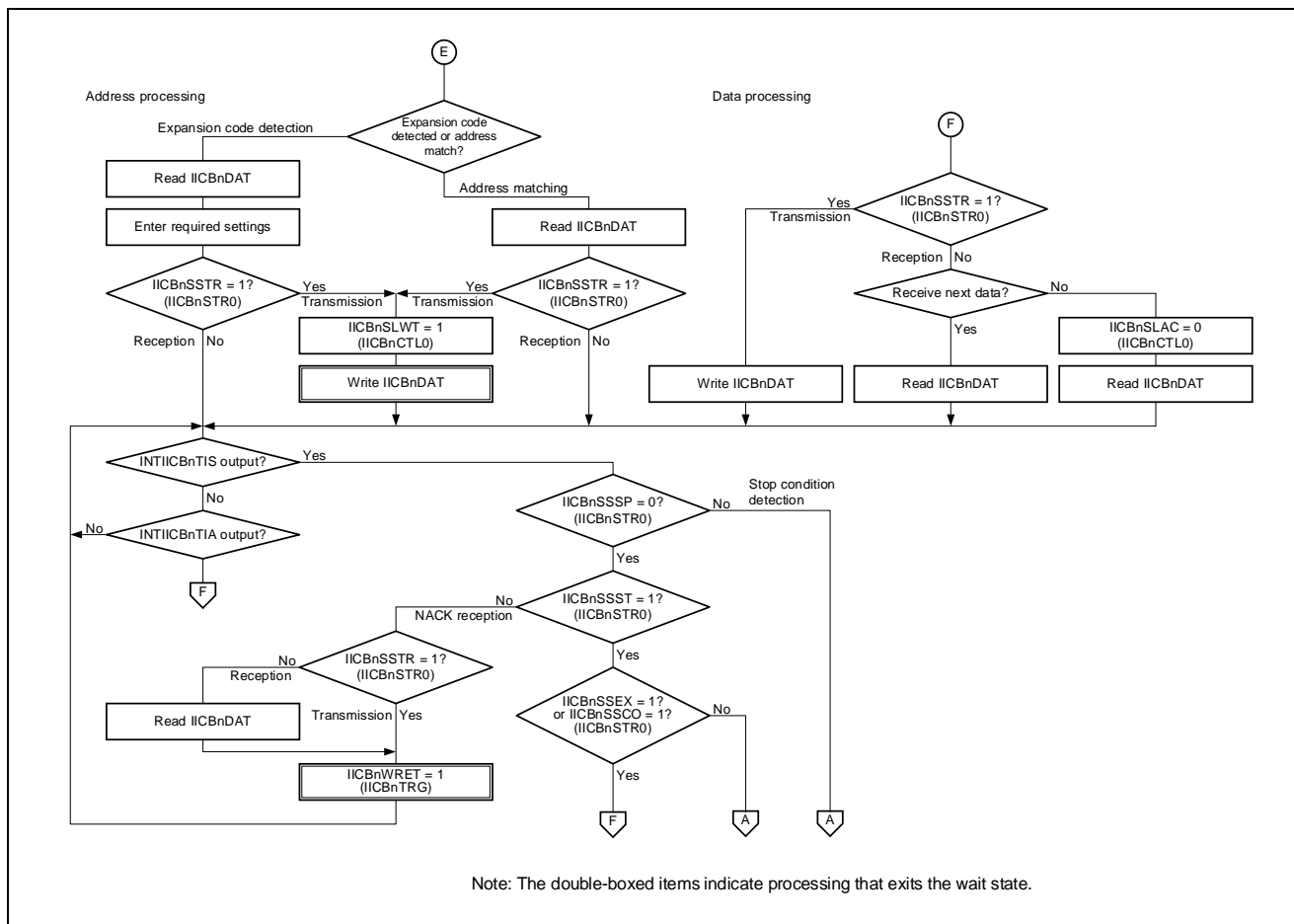


Figure 18.21 Continuous Transfer Mode Setting Procedure when Communication Reserve Function is Disabled (IICBnCTL1.IICBnSLRS bit = 1) (Multi-Master Environment) (2/2)

19. CAN Controller (FCN)

This section explains the CAN (Controller Area Network) controllers that comply with the CAN protocol as standardized in ISO 11898.

19.1 Features of FCN

This product has the following number of channels of the CAN controller.

Table 19.1 Channels of FCN

FCN	
Number of channels	2
Name	FCN0, FCN1

- Meaning of “n”:

Throughout 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 19.2 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), and for updating the real-time port pins (RP00-RP37).

Table 19.3 FCNn Interrupts and Requests for Peripheral Modules

FCNn Interrupt Signal	Function	Connected To
FCN0		
INTC0ERR	FCN0 error detection	<ul style="list-style-type: none"> • Interrupt controller INTFCN0ERR • HW-RTOS (Hardware ISR)
INTC0REC	FCN0 reception completion	<ul style="list-style-type: none"> • Interrupt controller INTFCN0REC • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTC0TRX	FCN0 transmission completion	<ul style="list-style-type: none"> • Interrupt controller INTFCN0TRX • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTC0WUP	FCN0 sleep wake-up/transmission abortion	<ul style="list-style-type: none"> • Interrupt controller INTFCN0WUP • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
FCN1		
INTC1ERR	FCN1 error detection	<ul style="list-style-type: none"> • Interrupt controller INTFCN1ERR • HW-RTOS (Hardware ISR)
INTC1REC	FCN1 reception completion	<ul style="list-style-type: none"> • Interrupt controller INTFCN1REC • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTC1TRX	FCN1 transmission completion	<ul style="list-style-type: none"> • Interrupt controller INTFCN1TRX • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)
INTC1WUP	FCN1 sleep wake-up/transmission abortion	<ul style="list-style-type: none"> • Interrupt controller INTFCN1WUP • DMA controller trigger (DTFR/RTDTFR) • Timer capture trigger (TMTFR) • Real-time port trigger (RPTFR) • HW-RTOS (Hardware ISR)

- I/O signals:

The I/O signals of the CAN controllers are listed in the Table 19.4.

Table 19.4 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)

19.2 Features

- Compliant with ISO 11898
- Standard frame and extended frame transmission/reception
- Transfer rate: up to 1 Mbps (If FCN clock input \geq 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.

19.2.1 Overview of Functions

Table 19.5 lists an overview of the CAN controller functions.

Table 19.5 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Remote frame processing by transmit message buffer • Remote frame processing by receive message buffer, when applying one of the 8 masks
Timestamping	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Forced release from bus-off possible by software. • No automatic release from bus-off (software must send recovery request).
Power save mode	<ul style="list-style-type: none"> • CAN sleep mode (can be woken up by CAN bus) • CAN stop mode (cannot be woken up by CAN bus)

19.2.2 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.

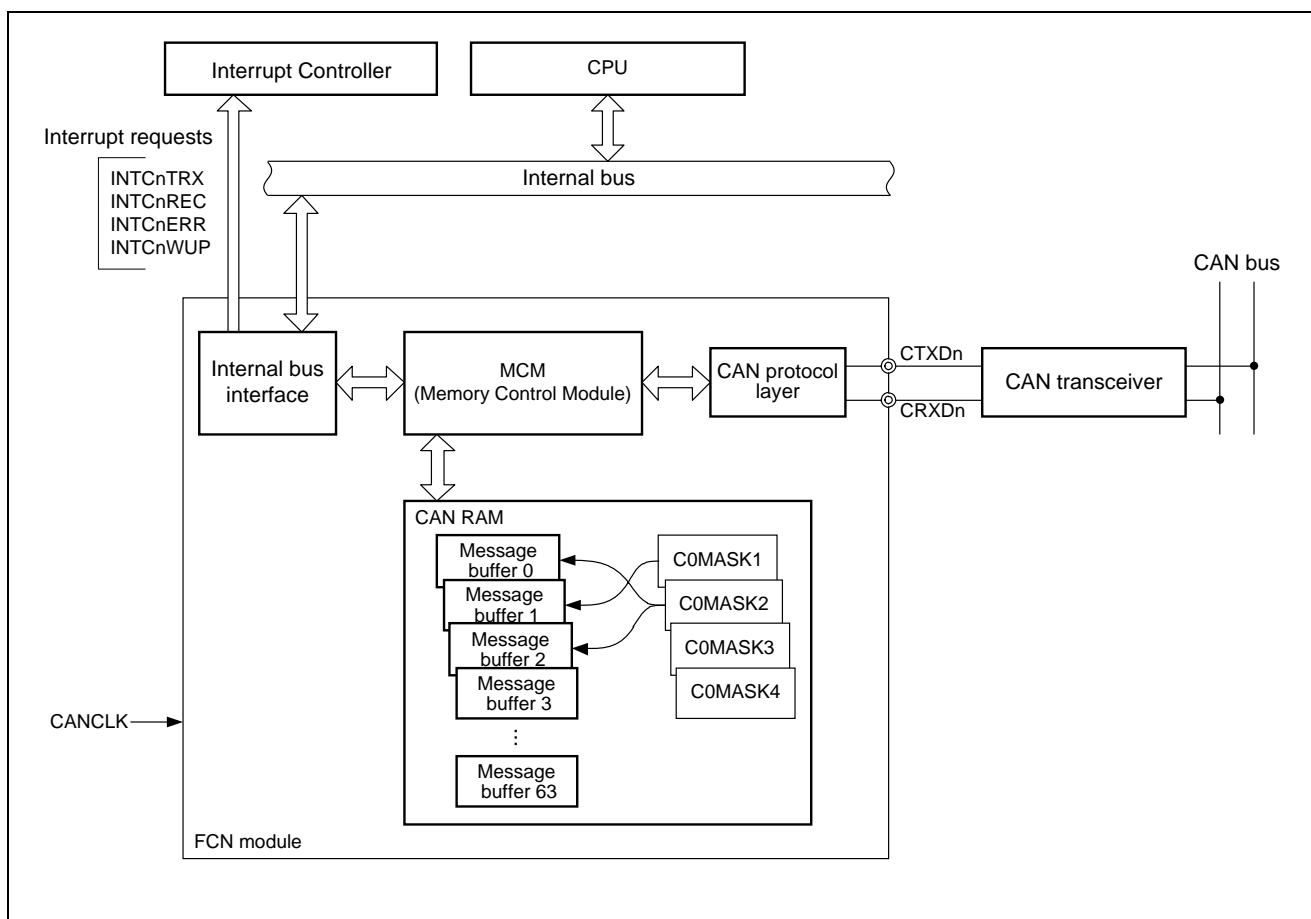


Figure 19.1 Block Diagram of the CAN Controller

19.3 Internal Registers of FCN

19.3.1 CAN Controller Configuration

Table 19.6 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 19.6 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)

19.3.2 CAN Controller Registers Overview

(1) FCNn Global and Module Registers

Table 19.7 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 ^{Note 1}
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	Note 2
4003 00D0H	FCN0 global data update bit monitor register 1	FCN0DNBMRX1	R	32	Note 2
4002 8300H	FCN0 module mask 1 register	FCN0CMMKCTL01H	R/W	16	Note 2
4002 8308H		FCN0CMMKCTL02H			
4003 0300H		FCN0CMMKCTL01W		32	
4002 8310H	FCN0 module mask 2 register	FCN0CMMKCTL03H	R/W	16	Note 2
4002 8318H		FCN0CMMKCTL04H			
4003 0310H		FCN0CMMKCTL03W		32	
4002 8320H	FCN0 module mask 3 register	FCN0CMMKCTL05H	R/W	16	Note 2
4002 8328H		FCN0CMMKCTL06H			
4003 0320H		FCN0CMMKCTL05W		32	
4002 8330H	FCN0 module mask 4 register	FCN0CMMKCTL07H	R/W	16	Note 2
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 19.7 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	FCN0 DAT0H	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 19.8 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 ^{Note 1}
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	Note 2
4005 00D0H	FCN1 global data update bit monitor register 1	FCN1DNBMRX1	R	32	Note 2
4004 8300H	FCN1 module mask 1 register	FCN1CMMKCTL01H	R/W	16	Note 2
4004 8308H		FCN1CMMKCTL02H			
4005 0300H		FCN1CMMKCTL01W		32	
4004 8310H	FCN1 module mask 2 register	FCN1CMMKCTL03H	R/W	16	Note 2
4004 8318H		FCN1CMMKCTL04H			
4005 0310H		FCN1CMMKCTL03W		32	
4004 8320H	FCN1 module mask 3 register	FCN1CMMKCTL05H	R/W	16	Note 2
4004 8328H		FCN1CMMKCTL06H			
4005 0320H		FCN1CMMKCTL05W		32	
4004 8330H	FCN1 module mask 4 register	FCN1CMMKCTL07H	R/W	16	Note 2
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 19.8 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.

19.3.3 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 19.9 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 19.10 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 19.11 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 19.12 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	FCNnCMCLCSTR (W)	0	0	0	0	0	0	0	0	
	FCNnCMCLCSTR (R)	0	0	0	0	0	FCN0CMCLCSSLC[2:0]			
0 024CH	FCNnCMINSTR	0	0	0	FCNnCMINBOFF	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
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 RGCLR	
		0	0	0	0	0	0	0	0	
	FCNnCMRGRX (R)	0	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	0	FCNnCM TGSSPM	FCNnCM TGTVFF
		FCNnCMTGSSPT[7:0]								
0 8298H	FCNnCMTSCTL (W)	0	0	0	0	0	FCNnCM TSCLK	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 19.13 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 x 40H	FCNnMmDAT0W	FCNnMmSSD[07:00]							
		FCNnMmSSD[17:10]							
		FCNnMmSSD[27:00]							
		FCNnMmSSD[37:30]							
0 9000H + m x 40H	FCNnMmDAT0H	FCNnMmSSD[07:00]							
		FCNnMmSSD[17:10]							
0 1000H + m x 40H	FCNnMmDAT0B	FCNnMmSSD[07:00]							
0 1004H + m x 40H	FCNnMmDAT1B	FCNnMmSSD[17:10]							
0 9008H + m x 40H	FCNnMmDAT2H	FCNnMmSSD[27:20]							
		FCNnMmSSD[37:30]							
0 1008H + m x 40H	FCNnMmDAT2B	FCNnMmSSD[27:20]							
0 100CH + m x 40H	FCNnMmDAT3B	FCNnMmSSD[37:30]							
1 1010H + m x 40H	FCNnMmDAT4W	FCNnMmSSD[47:40]							
		FCNnMmSSD[57:50]							
		FCNnMmSSD[67:60]							
		FCNnMmSSD[77:70]							
0 9010H + m x 40H	FCNnMmDAT4H	FCNnMmSSD[47:40]							
		FCNnMmSSD[57:50]							
0 1010H + m x 40H	FCNnMmDAT4B	FCNnMmSSD[47:40]							
0 1014H + m x 40H	FCNnMmDAT5B	FCNnMmSSD[57:50]							
0 9018H + m x 40H	FCNnMmDAT6H	FCNnMmSSD[67:60]							
		FCNnMmSSD[77:70]							
0 1018H + m x 40H	FCNnMmDAT6B	FCNnMmSSD[67:60]							
0 101CH + m x 40H	FCNnMmDAT7B	FCNnMmSSD[77:70]							

Table 19.13 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 x 40H	FCNnMmDTLGB	0				FCNnMmDTLG[3:0]				
0 1024H + m x 40H	FCNnMmSTRB	FCNnMmSSOW	FCNnMmSSMT[3:0]				FCNnMmSSRT	0	FCNnMmSSAM	
0 9028H + m x 40H	FCNnMmMID0H	FCNnMmSSID[7:0]								
		FCNnMmSSID[15:8]								
0 9030H + m x 40H	FCNnMmMID1H	FCNnMmSSID[23:16]								
		FCNnMmSSIE	0	0	FCNnMmSSID[28:24]					
1 1028H + m x 40H	FCNnMmMID0W	FCNnMmSSID[7:0]								
		FCNnMmSSID[15:8]								
		FCNnMmSSID[23:16]								
		FCNnMmSSIE	0	0	FCNnMmSSID[28:24]					
0 9038H + m x 40H	FCNnMmCTL (W)	0	FCNnMmCLNH	0	FCNnMmCLMW	FCNnMmCLIE	FCNnMmCLDN	FCNnMmCLTR	FCNnMmCLRY	
		0	FCNnMmSENH	0	0	FCNnMmSEIE	0	FCNnMmSETR	FCNnMmSERY	
	FCNnMmCTL (R)	0	FCNnMmNHMF	0	FCNnMmMOWF	FCNnMmIENF	FCNnMmDTNF	FCNnMmTRQF	FCNnMmRDYF	
		0	0	FCNnMmMUCF	0	0	0	FCNnMmTCPF	0	

19.4 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 19.14 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							

19.5 Control Registers

19.5.1 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 $\langle \text{FCNn_base} \rangle + 0\ 8000\text{H}$
- Initial Value $00\text{x}0\text{H}$ ^{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	FCNnGMCLE CCF	FCNnGM CLSORF	0	0	FCNnGM CLEUDE	FCNnGM CLPWOM

Bit Position	Bit Name	Description
15	FCNnGMCLSSMO	<p>Enables access to the FCN message buffer register and transmit/receive history registers</p> <p>0: Write access and read access to the FCN message buffer register and the transmit/receive history list registers are disabled.</p> <p>1: Write access and read access to the FCN message buffer register and the transmit/receive history list registers are enabled.</p> <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;"> <p>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.</p> <p>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.</p> </div> <div style="background-color: #e0ffff; padding: 5px; margin-top: 10px;"> <p>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.</p> <p>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.</p> </div>

(2/2)

Bit Position	Bit Name	Description
5	FCNnGMCLECCF	<p>Message buffer RAM read error detect bit</p> <p>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.</p>
4	FCNnGMCLSORF	<p>Software reset execution status bit</p> <p>0: No software reset 1: Software reset is ongoing.</p> <p>Remarks</p> <ol style="list-style-type: none"> 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. When FCNnGMCLCTL.FCNnGMCLSORF is set to 1, the initialization of the message buffer RAM starts. When FCNnGMCLCTL.FCNnGMCLSORF already set to 1 is set to 1 again, software reset processing does not restart, but continues. After releasing hardware reset, FCNnGMCLCTL.FCNnGMCLSORF is automatically set to 1 and initialization of the message buffer RAM starts. Clearing FCNnGMCLCTL.FCNnGMCLPWOM (0) and setting FCNnGMCLCTL.FCNnGMCLSORF (1) cannot proceed at the same time. If a hardware reset occurs while FCNnGMCLCTL.FCNnGMCLSORF = 1, then software reset processing is stopped (aborted), and a hardware reset starts.
1	FCNnGMCLESEDE	<p>Bit enabling forced shutdown</p> <p>0: Forced shutdown is disabled while FCNnGMCLCTL.FCNnGMCLPWOM = 0. 1: Forced shutdown is enabled while FCNnGMCLCTL.FCNnGMCLPWOM = 0.</p> <p>Caution: To request a forced shutdown, FCNnGMCLCTL.FCNnGMCLPWOM must be cleared to 0 immediately for access after FCNnGMCLCTL.FCNnGMCLESEDE has been set to 1. If any access to another register (including reading the FCNnGMCLCTL register) is executed without clearing FCNnGMCLPWOM immediately after FCNnGMCLESEDE has been set to 1, FCNnGMCLESEDE is forcibly cleared to 0, and the forced shutdown request is disabled.</p>
0	FCNnGMCLPWOM	<p>Global operation mode bit</p> <p>0: The FCN module is disabled. 1: The FCN module is enabled.</p> <p>Caution: FCNnGMCLCTL.FCNnGMCLPWOM can only be cleared in the initialization mode or immediately after FCNnGMCLCTL.FCNnGMCLESEDE has been set (forced shutdown).</p>

(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

Bit Position	Bit Name	Description												
12	FCNnGMCLSESR	Software reset start 0: No change 1: Start software reset.												
9	FCNnGMCLSESD	FCNnGMCLSESD bit setting 0: No change in the FCNnGMCLSEDE bit. 1: The FCNnGMCLSEDE bit set to 1.												
8, 0	FCNnGMCLSEOM, FCNnGMCLCLOM	FCNnGMCLPWOM bit setting <table border="1" style="width:100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width:25%;">FCNnGMCLSEOM</th> <th style="width:25%;">FCNnGMCLCLOM</th> <th style="width:50%;">FCNnGMCLPWOM Bit Setting</th> </tr> </thead> <tbody> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td>FCNnGMCLCTL.FCNnGMCLPWOM bit cleared to 0.</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>FCNnGMCLCTL.FCNnGMCLPWOM bit set to 1.</td> </tr> <tr> <td colspan="2" style="text-align:center;">Other than above</td> <td>No change in the FCNnGMCLCTL.FCNnGMCLPWOM bit.</td> </tr> </tbody> </table> <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;"> Caution: The FCNnGMCLCTL.FCNnGMCLPWOM and FCNnGMCLCTL.FCNnGMCLSEDE bits must be separately. </div>	FCNnGMCLSEOM	FCNnGMCLCLOM	FCNnGMCLPWOM Bit Setting	0	1	FCNnGMCLCTL.FCNnGMCLPWOM bit cleared to 0.	1	0	FCNnGMCLCTL.FCNnGMCLPWOM bit set to 1.	Other than above		No change in the FCNnGMCLCTL.FCNnGMCLPWOM bit.
FCNnGMCLSEOM	FCNnGMCLCLOM	FCNnGMCLPWOM Bit Setting												
0	1	FCNnGMCLCTL.FCNnGMCLPWOM bit cleared to 0.												
1	0	FCNnGMCLCTL.FCNnGMCLPWOM bit set to 1.												
Other than above		No change in the FCNnGMCLCTL.FCNnGMCLPWOM bit.												
1	FCNnGMCLCLMB	FCNnGMCLCTL.FCNnGMCLCECCF bit clear 0: No change in the FCNnGMCLCTL.FCNnGMCLCECCF bit. 1: The FCNnGMCLCTL.FCNnGMCLCECCF bit cleared to 0.												

(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}) <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 40%;">FCNnGMCSPRSC[3:0]</th> <th style="width: 60%;">FCN Module System Clock (f_{CANMOD})</th> </tr> </thead> <tbody> <tr><td>0000B</td><td>$f_{CAN} / 1$</td></tr> <tr><td>0001B</td><td>$f_{CAN} / 2$</td></tr> <tr><td>0010B</td><td>$f_{CAN} / 3$</td></tr> <tr><td>0011B</td><td>$f_{CAN} / 4$</td></tr> <tr><td>0100B</td><td>$f_{CAN} / 5$</td></tr> <tr><td>0101B</td><td>$f_{CAN} / 6$</td></tr> <tr><td>0110B</td><td>$f_{CAN} / 7$</td></tr> <tr><td>0111B</td><td>$f_{CAN} / 8$</td></tr> <tr><td>1000B</td><td>$f_{CAN} / 9$</td></tr> <tr><td>1001B</td><td>$f_{CAN} / 10$</td></tr> <tr><td>1010B</td><td>$f_{CAN} / 11$</td></tr> <tr><td>1011B</td><td>$f_{CAN} / 12$</td></tr> <tr><td>1100B</td><td>$f_{CAN} / 13$</td></tr> <tr><td>1101B</td><td>$f_{CAN} / 14$</td></tr> <tr><td>1110B</td><td>$f_{CAN} / 15$</td></tr> <tr><td>1111B</td><td>$f_{CAN} / 16$ (default value)</td></tr> </tbody> </table>	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)
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. <div style="background-color: #e0ffff; padding: 5px;"> 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. </div>
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.

Bit Position	Bit Name	Description												
1	FCNnGMABSEAC	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.												
8, 0	FCNnGMABSEAT, FCNnGMABCLAT	Automatic block transmission start bit <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>FCNnGMABSEAT</th> <th>FCNnGMABCLAT</th> <th>Automatic Block Transmission Start Bit</th> </tr> </thead> <tbody> <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> </tbody> </table>	FCNnGMABSEAT	FCNnGMABCLAT	Automatic Block Transmission Start Bit	0	1	Request automatic block transmission to be stopped.	1	0	Request automatic block transmission to be started.	Other than above		The FCNnGMABCTL.FCNnGMABABTT bit is not changed.
FCNnGMABSEAT	FCNnGMABCLAT	Automatic Block Transmission Start Bit												
0	1	Request automatic block transmission to be stopped.												
1	0	Request automatic block transmission to be started.												
Other than above		The FCNnGMABCTL.FCNnGMABABTT bit is not changed.												

(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} <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 30%;">FCNnGMADSSAD[3:0]</th> <th style="width: 70%;">Data Frame Interval during Automatic Block Transmission (in units of DBT)^{Note}</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0000B</td> <td>0 DBT (default value)</td> </tr> <tr> <td style="text-align: center;">0001B</td> <td>2⁵ DBT</td> </tr> <tr> <td style="text-align: center;">0010B</td> <td>2⁶ DBT</td> </tr> <tr> <td style="text-align: center;">0011B</td> <td>2⁷ DBT</td> </tr> <tr> <td style="text-align: center;">0100B</td> <td>2⁸ DBT</td> </tr> <tr> <td style="text-align: center;">0101B</td> <td>2⁹ DBT</td> </tr> <tr> <td style="text-align: center;">0110B</td> <td>2¹⁰ DBT</td> </tr> <tr> <td style="text-align: center;">0111B</td> <td>2¹¹ DBT</td> </tr> <tr> <td style="text-align: center;">1000B</td> <td>2¹² DBT</td> </tr> <tr> <td style="text-align: center;">Other than above</td> <td>Setting prohibited</td> </tr> </tbody> </table>	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
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)

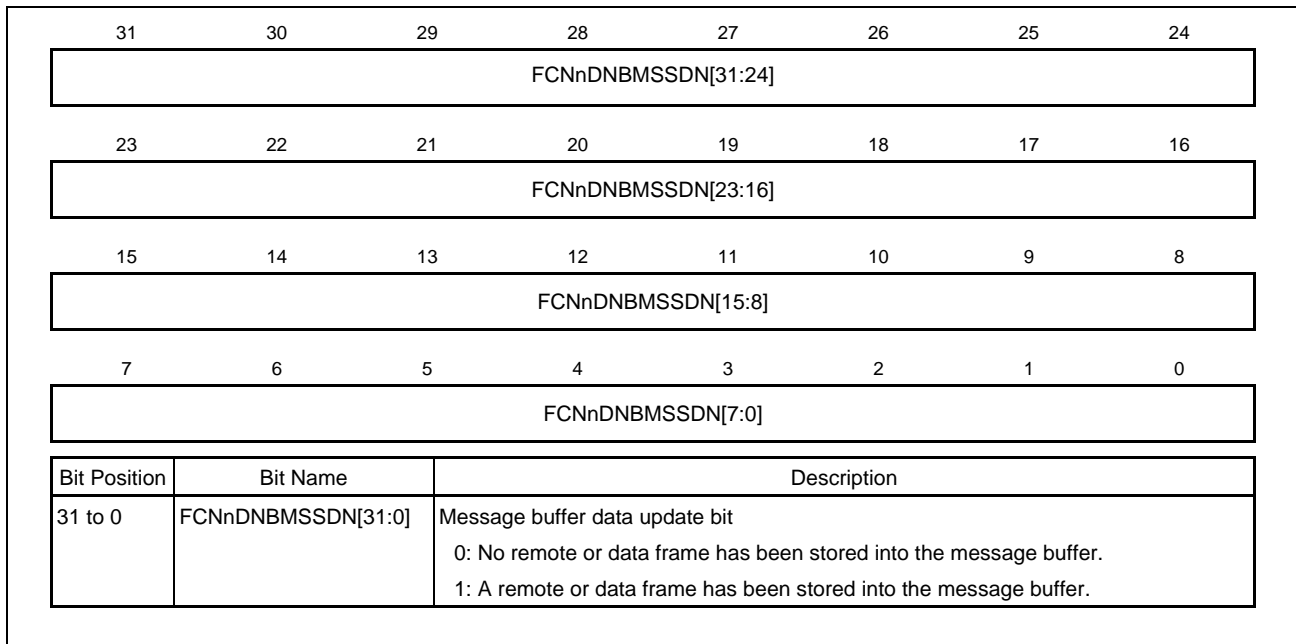
Cautions

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.



19.5.2 FCN Module Registers

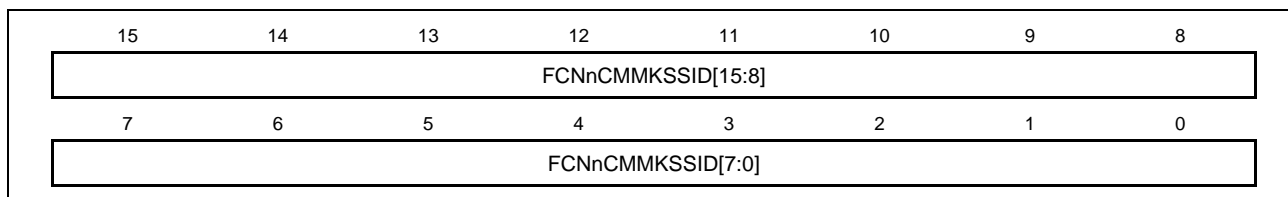
(1) FCNn Module Mask Control Register (FCNnCMMKCTLaH, FCNnCMMKCTLaW)

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.

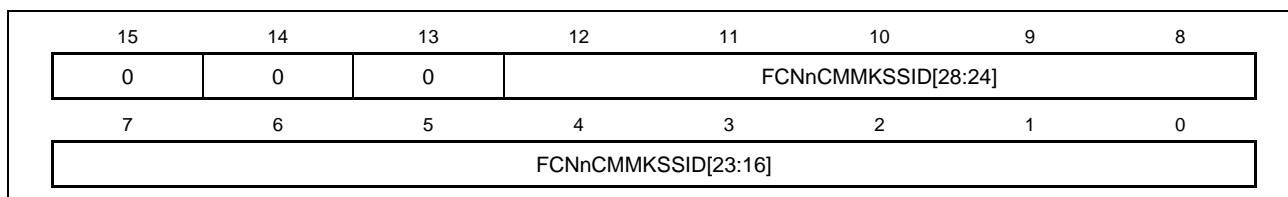
Two 16-bit registers FCNnCMMKCTLaH (a = 01 to 16) can also be accessed via a single 32-bit access to the registers FCNnCMMKCTLaW (a = 01, 03, 05, 07, 09, 11, 13, 15).

- 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.

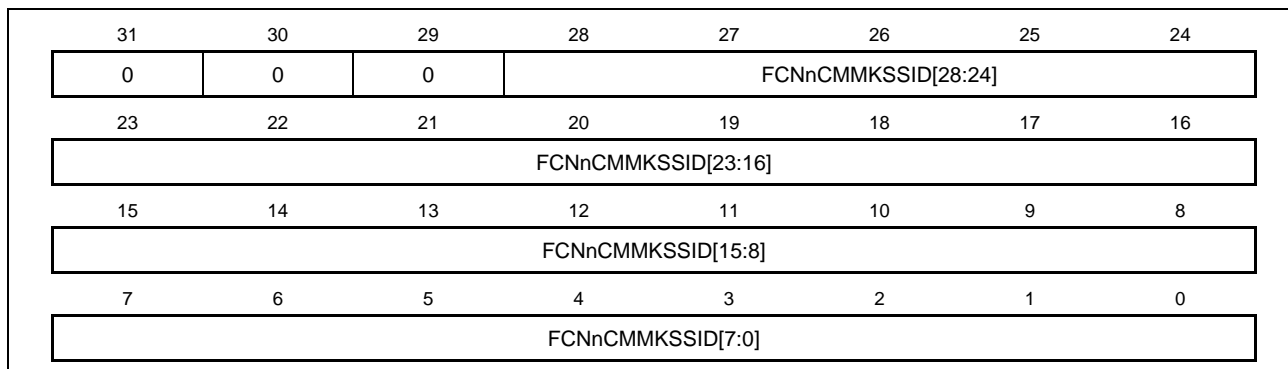
(a) FCNnCMMKCTLaH (a = 01, 03, 05, 07, 09, 11, 13, 15)



(b) FCNnCMMKCTLaH (a = 02, 04, 06, 08, 10, 12, 14, 16)



(c) FCNnCMMKCTLaW (a = 01, 03, 05, 07, 09, 11, 13, 15)



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) <ul style="list-style-type: none"> • 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) <ul style="list-style-type: none"> • 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) <ul style="list-style-type: none"> • The SOF bit of a transmission frame is detected 2. FCNnCMCLSSTS is cleared to 0 under the following conditions (timing) <ul style="list-style-type: none"> • 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	<p>Error counter clear bit</p> <p>0: The FCNnCMERCNT and FCNnCMINSTR registers are not cleared in the initialization mode.</p> <p>1: The FCNnCMERCNT and FCNnCMINSTR registers are cleared in the initialization mode.</p> <p>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):</p> <ul style="list-style-type: none"> - 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 19.24, 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 19.25, Transmission Abort Processing (in Normal Operation Mode with ABT) – Repeat Option for Aborted Message.) <p>Remarks</p> <ol style="list-style-type: none"> 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	<p>Bit to set operation in case of arbitration loss</p> <p>0: Re-transmission is not executed in case of an arbitration loss in the single-shot mode.</p> <p>1: Re-transmission is executed in case of an arbitration loss in the single-shot mode.</p> <p>Remark: FCNnCMCLALBF is valid only in the single-shot mode.</p>

(3/4)

Bit Position	Bit Name	Description										
5	FCNnCMCLVALF	<p>Valid receive message frame detection bit</p> <p>0: A valid message frame has not been received since FCNnCMCLVALF was last cleared to 0.</p> <p>1: A valid message frame has been received since FCNnCMCLVALF was last cleared to 0.</p> <p>Remarks</p> <ol style="list-style-type: none"> 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. 										
4, 3	FCNnCMCLMDPF[1:0]	<p>Power save mode</p> <table border="1"> <thead> <tr> <th>FCNnCMCLMDPF[1:0]</th> <th>Power Save Mode</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Cautions</p> <ol style="list-style-type: none"> 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. <p>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.</p>	FCNnCMCLMDPF[1:0]	Power Save Mode	00B	No power save mode is selected.	01B	FCN sleep mode	10B	Setting prohibited	11B	FCN stop mode
FCNnCMCLMDPF[1:0]	Power Save Mode											
00B	No power save mode is selected.											
01B	FCN sleep mode											
10B	Setting prohibited											
11B	FCN stop mode											

(4/4)

Bit Position	Bit Name	Description																
2 to 0	FCNnCMCLMDOF[2:0]	<p>Operation mode</p> <table border="1"> <thead> <tr> <th>FCNnCMCLMDOF[2:0]</th> <th>Operation Mode</th> </tr> </thead> <tbody> <tr> <td>000B</td> <td>No operation mode is selected (FCN module is in the initialization mode).</td> </tr> <tr> <td>001B</td> <td>Normal operation mode</td> </tr> <tr> <td>010B</td> <td>Normal operation mode with automatic block transmission (normal operation mode with ABT)</td> </tr> <tr> <td>011B</td> <td>Receive-only mode</td> </tr> <tr> <td>100B</td> <td>Single-shot mode</td> </tr> <tr> <td>101B</td> <td>Self-test mode</td> </tr> <tr> <td>Other than above</td> <td>Setting prohibited</td> </tr> </tbody> </table> <p>Cautions</p> <ol style="list-style-type: none"> 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. 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. <p>Remark: FCNnCM.FCNnCMCLMDOF[2:0] are read-only in the FCN sleep mode or FCN stop mode.</p>	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
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																	

(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]		

Bit Position	Bit Name	Description												
15	FCNnCMCLSERC	Setting of FCNnCMCLERCF bit 0: FCNnCMCLERCF is not changed. 1: FCNnCMCLERCF is set to 1.												
14, 6	FCNnCMCLSEAL, FCNnCMCLCLAL	<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:25%;">FCNnCMCLSEAL</th> <th style="width:25%;">FCNnCMCLCLAL</th> <th style="width:50%;">Setting of FCNnCMCLALBF Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td>FCNnCMCLALBF is cleared to 0.</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>FCNnCMCLALBF is set to 1.</td> </tr> <tr> <td colspan="2" style="text-align:center;">Other than above</td> <td>FCNnCMCLALBF is not changed.</td> </tr> </tbody> </table>	FCNnCMCLSEAL	FCNnCMCLCLAL	Setting of FCNnCMCLALBF Bit	0	1	FCNnCMCLALBF is cleared to 0.	1	0	FCNnCMCLALBF is set to 1.	Other than above		FCNnCMCLALBF is not changed.
FCNnCMCLSEAL	FCNnCMCLCLAL	Setting of FCNnCMCLALBF Bit												
0	1	FCNnCMCLALBF is cleared to 0.												
1	0	FCNnCMCLALBF is set to 1.												
Other than above		FCNnCMCLALBF is not changed.												
5	FCNnCMCLCLVL	Setting of FCNnCMCLVALF bit 0: FCNnCMCLVALF is not changed. 1: FCNnCMCLVALF is cleared to 0.												
11, 3	FCNnCMCLSEPS0, FCNnCMCLCLPS0	<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:25%;">FCNnCMCLSEPS0</th> <th style="width:25%;">FCNnCMCLCLPS0</th> <th style="width:50%;">Setting of FCNnCMCLMDPF0 Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td>FCNnCMCLMDPF0 is cleared to 0.</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>FCNnCMCLMDPF0 is set to 1.</td> </tr> <tr> <td colspan="2" style="text-align:center;">Other than above</td> <td>FCNnCMCLMDPF0 is not changed.</td> </tr> </tbody> </table>	FCNnCMCLSEPS0	FCNnCMCLCLPS0	Setting of FCNnCMCLMDPF0 Bit	0	1	FCNnCMCLMDPF0 is cleared to 0.	1	0	FCNnCMCLMDPF0 is set to 1.	Other than above		FCNnCMCLMDPF0 is not changed.
FCNnCMCLSEPS0	FCNnCMCLCLPS0	Setting of FCNnCMCLMDPF0 Bit												
0	1	FCNnCMCLMDPF0 is cleared to 0.												
1	0	FCNnCMCLMDPF0 is set to 1.												
Other than above		FCNnCMCLMDPF0 is not changed.												
12, 4	FCNnCMCLSEPS1, FCNnCMCLCLPS1	<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:25%;">FCNnCMCLSEPS1</th> <th style="width:25%;">FCNnCMCLCLPS1</th> <th style="width:50%;">Setting of FCNnCMCLMDPF1 Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td>FCNnCMCLMDPF1 is cleared to 0.</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>FCNnCMCLMDPF1 is set to 1</td> </tr> <tr> <td colspan="2" style="text-align:center;">Other than above</td> <td>FCNnCMCLMDPF1 is not changed.</td> </tr> </tbody> </table>	FCNnCMCLSEPS1	FCNnCMCLCLPS1	Setting of FCNnCMCLMDPF1 Bit	0	1	FCNnCMCLMDPF1 is cleared to 0.	1	0	FCNnCMCLMDPF1 is set to 1	Other than above		FCNnCMCLMDPF1 is not changed.
FCNnCMCLSEPS1	FCNnCMCLCLPS1	Setting of FCNnCMCLMDPF1 Bit												
0	1	FCNnCMCLMDPF1 is cleared to 0.												
1	0	FCNnCMCLMDPF1 is set to 1												
Other than above		FCNnCMCLMDPF1 is not changed.												
8, 0	FCNnCMCLSEOP0, FCNnCMCLCLOP0	<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:25%;">FCNnCMCLSEOP0</th> <th style="width:25%;">FCNnCMCLCLOP0</th> <th style="width:50%;">Setting of FCNnCMCLMDOF0 Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td>FCNnCMCLMDOF0 is cleared to 0.</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>FCNnCMCLMDOF0 is set to 1</td> </tr> <tr> <td colspan="2" style="text-align:center;">Other than above</td> <td>FCNnCMCLMDOF0 is not changed.</td> </tr> </tbody> </table>	FCNnCMCLSEOP0	FCNnCMCLCLOP0	Setting of FCNnCMCLMDOF0 Bit	0	1	FCNnCMCLMDOF0 is cleared to 0.	1	0	FCNnCMCLMDOF0 is set to 1	Other than above		FCNnCMCLMDOF0 is not changed.
FCNnCMCLSEOP0	FCNnCMCLCLOP0	Setting of FCNnCMCLMDOF0 Bit												
0	1	FCNnCMCLMDOF0 is cleared to 0.												
1	0	FCNnCMCLMDOF0 is set to 1												
Other than above		FCNnCMCLMDOF0 is not changed.												

(2/2)

Bit Position	Bit Name	Description												
9, 1	FCNnCMCLSEOP1, FCNnCMCLCLOP1	<table border="1"> <thead> <tr> <th>FCNnCMCLSEOP1</th> <th>FCNnCMCLCLOP1</th> <th>Setting of FCNnCMCLMDOF1 Bit</th> </tr> </thead> <tbody> <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> </tbody> </table>	FCNnCMCLSEOP1	FCNnCMCLCLOP1	Setting of FCNnCMCLMDOF1 Bit	0	1	FCNnCMCLMDOF1 is cleared to 0.	1	0	FCNnCMCLMDOF1 is set to 1	Other than above		FCNnCMCLMDOF1 is not changed.
		FCNnCMCLSEOP1	FCNnCMCLCLOP1	Setting of FCNnCMCLMDOF1 Bit										
		0	1	FCNnCMCLMDOF1 is cleared to 0.										
		1	0	FCNnCMCLMDOF1 is set to 1										
Other than above		FCNnCMCLMDOF1 is not changed.												
10, 2	FCNnCMCLSEOP2, FCNnCMCLCLOP2	<table border="1"> <thead> <tr> <th>FCNnCMCLSEOP2</th> <th>FCNnCMCLCLOP2</th> <th>Setting of FCNnCMCLMDOF2 Bit</th> </tr> </thead> <tbody> <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> </tbody> </table>	FCNnCMCLSEOP2	FCNnCMCLCLOP2	Setting of FCNnCMCLMDOF2 Bit	0	1	FCNnCMCLMDOF2 is cleared to 0.	1	0	FCNnCMCLMDOF2 is set to 1	Other than above		FCNnCMCLMDOF2 is not changed.
		FCNnCMCLSEOP2	FCNnCMCLCLOP2	Setting of FCNnCMCLMDOF2 Bit										
		0	1	FCNnCMCLMDOF2 is cleared to 0.										
		1	0	FCNnCMCLMDOF2 is set to 1										
Other than above		FCNnCMCLMDOF2 is not changed.												

(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	FCNnCMLCSSL[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	FCNnCMLCSSL[2:0]	FCNnCMLCSSL[2:0]	
		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	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.)				
3, 2	FCNnCMINSSTE[1:0]	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).				
1, 0	FCNnCMINSSRE[1:0]	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						
FCNnCMERRPSF		FCNnCMERRECF[6:0]											
7	6	5	4	3	2	1	0						
FCNnCMERTECF[7:0]													
Bit Position	Bit Name	Description											
15	FCNnCMERRPSF	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">FCNnCMERRPSF</th> <th style="text-align: center;">Reception Error Passive Status Bit</th> </tr> <tr> <td style="text-align: center;">0</td> <td>The reception error counter is not in the error passive range (< 128)</td> </tr> <tr> <td style="text-align: center;">1</td> <td>The reception error counter is in the error passive range (≥ 128)</td> </tr> </table>						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)
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)												
14 to 8	FCNnCMERRECF[6:0]	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">FCNnCMERRECF[6:0]</th> <th style="text-align: center;">Reception Error Counter Bit</th> </tr> <tr> <td style="text-align: center;">0-127</td> <td>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.</td> </tr> </table> <p style="color: green; text-align: center; margin-top: 10px;">Remark: FCNnCMERRECF[6:0] are invalid in the reception error passive state (FCNnCMINSTR.FCNnCMINSSRE[1:0] = 11B).</p>						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.		
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.												
7 to 0	FCNnCMERTECF[7:0]	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">FCNnCMERTECF[7:0]</th> <th style="text-align: center;">Transmission Error Counter Bit</th> </tr> <tr> <td style="text-align: center;">0-255</td> <td>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.</td> </tr> </table> <p style="color: green; text-align: center; margin-top: 10px;">Remark: FCNnCMERTECF[7:0] are invalid in the bus-off state (FCNnCMINSTR.FCNnCMINBOFF = 1).</p>						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.		
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.												

(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]	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">FCNnCMIEINTF[6:0]</th> <th style="text-align: center;">FCN Module Interrupt Enable Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Output of the interrupt corresponding to interrupt status register FCNnCMISCTL is disabled.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Output of the interrupt corresponding to interrupt status register FCNnCMISCTL is enabled.</td> </tr> </tbody> </table>						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.
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												
FCNnCMIESEIE[6:0]																			
7	6	5	4	3	2	1	0												
FCNnCMIECLIE[6:0]																			
Bit Position	Bit Name	Description																	
14 to 8, 6 to 0	FCNnCMIESEIE[6:0], FCNnCMIECLIE[6:0]	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">FCNnCMIESEIE[6:0]</th> <th style="text-align: center;">FCNnCMIECLIE[6:0]</th> <th style="text-align: center;">Setting of FCNnCMIEINTF[6:0] Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>FCNnCMIEINTF[6:0] bit is cleared to 0.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>FCNnCMIEINTF[6:0] bit is set to 1.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Other than above</td> <td>FCNnCMIEINTF[6:0] bit is not to change.</td> </tr> </tbody> </table>						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.
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]	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">FCNnCMISITSF[6:0]</th> <th style="width: 50%;">FCN Interrupt Status Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>No related interrupt source event is pending</td> </tr> <tr> <td style="text-align: center;">1</td> <td>A related interrupt source event is pending</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Interrupt Status Bit</th> <th style="width: 70%;">Related Interrupt Source Event</th> </tr> </thead> <tbody> <tr> <td>FCNnCMISITSF6</td> <td>FCN module transmission abort interrupt status bit</td> </tr> <tr> <td>FCNnCMISITSF5</td> <td>Wakeup interrupt from FCN sleep mode^{Note}</td> </tr> <tr> <td>FCNnCMISITSF4</td> <td>Arbitration loss interrupt</td> </tr> <tr> <td>FCNnCMISITSF3</td> <td>CAN protocol error interrupt</td> </tr> <tr> <td>FCNnCMISITSF2</td> <td>CAN error status interrupt</td> </tr> <tr> <td>FCNnCMISITSF1</td> <td>Interrupt on completion of reception of valid message frame to message buffer m</td> </tr> <tr> <td>FCNnCMISITSF0</td> <td>Interrupt on normal completion of transmission of message frame from message buffer m</td> </tr> </tbody> </table>	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
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																							
<p>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.</p>																								

(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]	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">FCNnCMISCLTS[6:0]</td> <td style="width: 50%;">Clearing of FCNnCMISITSF[6:0]</td> </tr> <tr> <td style="text-align: center;">0</td> <td>FCNnCMISITSF[6:0] bits are not changed.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>FCNnCMISITSF[6:0] bits are cleared to 0.</td> </tr> </table>	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.
		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.							
<p>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.</p>								

(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 $\langle \text{FCNn_base} \rangle + 0\ 0268\text{H}$
- 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]	<table border="1" style="width: 100%;"> <tr> <th>FCNnCMBRPRS[7:0]</th> <th>CAN Protocol Layer Basic System Clock (f_{TQ})</th> </tr> <tr> <td>0x00</td> <td>$f_{\text{CANMOD}} / 1$</td> </tr> <tr> <td>0x01</td> <td>$f_{\text{CANMOD}} / 2$</td> </tr> <tr> <td>n</td> <td>$f_{\text{CANMOD}} / (n+1)$</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>0xff</td> <td>$f_{\text{CANMOD}} / 256$(default value)</td> </tr> </table>						FCNnCMBRPRS[7:0]	CAN Protocol Layer Basic System Clock (f_{TQ})	0x00	$f_{\text{CANMOD}} / 1$	0x01	$f_{\text{CANMOD}} / 2$	n	$f_{\text{CANMOD}} / (n+1)$:	:	0xff	$f_{\text{CANMOD}} / 256$ (default value)
FCNnCMBRPRS[7:0]	CAN Protocol Layer Basic System Clock (f_{TQ})																		
0x00	$f_{\text{CANMOD}} / 1$																		
0x01	$f_{\text{CANMOD}} / 2$																		
n	$f_{\text{CANMOD}} / (n+1)$																		
:	:																		
0xff	$f_{\text{CANMOD}} / 256$ (default value)																		

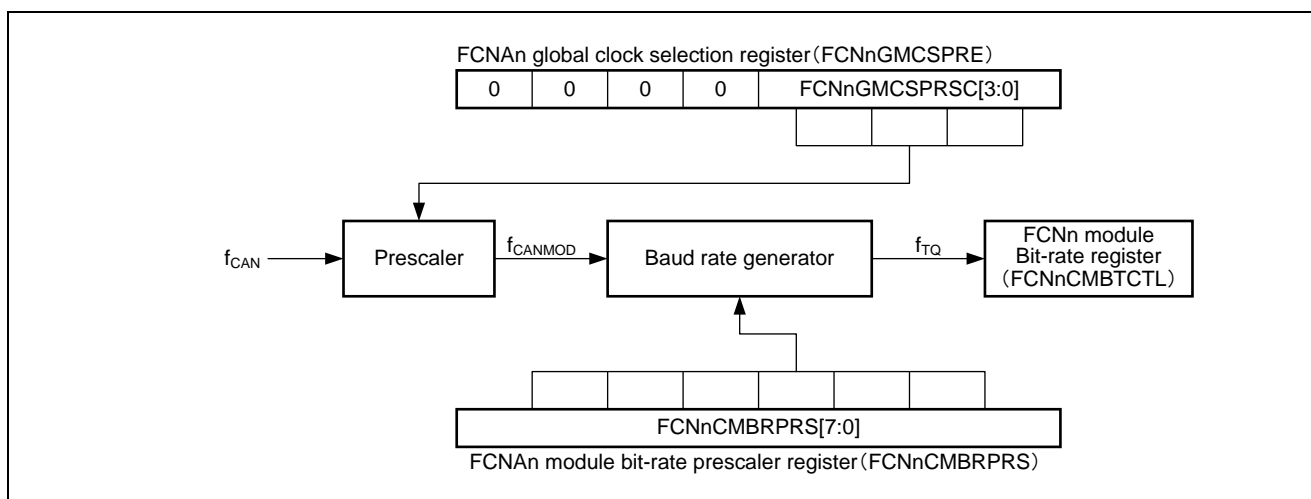


Figure 19.2 FCN Module Clock

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.

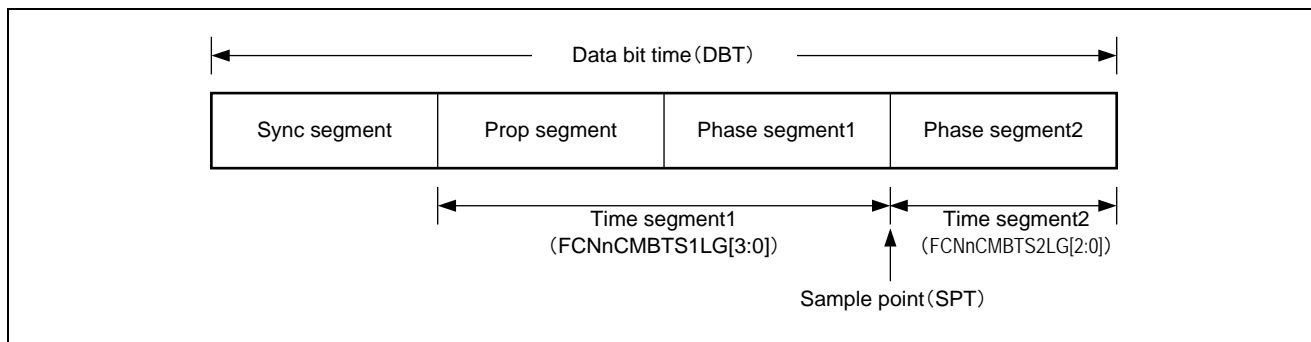


Figure 19.3 Data Bit Time

(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]	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]	<table border="1"> <thead> <tr> <th>FCNnCMBTS2LG[2:0]</th> <th>Length of Time Segment 2</th> </tr> </thead> <tbody> <tr> <td>000B</td> <td>1T_Q</td> </tr> <tr> <td>001B</td> <td>2T_Q</td> </tr> <tr> <td>010B</td> <td>3T_Q</td> </tr> <tr> <td>011B</td> <td>4T_Q</td> </tr> <tr> <td>100B</td> <td>5T_Q</td> </tr> <tr> <td>101B</td> <td>6T_Q</td> </tr> <tr> <td>110B</td> <td>7T_Q</td> </tr> <tr> <td>111B</td> <td>8T_Q (Initial value)</td> </tr> </tbody> </table>	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)																
		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]	<table border="1"> <thead> <tr> <th>FCNnCMBTS1LG[3:0]</th> <th>Length of Time Segment 1</th> </tr> </thead> <tbody> <tr> <td>0000B</td> <td>Setting prohibited</td> </tr> <tr> <td>0001B</td> <td>Setting prohibited</td> </tr> <tr> <td>0010B</td> <td>Setting prohibited</td> </tr> <tr> <td>0011B</td> <td>4T_Q</td> </tr> <tr> <td>0100B</td> <td>5T_Q</td> </tr> <tr> <td>0101B</td> <td>6T_Q</td> </tr> <tr> <td>0110B</td> <td>7T_Q</td> </tr> <tr> <td>0111B</td> <td>8T_Q</td> </tr> <tr> <td>1000B</td> <td>9T_Q</td> </tr> <tr> <td>1001B</td> <td>10T_Q</td> </tr> <tr> <td>1010B</td> <td>11T_Q</td> </tr> <tr> <td>1011B</td> <td>12T_Q</td> </tr> <tr> <td>1100B</td> <td>13T_Q</td> </tr> <tr> <td>1101B</td> <td>14T_Q</td> </tr> <tr> <td>1110B</td> <td>15T_Q</td> </tr> <tr> <td>1111B</td> <td>16T_Q (Initial value)</td> </tr> </tbody> </table>	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)
		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.				
<p>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.</p>							

(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 FCNnCMRGSSPT[7:0] is invalid while FCNnCMRGSSPM = 1.					

(2/2)

Bit Position	Bit Name	Description						
0	FCNnCMRGRVFF ^{Note 1}	<table border="1"> <thead> <tr> <th>FCNnCMRGRVFF^{Note 1}</th> <th>Receive History List Overflow Bit^{Note 2}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>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).</td> </tr> <tr> <td>1</td> <td>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.</td> </tr> </tbody> </table>	FCNnCMRGRVFF ^{Note 1}	Receive History List Overflow Bit ^{Note 2}	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.
		FCNnCMRGRVFF ^{Note 1}	Receive History List Overflow Bit ^{Note 2}					
		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.							
<p>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.</p> <p>2. i = 47</p>								

(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	FCNnCMRGCLR

Bit Position	Bit Name	Description						
0	FCNnCMRGCLR	<table border="1"> <thead> <tr> <th>FCNnCMRGCLR</th> <th>Clearing of FCNnCMRGRVFF Bit</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>The FCNnCMRGRVFF bit is not changed</td> </tr> <tr> <td>1</td> <td>The FCNnCMRGRVFF bit is cleared to 0.</td> </tr> </tbody> </table>	FCNnCMRGCLR	Clearing of FCNnCMRGRVFF Bit	0	The FCNnCMRGRVFF bit is not changed	1	The FCNnCMRGRVFF bit is cleared to 0.
		FCNnCMRGCLR	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									
	FCNnCMLOSSLT[7:0]	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">FCNnCMLOSSLT[7:0]</td> <td>Last Out-Pointer of Transmit History List</td> </tr> <tr> <td style="text-align: center;">0-63</td> <td>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.</td> </tr> </table>						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.
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]	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">FCNnCMTGSSPT[7:0]</th> <th style="text-align: center;">Transmit History List Read Pointer</th> </tr> <tr> <td style="text-align: center;">0-63</td> <td>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.</td> </tr> </table>						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.		
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}	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">FCNnCMTGSSPM^{Note}</th> <th style="text-align: center;">Transmit History Pointer Match</th> </tr> <tr> <td style="text-align: center;">0</td> <td>The transmit history list has at least one message buffer number that has not been read.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>The transmit history list has no message buffer numbers that have not been read.</td> </tr> </table> <p style="color: red; font-weight: bold; margin-top: 10px;">Note. The read value of FCNnCMTGSSPT[7:0] is invalid when FCNnCMTGSSPM = 1.</p>						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.
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.												

(2/2)

Bit Position	Bit Name	Description						
0	FCNnCMTGTVFF ^{Note 1}	<table border="1"> <thead> <tr> <th>FCNnCMTGTVFF^{Note 1}</th> <th>Transmit History List Overflow Bit^{Note 2}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>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).</td> </tr> <tr> <td>1</td> <td>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.</td> </tr> </tbody> </table>	FCNnCMTGTVFF ^{Note 1}	Transmit History List Overflow Bit ^{Note 2}	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.
		FCNnCMTGTVFF ^{Note 1}	Transmit History List Overflow Bit ^{Note 2}					
		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.							
<p>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.</p> <p>2. i = 15</p>								
<p>Remark: Transmission from the following message buffers is not recorded in the transmit history list in the normal operation mode with ABT.</p> <ul style="list-style-type: none"> • 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	FCNnCM TGCLTV

Bit Position	Bit Name	Description						
0	FCNnCMTGCLTV	<table border="1"> <thead> <tr> <th>FCNnCMTGCLTV</th> <th>Setting of FCNnCMTGTVFF Bit</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FCNnCMTGTVFF bit is not changed.</td> </tr> <tr> <td>1</td> <td>FCNnCMTGTVFF bit is cleared to 0.</td> </tr> </tbody> </table>	FCNnCMTGCLTV	Setting of FCNnCMTGTVFF Bit	0	FCNnCMTGTVFF bit is not changed.	1	FCNnCMTGTVFF bit is cleared to 0.
		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	<table border="1"> <tr> <th>FCNnCMTSLOKE</th> <th>Timestamp Locking Enable Bit</th> </tr> <tr> <td>0</td> <td>Timestamp locking is stopped. The TSOUT signal is toggled each time the selected timestamp capture event occurs.</td> </tr> <tr> <td>1</td> <td>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}</td> </tr> </table>	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}
		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}							
<p>Note: FCNnCMTSTSGE is automatically cleared to 0.</p>								
1	FCNnCMTSSELE	<table border="1"> <tr> <th>FCNnCMTSSELE</th> <th>Timestamp Capture Event Selection Bit</th> </tr> <tr> <td>0</td> <td>The timestamp capture event is SOF.</td> </tr> <tr> <td>1</td> <td>The timestamp capture event is the last bit of EOF.</td> </tr> </table>	FCNnCMTSSELE	Timestamp Capture Event Selection Bit	0	The timestamp capture event is SOF.	1	The timestamp capture event is the last bit of EOF.
		FCNnCMTSSELE	Timestamp Capture Event Selection Bit					
		0	The timestamp capture event is SOF.					
1	The timestamp capture event is the last bit of EOF.							
0	FCNnCMTSTSGE	<table border="1"> <tr> <th>FCNnCMTSTSGE</th> <th>TSOUT Operation Setting Bit</th> </tr> <tr> <td>0</td> <td>TSOUT toggle operation is disabled.</td> </tr> <tr> <td>1</td> <td>TSOUT toggle operation is enabled.</td> </tr> </table>	FCNnCMTSTSGE	TSOUT Operation Setting Bit	0	TSOUT toggle operation is disabled.	1	TSOUT toggle operation is enabled.
		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 TSCLK	FCNnCM TSCLSL	FCNnCM TSCLTS

Bit Position	Bit Name	Description												
10, 2	FCNnCMTSSELK, FCNnCMTSCLK	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">FCNnCMTSSELK</th> <th style="width: 33%;">FCNnCMTSCLK</th> <th style="width: 34%;">Setting of FCNnCMTSLOKE Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>FCNnCMTSLOKE is cleared to 0.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>FCNnCMTSLOKE is set to 1.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Other than above</td> <td>FCNnCMTSLOKE is not changed.</td> </tr> </tbody> </table>	FCNnCMTSSELK	FCNnCMTSCLK	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.
		FCNnCMTSSELK	FCNnCMTSCLK	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.												
9, 1	FCNnCMTSSESL, FCNnCMTSCLSL	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">FCNnCMTSSESL</th> <th style="width: 33%;">FCNnCMTSCLSL</th> <th style="width: 34%;">Setting of FCNnCMTSSELE Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>FCNnCMTSSELE is cleared to 0.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>FCNnCMTSSELE is set to 1</td> </tr> <tr> <td colspan="2" style="text-align: center;">Other than above</td> <td>FCNnCMTSSELE is not changed.</td> </tr> </tbody> </table>	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.
		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.												
8, 0	FCNnCMTSSETS, FCNnCMTSCLTS	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">FCNnCMTSSETS</th> <th style="width: 33%;">FCNnCMTSCLTS</th> <th style="width: 34%;">Setting of FCNnCMTSTSGE Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>FCNnCMTSTSGE is cleared to 0.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>FCNnCMTSTSGE is set to 1</td> </tr> <tr> <td colspan="2" style="text-align: center;">Other than above</td> <td>FCNnCMTSTSGE is not changed.</td> </tr> </tbody> </table>	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.
		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.												

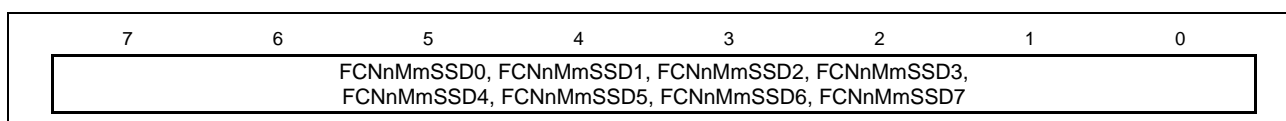
19.5.3 FCN Message Buffer Registers

(1) FCNn Message Data Byte Registers (FCNnMmDATxB/H/W)

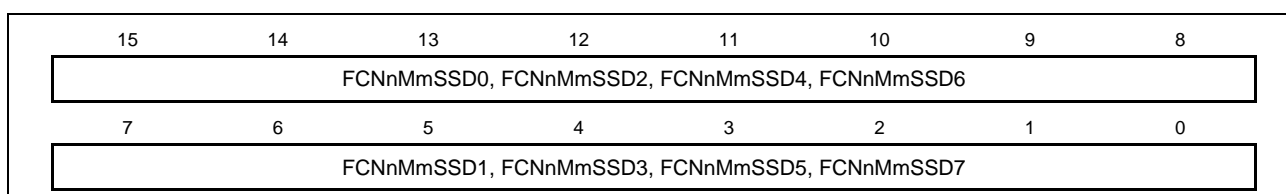
These registers are used to store the data of transmit/receive messages.

- 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 x 40H
 - FCNnMmDAT1B: <FCNn_base> + 0 1004H + m x 40H
 - FCNnMmDAT2B: <FCNn_base> + 0 1008H + m x 40H
 - FCNnMmDAT3B: <FCNn_base> + 0 100CH + m x 40H
 - FCNnMmDAT4B: <FCNn_base> + 0 1010H + m x 40H
 - FCNnMmDAT5B: <FCNn_base> + 0 1014H + m x 40H
 - FCNnMmDAT6B: <FCNn_base> + 0 1018H + m x 40H
 - FCNnMmDAT7B: <FCNn_base> + 0 101CH + m x 40H
 - FCNnMmDAT0H: <FCNn_base> + 0 9000H + m x 40H
 - FCNnMmDAT2H: <FCNn_base> + 0 9008H + m x 40H
 - FCNnMmDAT4H: <FCNn_base> + 0 9010H + m x 40H
 - FCNnMmDAT6H: <FCNn_base> + 0 9018H + m x 40H
 - FCNnMmDAT0W: <FCNn_base> + 1 1000H + m x 40H
 - FCNnMmDAT4W: <FCNn_base> + 1 1010H + m x 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)



(b) FCNnCMmDATxH (x = 0, 2, 4, 6)



(c) FCNnCMmDATxW (x = 0, 4)



(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 x 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]	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	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}				
		1010B					
		1011B					
		1100B					
1101B							
1110B							
1111B							

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 x 40H
- Initial Value 00H. This register is initialized by any reset.

(1/3)

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	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">FCNnMmSSOW</th> <th style="text-align: center;">Overwrite Control Bit</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>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.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>The message buffer that has already received a data frame^{Note} is overwritten by a newly received data frame.</td> </tr> </tbody> </table> <p style="color: #800000; font-weight: bold; margin-top: 10px;">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.</p> <p style="color: #008000; font-weight: bold; margin-top: 10px;">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).</p>						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.
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.												

Bit Position	Bit Name	Description																								
6 to 3	FCNnMmSSMT[3:0]	<table border="1"> <thead> <tr> <th>FCNnMmSSMT[3:0]</th> <th>Message Buffer Type Setting Bit</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>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.</p>	FCNnMmSSMT[3:0]	Message Buffer Type Setting Bit	0000B	Transmit message buffer	0001B	Receive message buffer (no mask setting)	0010B	Receive message buffer (mask 1 set)	0011B	Receive message buffer (mask 2 set)	0100B	Receive message buffer (mask 3 set)	0101B	Receive message buffer (mask 4 set)	0110B	Receive message buffer (mask 5 set)	0111B	Receive message buffer (mask 6 set)	1000B	Receive message buffer (mask 7 set)	1001B	Receive message buffer (mask 8 set)	Other than above	Setting prohibited
FCNnMmSSMT[3:0]	Message Buffer Type Setting Bit																									
0000B	Transmit message buffer																									
0001B	Receive message buffer (no mask setting)																									
0010B	Receive message buffer (mask 1 set)																									
0011B	Receive message buffer (mask 2 set)																									
0100B	Receive message buffer (mask 3 set)																									
0101B	Receive message buffer (mask 4 set)																									
0110B	Receive message buffer (mask 5 set)																									
0111B	Receive message buffer (mask 6 set)																									
1000B	Receive message buffer (mask 7 set)																									
1001B	Receive message buffer (mask 8 set)																									
Other than above	Setting prohibited																									
2	FCNnMmSSRT	<p>Specifies the type of message frame for transmission to or reception from a message buffer.</p> <table border="1"> <thead> <tr> <th>FCNnMmSSRT</th> <th>Remote Frame Request Bit</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>	FCNnMmSSRT	Remote Frame Request Bit	0	Transmit or receive a data frame.	1	Transmit or receive a remote frame.																		
FCNnMmSSRT	Remote Frame Request Bit																									
0	Transmit or receive a data frame.																									
1	Transmit or receive a remote frame.																									

(3/3)

Bit Position	Bit Name	Description						
0	FCNnMmSSAM	<table border="1"><thead><tr><th>FCNnMmSSAM</th><th>Message Buffer Assignment Bit</th></tr></thead><tbody><tr><td>0</td><td>Message buffer not used.</td></tr><tr><td>1</td><td>Message buffer used.</td></tr></tbody></table>	FCNnMmSSAM	Message Buffer Assignment Bit	0	Message buffer not used.	1	Message buffer used.
FCNnMmSSAM	Message Buffer Assignment Bit							
0	Message buffer not used.							
1	Message buffer used.							

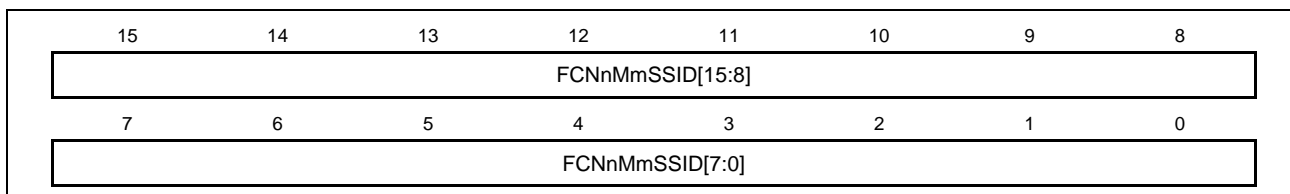
Caution: Be sure to write 0 to bit 1.

(4) FCNn Message ID Register m (FCNnMmMID0H, FCNnMmMID1H, FCNnMmMID0W)

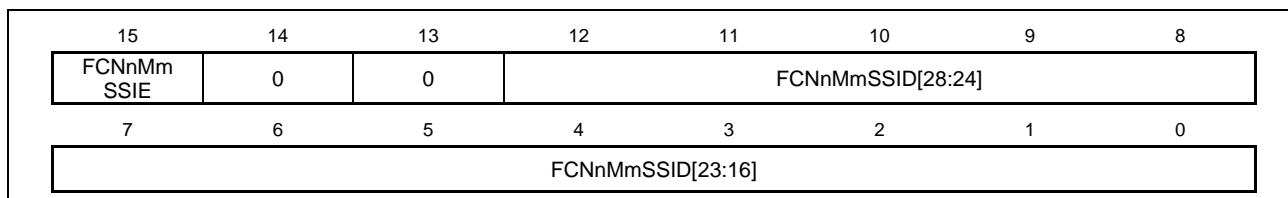
These registers are used to set an identifier (ID).

- 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 x 40H
FCNnMmMID1H: <FCNn_base> + 0 9030H + m x 40H
FCNnMmMID0W: <FCNn_base> + 1 1028H + m x 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.

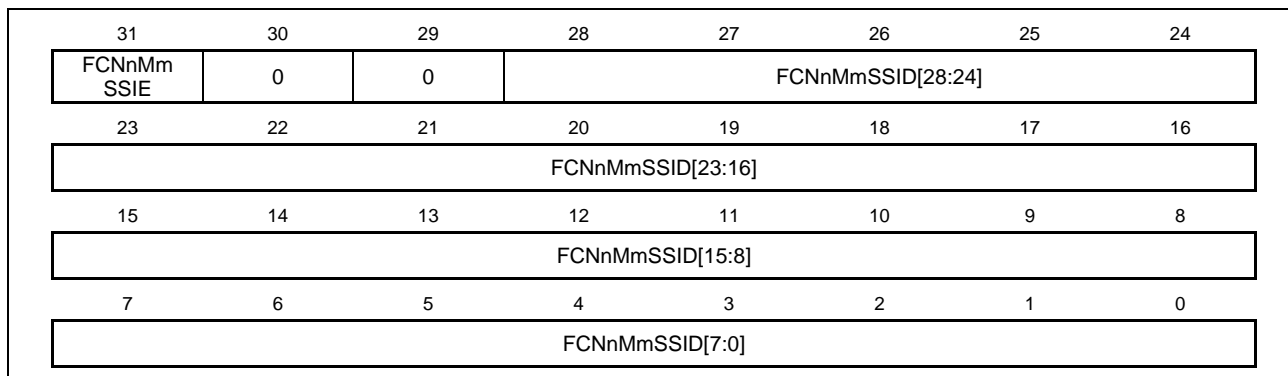
(a) FCNnMmMID0H



(b) FCNnMmMID1H



(c) FCNnCMmMID0W



Bit Position	Bit Name	Description						
31	FCNnMmSSIE	<table border="1"> <thead> <tr> <th>FCNnMmSSIE</th> <th>Format Mode Specification Bit</th> </tr> </thead> <tbody> <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> </tbody> </table>	FCNnMmSSIE	Format Mode Specification Bit	0	Standard format mode (FCNnMmSSID[28:18]: 11 bits. FCNnMmSSID[17:0] are not used)	1	Extended format mode (FCNnMmSSID[28:0]: 29 bits)
		FCNnMmSSIE	Format Mode Specification Bit					
		0	Standard format mode (FCNnMmSSID[28:18]: 11 bits. FCNnMmSSID[17:0] are not used)					
		1	Extended format mode (FCNnMmSSID[28:0]: 29 bits)					
28 to 0	FCNnMmSSID[28:0]	<table border="1"> <thead> <tr> <th>FCNnMmSSID[28:0]</th> <th>Message ID</th> </tr> </thead> <tbody> <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> </tbody> </table>	FCNnMmSSID[28:0]	Message ID	FCNnMmSSID[28:18]	11 bits of standard ID value (FCNnMmSSIE = 0)	FCNnMmSSID[28:0]	29 bits of extended ID value (FCNnMmSSIE = 1)
		FCNnMmSSID[28:0]	Message ID					
		FCNnMmSSID[28:18]	11 bits of standard ID value (FCNnMmSSIE = 0)					
FCNnMmSSID[28:0]	29 bits of extended ID value (FCNnMmSSIE = 1)							

- 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 x 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 border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:15%;">FCNnMmMUCF</td> <td>Bit indicating that Message Buffer Data is being Updated</td> </tr> <tr> <td style="text-align:center;">0</td> <td>The FCN module is not updating the message buffer (no data is being received and stored).</td> </tr> <tr> <td style="text-align:center;">1</td> <td>The FCN module is updating the message buffer (data is being received and stored).</td> </tr> </table>	FCNnMmMUCF	Bit indicating that Message Buffer Data is being Updated	0	The FCN module is not updating the message buffer (no data is being received and stored).	1	The FCN module is updating the message buffer (data is being received and stored).
FCNnMmMUCF	Bit indicating that Message Buffer Data is being Updated							
0	The FCN module is not updating the message buffer (no data is being received and stored).							
1	The FCN module is updating the message buffer (data is being received and stored).							
9	FCNnMmTCPF ^{Note 1}	<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:15%;">FCNnMmTCPF^{Note 1}</td> <td>Transmission Complete Flag</td> </tr> <tr> <td style="text-align:center;">0</td> <td>Transmission failed.^{Note 2}</td> </tr> <tr> <td style="text-align:center;">1</td> <td>Transmission is complete.</td> </tr> </table> <p style="margin-top: 10px;">Notes 1. FCNnMmTCPF is cleared if FCNnMmRDYF is changed or FCNnMmTRQF is set.</p> <p style="margin-top: 5px;">2. If transmission abort was requested by clearing the FCNnMmTRQF flag by the application, FCNnMmTCPF = 0 indicates that the transmission has been successfully aborted.</p>	FCNnMmTCPF ^{Note 1}	Transmission Complete Flag	0	Transmission failed. ^{Note 2}	1	Transmission is complete.
FCNnMmTCPF ^{Note 1}	Transmission Complete Flag							
0	Transmission failed. ^{Note 2}							
1	Transmission is complete.							
6	FCNnMmNHMF	<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:15%;">FCNnMmNHMF</td> <td>History Mask Flag^{Note 3}</td> </tr> <tr> <td style="text-align:center;">0</td> <td>Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is not masked.</td> </tr> <tr> <td style="text-align:center;">1</td> <td>Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is masked.</td> </tr> </table> <p style="margin-top: 10px;">3. If updating is masked, transmit and receive history lists are not updated even when reception or transmission on the corresponding message buffer finishes.</p>	FCNnMmNHMF	History Mask Flag ^{Note 3}	0	Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is not masked.	1	Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is masked.
FCNnMmNHMF	History Mask Flag ^{Note 3}							
0	Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is not masked.							
1	Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is masked.							

(2/3)

Bit Position	Bit Name	Description						
4	FCNnMmMOWF	<table border="1"> <thead> <tr> <th>FCNnMmMOWF</th> <th>Message Buffer Overwrite Status Bit</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Remark: This bit will not be set (1) if a remote frame is received and stored in a transmit message buffer with FCNnMmDTNF = 1.</p>	FCNnMmMOWF	Message Buffer Overwrite Status Bit	0	The message buffer is not overwritten by a newly received data or remote frame.	1	The message buffer is overwritten by a newly received data or remote frame.
FCNnMmMOWF	Message Buffer Overwrite Status Bit							
0	The message buffer is not overwritten by a newly received data or remote frame.							
1	The message buffer is overwritten by a newly received data or remote frame.							
3	FCNnMmIENF	<table border="1"> <thead> <tr> <th>FCNnMmIENF</th> <th>Message Buffer Interrupt Request Enable Bit</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Caution: Always set FCNnMmIENF and FCNnMmRDYF separately.</p>	FCNnMmIENF	Message Buffer Interrupt Request Enable Bit	0	Receive message buffer: Valid message reception completion interrupt is disabled. Transmit message buffer: Normal message transmission completion interrupt and transmit abort interrupt are disabled.	1	Receive message buffer: Valid message reception completion interrupt is enabled. Transmit message buffer: Normal message transmission completion interrupt and transmit abort interrupt are enabled.
FCNnMmIENF	Message Buffer Interrupt Request Enable Bit							
0	Receive message buffer: Valid message reception completion interrupt is disabled. Transmit message buffer: Normal message transmission completion interrupt and transmit abort interrupt are disabled.							
1	Receive message buffer: Valid message reception completion interrupt is enabled. Transmit message buffer: Normal message transmission completion interrupt and transmit abort interrupt are enabled.							
2	FCNnMmDTNF	<table border="1"> <thead> <tr> <th>FCNnMmDTNF</th> <th>Message Buffer Data Update Bit</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Caution: Do not set FCNnMmDTNF to 1 by software. Be sure to write 0 to bit 10.</p>	FCNnMmDTNF	Message Buffer Data Update Bit	0	No new data frame or remote frame has been stored in the message buffer.	1	A new data frame or remote frame has been stored in the message buffer.
FCNnMmDTNF	Message Buffer Data Update Bit							
0	No new data frame or remote frame has been stored in the message buffer.							
1	A new data frame or remote frame has been stored in the message buffer.							

Bit Position	Bit Name	Description						
1	FCNnMmTRQF	<table border="1"> <thead> <tr> <th>FCNnMmTRQF</th> <th>Message Buffer Transmission Request Bit</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Cautions</p> <ol style="list-style-type: none"> Do not set FCNnMmTRQF and FCNnMmRDYF to 1 at the same time. Set FCNnMmRDYF = 1 before setting FCNnMmTRQF = 1. Only set FCNnMmTRQF to 1 for buffers other than transmit message buffers (buffers with FCNnMmSSMT[3:0] ≠ 4'b0000 or FCNnMmSSAM = 0). 	FCNnMmTRQF	Message Buffer Transmission Request Bit	0	No message frame transmitting request that is pending or being transmitted is in the message buffer.	1	The message buffer is holding a message frame pending for transmission or is transmitting a message frame.
FCNnMmTRQF	Message Buffer Transmission Request Bit							
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.							
1	The message buffer is holding a message frame pending for transmission or is transmitting a message frame.							
0	FCNnMmRDYF	<table border="1"> <thead> <tr> <th>FCNnMmRDYF</th> <th>Message Buffer Ready Bit</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Cautions</p> <ol style="list-style-type: none"> Always set FCNnMmIENF and FCNnMmRDYF separately. Do not set FCNnMmTRQF and FCNnMmRDYF to 1 at the same time. Set FCNnMmRDYF = 1 before setting FCNnMmTRQF = 1. Do not clear FCNnMmRDYF to "0" during message transmission. Execute transmission abort processing to clear FCNnMmRDYF to redefine the message buffer. 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. Do not write to another FCN message buffer register until the clearing of FCNnMmRDYF is confirmed by checking its state. 	FCNnMmRDYF	Message Buffer Ready Bit	0	The message buffer can be written by software. The FCN module cannot write to the message buffer.	1	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.
FCNnMmRDYF	Message Buffer Ready Bit							
0	The message buffer can be written by software. The FCN module cannot write to the message buffer.							
1	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.							

(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 border="1" style="width:100%; border-collapse: collapse;"> <tr> <th style="width:20%;">FCNnMmSENH</th> <th style="width:20%;">FCNnMmCLNH</th> <th style="width:60%;">Setting of FCNnMmNHMF Bit</th> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>FCNnMmNHMF is cleared (0).</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>FCNnMmNHMF is set (1).</td> </tr> <tr> <td colspan="2" style="text-align: center;">Other than above</td> <td>FCNnMmNHMF is not changed.</td> </tr> </table>	FCNnMmSENH	FCNnMmCLNH	Setting of FCNnMmNHMF Bit	0	1	FCNnMmNHMF is cleared (0).	1	0	FCNnMmNHMF is set (1).	Other than above		FCNnMmNHMF is not changed.
		FCNnMmSENH	FCNnMmCLNH	Setting of FCNnMmNHMF Bit										
		0	1	FCNnMmNHMF is cleared (0).										
		1	0	FCNnMmNHMF is set (1).										
Other than above		FCNnMmNHMF is not changed.												
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <th style="width:20%;">FCNnMmCLMW</th> <th style="width:80%;">Setting of FCNnMmMOWF Bit</th> </tr> <tr> <td style="text-align: center;">0</td> <td>FCNnMmMOWF is not changed.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>FCNnMmMOWF is cleared (0).</td> </tr> </table>	FCNnMmCLMW	Setting of FCNnMmMOWF Bit	0	FCNnMmMOWF is not changed.	1	FCNnMmMOWF is cleared (0).								
FCNnMmCLMW	Setting of FCNnMmMOWF Bit													
0	FCNnMmMOWF is not changed.													
1	FCNnMmMOWF is cleared (0).													
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <th style="width:20%;">FCNnMmSEIE</th> <th style="width:20%;">FCNnMmCLIE</th> <th style="width:60%;">Setting of FCNnMmIENF Bit</th> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>FCNnMmIENF is cleared (0).</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>FCNnMmIENF is set (1).</td> </tr> <tr> <td colspan="2" style="text-align: center;">Other than above</td> <td>FCNnMmIENF is not changed.</td> </tr> </table>	FCNnMmSEIE	FCNnMmCLIE	Setting of FCNnMmIENF Bit	0	1	FCNnMmIENF is cleared (0).	1	0	FCNnMmIENF is set (1).	Other than above		FCNnMmIENF is not changed.		
FCNnMmSEIE	FCNnMmCLIE	Setting of FCNnMmIENF Bit												
0	1	FCNnMmIENF is cleared (0).												
1	0	FCNnMmIENF is set (1).												
Other than above		FCNnMmIENF is not changed.												
2	FCNnMmCLDN	<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <th style="width:20%;">FCNnMmCLDN</th> <th style="width:80%;">Setting of FCNnMmDTNF Bit</th> </tr> <tr> <td style="text-align: center;">0</td> <td>FCNnMmDTNF is not changed.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>FCNnMmDTNF is cleared (0).</td> </tr> </table> <p style="color: green; font-weight: bold; margin-top: 10px;">Remark: If FCNnMmDTNF is cleared at the end of ID field reception, the frames being received will be saved into the corresponding message buffer.</p>	FCNnMmCLDN	Setting of FCNnMmDTNF Bit	0	FCNnMmDTNF is not changed.	1	FCNnMmDTNF is cleared (0).						
FCNnMmCLDN	Setting of FCNnMmDTNF Bit													
0	FCNnMmDTNF is not changed.													
1	FCNnMmDTNF is cleared (0).													
9, 1	FCNnMmSETR, FCNnMmCLTR	<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <th style="width:20%;">FCNnMmSETR</th> <th style="width:20%;">FCNnMmCLTR</th> <th style="width:60%;">Setting of FCNnMmTRQF Bit</th> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>FCNnMmTRQF is cleared (0).</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>FCNnMmTRQF is set (1).</td> </tr> <tr> <td colspan="2" style="text-align: center;">Other than above</td> <td>FCNnMmTRQF is not changed.</td> </tr> </table>	FCNnMmSETR	FCNnMmCLTR	Setting of FCNnMmTRQF Bit	0	1	FCNnMmTRQF is cleared (0).	1	0	FCNnMmTRQF is set (1).	Other than above		FCNnMmTRQF is not changed.
FCNnMmSETR	FCNnMmCLTR	Setting of FCNnMmTRQF Bit												
0	1	FCNnMmTRQF is cleared (0).												
1	0	FCNnMmTRQF is set (1).												
Other than above		FCNnMmTRQF is not changed.												
8, 0	FCNnMmSERY, FCNnMmCLRY	<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <th style="width:20%;">FCNnMmSERY</th> <th style="width:20%;">FCNnMmCLRY</th> <th style="width:60%;">Setting of FCNnMmRDYF Bit</th> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>FCNnMmRDYF is cleared (0).</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>FCNnMmRDYF is set (1).</td> </tr> <tr> <td colspan="2" style="text-align: center;">Other than above</td> <td>FCNnMmRDYF is not changed.</td> </tr> </table>	FCNnMmSERY	FCNnMmCLRY	Setting of FCNnMmRDYF Bit	0	1	FCNnMmRDYF is cleared (0).	1	0	FCNnMmRDYF is set (1).	Other than above		FCNnMmRDYF is not changed.
FCNnMmSERY	FCNnMmCLRY	Setting of FCNnMmRDYF Bit												
0	1	FCNnMmRDYF is cleared (0).												
1	0	FCNnMmRDYF is set (1).												
Other than above		FCNnMmRDYF is not changed.												

19.6 Initialization of CAN Controller

19.6.1 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 19.14, Operation of the CAN Controller.

19.6.2 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.

19.6.3 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 19.17, 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 19.8.4(1) Aborting Transmission Other than Automatic Block Transmission (ABT) and 19.8.4(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.

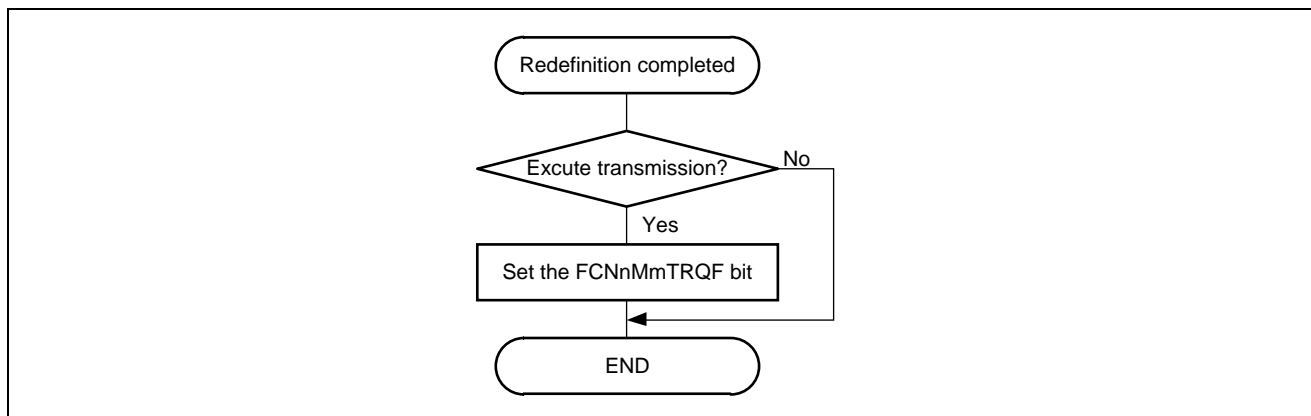


Figure 19.4 Setting Transmission Request (FCNnMmCTL.FCNnMmTRQF) to Transmit Message Buffer after Redefinition

- 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 19.17, 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 19.4, 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.

19.6.4 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

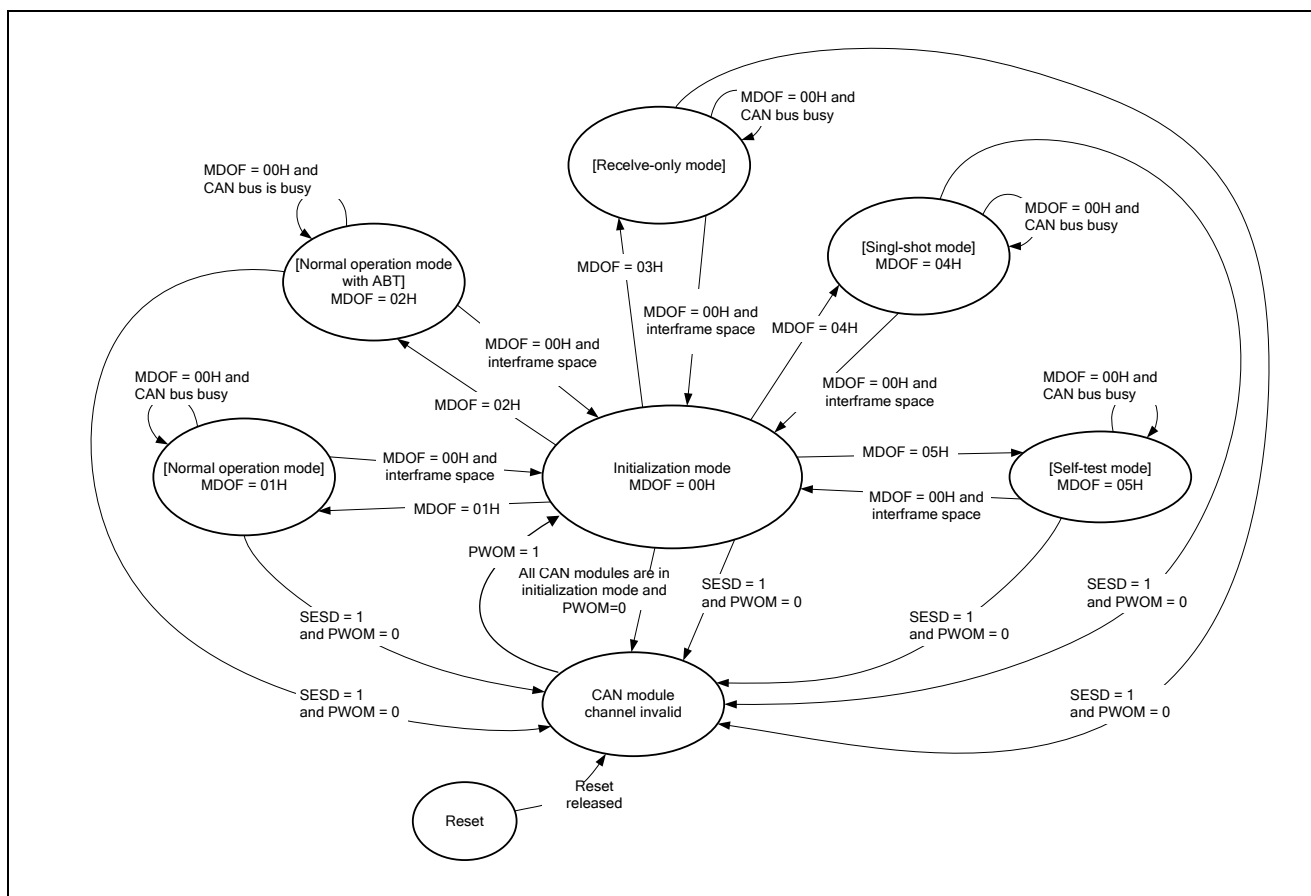


Figure 19.5 Transition to Operation Mode

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 19.14, Re-initialization without Using the Software Reset).

19.7 Message Reception

19.7.1 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 19.15 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

19.7.2 Receive Data Read

To keep data consistency when reading FCN message buffers, perform the data reading according to Figure 19.31, Reception via Interrupt (Using FCNnCMLISTR Register), to Figure 19.34, 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 19.6, 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.

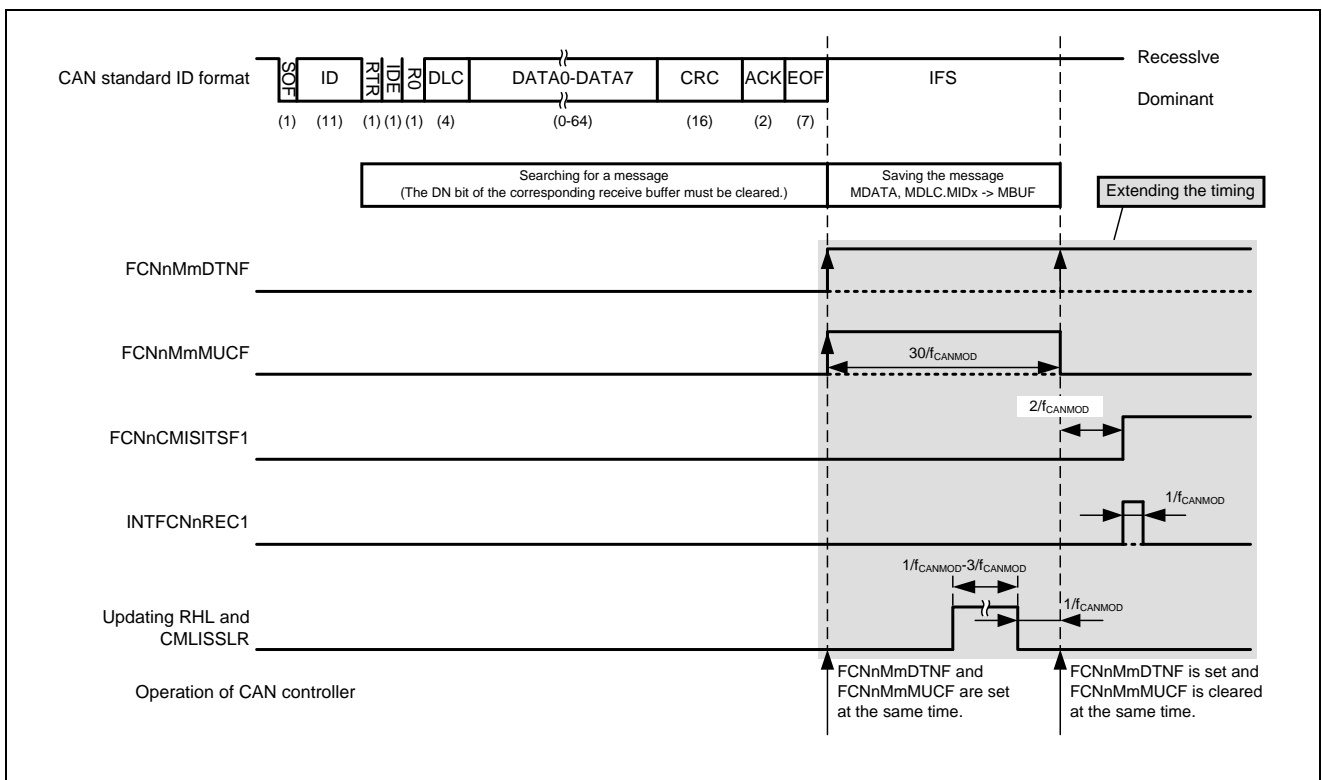


Figure 19.6 Reception Timing

19.7.3 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.

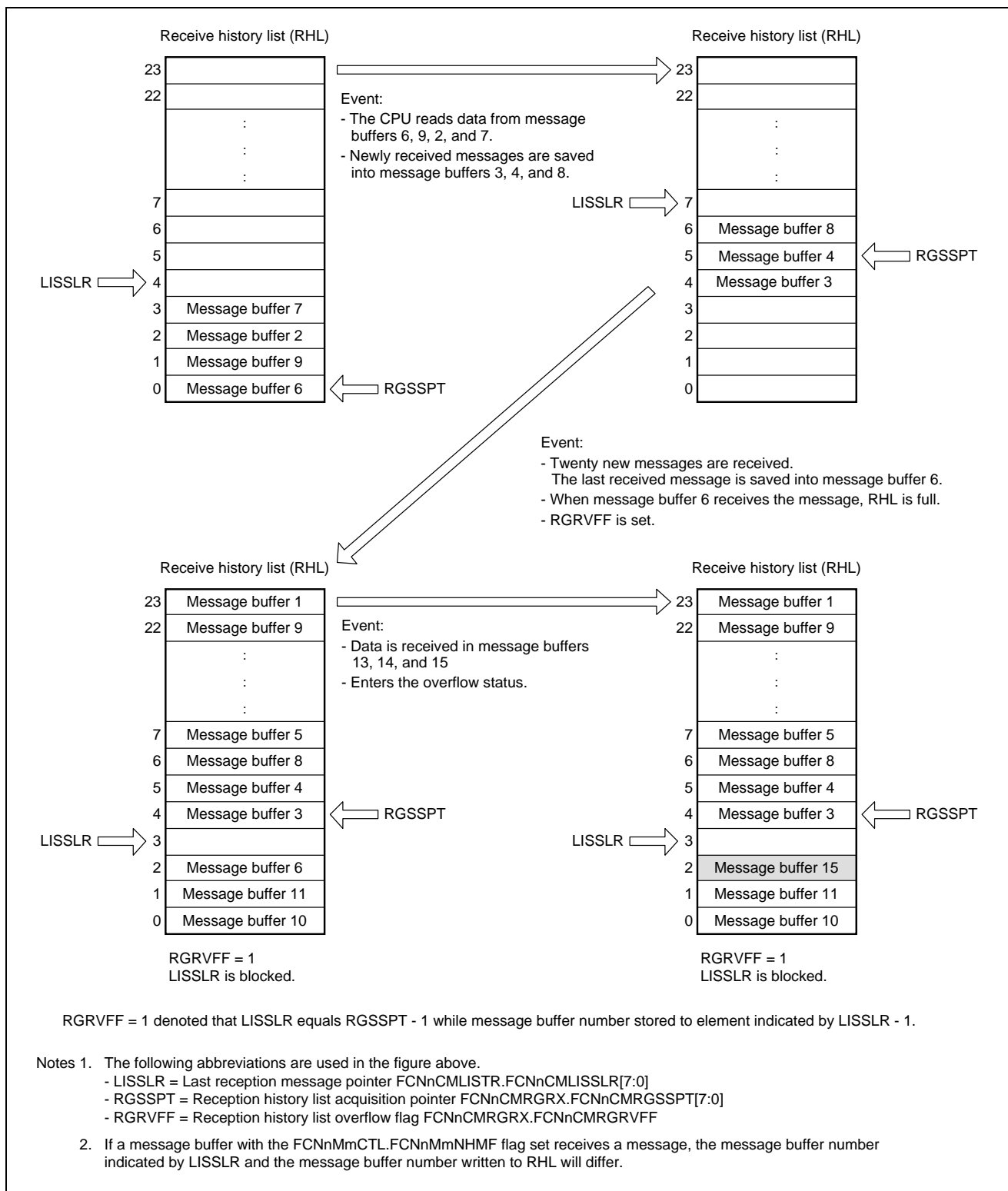


Figure 19.7 Receive History List

19.7.4 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)

FCNnCMMKSSID[.]

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.

19.7.5 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 19.16, List of FCN Module Interrupt Sources.**

19.7.6 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.

19.8 Message Transmission

19.8.1 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).

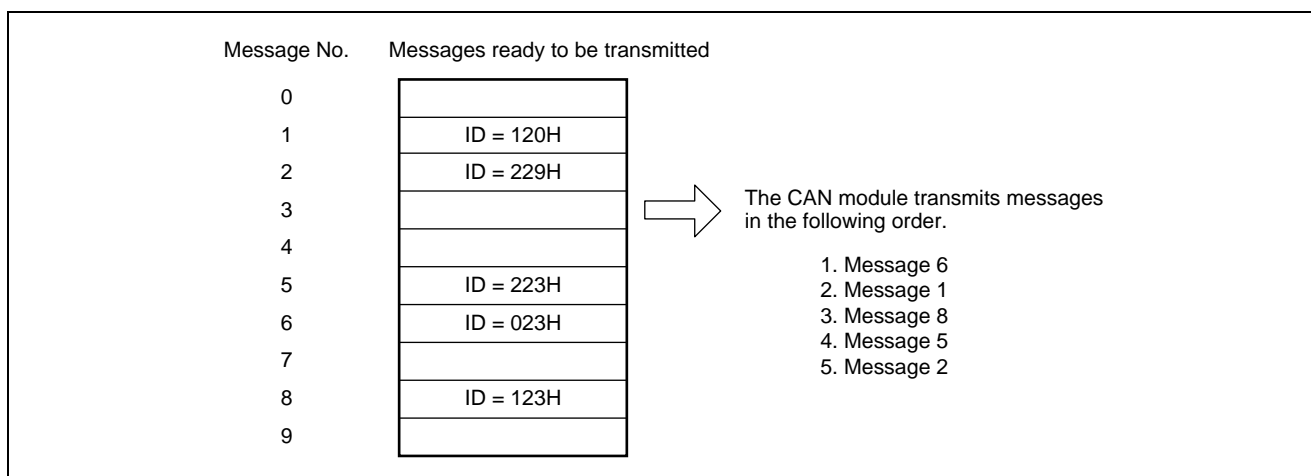


Figure 19.8 Message Processing Example

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 FCNnCMmMIDOW 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 to 1 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.

19.8.2 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 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.

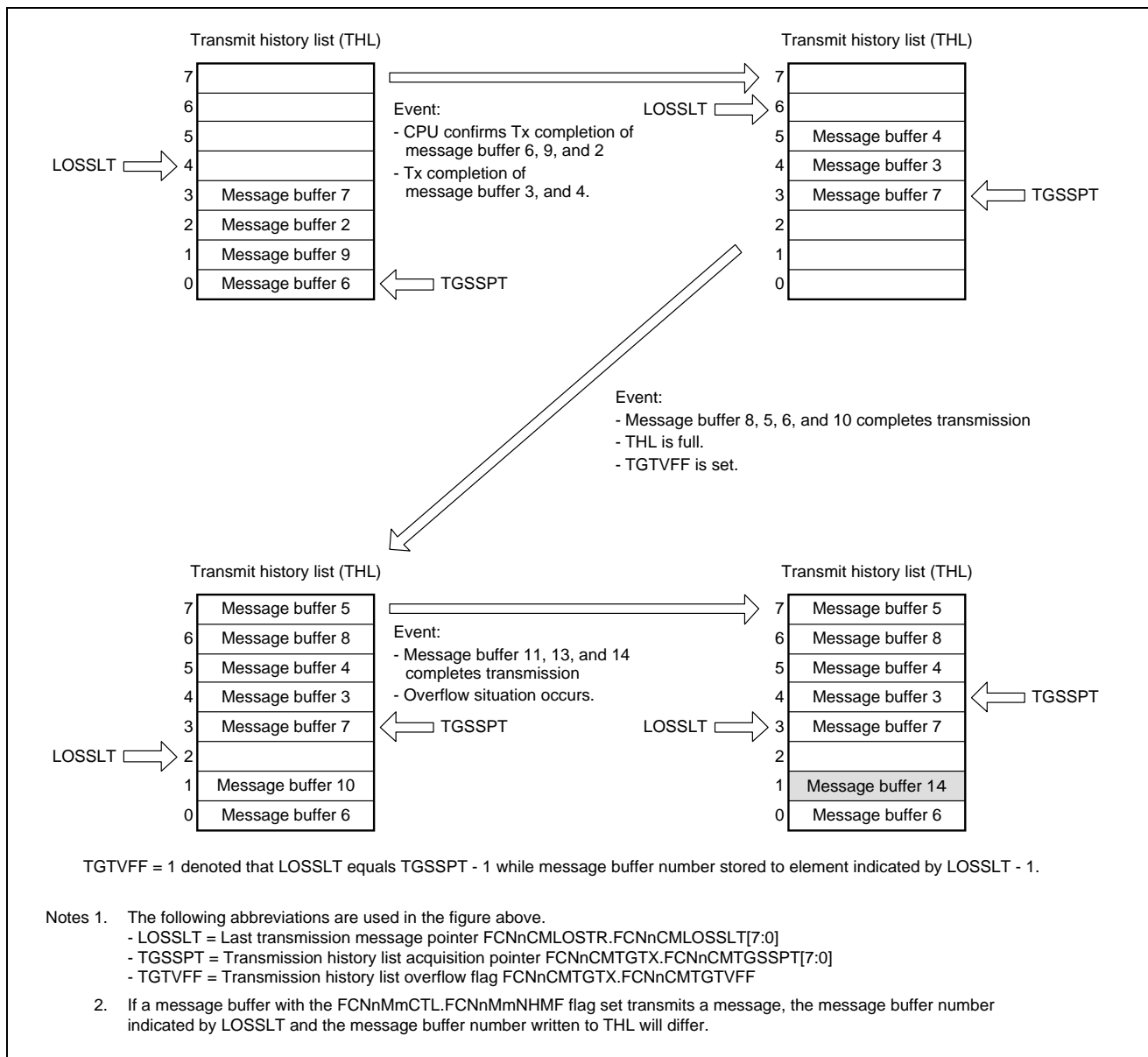


Figure 19.9 Transmit History List

19.8.3 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 = 1 in 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.FCNnGMABCLRFBIT 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 FCNnGMABABTT 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.FCNnGMABSEAC 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.

19.8.4 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 FCNnCMGTGX register, or the FCNnMmCTL.FCNnMmTCPF flag, which indicate the transmission status on the CAN bus (for details, refer to the processing in Figure 19.24, 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 19.26, 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.

19.8.5 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.

19.9 Power Saving Modes

19.9.1 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 $FCN_nCMCLCTL.FCN_nCMCLMDPF[1:0] = 01B$.

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 FCNnCMCLISTR, 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.

19.9.2 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 FCNnCMCLISTR, FCNnCMRGRX, FCNnCMLOSTR, and FCNnCMGTGX 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.

19.9.3 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).

19.10 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 19.16 List of FCN Module Interrupt Sources

No.	Interrupt Status Bit FCNnCMISCTL.	Interrupt Enable Bit FCNnCMIESEIE ^{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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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.FCnMmIENF of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

19.11 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.

19.11.1 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 $FCNnCMCLCTL.FCNnCMCLVALF = 1$.

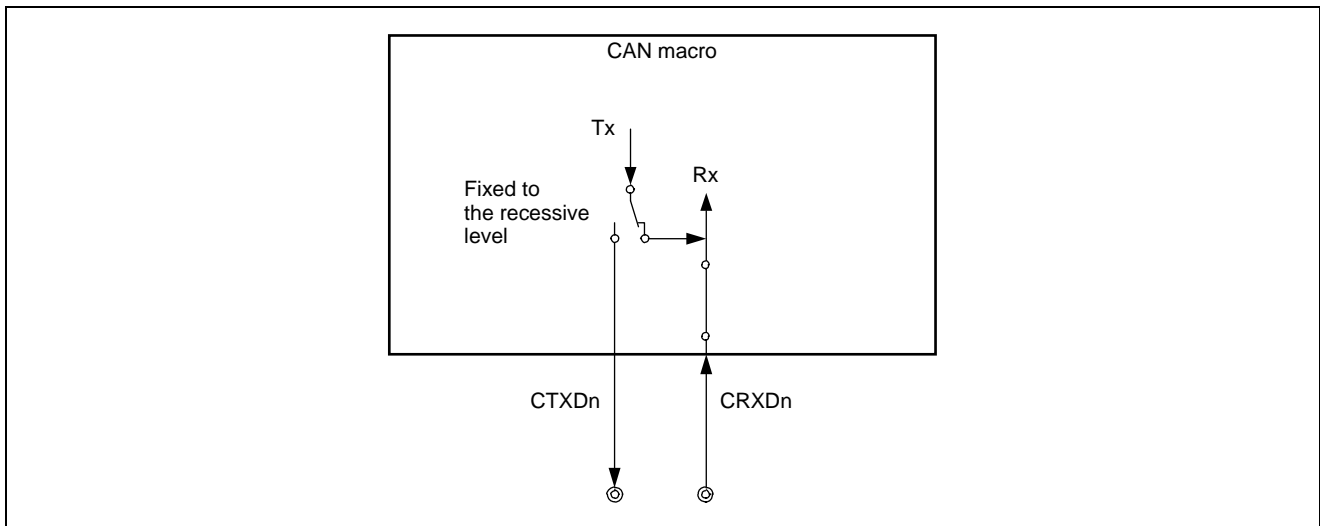


Figure 19.10 FCN Module Terminal Connection in Receive-Only Mode

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.

19.11.2 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.

19.11.3 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.

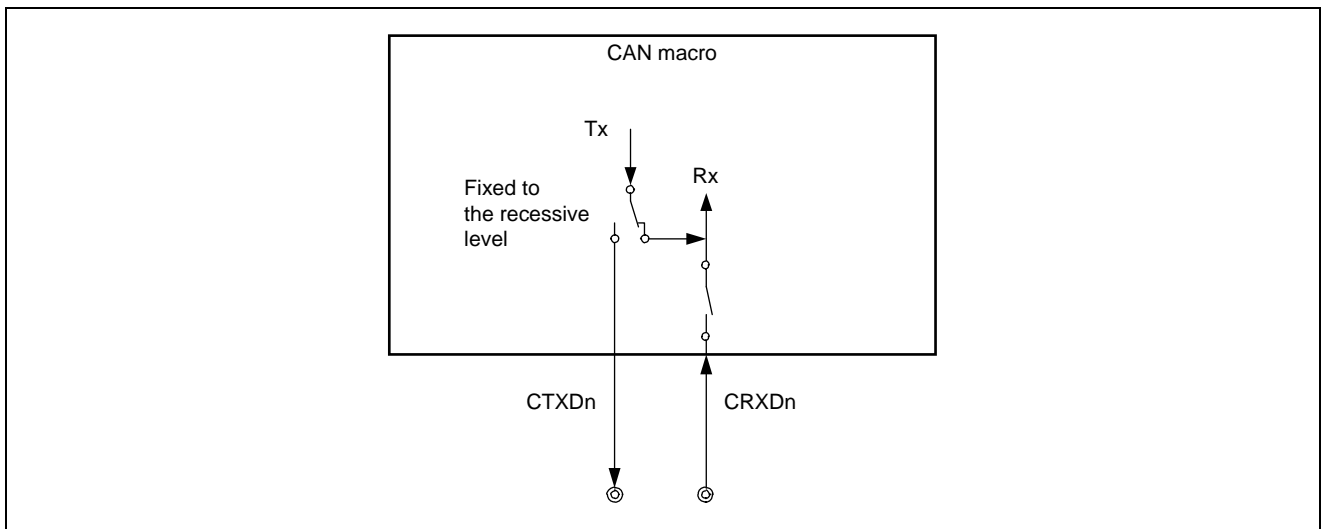


Figure 19.11 FCN Module Terminal Connection in Self-Test Mode

19.11.4 Receive/Transmit Operation in Each Operation Mode

The following table shows the outline of the receive/transmit operation in each operation mode.

Table 19.17 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 ^{Note 1}	No	Yes	Yes
Self-test mode	Yes ^{Note 2}	Yes ^{Note 2}	Yes ^{Note 2}	Yes ^{Note 2}	No	Yes ^{Note 2}	Yes ^{Note 2}

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.

19.12 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.

19.12.1 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.

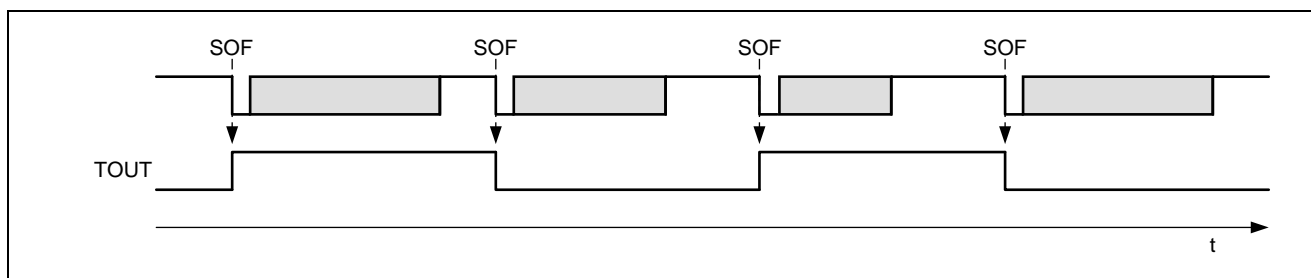


Figure 19.12 Timing Diagram of Capture Signal TSOUT

The TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in Figure 19.12, 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 FCNnCMSTLOKE bit of the FCNnCMSTCTL register. When FCNnCMSTLOKE is cleared to 0, the TSOUT signal toggles upon occurrence of the selected event. If FCNnCMSTLOKE is set to 1, the TSOUT signal toggles upon occurrence of the selected event, but the toggle is stopped as FCNnCMSTCTL.FCNnCMSTSTSGE 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 FCNnCMSTLOKE 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 FCNnCMSTLOKE bit cannot be used.

19.13 Baud Rate Settings

19.13.1 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.

- $5 TQ \leq SPT$ (sampling point) $\leq 17 TQ$
 $SPT = FCNnCMBTS1LG[3:0] + 1$
- $8 TQ \leq DBT$ (data bit time) $\leq 25 TQ$
 $DBT = FCNnCMBTS1LG[3:0] + FCNnCMBTS2LG[2:0] + 1 TQ = FCNnCMBTS2LG[2:0] + SPT$
- $1 TQ \leq FCNnCMBTJWLG[1:0]$ (synchronization jump width) $\leq 4 TQ$
 $FCNnCMBTJWLG[1:0] \leq DBT - SPT$
- $4 TQ \leq TSEG1 \leq 16 TQ$ [$3 \leq FCNnCMBTS1LG[3:0] \leq 15$]
- $1 \leq TSEG2[2:0] \leq 8$ [$0 \leq FCNnCMBTS2LG[2:0] \leq 7$]
- $75 \text{ [nsec]}^{\text{Note}} < 5 \text{ [nsec]} + 1 TQ - 20 \text{ [nsec]}^{\text{Note}}$

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 19.18, Combinations of Available Bit Rate Settings (1/3), shows the combinations of bit rates that satisfy the above conditions.

Table 19.18 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 19.18 Combinations of Available Bit Rate Settings (2/3)

DBT Length	Valid Bit Rate Setting				FCNnCMBTCTL Register Setting Value		Sampling Point (unit: %)
	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 19.18 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 19.18, Combinations of Available Bit Rate Settings (1/3), 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.

19.13.2 Representative Examples of Baud Rate Settings

Table 19.19 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	2	2	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	2	2	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 19.19 Representative Examples of Baud Rate Settings ($f_{CANMOD} = 20 \text{ MHz}$) (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 19.19, Representative Examples of Baud Rate Settings ($f_{CANMOD} = 20 \text{ MHz}$) (1/2), 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.

19.14 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.

19.14.1 Initialization

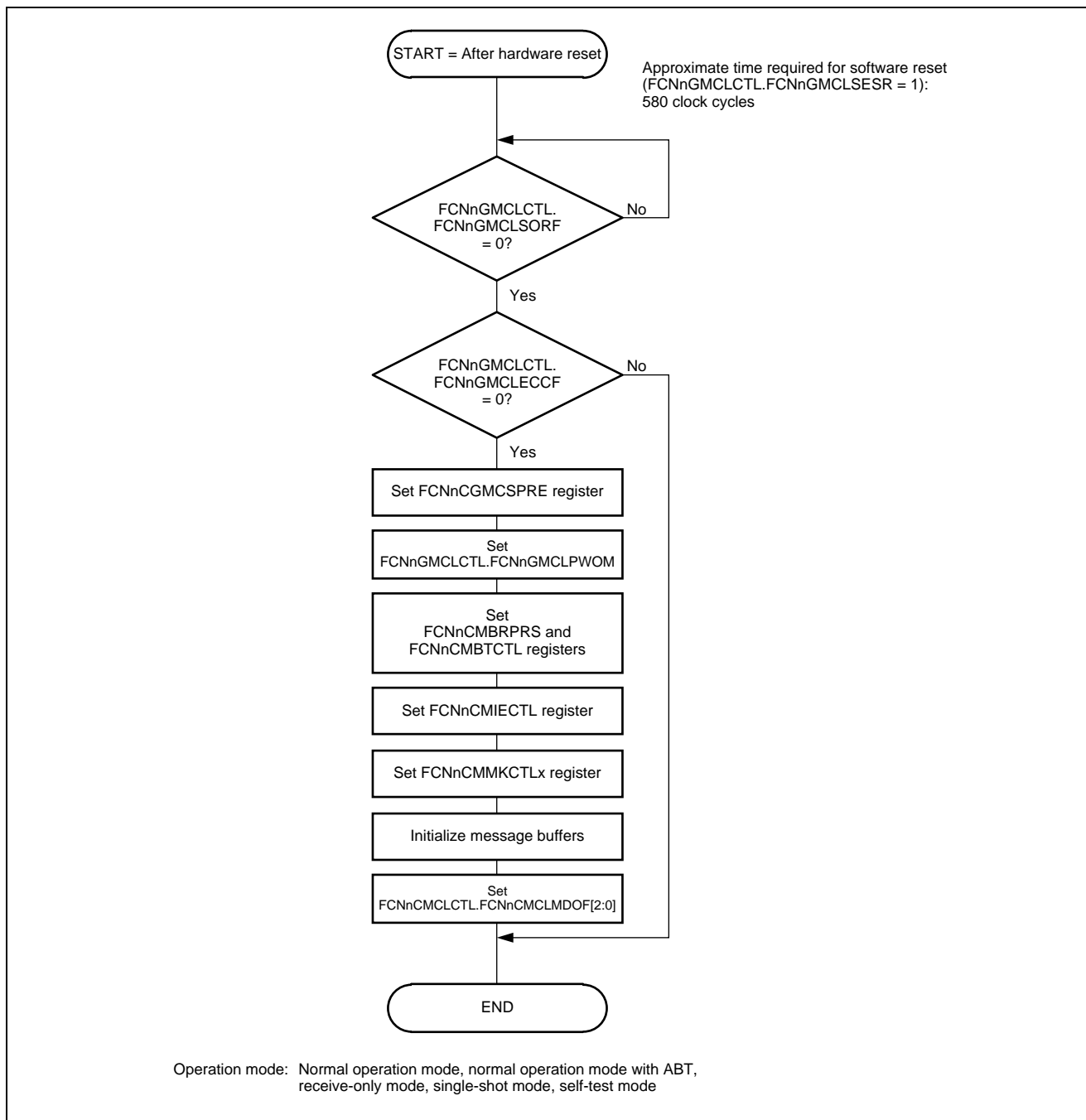


Figure 19.13 Initialization

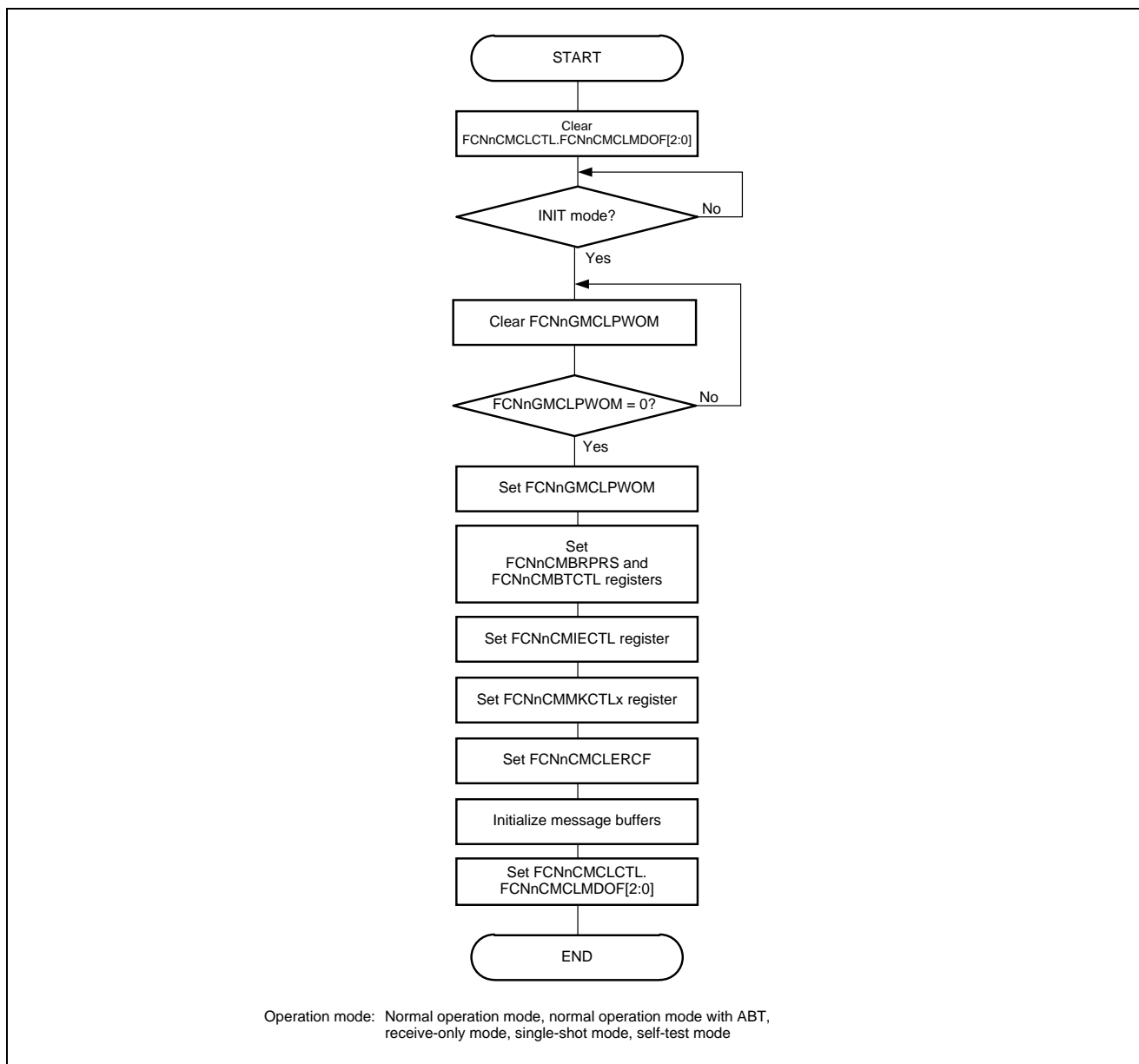


Figure 19.14 Re-initialization without Using the Software Reset

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 19.24, 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 19.25, Transmission Abort Processing (in Normal Operation Mode with ABT) – Repeat Option for Aborted Message, in the normal operation mode with ABT).

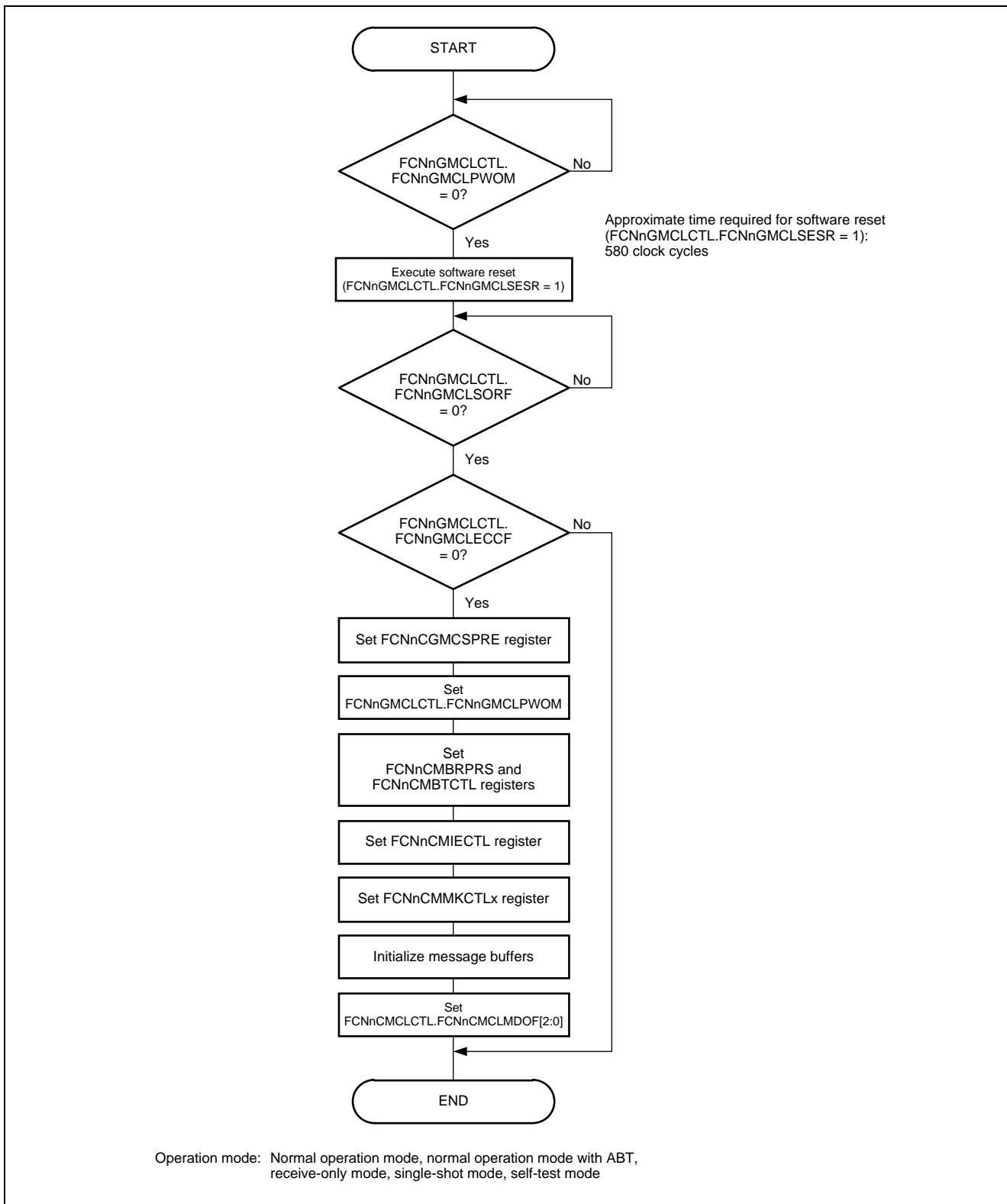


Figure 19.15 Re-Initialization with Software Reset

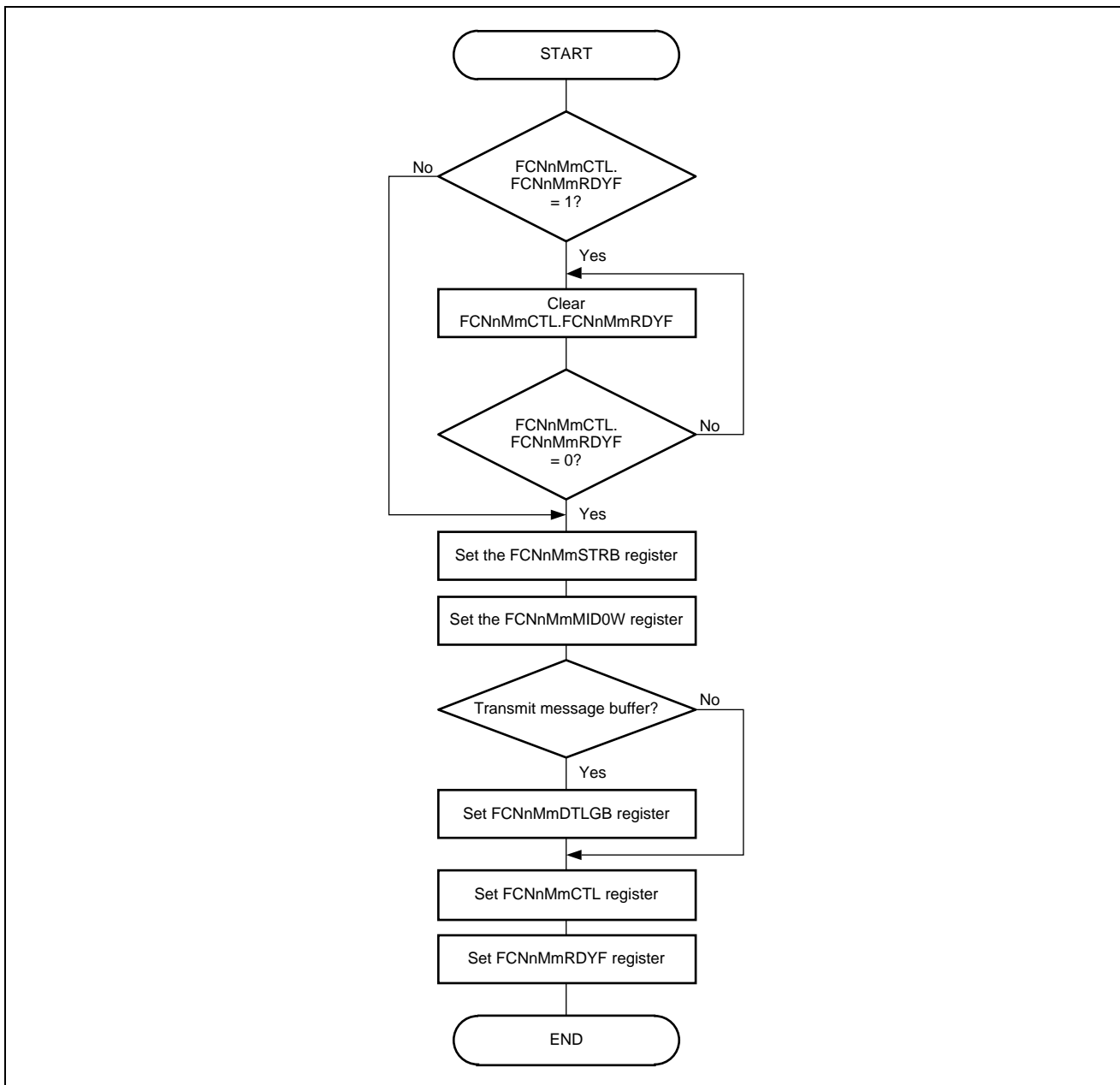


Figure 19.16 Message Buffer Initialization

- 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 19.17, Message Buffer Redefinition during Reception, shows the processing for a receive message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001B to 1000B).

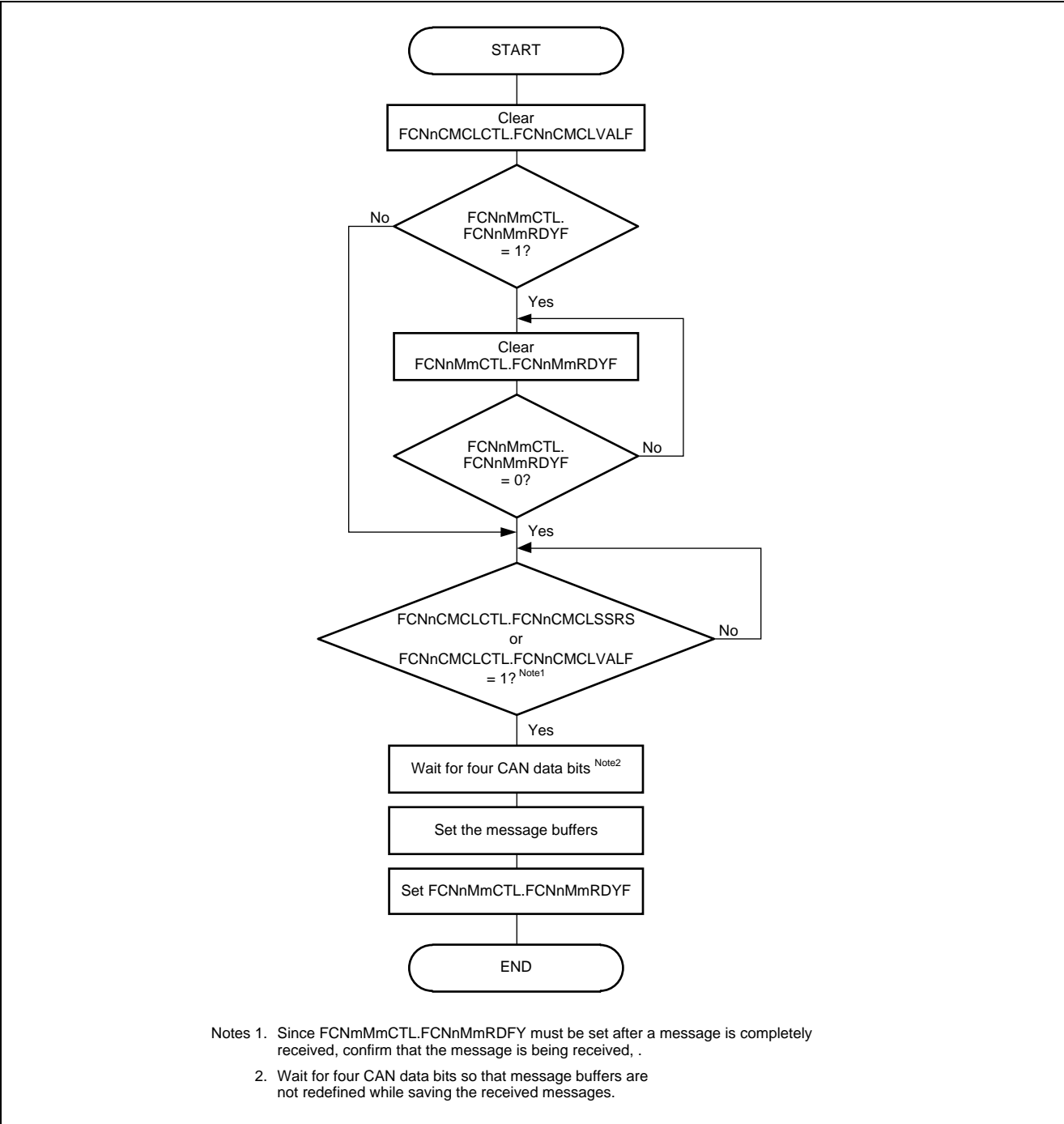


Figure 19.17 Message Buffer Redefinition during Reception

Figure 19.18, Message Buffer Redefinition during Transmission, shows the processing for a transmit message buffer during transmission (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000B).

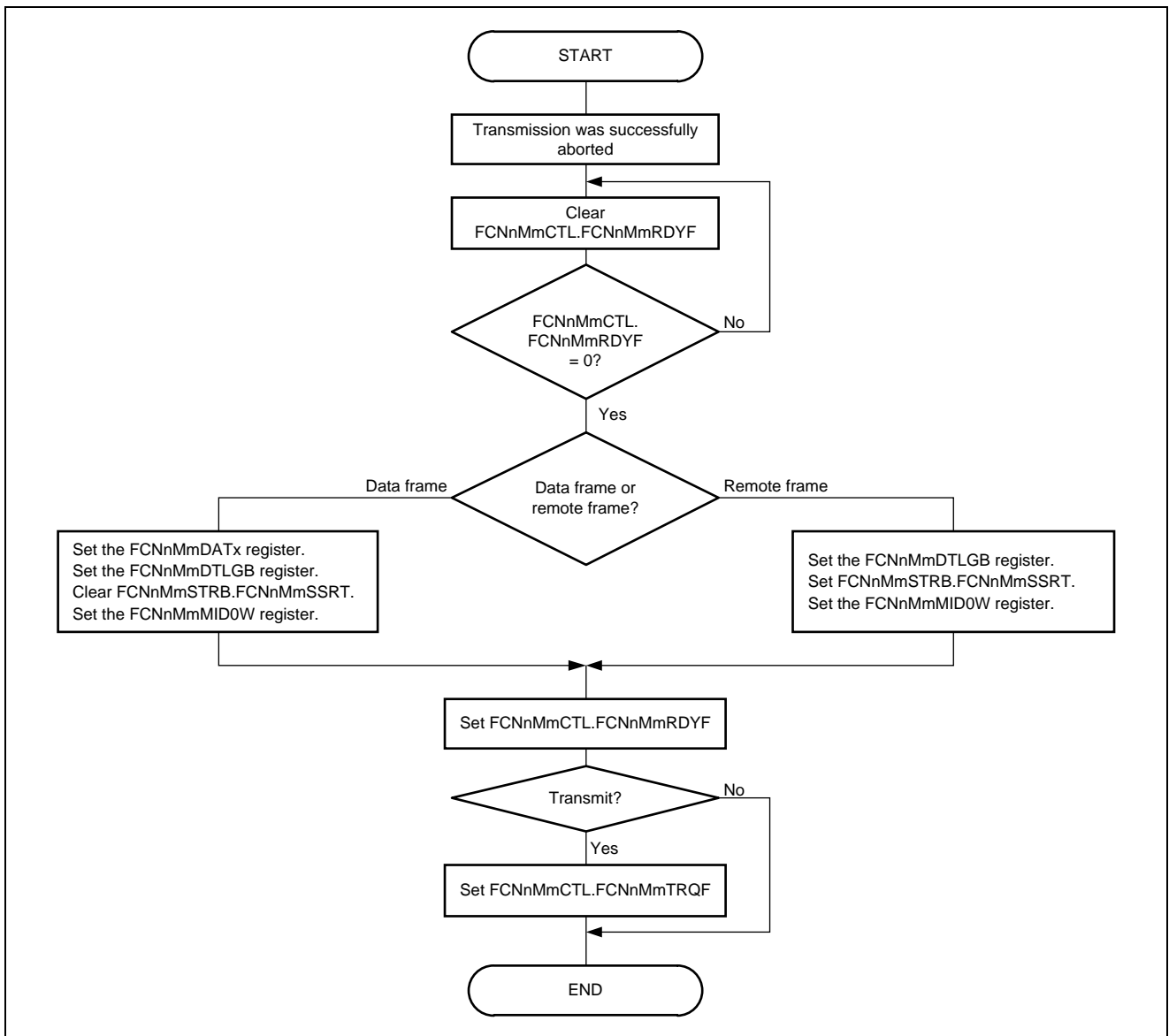


Figure 19.18 Message Buffer Redefinition during Transmission

19.14.2 Message Transmission

Figure 19.19, Message Transmit Processing, shows the processing for a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000B).

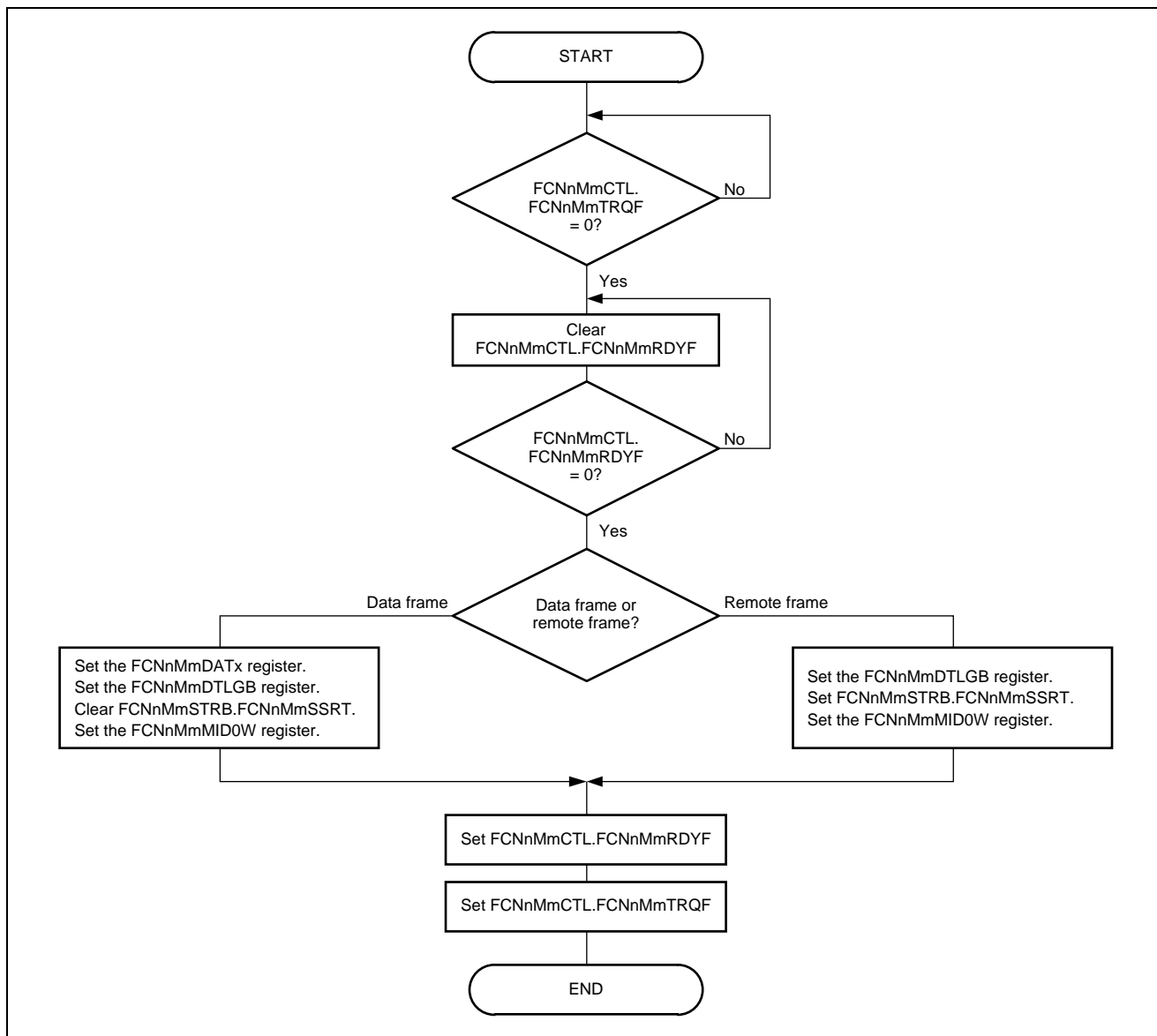


Figure 19.19 Message Transmit Processing

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 19.20, ABT Message Transmit Processing, shows the processing for a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000B)

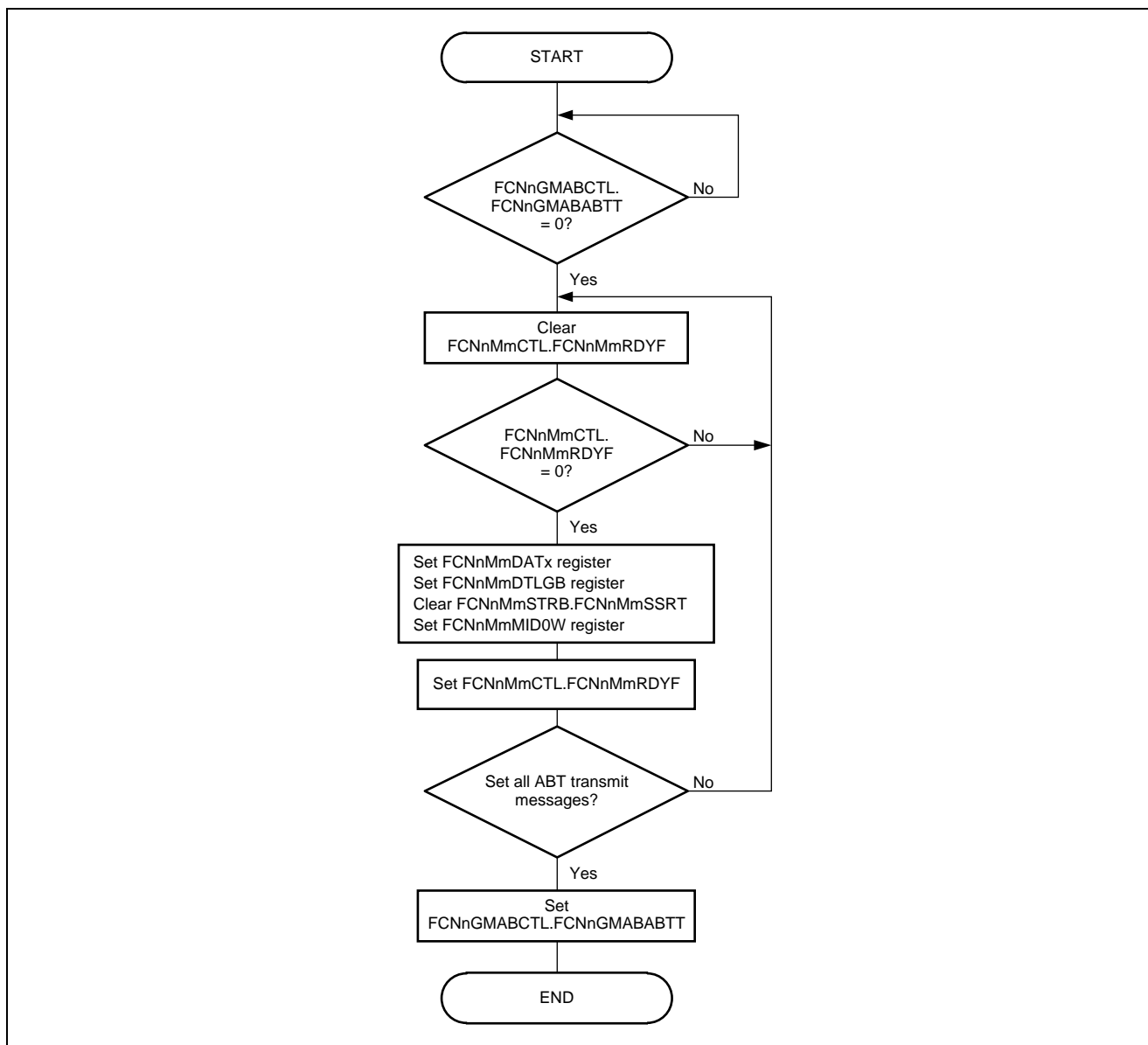


Figure 19.20 ABT Message Transmit Processing

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 19.19, Message Transmit Processing.

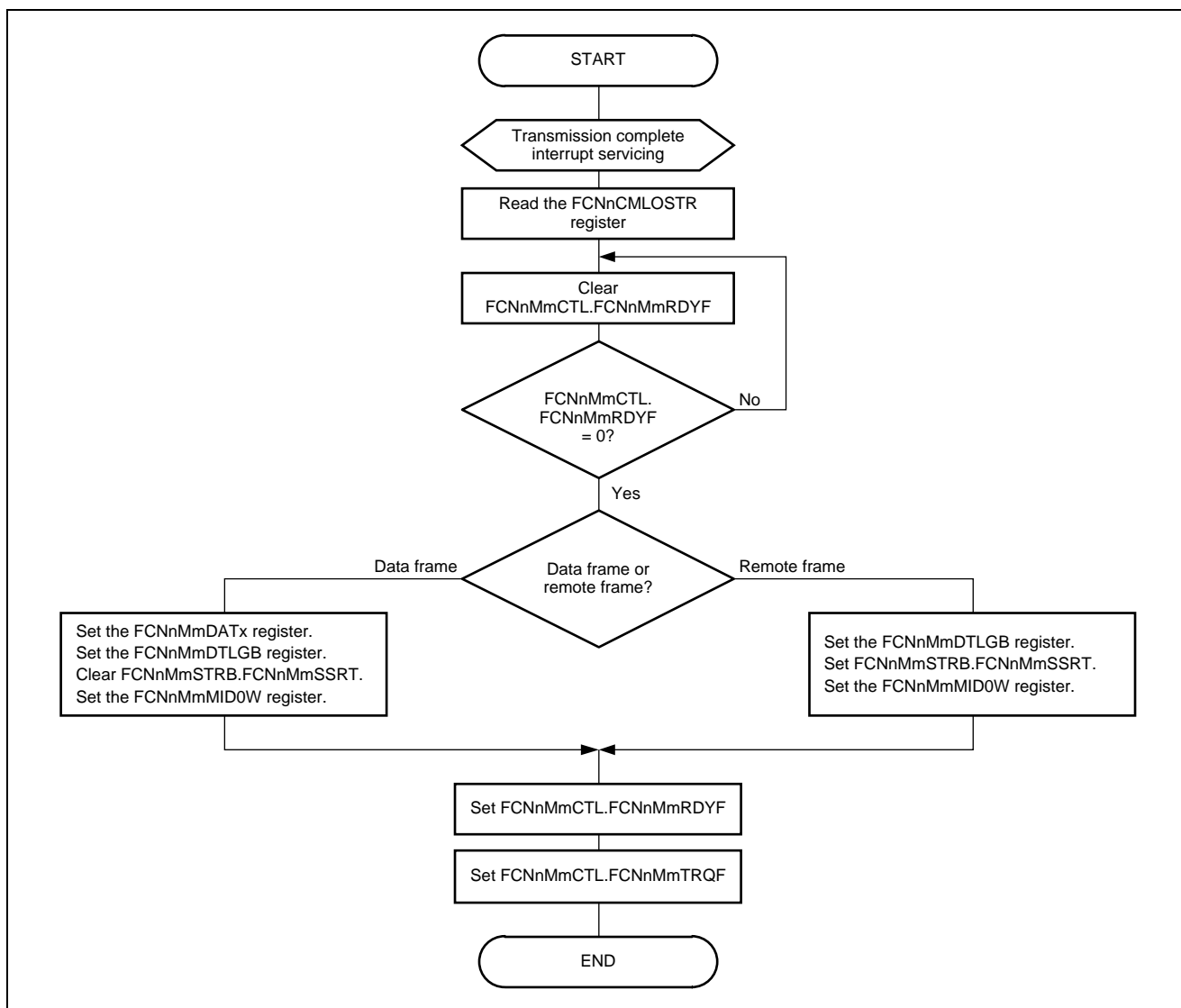


Figure 19.21 Transmission via Interrupt (Using FCNnCMLOSTR Register)

- Cautions 1.** FCNnMmCTL.FCnMmTRQF should be set after FCNnMmCTL.FCnMmRDYF is set.
- 2.** FCNnMmCTL.FCnMmRDYF and FCNnMmCTL.FCnMmTRQF 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.

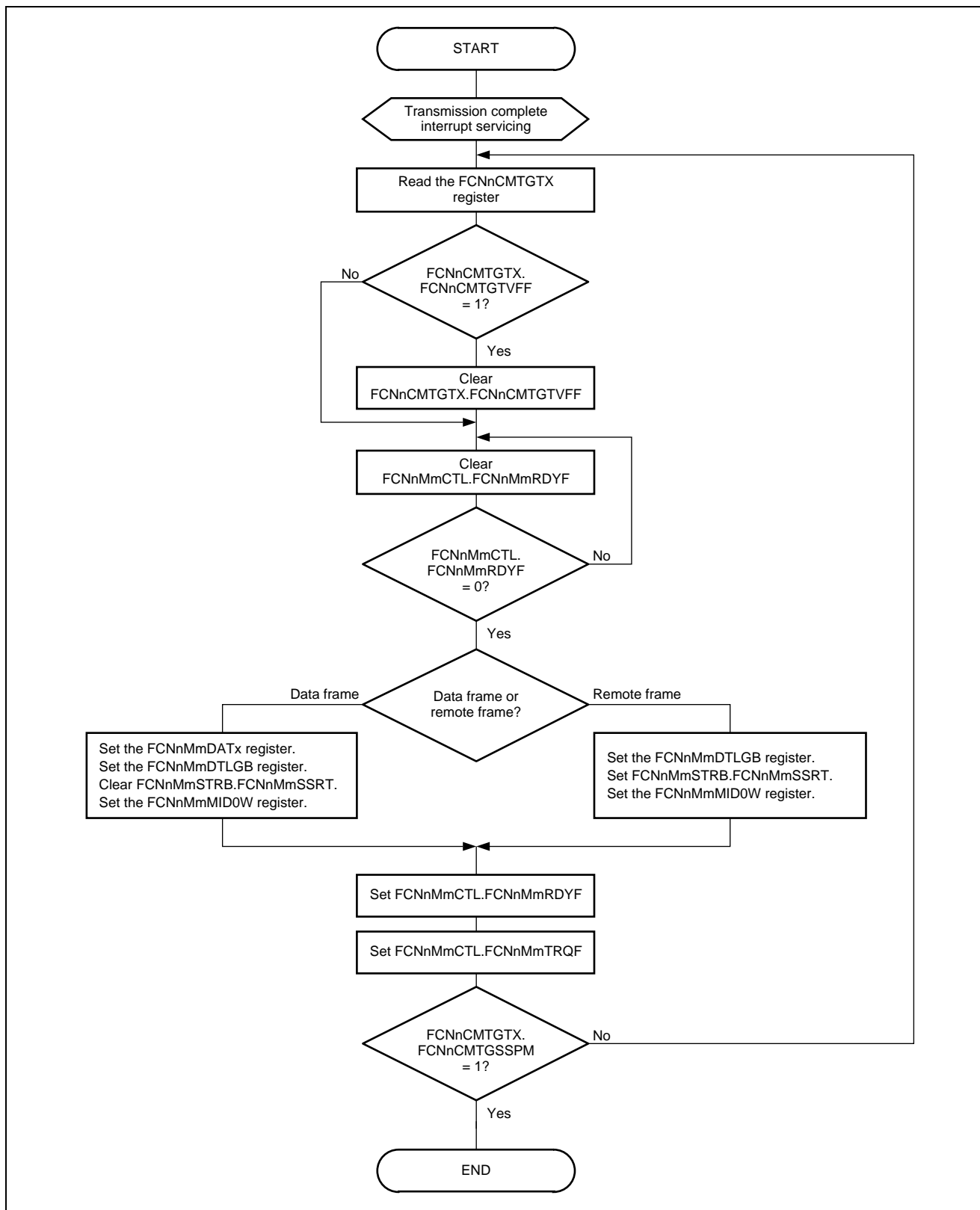


Figure 19.22 Transmission via Interrupt (Using FCNnCMTGTX Register)

- 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.**

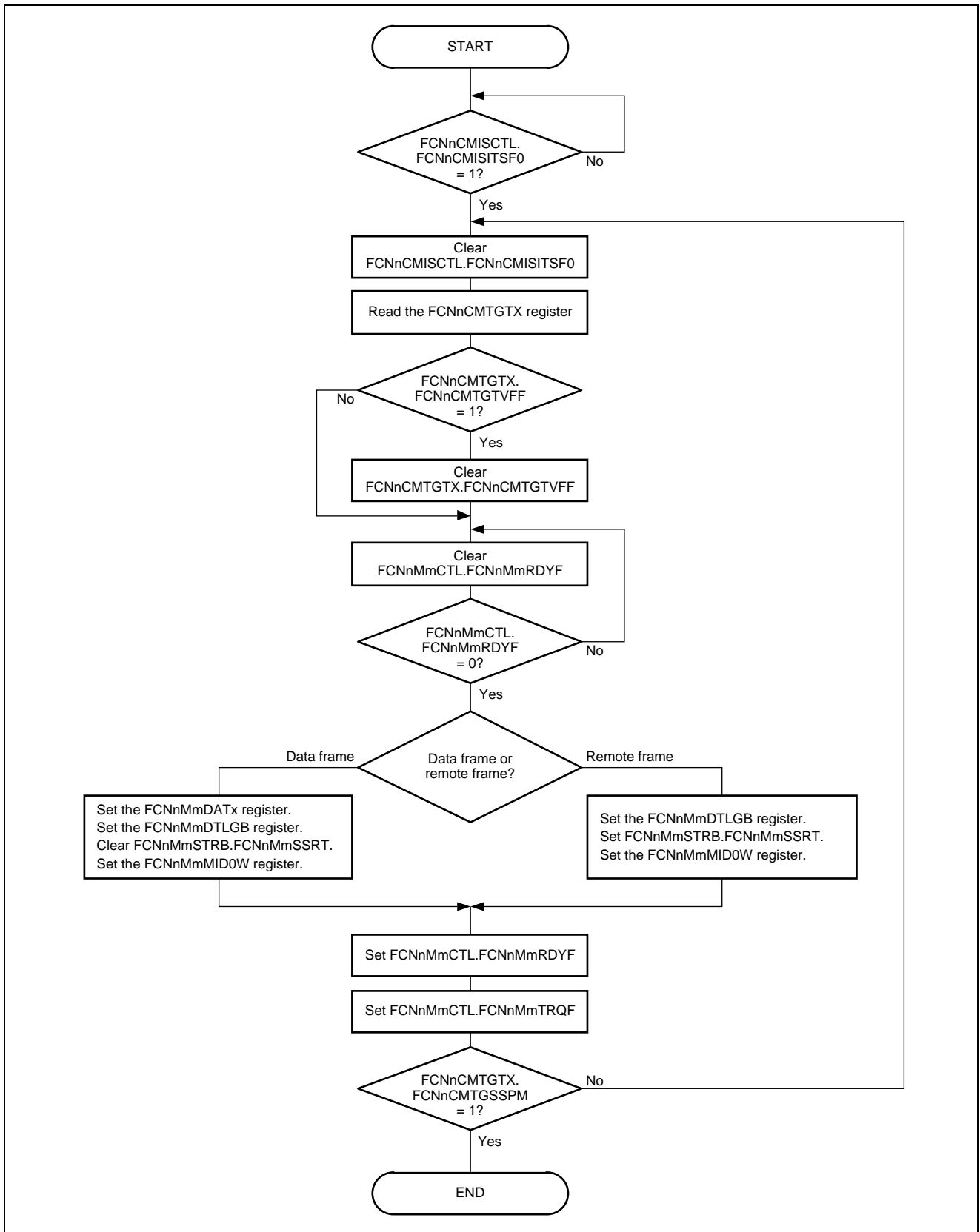


Figure 19.23 Transmission via Software Polling

- 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.

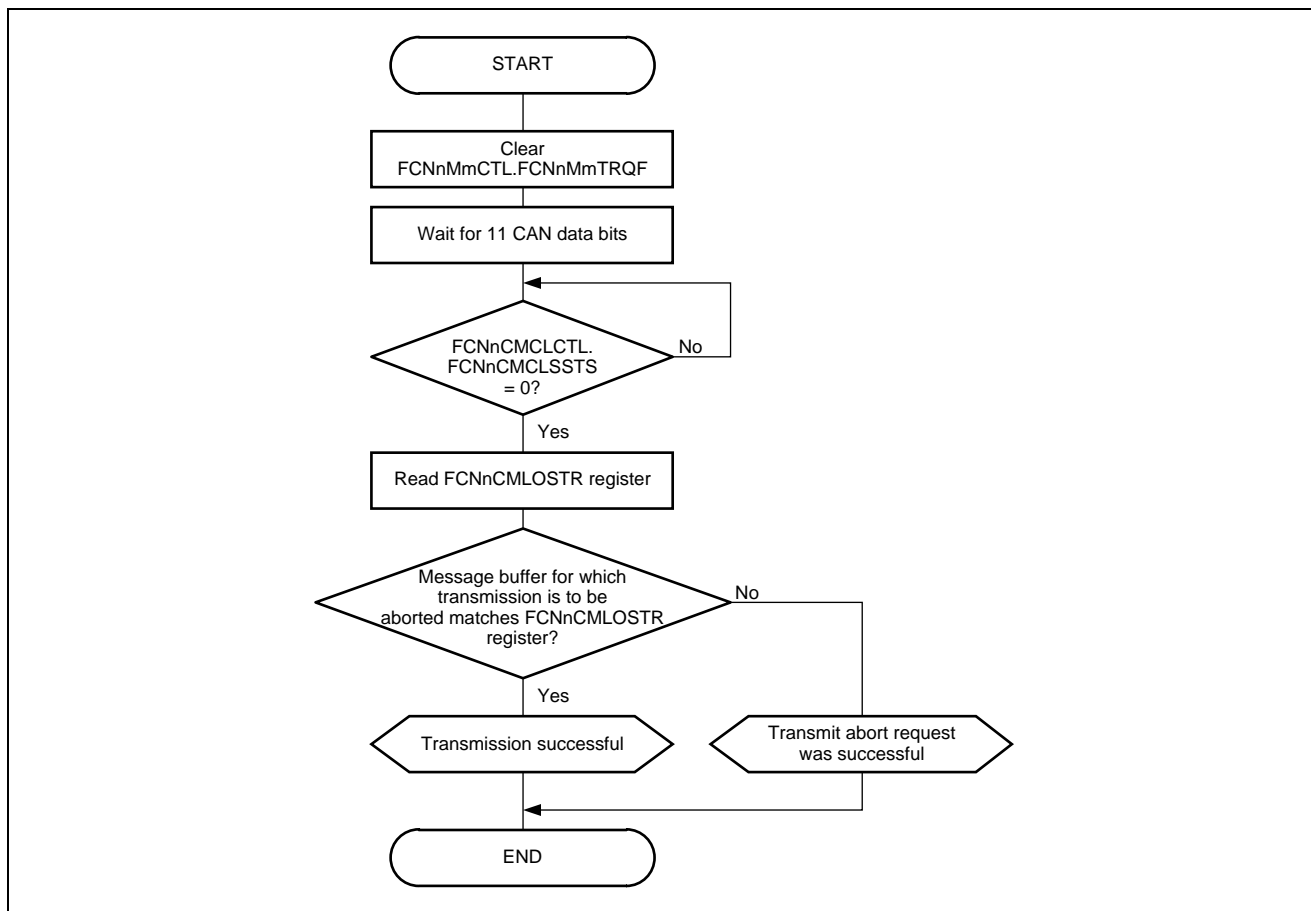


Figure 19.24 Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed)

- 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.**

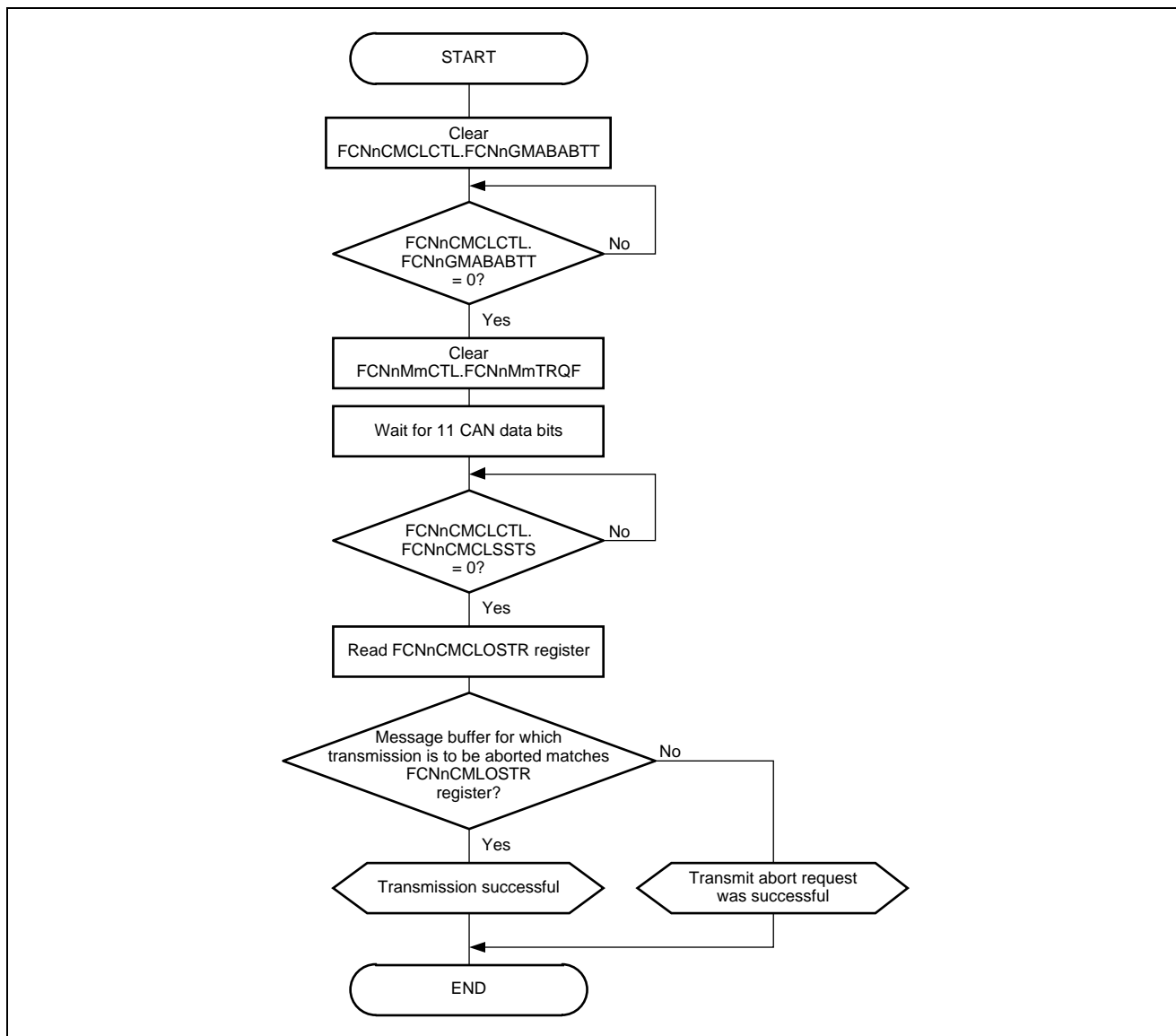


Figure 19.25 Transmission Abort Processing (in Normal Operation Mode with ABT) – Repeat Option for Aborted Message

- 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 19.26, 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.

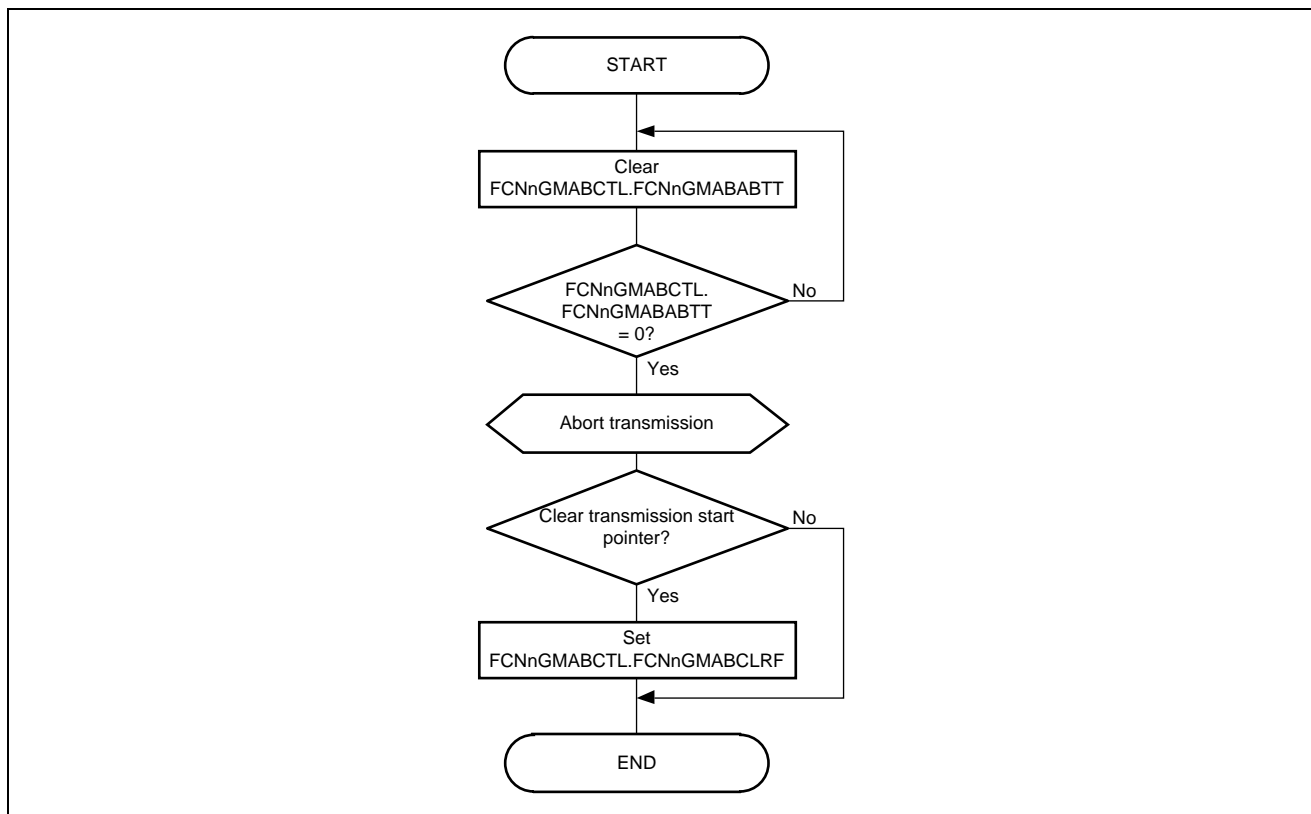


Figure 19.26 ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1)

- 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 19.26, ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1), or Figure 19.27, 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 19.24, Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed).

Figure 19.27, 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.

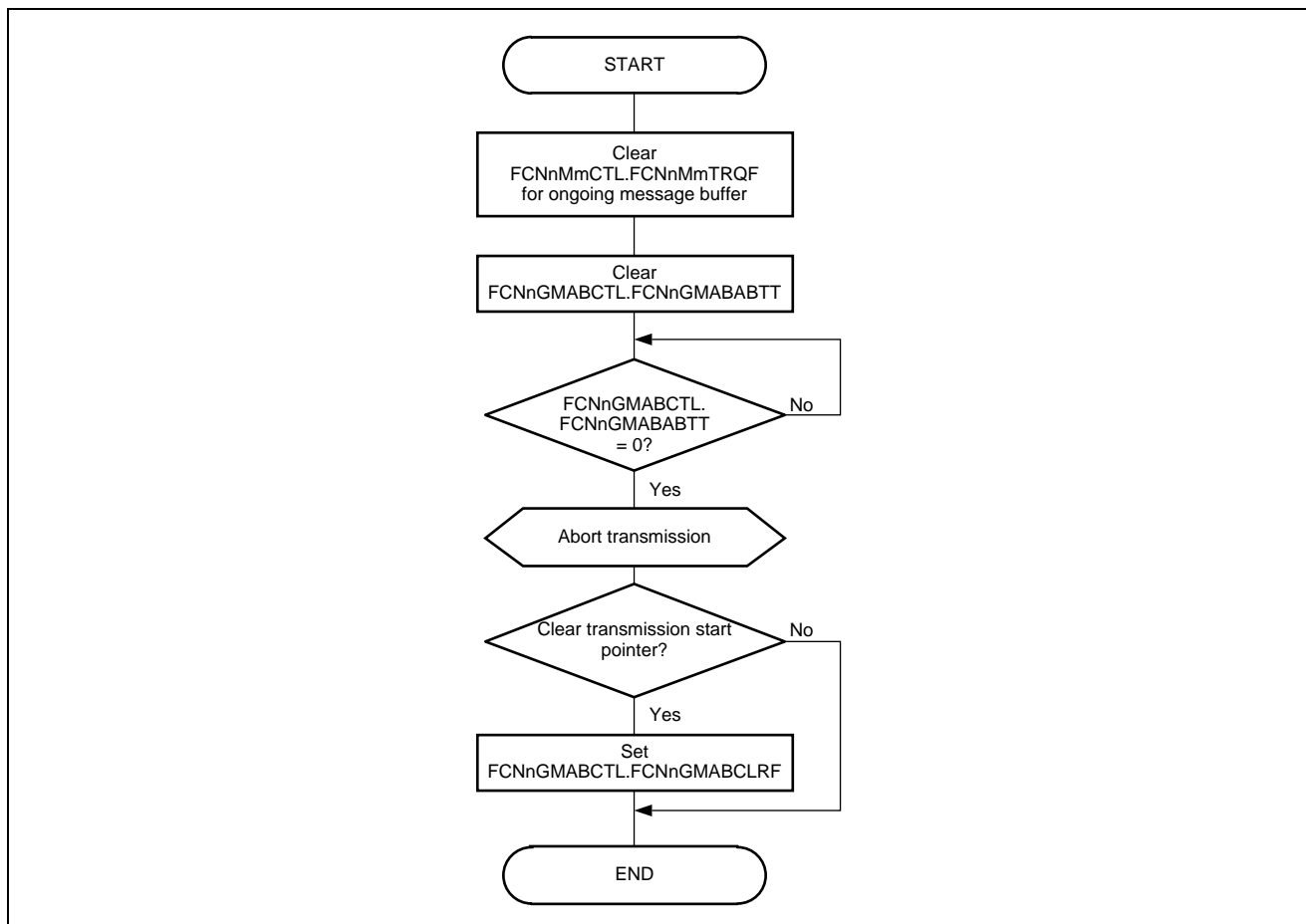


Figure 19.27 ABT Transmission Request Abort Processing (In Normal Operation Mode with ABT) (2)

- 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 19.26, ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1), or Figure 19.27, 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 19.24, Transmission Abort Processing (except when Normal Operation Mode with ABT is being executed).

Figure 19.28 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.

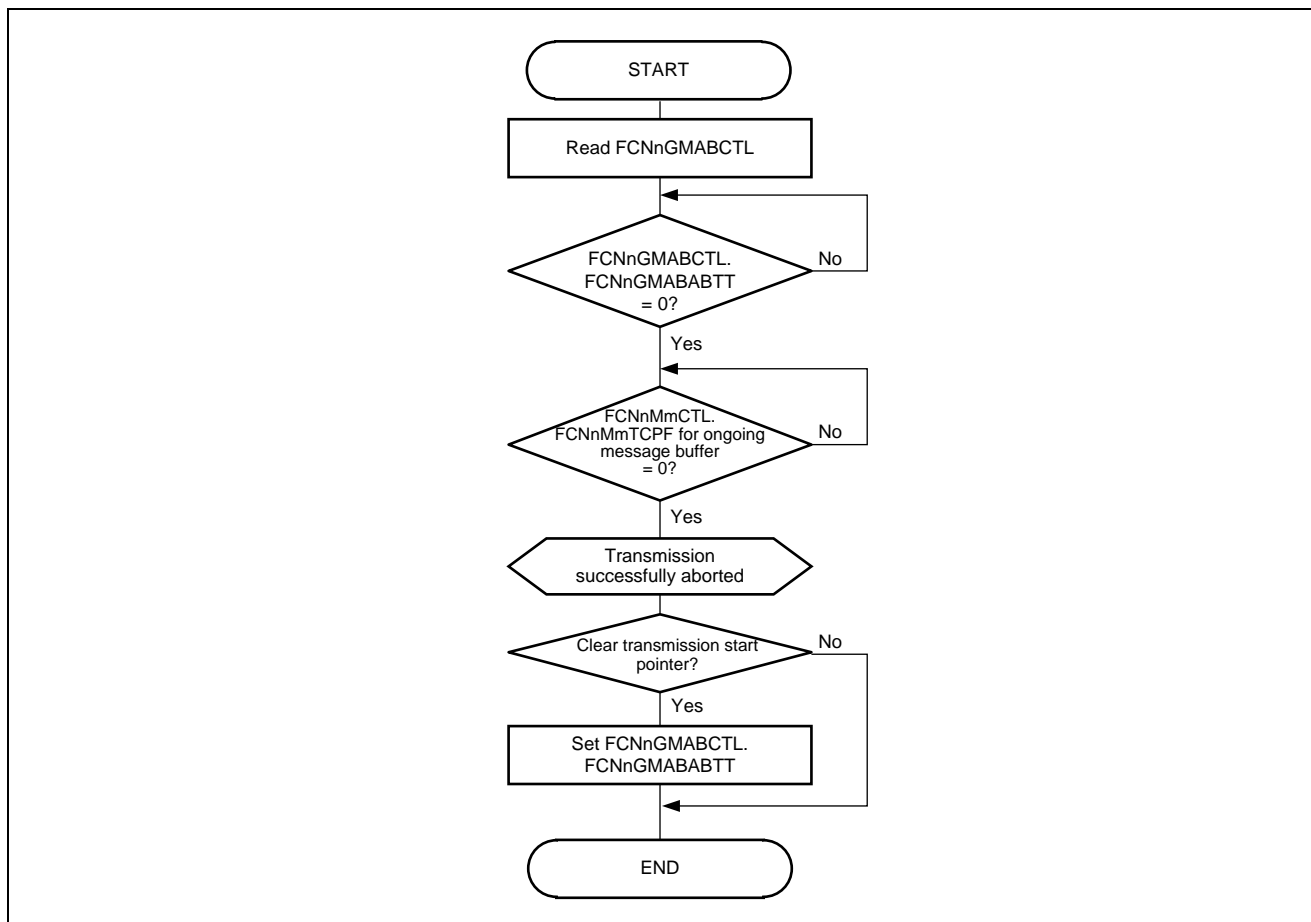


Figure 19.28 ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) with Transmission Complete Flag

- 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 19.26, ABT Transmission Request Abort Processing (in Normal Operation Mode with ABT) (1), or Figure 19.27, 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 19.24, 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 19.29, Transmission Abort Processing with Transmission Abort Interrupt and Transmission Complete Flag, shows the processing when using the Transmit Abort functionality (transmission abort interrupt).

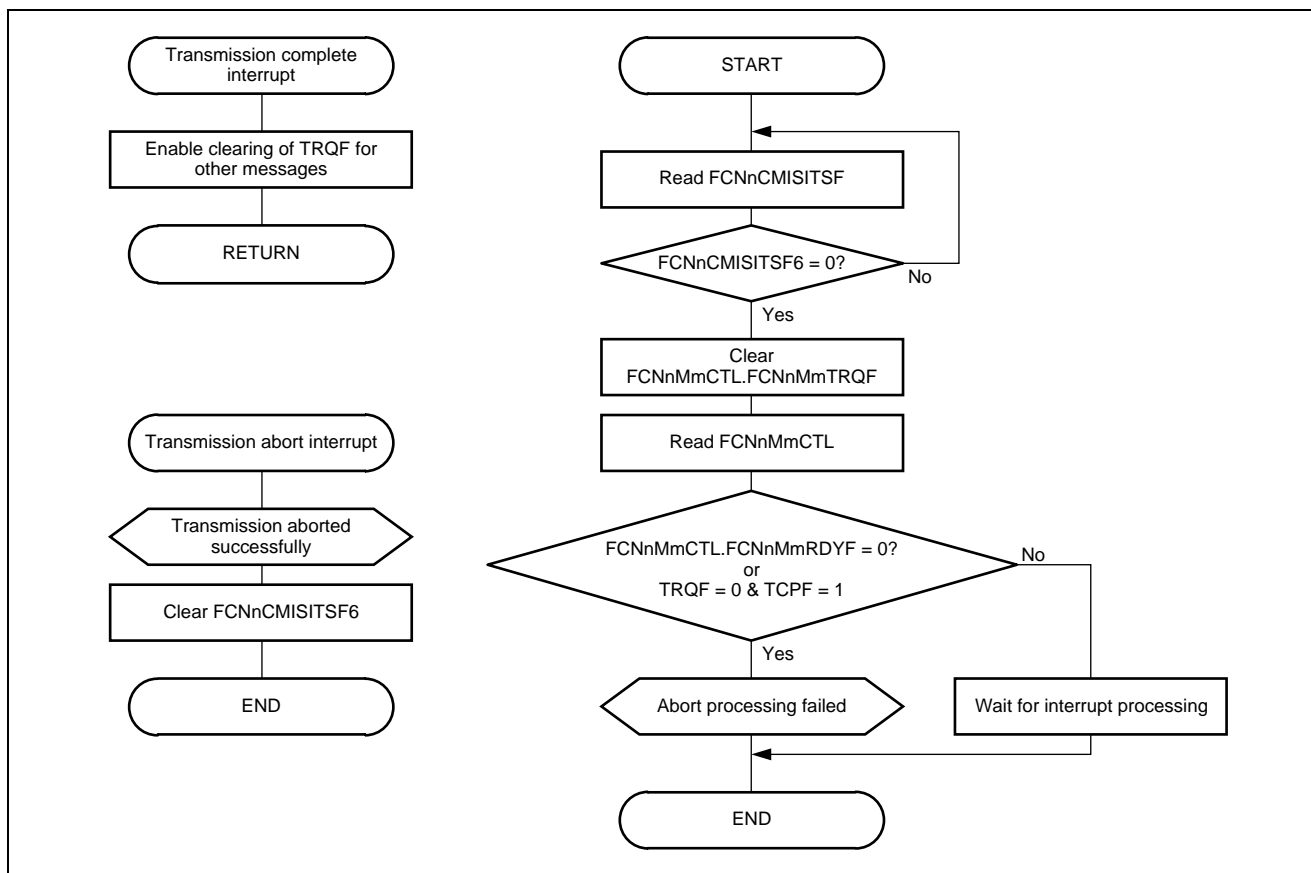


Figure 19.29 Transmission Abort Processing with Transmission Abort Interrupt and Transmission Complete Flag

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 19.30, Transmission Abort Processing with Transmission Complete Flag, shows the processing when using the Transmit Abort functionality (transmission complete flag FCNnMmTCPF).

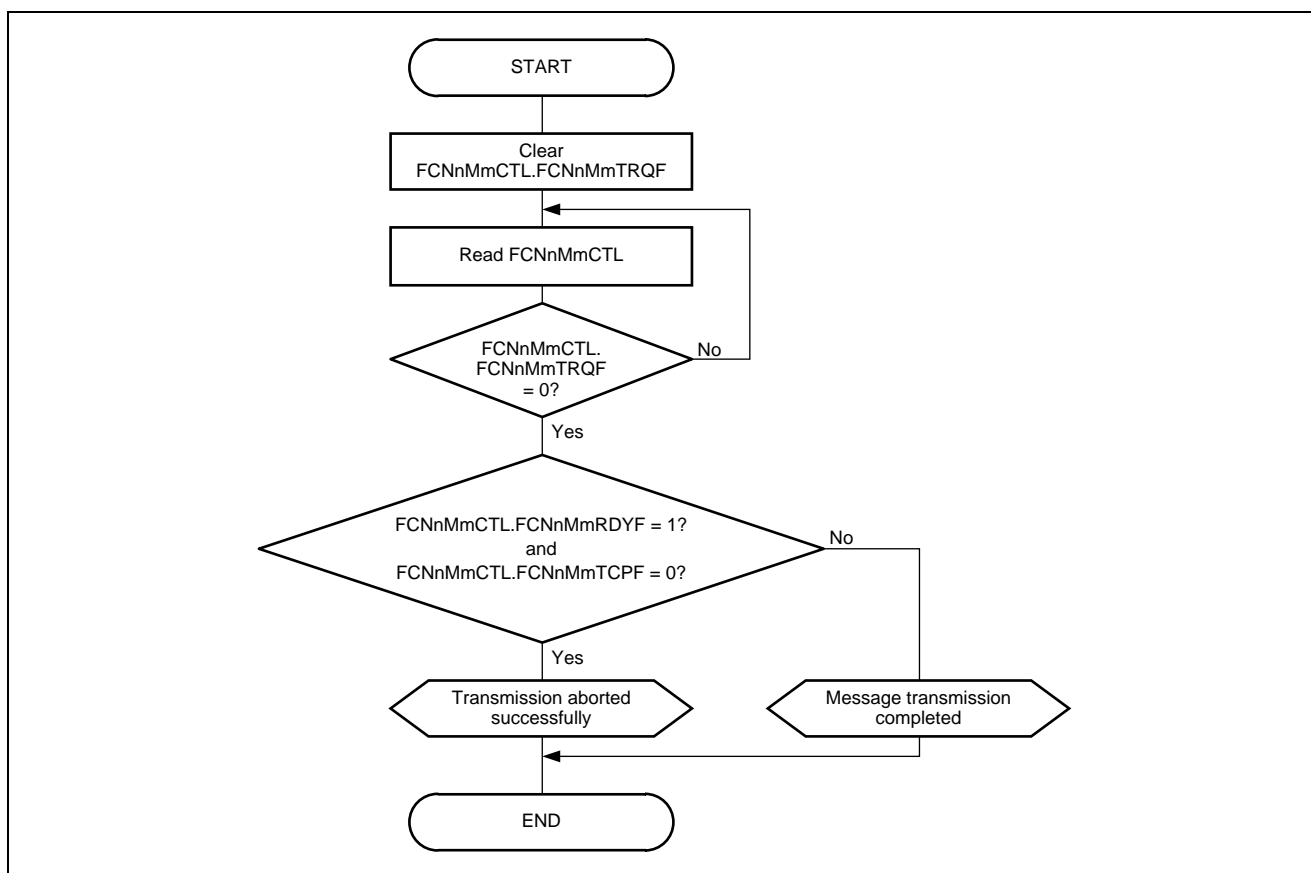


Figure 19.30 Transmission Abort Processing with Transmission Complete Flag

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.

19.14.3 Message Reception

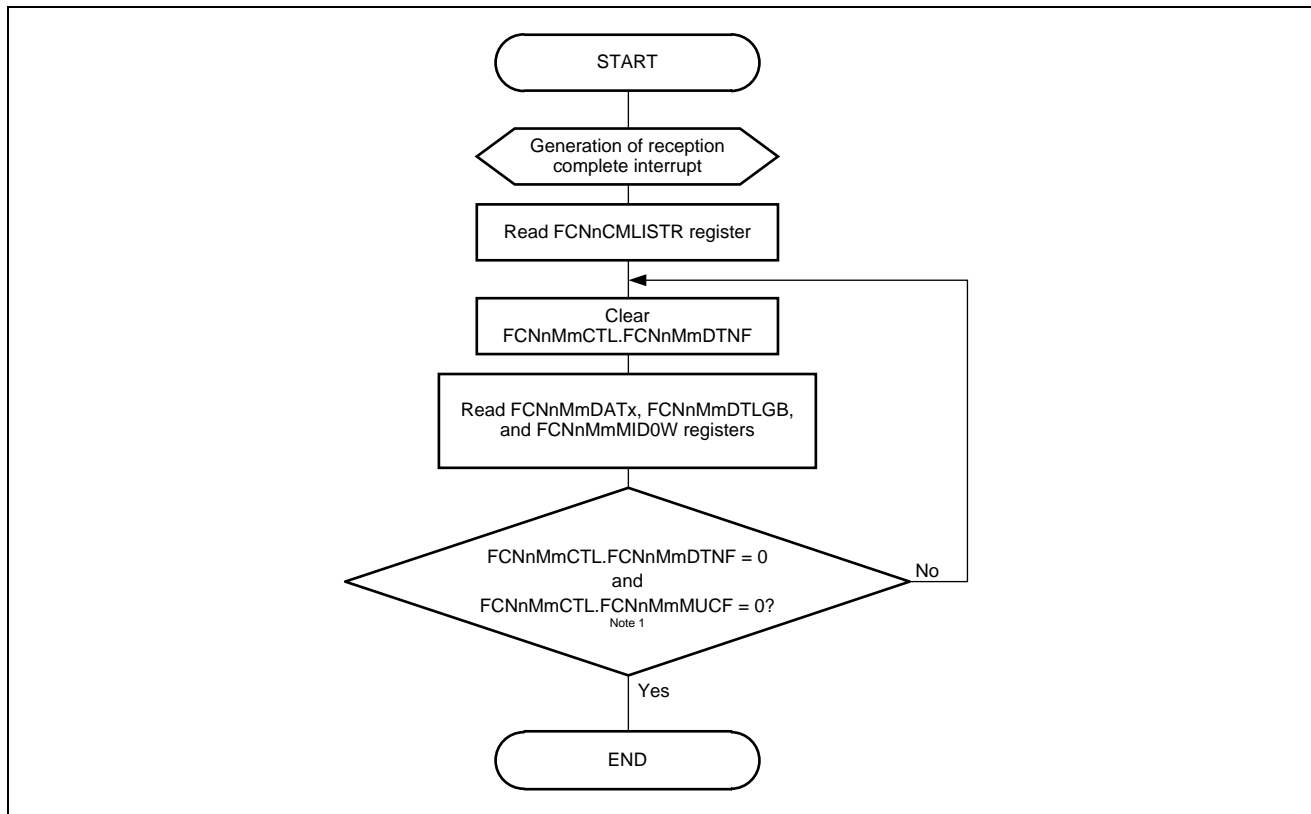


Figure 19.31 Reception via Interrupt (Using FCNnCMLISTR Register)

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.

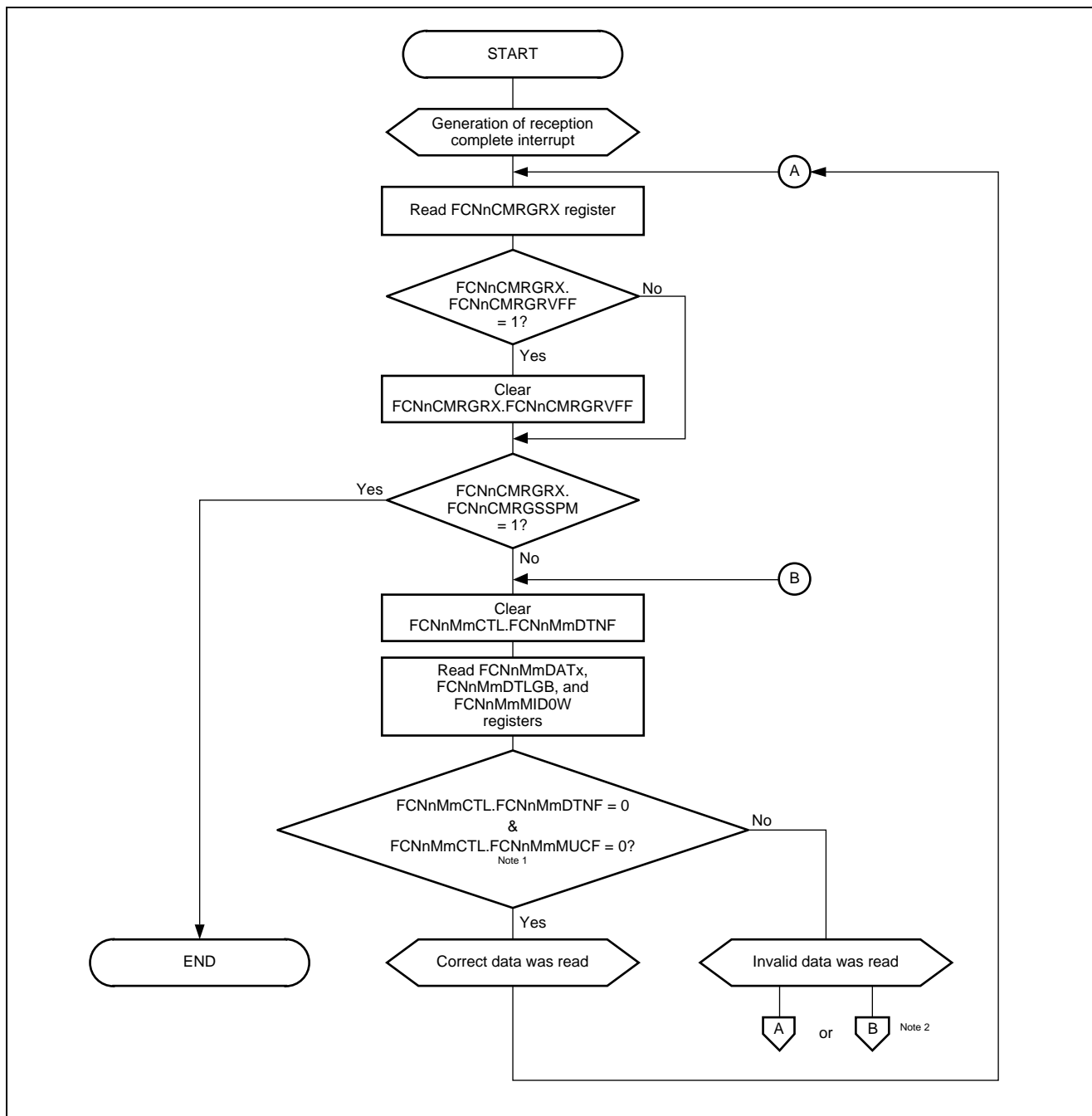


Figure 19.32 Reception via Interrupt (Using FCNnCMRGRX Register)

- 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 19.32, Reception via Interrupt (Using FCNnCMRGRX Register), the processing shown in Figure 19.33, Another Way of Reception via Interrupt (Using FCNnCMRGRX Register), can be used.

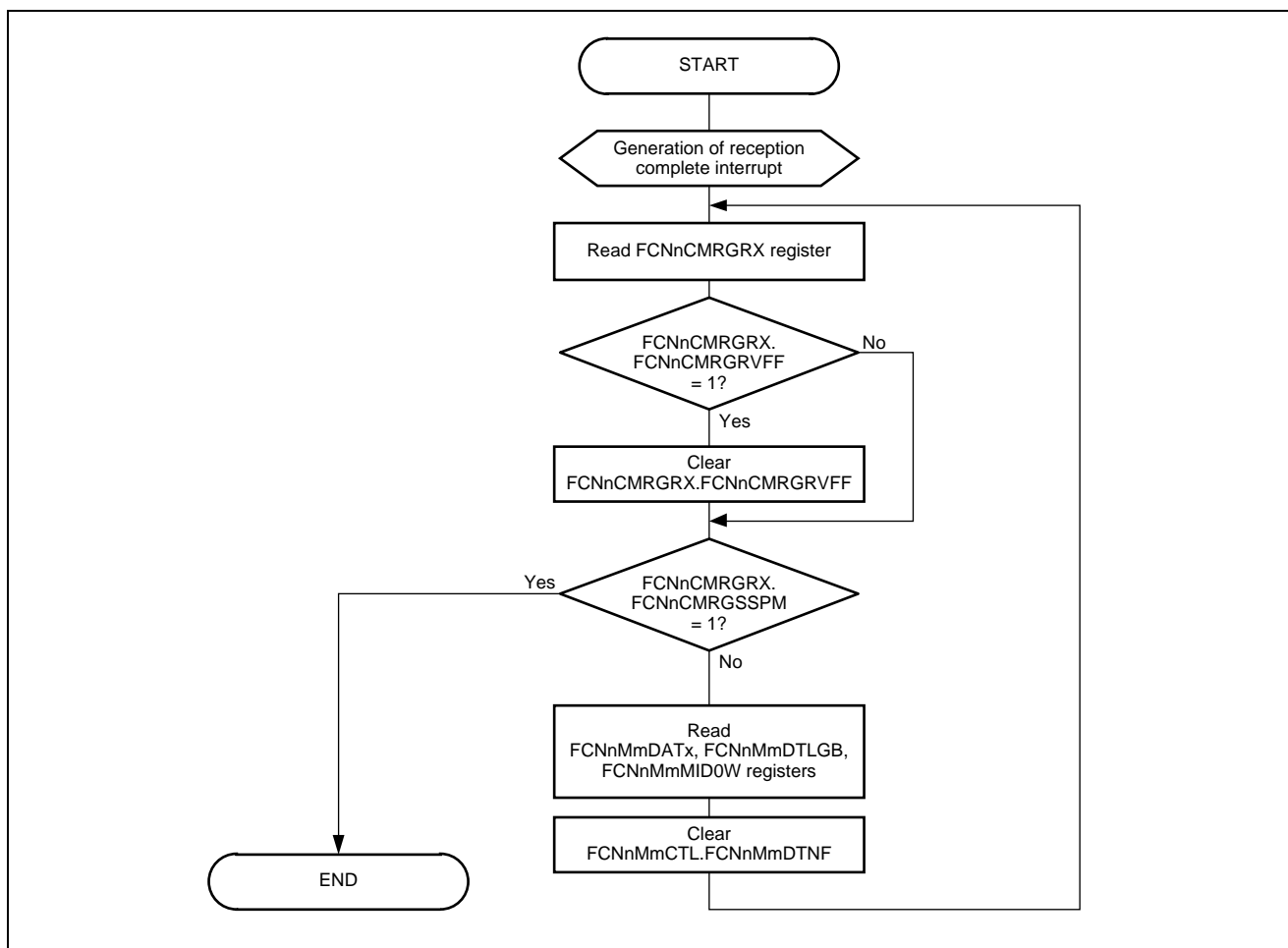


Figure 19.33 Another Way of Reception via Interrupt (Using FCNnCMRGRX Register)

- 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.

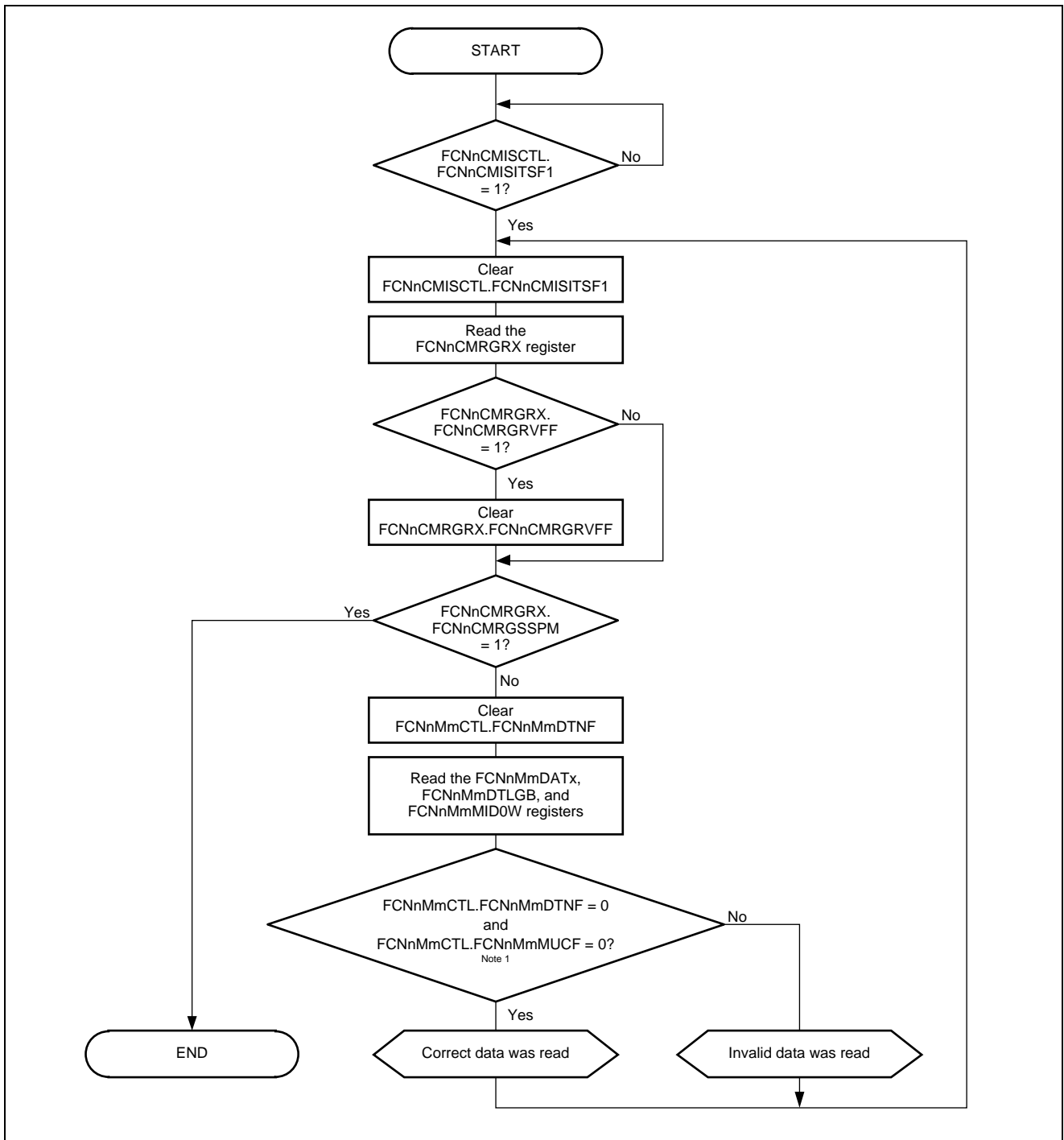


Figure 19.34 Reception via Software Polling

- 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.**

19.14.4 Power Safe Mode

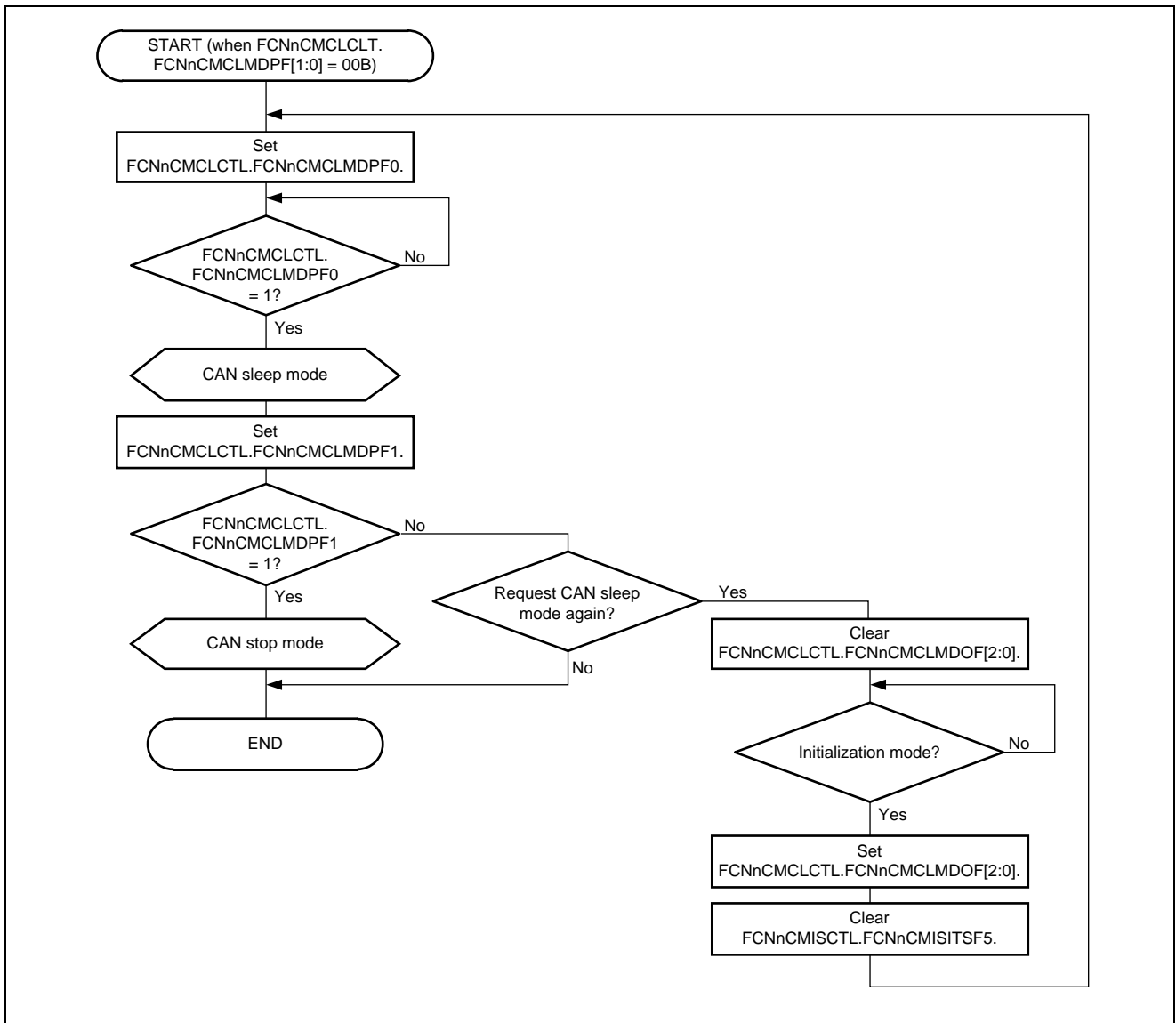


Figure 19.35 Setting FCN Sleep Mode/Stop Mode

Caution: To abort the transmission before issuing a request for transition to FCN sleep mode, perform transmission abort processing according to the previous flowcharts.

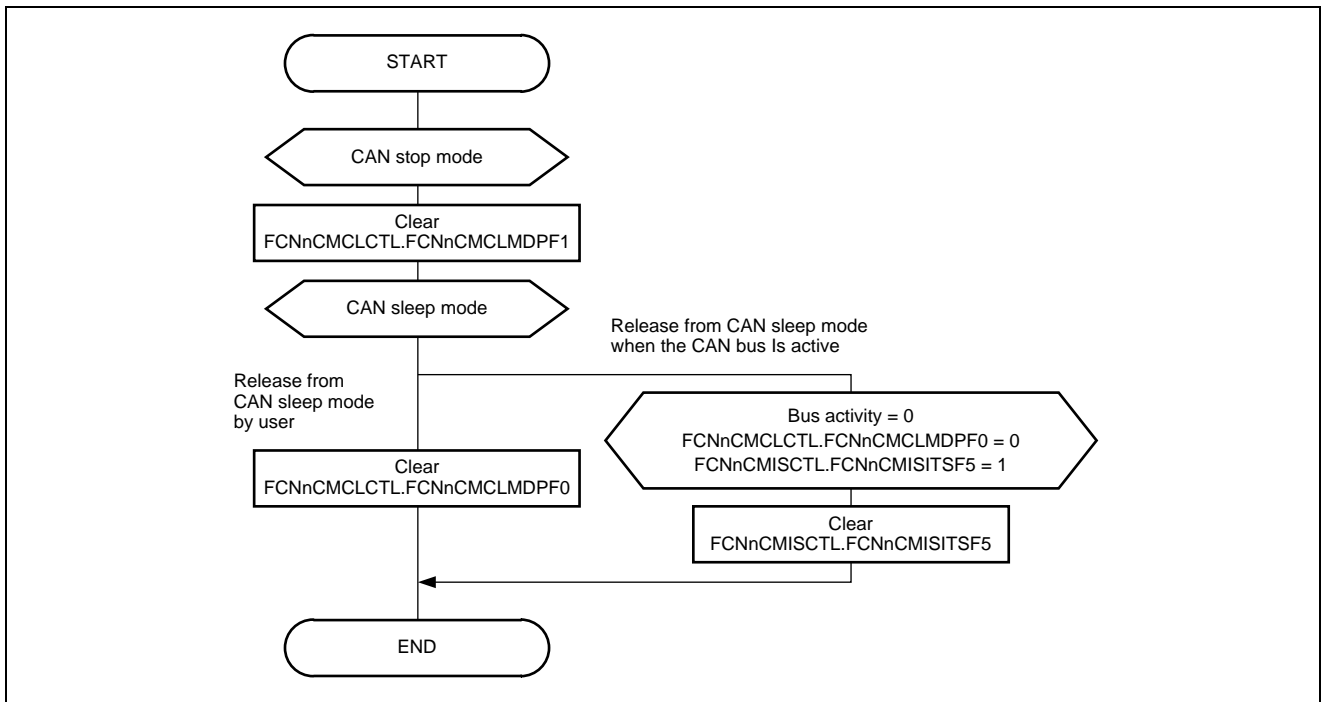


Figure 19.36 Release from FCN Sleep/Stop Mode

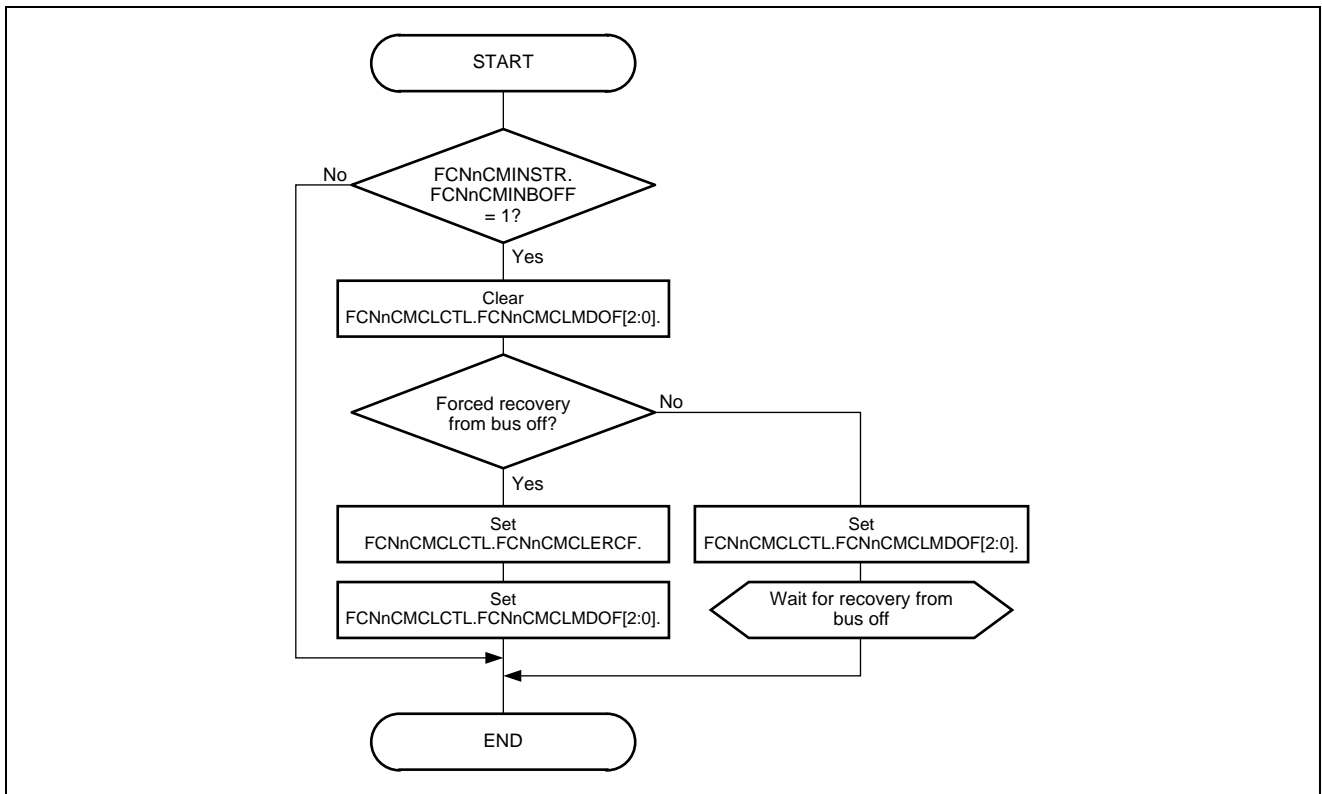


Figure 19.37 Recovery from Bus-Off

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.

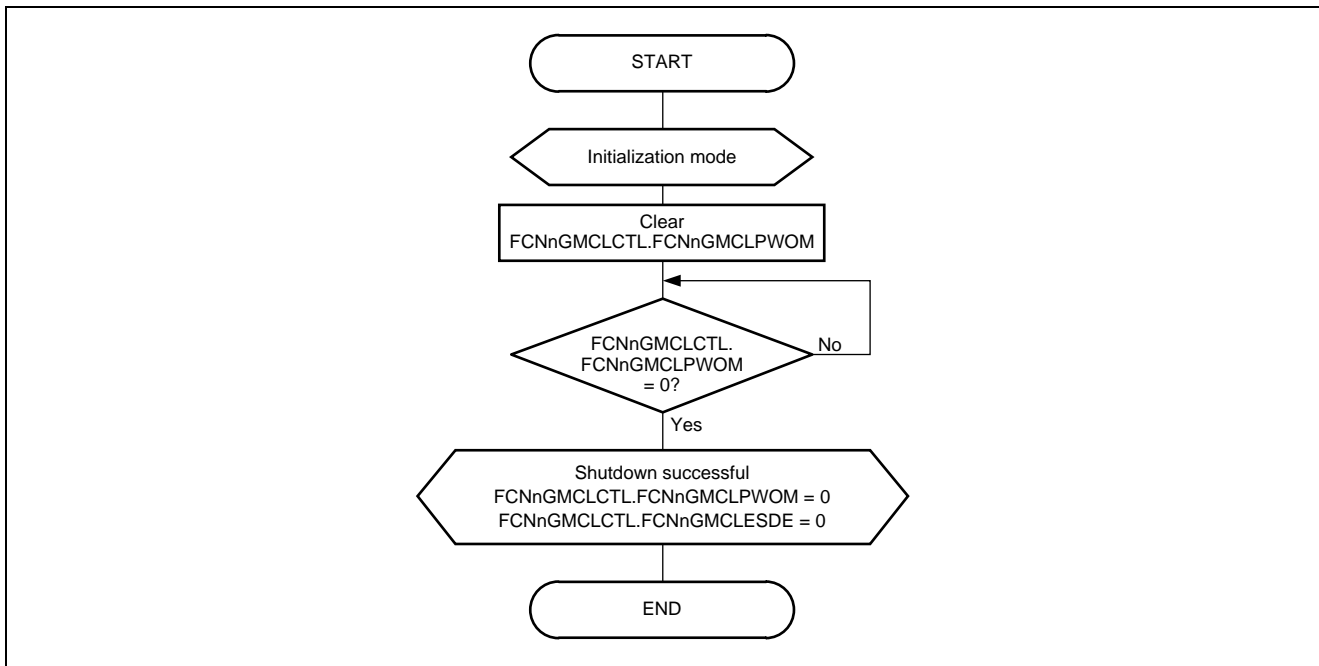


Figure 19.38 Normal Shutdown Processing

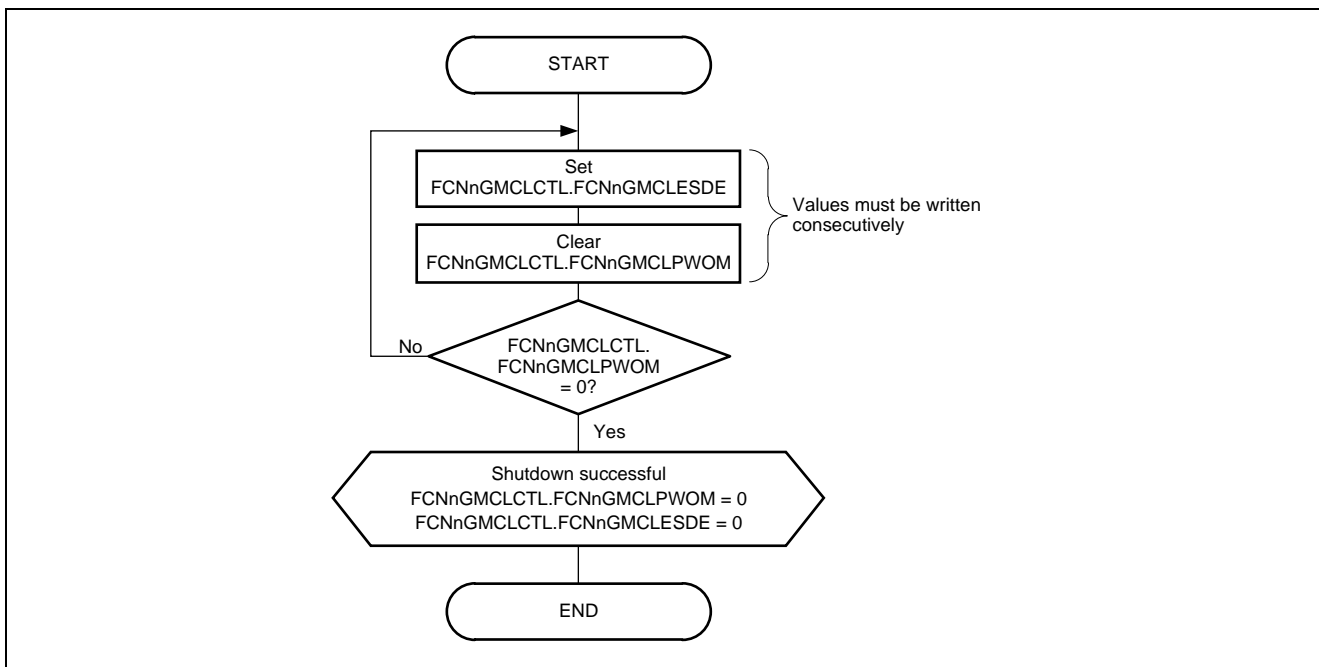


Figure 19.39 Forced Shutdown Processing

Caution: Do not read or write any registers by software between setting of the FCNnGMCLPWOM bit and clearing of the FCNnGMCLCTL bit.

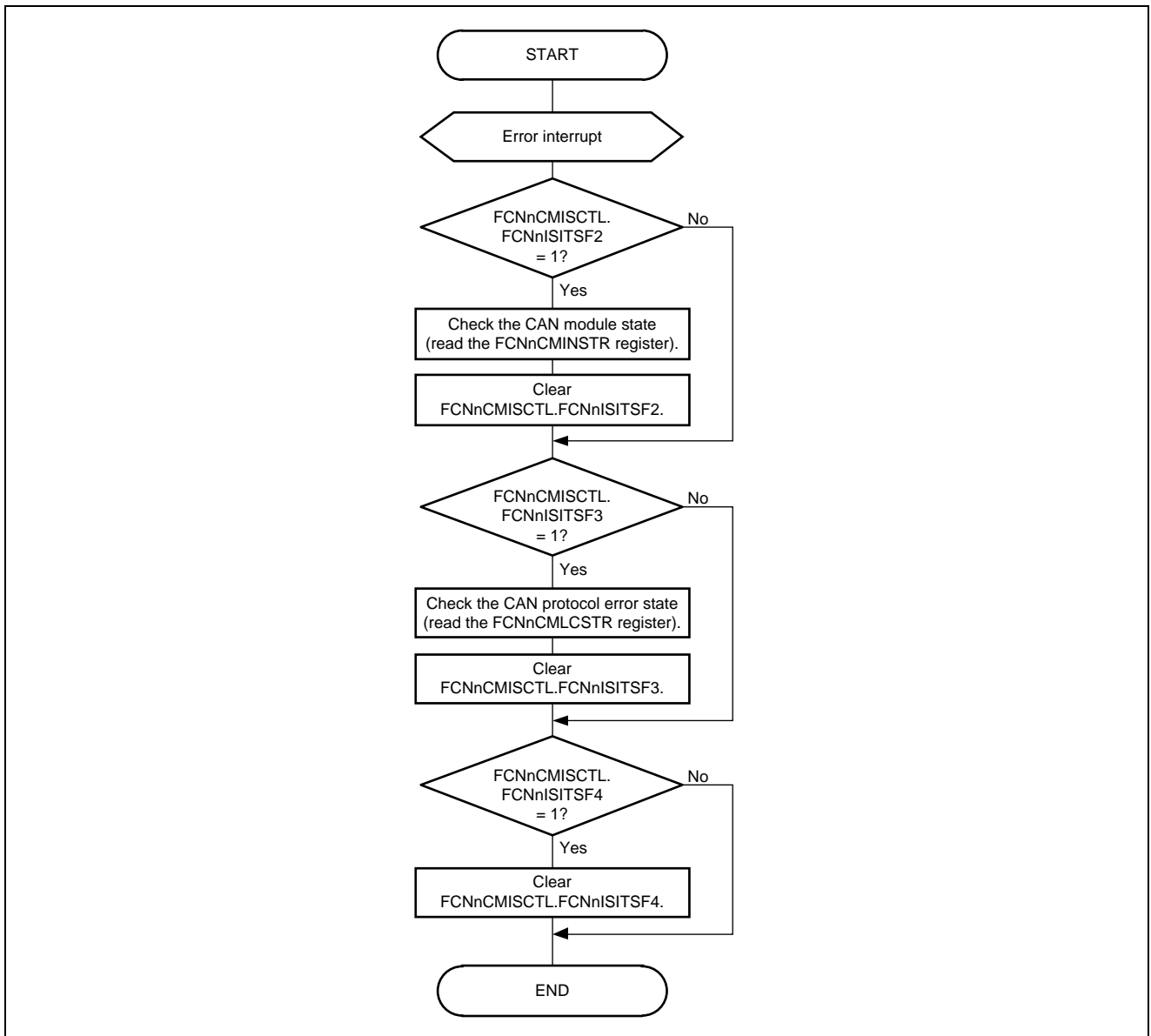


Figure 19.40 Error Handling

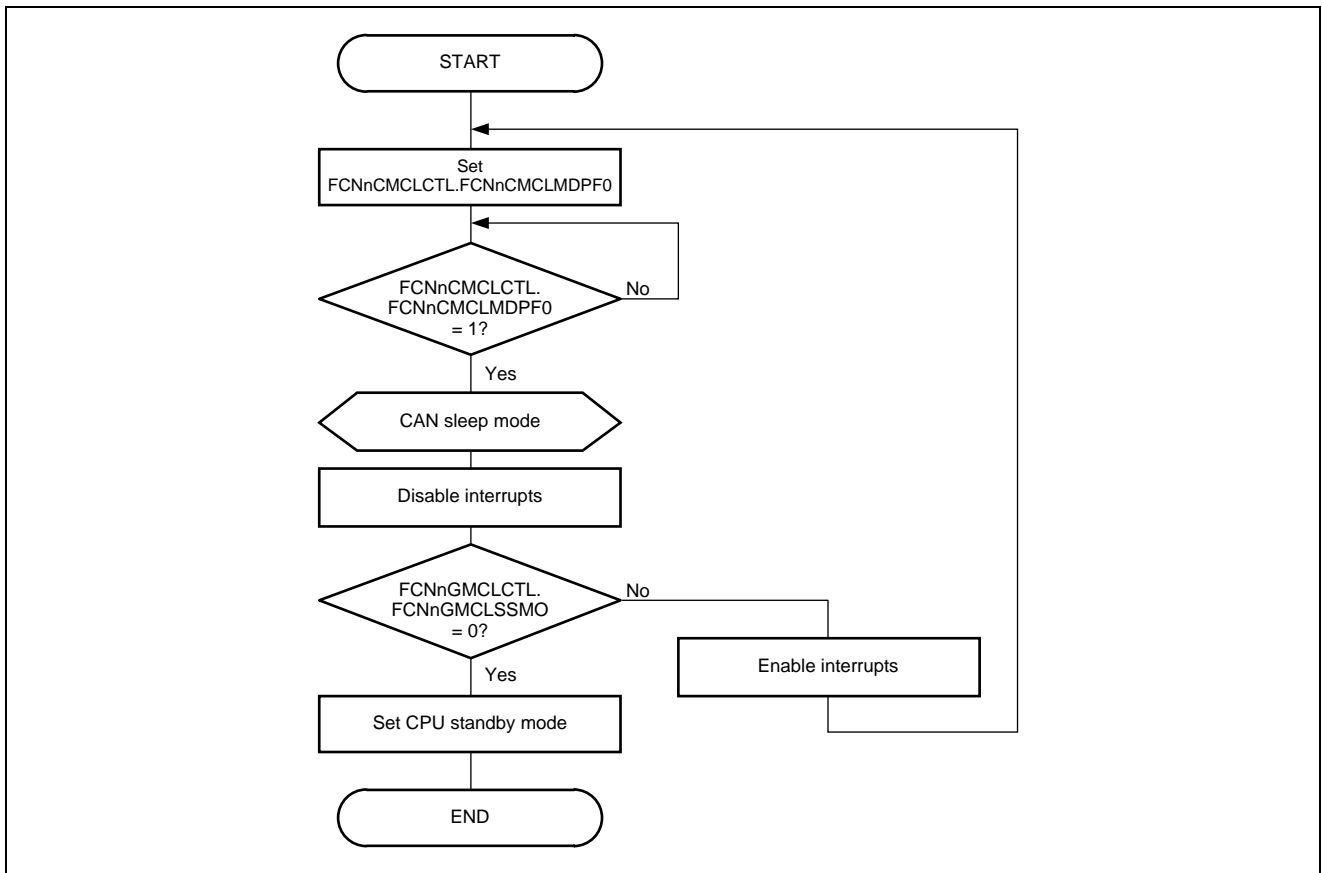


Figure 19.41 Setting CPU Standby (from FCN Sleep Mode)

- 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.

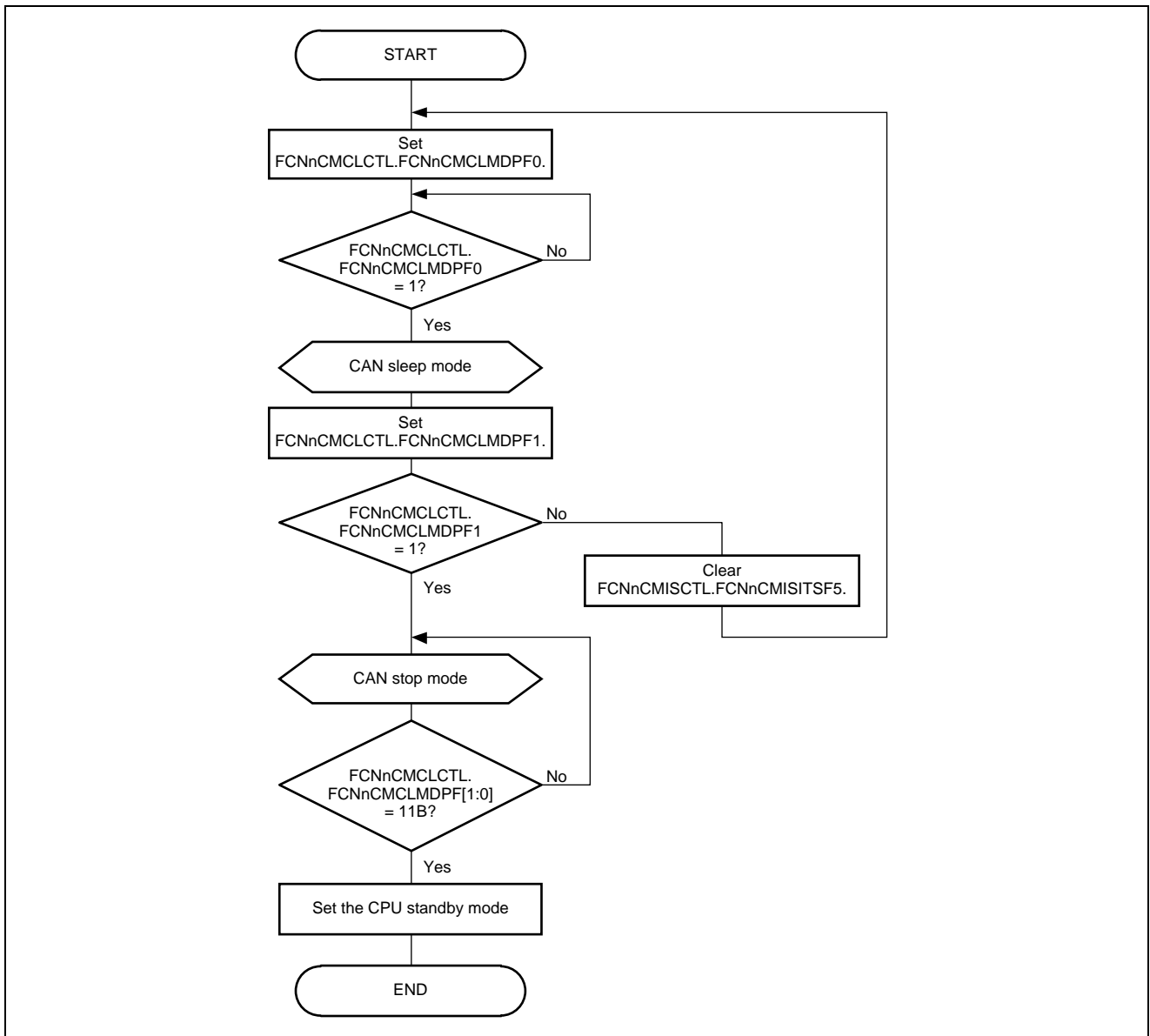


Figure 19.42 Setting CPU Standby (from FCN Stop Mode)

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.

20. CC-Link Interface

The specifications of CC-Link in outline are as follows. For the detailed specifications of CC-Link, see the website of the CC-Link Partner Association at the following URL.

<http://www.cc-link.org/eng/cclink/index.html>

Table 20.1 CC-Link Outline Specifications

Item	Specification
Supported versions	Ver.1.10 / Ver.2.00
Supported stations	Intelligent device station, Remote device station
Maximum number of link points	Remote I/O: 8192 points each, Remote register: 2048 words
Total number of slave stations	64 units
Communication speed and max. overall cable extension length	10 Mbps: 100m 5 Mbps: 160m 2.5 Mbps: 400m 625 kbps: 900m 156 kbps: 1200m
Communication system	Broadcast polling system

20.1 Registers

20.1.1 List of Registers

Register Name	Symbol	Address
CC-Link bus size control register	CCBSC	400A 4404H
CC-Link bus bridge control register 0	CCSMC0	400A 4408H
CC-Link bus bridge control register 1	CCSMC1	400A 440CH
CC-Link monitor register 1	CCSMON	BASE + 080CH
CC-Link RUN LED control register	CCSRUN	BASE + 0810H
CC-Link reset register	CCRES	BASE + 0814H
CC-Link remote device operating mode setting register	CCSMD	BASE + 0818H

20.1.2 CC-Link Bus Size Control Register (CCBSC)

The CCBSC register sets up the data bus width for access to CC-Link (intelligent device station, remote device station). When you are using CC-Link (intelligent device station, remote device station), set 0000 5575H to this register.

- 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						
CCBSC	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	CCBSC15 CCBSC14 CCBSC13 CCBSC12 CCBSC11 CCBSC10 CCBSC9 CCBSC8 CCBSC7 CCBSC6 CCBSC5 CCBSC4 CCBSC3 CCBSC2 CCBSC1 CCBSC0	400A 4404H Initial value 0000 5555H						
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	R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Bit Position</th> <th style="width: 15%;">Bit Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>15 to 0</td> <td>CCBSC15-0</td> <td>Set these bits to 5575H.</td> </tr> </tbody> </table>					Bit Position	Bit Name	Description	15 to 0	CCBSC15-0	Set these bits to 5575H.
Bit Position	Bit Name	Description								
15 to 0	CCBSC15-0	Set these bits to 5575H.								

20.1.3 CC-Link Bus Bridge Control Register 0 (CCSMC0)

The CCSMC0 register controls access to CC-Link (intelligent device station). When you are using CC-Link (intelligent device station), be sure to set 0000 11B1H to this register.

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

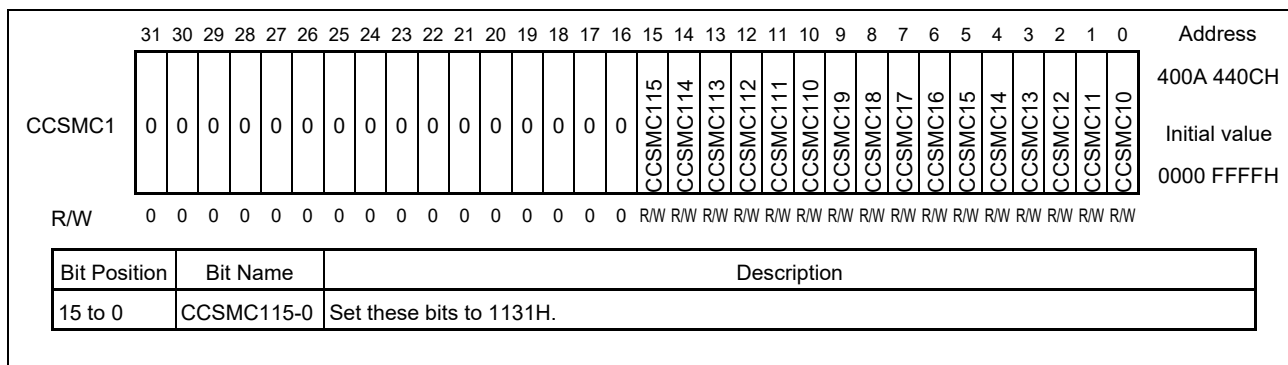
	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16		15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Address						
CCSMC0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	CCSMC15 CCSMC14 CCSMC13 CCSMC12 CCSMC11 CCSMC10 CCSMC09 CCSMC08 CCSMC07 CCSMC06 CCSMC05 CCSMC04 CCSMC03 CCSMC02 CCSMC01 CCSMC00	400A 4408H Initial value 0000 FFFFH						
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	R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Bit Position</th> <th style="width: 15%;">Bit Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>15 to 0</td> <td>CCSMC015-0</td> <td>Set these bits to 11B1H.</td> </tr> </tbody> </table>					Bit Position	Bit Name	Description	15 to 0	CCSMC015-0	Set these bits to 11B1H.
Bit Position	Bit Name	Description								
15 to 0	CCSMC015-0	Set these bits to 11B1H.								

20.1.4 CC-Link Bus Bridge Control Register 1 (CCSMC1)

The CCSMC1 register controls access to CC-Link (remote device station).

When you are using CC-Link (remote device station), be sure to set 0000 1131H to this register.

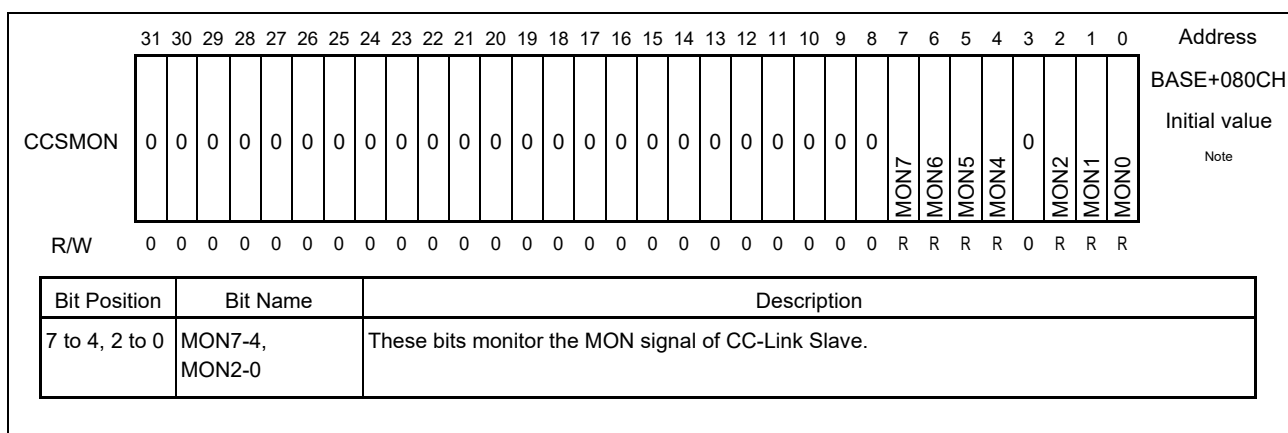
- Access This register can be read or written in 32-bit units.



20.1.5 CC-Link Monitor Register (CCSMON)

The CCSMON register is used to monitor the MON signal of CC-Link Slave (remote device station).

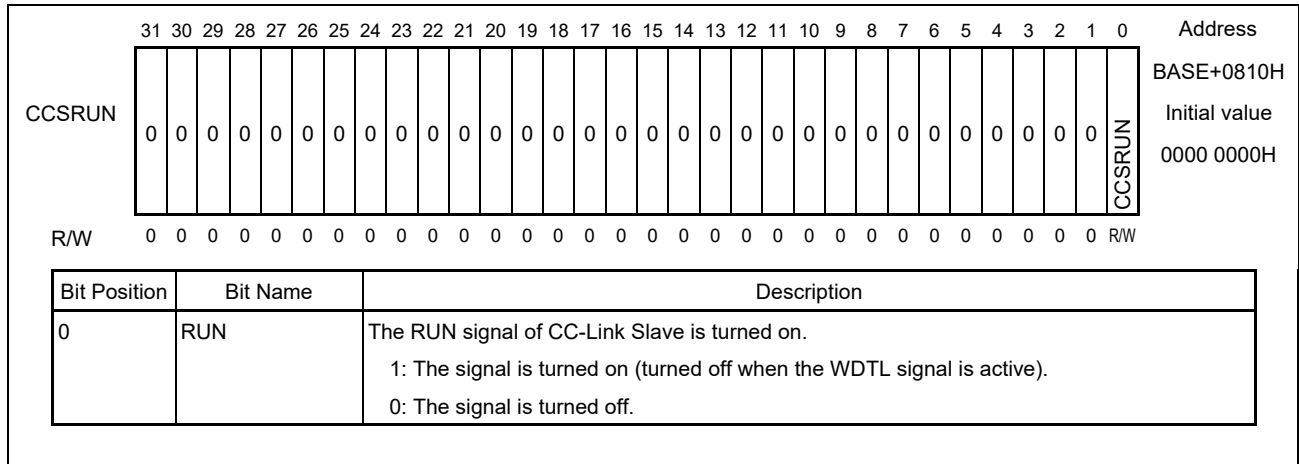
This register is only readable in 32- or 16-bit units.



Note: The initial value depends on the MON signal.

20.1.6 CC-Link Slave RUN LED Control Register (CCSRUN)

The CCSRUN register generates the RUN (P26) signal for the CC-Link slave (remote device station). This register can be read or written in 32- or 16-bit units.



The figure below shows the circuit configuration of the CCSRUN register, CC-Link (intelligent device station and remote device station) RUN signals, and port pins.

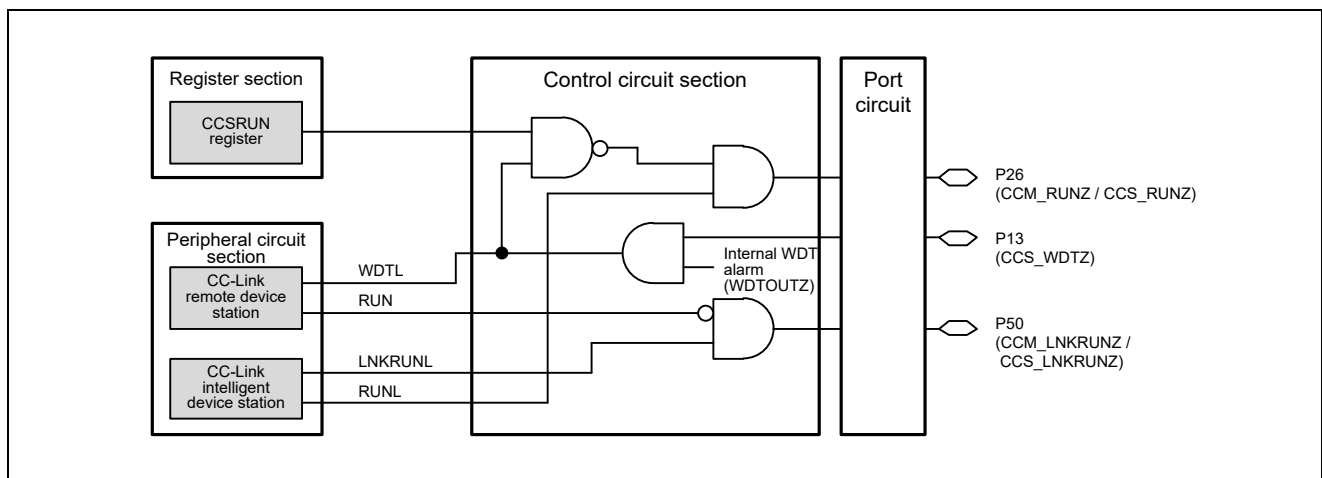


Figure 20.1 Configuration of the CCSRUN Register and CC-Link (Intelligent Device Station and Remote Device Station) RUN Signals

20.1.7 CC-Link Reset Register (CCRES)

The CCRES register is used to generate a reset signal for CC-Link (intelligent device station, remote device station). The initial value starts CC-Link in the reset state and the reset signal is de-asserted by using this register after setting the operating mode for the port function

- 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 how to release protection, see the description of the system protection 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																															
CCRES	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> </table>		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BASE+0814H Initial value 0000 0000H
	0	0	0	0	0	0	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 border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> </table>		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CCMRES CCSRES 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			

Bit Position	Bit Name	Description
1	CCMRES	Releases CC-Link (intelligent device station) from the reset state. 1: Release from the reset state 0: Reset state (initial value)
0	CCSRES	Releases CC-Link (remote device station) from the reset state. 1: Release from the reset state 0: Reset state (initial value)

20.1.8 CC-Link Slave Operating Mode Setting Register (CCSMD)

The CCSMD register sets the operating mode of CC-Link (remote device station).

Bits 14 to 8 are used to select whether the external pin settings or the settings in this register are effective.

- 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	
CCSMD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IOTENSUSEL	SENYU1SEL	SENYU0SEL	BS4SEL	BS3SEL	BS2SEL	BS1SEL	0	IOTENNSU	SENYU1	SENYU0	BS4	BS3	BS2	BS1	BASE+0818H
																		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/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	

Bit Position	Bit Name	Description
14 to 8	xxSEL	These bits select whether the external pin settings or the settings in this register are effective. 1: The settings in this register are effective. 0: The external pin settings are effective.
6	IOTENSU	Sets a value for the IOTENSU pin.
5	SENYU1	Sets a value for the SENYU1 pin.
4	SENYU0	Sets a value for the SENYU0 pin.
3	BS4	Sets a value for the BS4 pin.
2	BS3	Sets a value for the BS3 pin.
1	BS2	Sets a value for the BS2 pin.
0	BS1	Sets a value for the BS1 pin.

21. System Registers (APB Peripheral Registers Area)

The system registers of the R-IN32M3 series are listed below. These registers can be accessed via the external MCU interface. Some registers cannot be accessed if a 16-bit bus is used for the external MCU interface. For details, see section 21.1, 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

21.1 List of Registers

Register Name	Symbol	Address	Protection	Unit of Access		Access by external MCU
				16	32	
Operating mode monitor register	MDMNT	BASE+0000H	—	✓	✓	✓
IDCODE register	IDCODE	BASE+0004H	—	X	✓	✓
Version register	RINVER	BASE+0008H	—	✓	✓	✓
Watchdog timer input clock selection register	WDTCLKCFG	BASE+0180H	✓	X	✓	X
CPURESET register	CPURESET	BASE+0210H	✓	✓	✓	✓
System protect command register	SYSPCMD	BASE+0300H	—	✓	✓	✓
HW-RTOS reset register	RTOS_SOFTRST	BASE+0400H	✓	X	✓	✓
Timer input function selection register	SELCNT	BASE+0500H	✓	X	✓	X
Timer trigger source register 0	TMTFR0	BASE+0530H	✓	X	✓	X
Timer trigger source register 1	TMTFR1	BASE+0534H	✓	X	✓	X
Timer trigger source register 2	TMTFR2	BASE+0538H	✓	X	✓	X
Timer trigger source register 3	TMTFR3	BASE+053CH	✓	X	✓	X
Noise filter setting register 0	NFC0	BASE+0700H	✓	X	✓	X
Noise filter setting register 1	NFC1	BASE+0704H	✓	X	✓	X
Noise filter setting register 2	NFC2	BASE+0708H	✓	X	✓	X
Noise filter setting register 3	NFC3	BASE+070CH	✓	X	✓	X
External interrupt mode register 0	INTM0	BASE+0710H	✓	X	✓	X
External interrupt mode register 1	INTM1	BASE+0714H	✓	X	✓	X
External interrupt mode register 2	INTM2	BASE+0718H	✓	X	✓	X

Register Name	Symbol	Address	Protection	Unit of Access		Access by external MCU
				16	32	
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	—	✓	✓	✓
Scratch register D	SCRATCHD	BASE+0934H	—	✓	✓	✓
Trigger-synchronous port control mode register	RPTRGMD	BASE+0A00H	✓	X	✓	X
Trigger-synchronous port source register 0	RP0TFR	BASE+0A30H	✓	X	✓	X
Trigger-synchronous port source register 1	RP1TFR	BASE+0A34H	✓	X	✓	X
Trigger-synchronous port source register 2	RP2TFR	BASE+0A38H	✓	X	✓	X
Trigger-synchronous port source register 3	RP3TFR	BASE+0A3CH	✓	X	✓	X
CPU bus operating mode register	CPUBUSMD	BASE+0214H	—	✓	✓	✓
SRAM bridge select register	SRAMBRSEL	BASE+0804H	✓	✓	✓	✓

21.2 Operating Mode Monitor Register (MDMNT)

This register is used to monitor the level of the operating mode setting pins. Identifying the R-IN32M3-EC and the R-IN32M3-CL is also possible.

- Access 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			
MDMNT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FRU	FRU	FRU	MODE	FRU	MEMCSEL	0	BOOT1	BOOT0	OSCTH	JTAGSEL	HWRZSEL	HIFSYNC	ADMUXMODE	MEMIFSEL	BUS32EN	BASE+0000H			
																	R	R	R	R	R	R	0	R	R	R	R	R	R	R	R	R	R	R	Initial value	
R/W																	R	R	R	R	R	R	0	R	R	R	R	R	R	R	R	R	R	R	R	Pin setting

Bit Position	Bit Name	Description
15 to 13, 11	FRU	Reserved. When read, X is returned.
12	MODE	In the R-IN32M3-CL, this bit is read as "1". In the R-IN32M3-EC, it is read as "0".
10, 8 to 0	MEMCSEL, BOOT1, BOOT0, OSCTH, JTAGSEL, HWRZSEL, HIFSYNC, ADMUXMODE, MEMIFSEL, BUS32EN	Use this field to read the level of the operating mode setting pins. The values change dynamically. Changing these values during operation is prohibited.
30 to 16, 9	—	Reserved. When read, 0 is returned.

21.3 IDCODE Register (IDCODE)

This register is used to identify an R-IN32M3. When this register is read, RIN1 is returned in ASCII code.

- 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																		
IDCODE	IDCODE31	IDCODE30	IDCODE29	IDCODE28	IDCODE27	IDCODE26	IDCODE25	IDCODE24	IDCODE23	IDCODE22	IDCODE21	IDCODE20	IDCODE19	IDCODE18	IDCODE17	IDCODE16	IDCODE15	IDCODE14	IDCODE13	IDCODE12	IDCODE11	IDCODE10	IDCODE9	IDCODE8	IDCODE7	IDCODE6	IDCODE5	IDCODE4	IDCODE3	IDCODE2	IDCODE1	IDCODE0	BASE + 0004H																		
																	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	Initial value		
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	R	R	5249 4E31H

Bit Position	Bit Name	Description
31 to 0	IDCODE31-0	This register is used to identify an R-IN32M3. When this register is read, RIN1 is returned in ASCII code.

21.4 Version Register (RINVER)

This register is used to identify the version number of an R-IN32M3. If read, 0000 0002H is returned.

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

RINVER	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+0008H Initial value 0000 0002H
	RINVER31	RINVER30	RINVER29	RINVER28	RINVER27	RINVER26	RINVER25	RINVER24	RINVER23	RINVER22	RINVER21	RINVER20	RINVER19	RINVER18	RINVER17	RINVER16	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	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit Position	Bit Name		Description																														
31 to 0	RINVER31-0		These bits are for identifying the version number of an R-IN32M3. When read, 0000 0002H is returned.																														

Remark: In the old products, the value read from this register is 0000 0001H.
 For details of the old products, see section 1.1, Type Names of R-IN32M3-Series Products.

21.5 Watchdog Timer Input Clock Selection Register (WDTCLKCFG)

This register is used to select the division factor of the timer count clock input to the watchdog timer.

- 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 how to release protection, see the description of the system protection 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	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 100%;">0 0</td> </tr> </table>	0 0	BASE + 0180H
0 0			
	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 100%;">WDTCNF3 WDTCNF2 WDTCNF1 WDTCNF0</td> </tr> </table>	WDTCNF3 WDTCNF2 WDTCNF1 WDTCNF0	Initial value 0000 0000H
WDTCNF3 WDTCNF2 WDTCNF1 WDTCNF0			
R/W	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 100%;">0 R/W R/W R/W R/W</td> </tr> </table>	0 R/W R/W R/W R/W	
0 R/W R/W R/W R/W			

Bit Position	Bit Name	Description
31 to 4	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
3 to 0	WDTCNF3 to WDTCNF0	Specify 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

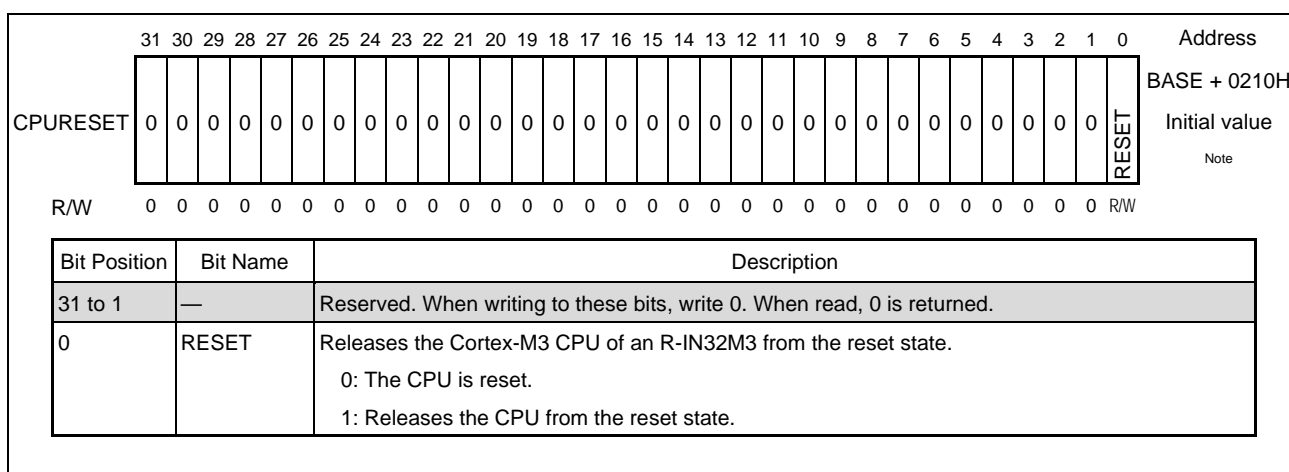
21.6 CPURESET Register (CPURESET)

This register is used to release the Cortex-M3 CPU of an R-IN32M3 from the reset state by the host CPU when the host is booted.

The initial value after release from the reset state depends on the setting of the BOOT1 and BOOT0 pins. This register can be used to release the CPU from the reset state, but it cannot reset the CPU again.

- Access This register can be read or written in 32- or 16-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 how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.



Note: The initial value changes with the state of the BOOT1 and BOOT0 pins.

BOOT1, BOOT0	Initial value
00	1
01	1
10	0
11	1

21.7 System Protect Command Register (SYSPCMD)

The SYSPCMD register is a 32-bit register for use in protecting against inadvertent access to registers to which writing raises the possibility of serious effects on application systems, such as programs crashing and the like. The SYSPCMD register can be read or written in 32- or 16-bit units.

Write-protected registers cannot be written unless the PROT bit is set (1).

When the PROT bit is set (1), writing to write-protected registers is only allowed in the sequence described below. No special sequence is required for clearing or reading these registers.

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

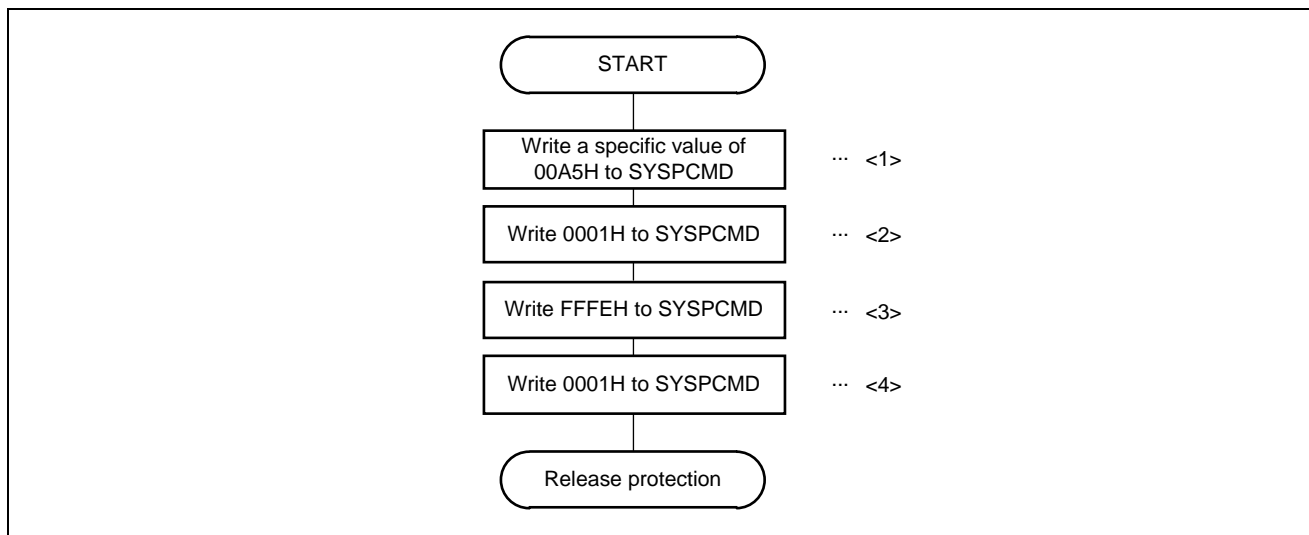
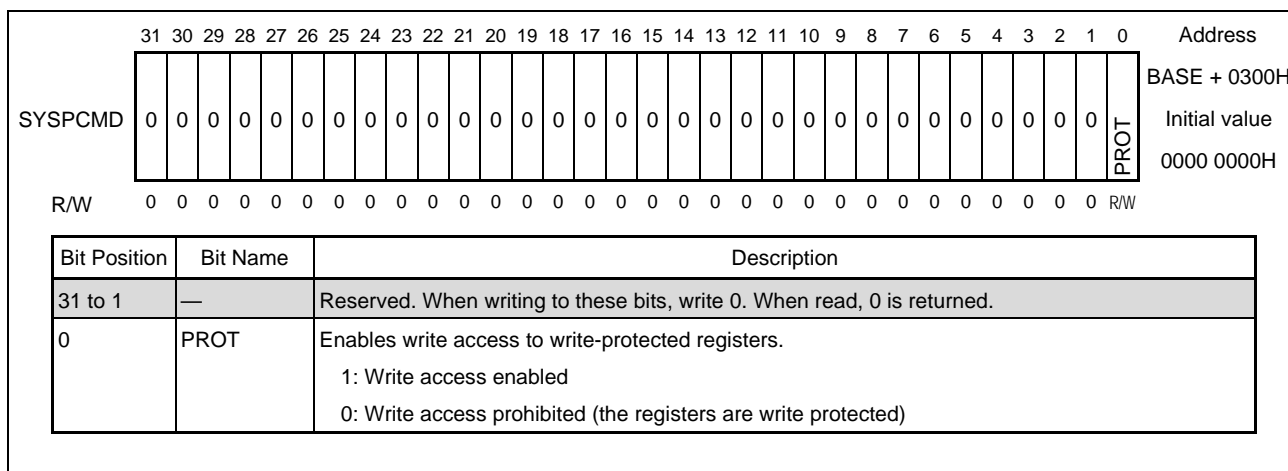


Figure 21.1 Protection Release Sequence

Cautions 1. A value is not written to the register in steps <1>, <2> and <3>.

2. Be sure to clear this bit to 0 (0: setting for protection) after the completion of writing to an applicable register.

21.9 Timer Input Selection

In an R-IN32M3, the signal to be input to the on-chip 32-bit timer array unit (TAUJ2) can be selected from the TSOUT signal from the CAN or an interrupt signal from a peripheral module. The signal input to the timer input pins (TIN0 to TIN3) is selected by using the SELCNT and TMTFR3 to TMTFR0 registers.

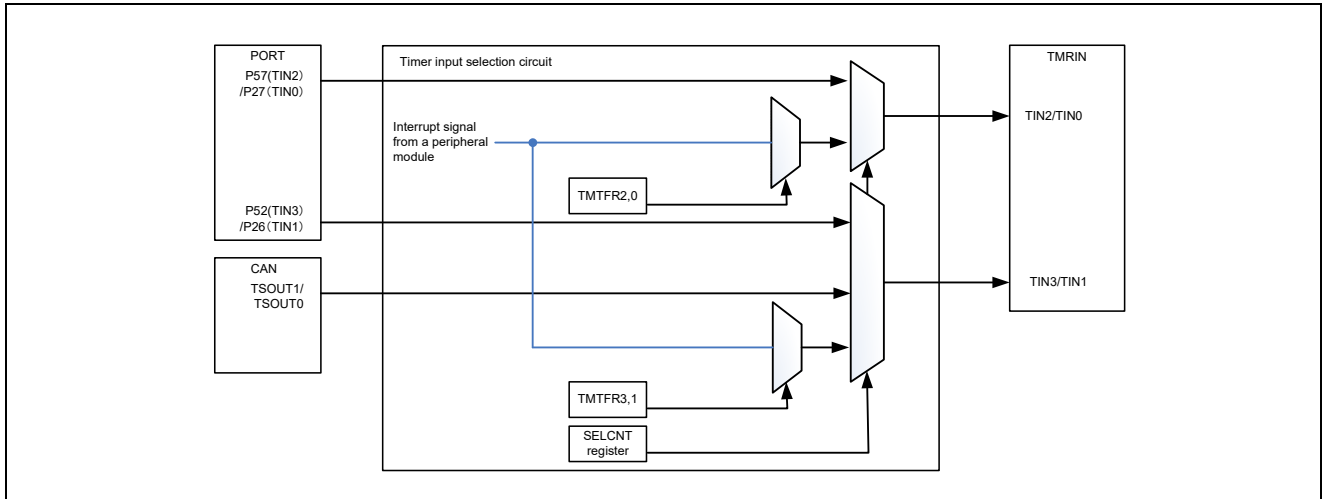


Figure 21.2 Configuration of Timer Input Selection

21.9.1 Timer Input Function Selection Register (SELCNT)

This register is used to select the mode for capture trigger input to TAUJ2 (4 channels). Timestamping is supported by the input of the trigger output signal (TSOUT) of the CAN macro to TINx. Furthermore, the internal peripheral interrupt can be selected as a capture trigger. To select the internal peripheral interrupt, use the timer source select register.

- 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 how to release protection, see the description of the system protection 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																																
SELCNT	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td> </tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BASE + 0500H Initial value 0000 0000H
0	0	0	0	0	0	0	0	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 border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td><td style="width: 20px;">0</td> </tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	0	0	0	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																																
31 to 8	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.																																
7, 6	ISEL31, ISEL30	Specify the signal input to TIN3 (TAUJ2 ch3). <table border="1" style="margin: 5px auto; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ISEL31</th> <th style="width: 10%;">ISEL30</th> <th style="width: 80%;">Selection of TIN3 input signal (TAUJ ch3)</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>TIN3 (P52 pin)</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Interrupt signal selected in the TMTFR3 register</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>TSOUT signal of CAN1</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Setting prohibited (equivalent to setting 00)</td> </tr> </tbody> </table>	ISEL31	ISEL30	Selection of TIN3 input signal (TAUJ ch3)	0	0	TIN3 (P52 pin)	0	1	Interrupt signal selected in the TMTFR3 register	1	0	TSOUT signal of CAN1	1	1	Setting prohibited (equivalent to setting 00)																	
ISEL31	ISEL30	Selection of TIN3 input signal (TAUJ ch3)																																
0	0	TIN3 (P52 pin)																																
0	1	Interrupt signal selected in the TMTFR3 register																																
1	0	TSOUT signal of CAN1																																
1	1	Setting prohibited (equivalent to setting 00)																																
5, 4	ISEL21, ISEL20	Specify the signal input to TIN2 (TAUJ2 ch2). <table border="1" style="margin: 5px auto; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ISEL21</th> <th style="width: 10%;">ISEL20</th> <th style="width: 80%;">Selection of TIN2 input signal (TAUJ ch2)</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>TIN2 (P57 pin)</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Interrupt signal selected in the TMTFR2 register</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Setting prohibited (equivalent to setting 00)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Setting prohibited (equivalent to setting 00)</td> </tr> </tbody> </table>	ISEL21	ISEL20	Selection of TIN2 input signal (TAUJ ch2)	0	0	TIN2 (P57 pin)	0	1	Interrupt signal selected in the TMTFR2 register	1	0	Setting prohibited (equivalent to setting 00)	1	1	Setting prohibited (equivalent to setting 00)																	
ISEL21	ISEL20	Selection of TIN2 input signal (TAUJ ch2)																																
0	0	TIN2 (P57 pin)																																
0	1	Interrupt signal selected in the TMTFR2 register																																
1	0	Setting prohibited (equivalent to setting 00)																																
1	1	Setting prohibited (equivalent to setting 00)																																

Bit Position	Bit Name	Description															
3, 2	ISEL11, ISEL10	Specify the signal input to TIN1 (TAUJ2 ch1). <table border="1" data-bbox="496 400 1265 607"> <thead> <tr> <th>ISEL11</th> <th>ISEL10</th> <th>Selection of TIN1 input signal (TAUJ ch1)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>TIN1 (P26 pin)</td> </tr> <tr> <td>0</td> <td>1</td> <td>Interrupt signal selected in 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 (equivalent to setting 00)</td> </tr> </tbody> </table>	ISEL11	ISEL10	Selection of TIN1 input signal (TAUJ ch1)	0	0	TIN1 (P26 pin)	0	1	Interrupt signal selected in the TMTFR1 register	1	0	TSOUT signal of CAN0	1	1	Setting prohibited (equivalent to setting 00)
ISEL11	ISEL10	Selection of TIN1 input signal (TAUJ ch1)															
0	0	TIN1 (P26 pin)															
0	1	Interrupt signal selected in the TMTFR1 register															
1	0	TSOUT signal of CAN0															
1	1	Setting prohibited (equivalent to setting 00)															
1, 0	ISEL01, ISEL00	Specify the signal input to TIN0 (TAUJ2 ch0). <table border="1" data-bbox="496 696 1265 902"> <thead> <tr> <th>ISEL01</th> <th>ISEL00</th> <th>Selection of TIN0 input signal (TAUJ ch0)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>TIN0 (P27 pin)</td> </tr> <tr> <td>0</td> <td>1</td> <td>Interrupt signal selected in the TMTFR0 register</td> </tr> <tr> <td>1</td> <td>0</td> <td>Setting prohibited (equivalent to setting 00)</td> </tr> <tr> <td>1</td> <td>1</td> <td>Setting prohibited (equivalent to setting 00)</td> </tr> </tbody> </table>	ISEL01	ISEL00	Selection of TIN0 input signal (TAUJ ch0)	0	0	TIN0 (P27 pin)	0	1	Interrupt signal selected in the TMTFR0 register	1	0	Setting prohibited (equivalent to setting 00)	1	1	Setting prohibited (equivalent to setting 00)
ISEL01	ISEL00	Selection of TIN0 input signal (TAUJ ch0)															
0	0	TIN0 (P27 pin)															
0	1	Interrupt signal selected in the TMTFR0 register															
1	0	Setting prohibited (equivalent to setting 00)															
1	1	Setting prohibited (equivalent to setting 00)															

Bit Position	Bit Name	Description																																																																				
16 to 0	IFC6 to IFC0	Specify the trigger source for timer channel n.																																																																				
		<table border="1"> <thead> <tr> <th>IFC6 to IFC0</th> <th>Selection of timer count trigger source</th> </tr> </thead> <tbody> <tr><td>0EH</td><td>CSI0 job completion interrupt</td></tr> <tr><td>0FH</td><td>CSI1 communication status interrupt</td></tr> <tr><td>10H</td><td>CSI1 reception status interrupt</td></tr> <tr><td>11H</td><td>CSI1 job completion interrupt</td></tr> <tr><td>12H</td><td>IICB0 data transmission/reception interrupt request signal</td></tr> <tr><td>13H</td><td>IICB1 data transmission/reception interrupt request signal</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/wakeup or transmission stop interrupt</td></tr> <tr><td>17H</td><td>FCN1 reception completion interrupt</td></tr> <tr><td>18H</td><td>FCN1 transmission completion interrupt</td></tr> <tr><td>19H</td><td>FCN1 sleep wakeup/transmission stop interrupt</td></tr> <tr><td>1AH</td><td>General-purpose DMAC channel 0 transfer completion interrupt</td></tr> <tr><td>1BH</td><td>General-purpose DMAC channel 1 transfer completion interrupt</td></tr> <tr><td>1CH</td><td>General-purpose DMAC channel 2 transfer completion interrupt</td></tr> <tr><td>1DH</td><td>General-purpose DMAC channel 3 transfer completion interrupt</td></tr> <tr><td>1EH</td><td>Real-time port DMAC transfer completion interrupt</td></tr> <tr><td>1FH, 20H</td><td>Reserved (setting prohibited)</td></tr> <tr><td>21H</td><td>EtherCAT interrupt ^{Note}</td></tr> <tr><td>22H</td><td>EtherCAT SOF interrupt ^{Note}</td></tr> <tr><td>23H</td><td>EtherCAT EOF interrupt ^{Note}</td></tr> <tr><td>24H</td><td>Inter-buffer DMA transfer completion</td></tr> <tr><td>25H</td><td>Ethernet PHY interrupt 0</td></tr> <tr><td>26H</td><td>Ethernet PHY interrupt 1</td></tr> <tr><td>27H</td><td>Ethernet MII management access completion interrupt</td></tr> <tr><td>28H</td><td>Ethernet pause packet transmission completion</td></tr> <tr><td>29H</td><td>Ethernet transmission completion interrupt</td></tr> <tr><td>2AH</td><td>Ethernet switch interrupt</td></tr> <tr><td>2BH</td><td>Ethernet switch DLR interrupt</td></tr> <tr><td>2CH</td><td>Ethernet switch SEC interrupt</td></tr> <tr><td>2DH, 2EH</td><td>Reserved (setting prohibited)</td></tr> <tr><td>2FH</td><td>Ethernet MACDMA reception completion</td></tr> <tr><td>30H</td><td>Ethernet MACDMA transmission completion</td></tr> </tbody> </table>	IFC6 to IFC0	Selection of timer count trigger source	0EH	CSI0 job completion interrupt	0FH	CSI1 communication status interrupt	10H	CSI1 reception status interrupt	11H	CSI1 job completion interrupt	12H	IICB0 data transmission/reception interrupt request signal	13H	IICB1 data transmission/reception interrupt request signal	14H	FCN0 reception completion interrupt	15H	FCN0 transmission completion interrupt	16H	FCN0 sleep/wakeup or transmission stop interrupt	17H	FCN1 reception completion interrupt	18H	FCN1 transmission completion interrupt	19H	FCN1 sleep wakeup/transmission stop interrupt	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 port DMAC transfer completion interrupt	1FH, 20H	Reserved (setting prohibited)	21H	EtherCAT interrupt ^{Note}	22H	EtherCAT SOF interrupt ^{Note}	23H	EtherCAT EOF interrupt ^{Note}	24H	Inter-buffer DMA transfer completion	25H	Ethernet PHY interrupt 0	26H	Ethernet PHY interrupt 1	27H	Ethernet MII management access completion interrupt	28H	Ethernet pause packet transmission completion	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	30H	Ethernet MACDMA transmission completion
IFC6 to IFC0	Selection of timer count trigger source																																																																					
0EH	CSI0 job completion interrupt																																																																					
0FH	CSI1 communication status interrupt																																																																					
10H	CSI1 reception status interrupt																																																																					
11H	CSI1 job completion interrupt																																																																					
12H	IICB0 data transmission/reception interrupt request signal																																																																					
13H	IICB1 data transmission/reception interrupt request signal																																																																					
14H	FCN0 reception completion interrupt																																																																					
15H	FCN0 transmission completion interrupt																																																																					
16H	FCN0 sleep/wakeup or transmission stop interrupt																																																																					
17H	FCN1 reception completion interrupt																																																																					
18H	FCN1 transmission completion interrupt																																																																					
19H	FCN1 sleep wakeup/transmission stop interrupt																																																																					
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 port DMAC transfer completion interrupt																																																																					
1FH, 20H	Reserved (setting prohibited)																																																																					
21H	EtherCAT interrupt ^{Note}																																																																					
22H	EtherCAT SOF interrupt ^{Note}																																																																					
23H	EtherCAT EOF interrupt ^{Note}																																																																					
24H	Inter-buffer DMA transfer completion																																																																					
25H	Ethernet PHY interrupt 0																																																																					
26H	Ethernet PHY interrupt 1																																																																					
27H	Ethernet MII management access completion interrupt																																																																					
28H	Ethernet pause packet transmission completion																																																																					
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																																																																					
30H	Ethernet MACDMA transmission completion																																																																					

Note: Only supported in the R-IN32M3-EC.

Bit Position	Bit Name	Description																																																																
16 to 0	IFC6 to IFC0	Specify the trigger source for timer channel n.																																																																
		<table border="1"> <thead> <tr> <th>IFC6 to IFC0</th> <th>Selection of timer count trigger source</th> </tr> </thead> <tbody> <tr> <td>31H</td> <td>Reception frame normal interrupt</td> </tr> <tr> <td>32H</td> <td>Reserved (setting prohibited)</td> </tr> <tr> <td>33H</td> <td>INTPZ0 input <small>Note</small></td> </tr> <tr> <td>34H</td> <td>INTPZ1 input <small>Note</small></td> </tr> <tr> <td>35H</td> <td>INTPZ2 input <small>Note</small></td> </tr> <tr> <td>36H</td> <td>INTPZ3 input <small>Note</small></td> </tr> <tr> <td>37H</td> <td>INTPZ4 input <small>Note</small></td> </tr> <tr> <td>38H</td> <td>INTPZ5 input <small>Note</small></td> </tr> <tr> <td>39H</td> <td>INTPZ6 input <small>Note</small></td> </tr> <tr> <td>3AH</td> <td>INTPZ7 input <small>Note</small></td> </tr> <tr> <td>3BH</td> <td>INTPZ8 input <small>Note</small></td> </tr> <tr> <td>3CH</td> <td>INTPZ9 input <small>Note</small></td> </tr> <tr> <td>3DH</td> <td>INTPZ10 input <small>Note</small></td> </tr> <tr> <td>3EH</td> <td>INTPZ11 input <small>Note</small></td> </tr> <tr> <td>3FH</td> <td>INTPZ12 input <small>Note</small></td> </tr> <tr> <td>40H</td> <td>INTPZ13 input <small>Note</small></td> </tr> <tr> <td>41H</td> <td>INTPZ14 input <small>Note</small></td> </tr> <tr> <td>42H</td> <td>INTPZ15 input <small>Note</small></td> </tr> <tr> <td>43H</td> <td>INTPZ16 input <small>Note</small></td> </tr> <tr> <td>44H</td> <td>INTPZ17 input <small>Note</small></td> </tr> <tr> <td>45H</td> <td>INTPZ18 input <small>Note</small></td> </tr> <tr> <td>46H</td> <td>INTPZ19 input <small>Note</small></td> </tr> <tr> <td>47H</td> <td>INTPZ20 input <small>Note</small></td> </tr> <tr> <td>48H</td> <td>INTPZ21 input <small>Note</small></td> </tr> <tr> <td>49H</td> <td>INTPZ22 input <small>Note</small></td> </tr> <tr> <td>4AH</td> <td>INTPZ23 input <small>Note</small></td> </tr> <tr> <td>4BH</td> <td>INTPZ24 input <small>Note</small></td> </tr> <tr> <td>4CH</td> <td>INTPZ25 input <small>Note</small></td> </tr> <tr> <td>4DH</td> <td>INTPZ26 input <small>Note</small></td> </tr> <tr> <td>4EH</td> <td>INTPZ27 input <small>Note</small></td> </tr> <tr> <td>4FH</td> <td>INTPZ28 input <small>Note</small></td> </tr> </tbody> </table>	IFC6 to IFC0	Selection of timer count trigger source	31H	Reception frame normal interrupt	32H	Reserved (setting prohibited)	33H	INTPZ0 input <small>Note</small>	34H	INTPZ1 input <small>Note</small>	35H	INTPZ2 input <small>Note</small>	36H	INTPZ3 input <small>Note</small>	37H	INTPZ4 input <small>Note</small>	38H	INTPZ5 input <small>Note</small>	39H	INTPZ6 input <small>Note</small>	3AH	INTPZ7 input <small>Note</small>	3BH	INTPZ8 input <small>Note</small>	3CH	INTPZ9 input <small>Note</small>	3DH	INTPZ10 input <small>Note</small>	3EH	INTPZ11 input <small>Note</small>	3FH	INTPZ12 input <small>Note</small>	40H	INTPZ13 input <small>Note</small>	41H	INTPZ14 input <small>Note</small>	42H	INTPZ15 input <small>Note</small>	43H	INTPZ16 input <small>Note</small>	44H	INTPZ17 input <small>Note</small>	45H	INTPZ18 input <small>Note</small>	46H	INTPZ19 input <small>Note</small>	47H	INTPZ20 input <small>Note</small>	48H	INTPZ21 input <small>Note</small>	49H	INTPZ22 input <small>Note</small>	4AH	INTPZ23 input <small>Note</small>	4BH	INTPZ24 input <small>Note</small>	4CH	INTPZ25 input <small>Note</small>	4DH	INTPZ26 input <small>Note</small>	4EH	INTPZ27 input <small>Note</small>	4FH	INTPZ28 input <small>Note</small>
IFC6 to IFC0	Selection of timer count trigger source																																																																	
31H	Reception frame normal interrupt																																																																	
32H	Reserved (setting prohibited)																																																																	
33H	INTPZ0 input <small>Note</small>																																																																	
34H	INTPZ1 input <small>Note</small>																																																																	
35H	INTPZ2 input <small>Note</small>																																																																	
36H	INTPZ3 input <small>Note</small>																																																																	
37H	INTPZ4 input <small>Note</small>																																																																	
38H	INTPZ5 input <small>Note</small>																																																																	
39H	INTPZ6 input <small>Note</small>																																																																	
3AH	INTPZ7 input <small>Note</small>																																																																	
3BH	INTPZ8 input <small>Note</small>																																																																	
3CH	INTPZ9 input <small>Note</small>																																																																	
3DH	INTPZ10 input <small>Note</small>																																																																	
3EH	INTPZ11 input <small>Note</small>																																																																	
3FH	INTPZ12 input <small>Note</small>																																																																	
40H	INTPZ13 input <small>Note</small>																																																																	
41H	INTPZ14 input <small>Note</small>																																																																	
42H	INTPZ15 input <small>Note</small>																																																																	
43H	INTPZ16 input <small>Note</small>																																																																	
44H	INTPZ17 input <small>Note</small>																																																																	
45H	INTPZ18 input <small>Note</small>																																																																	
46H	INTPZ19 input <small>Note</small>																																																																	
47H	INTPZ20 input <small>Note</small>																																																																	
48H	INTPZ21 input <small>Note</small>																																																																	
49H	INTPZ22 input <small>Note</small>																																																																	
4AH	INTPZ23 input <small>Note</small>																																																																	
4BH	INTPZ24 input <small>Note</small>																																																																	
4CH	INTPZ25 input <small>Note</small>																																																																	
4DH	INTPZ26 input <small>Note</small>																																																																	
4EH	INTPZ27 input <small>Note</small>																																																																	
4FH	INTPZ28 input <small>Note</small>																																																																	

Note: When using an external interrupt as a timer trigger source, be sure to specify edge detection (do not specify level detection).

Bit Position	Bit Name	Description																						
16 to 0	IFC6 to IFC0	Specify the trigger source for timer channel n. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>IFC6 to IFC0</th> <th>Selection of timer count trigger source</th> </tr> </thead> <tbody> <tr> <td>50H to 6EH</td> <td>Reserved (setting prohibited)</td> </tr> <tr> <td>6FH</td> <td>CC-Link IE Field Network NMIZ interrupt ^{Note}</td> </tr> <tr> <td>70H</td> <td>CC-Link IE Field WDTZ interrupt ^{Note}</td> </tr> <tr> <td>71H</td> <td>CC-Link IE Field INTZ interrupt ^{Note}</td> </tr> <tr> <td>72H</td> <td>CC-Link IE Field Network CLKLOSSZ interrupt ^{Note}</td> </tr> <tr> <td>73H to 76H</td> <td>Reserved (setting prohibited)</td> </tr> <tr> <td>77H</td> <td>CC-Link master IRZ interrupt</td> </tr> <tr> <td>78H</td> <td>CC-Link REFSTB interrupt</td> </tr> <tr> <td>79H</td> <td>CC-Link MON3 interrupt</td> </tr> <tr> <td>7AH to 7FH</td> <td>Reserved (setting prohibited)</td> </tr> </tbody> </table>	IFC6 to IFC0	Selection of timer count trigger source	50H to 6EH	Reserved (setting prohibited)	6FH	CC-Link IE Field Network NMIZ interrupt ^{Note}	70H	CC-Link IE Field WDTZ interrupt ^{Note}	71H	CC-Link IE Field INTZ interrupt ^{Note}	72H	CC-Link IE Field Network CLKLOSSZ interrupt ^{Note}	73H to 76H	Reserved (setting prohibited)	77H	CC-Link master IRZ interrupt	78H	CC-Link REFSTB interrupt	79H	CC-Link MON3 interrupt	7AH to 7FH	Reserved (setting prohibited)
IFC6 to IFC0	Selection of timer count trigger source																							
50H to 6EH	Reserved (setting prohibited)																							
6FH	CC-Link IE Field Network NMIZ interrupt ^{Note}																							
70H	CC-Link IE Field WDTZ interrupt ^{Note}																							
71H	CC-Link IE Field INTZ interrupt ^{Note}																							
72H	CC-Link IE Field Network CLKLOSSZ interrupt ^{Note}																							
73H to 76H	Reserved (setting prohibited)																							
77H	CC-Link master IRZ interrupt																							
78H	CC-Link REFSTB interrupt																							
79H	CC-Link MON3 interrupt																							
7AH to 7FH	Reserved (setting prohibited)																							

Note: Only supported in the R-IN32M3-CL.

21.10 Noise Eliminator

Digital noise filtering is used to eliminate noise from the external interrupt input signals, timer array input signals, and UART serial data input signals.

Use noise filter setting registers 0 to 3 (NFC0 to NFC3) to make settings for noise elimination.

Caution: This function can only be set for the CPU of an R-IN32M3, but is not available for applications that do not use the internal CPU.

Table 21.1 Signals Subject to Noise Elimination

Signals	Internally Connected Unit	Function of Signal
NMIZ	Interrupt controller	Non-maskable external interrupt input
INTPZ0 to INTPZ28	Interrupt controller	Maskable external interrupt input
TIN0 to TIN3	Timer array unit (TAUJ2)	Timer input
RXD0, RXD1	Asynchronous serial interface (UARTJ)	UART serial data input

21.10.1 Noise Filter Setting Registers 0 to 3 (NFC0 to NFC3)

These registers are used to specify the width of the noise eliminated from the input signals shown in Table 21.1.

- Access These registers can be read or written in 32-bit units.

- Cautions 1.** If the input pulse width is between the setting of NFC0 to NFC3 and the setting of NFC0 to NFC3 minus 1, it is undefined whether the signal is handled as a valid signal or eliminated as noise.
2. The interrupt input signals (INTPZ0 to INTPZ28 and NMIZ) are input via an edge specification circuit when operating as interrupt signals, but not when operating as an alternate function. The effective edge of the timer array unit input pins is specified by the timer array unit edge specification register. It is not possible to specify an effective edge for the RXD0 and RXD1 input pins.
 3. An interrupt might be inadvertently generated for each register when changing the settings of the NFC0 to NFC3 registers. Therefore, be sure to disable interrupt requests before changing the settings of these registers and then clear the interrupt pending bit of the corresponding register after changing the setting.
 4. INTPZ0 to INTPZ28 and NMI are input via an edge specification circuit in synchronization with the input to the CPU. A delay will therefore always occur even if the number of filter stages is set to 0.
 5. These registers are only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.

NFC1	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																																
	<table border="1" style="width:100%; border-collapse: collapse;"> <tr><td style="width:100%;">NFP151</td></tr> <tr><td>NFP150</td></tr> <tr><td>NFP141</td></tr> <tr><td>NFP140</td></tr> <tr><td>NFP131</td></tr> <tr><td>NFP130</td></tr> <tr><td>NFP121</td></tr> <tr><td>NFP120</td></tr> <tr><td>NFP111</td></tr> <tr><td>NFP110</td></tr> <tr><td>NFP101</td></tr> <tr><td>NFP100</td></tr> <tr><td>NFP91</td></tr> <tr><td>NFP90</td></tr> <tr><td>NFP81</td></tr> <tr><td>NFP80</td></tr> <tr><td>NFP71</td></tr> <tr><td>NFP70</td></tr> <tr><td>NFP61</td></tr> <tr><td>NFP60</td></tr> <tr><td>NFP51</td></tr> <tr><td>NFP50</td></tr> <tr><td>NFP41</td></tr> <tr><td>NFP40</td></tr> <tr><td>NFP31</td></tr> <tr><td>NFP30</td></tr> <tr><td>NFP21</td></tr> <tr><td>NFP20</td></tr> <tr><td>NFP11</td></tr> <tr><td>NFP10</td></tr> <tr><td>NFP01</td></tr> <tr><td>NFP00</td></tr> </table>	NFP151	NFP150	NFP141	NFP140	NFP131	NFP130	NFP121	NFP120	NFP111	NFP110	NFP101	NFP100	NFP91	NFP90	NFP81	NFP80	NFP71	NFP70	NFP61	NFP60	NFP51	NFP50	NFP41	NFP40	NFP31	NFP30	NFP21	NFP20	NFP11	NFP10	NFP01	NFP00	BASE + 0704H
NFP151																																		
NFP150																																		
NFP141																																		
NFP140																																		
NFP131																																		
NFP130																																		
NFP121																																		
NFP120																																		
NFP111																																		
NFP110																																		
NFP101																																		
NFP100																																		
NFP91																																		
NFP90																																		
NFP81																																		
NFP80																																		
NFP71																																		
NFP70																																		
NFP61																																		
NFP60																																		
NFP51																																		
NFP50																																		
NFP41																																		
NFP40																																		
NFP31																																		
NFP30																																		
NFP21																																		
NFP20																																		
NFP11																																		
NFP10																																		
NFP01																																		
NFP00																																		
R/W	R/W R/W																																	
NFC2	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																																
	<table border="1" style="width:100%; border-collapse: collapse;"> <tr><td style="width:100%;">0</td></tr> <tr><td>0</td></tr> <tr><td>0</td></tr> <tr><td>0</td></tr> <tr><td>0</td></tr> <tr><td>0</td></tr> <tr><td>NFP281</td></tr> <tr><td>NFP280</td></tr> <tr><td>NFP271</td></tr> <tr><td>NFP270</td></tr> <tr><td>NFP261</td></tr> <tr><td>NFP260</td></tr> <tr><td>NFP251</td></tr> <tr><td>NFP250</td></tr> <tr><td>NFP241</td></tr> <tr><td>NFP240</td></tr> <tr><td>NFP231</td></tr> <tr><td>NFP230</td></tr> <tr><td>NFP221</td></tr> <tr><td>NFP220</td></tr> <tr><td>NFP211</td></tr> <tr><td>NFP210</td></tr> <tr><td>NFP201</td></tr> <tr><td>NFP200</td></tr> <tr><td>NFP191</td></tr> <tr><td>NFP190</td></tr> <tr><td>NFP181</td></tr> <tr><td>NFP180</td></tr> <tr><td>NFP171</td></tr> <tr><td>NFP170</td></tr> <tr><td>NFP161</td></tr> <tr><td>NFP160</td></tr> </table>	0	0	0	0	0	0	NFP281	NFP280	NFP271	NFP270	NFP261	NFP260	NFP251	NFP250	NFP241	NFP240	NFP231	NFP230	NFP221	NFP220	NFP211	NFP210	NFP201	NFP200	NFP191	NFP190	NFP181	NFP180	NFP171	NFP170	NFP161	NFP160	BASE + 0708H
0																																		
0																																		
0																																		
0																																		
0																																		
0																																		
NFP281																																		
NFP280																																		
NFP271																																		
NFP270																																		
NFP261																																		
NFP260																																		
NFP251																																		
NFP250																																		
NFP241																																		
NFP240																																		
NFP231																																		
NFP230																																		
NFP221																																		
NFP220																																		
NFP211																																		
NFP210																																		
NFP201																																		
NFP200																																		
NFP191																																		
NFP190																																		
NFP181																																		
NFP180																																		
NFP171																																		
NFP170																																		
NFP161																																		
NFP160																																		
R/W	0 0 0 0 0 0 R/W																																	
Bit Position	Bit Name	Description																																
31 to 0	NFPn1, NFPn0	Specify the number of noise filter stages for the INTPZn input based on the number of cycles of the internal system clock.																																
		<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:10%;">NFPn1</th> <th style="width:10%;">NFPn0</th> <th style="width:80%;">Noise Filter Stage</th> </tr> </thead> <tbody> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">0</td> <td>0 HCLK cycles</td> </tr> <tr> <td style="text-align:center;">0</td> <td style="text-align:center;">1</td> <td>4 HCLK cycles</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">0</td> <td>8 HCLK cycles</td> </tr> <tr> <td style="text-align:center;">1</td> <td style="text-align:center;">1</td> <td>16 HCLK cycles</td> </tr> </tbody> </table>		NFPn1	NFPn0	Noise Filter Stage	0	0	0 HCLK cycles	0	1	4 HCLK cycles	1	0	8 HCLK cycles	1	1	16 HCLK cycles																
NFPn1	NFPn0	Noise Filter Stage																																
0	0	0 HCLK cycles																																
0	1	4 HCLK cycles																																
1	0	8 HCLK cycles																																
1	1	16 HCLK cycles																																

Remark: n = 0 to 28

NFC3	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
	0 0	BASE + 070CH
	NFRX11 NFRX10 NFRX01 NFRX00 0	Initial value
	R/W 0 0 0 0 0 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 R/W R/W R/W R/W R/W R/W R/W R/W	0000 0000H

Bit Position	Bit Name	Description															
31 to 20, 15 to 8	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.															
19 to 16	NFRXn1, NFRXn0	Specify the number of noise filter stages for the RXD1 and RXD0 inputs based on the number of cycles of the internal system clock. <table border="1" style="margin: 10px auto; border-collapse: collapse; width: 80%;"> <thead> <tr> <th>NFRXn1</th> <th>NFRXn0</th> <th>Noise Filter Stage</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>0 HCLK cycles</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>4 HCLK cycles</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>8 HCLK cycles</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>16 HCLK cycles</td> </tr> </tbody> </table>	NFRXn1	NFRXn0	Noise Filter Stage	0	0	0 HCLK cycles	0	1	4 HCLK cycles	1	0	8 HCLK cycles	1	1	16 HCLK cycles
NFRXn1	NFRXn0	Noise Filter Stage															
0	0	0 HCLK cycles															
0	1	4 HCLK cycles															
1	0	8 HCLK cycles															
1	1	16 HCLK cycles															
7 to 0	NFTINm1, NFTINm0	Specify the number of noise filter stages for the TIN3 to TIN0 inputs based on the number of cycles of the internal system clock. <table border="1" style="margin: 10px auto; border-collapse: collapse; width: 80%;"> <thead> <tr> <th>NFTINm1</th> <th>NTINm0</th> <th>Noise Filter Stage</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>0 HCLK cycles</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>4 HCLK cycles</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>8 HCLK cycles</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>16 HCLK cycles</td> </tr> </tbody> </table>	NFTINm1	NTINm0	Noise Filter Stage	0	0	0 HCLK cycles	0	1	4 HCLK cycles	1	0	8 HCLK cycles	1	1	16 HCLK cycles
NFTINm1	NTINm0	Noise Filter Stage															
0	0	0 HCLK cycles															
0	1	4 HCLK cycles															
1	0	8 HCLK cycles															
1	1	16 HCLK cycles															

Remark: n = 1 or 0, m = 3 to 0

21.10.2 Noise Filtering Operation

The input signals shown in Table 21.1 are sampled based on a clock with the same frequency as the internal bus clock HCLK and noise of the width specified by the noise filter setting registers (NFC0 to NFC3) is eliminated. Because this sampling clock does not stop in standby mode, the NMI and INTPZ0 to INTPZ28 external interrupts can be used to release standby mode. Rising edges, falling edges, rising and falling edges, or active-low level of the INTPZ0 to INTPZ28 signals can also be selected as effective triggers.

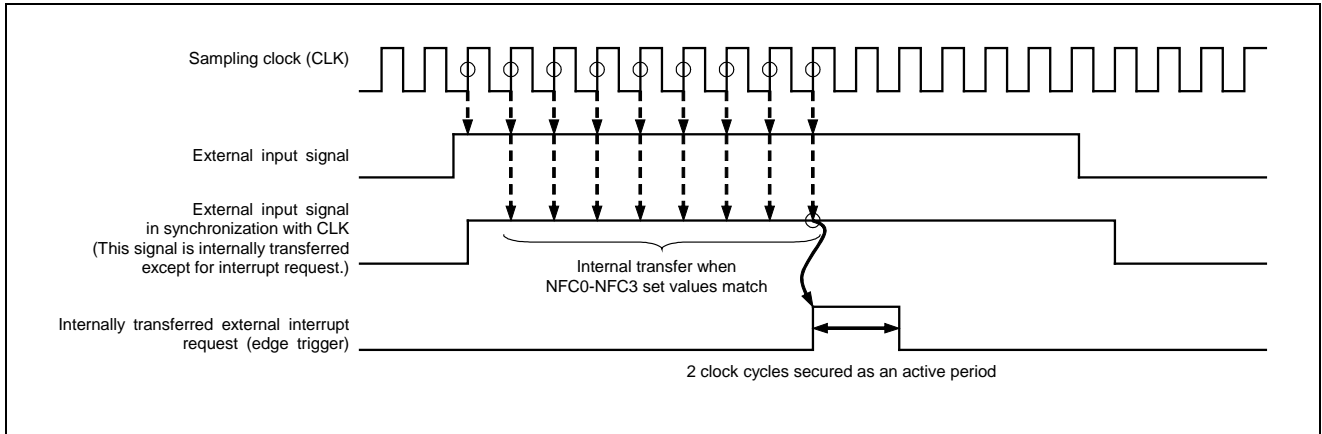


Figure 21.3 Digital Noise Filtering on Interrupt Signals (when the signal edge is used as the trigger)

21.11 External Interrupt Mode Registers 0, 1, 2 (INTM0, INTM1, INTM2)

These registers are used to specify the trigger mode for the external interrupt requests input via external pins (NMIZ and INTPZ0 to INTPZ28, ETH1_GE_INT, ETH0_GE_INT). The external interrupt requests controlled by each of these registers are shown below.

- Access These registers can be read or written in 32-bit units.
 - INTM0: NMIZ, ETH1_GE_INT, ETH0_GE_INT
 - INTM1: INTPZ0 to INTPZ15
 - INTM2: INTPZ16 to INTPZ28

Cautions 1. These registers are only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.

2. INTPZ0 to INTPZ28 are multiplexed with the port pins. When these pins are specified by the PMCM register to operate as interrupt pins, an unnecessary interrupt might be generated, depending on the immediately preceding status. To prevent this from happening, be sure to mask each interrupt request before using the PMCM register to specify these pins as interrupt pins, and then clear the interrupt request flag after specifying the setting.

		Address																																																																
		BASE + 0710H																																																																
		Initial value																																																																
		0000 0002H																																																																
INTMO	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td><td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">E</td><td style="text-align: center;">E</td><td style="text-align: center;">E</td><td style="text-align: center;">E</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">E</td><td style="text-align: center;">E</td> </tr> </table>	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	E	E	E	0	0	0	0	0	0	0	0	0	0	E	E
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																																			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	E	E	E	0	0	0	0	0	0	0	0	0	0	E	E																																		
R/W	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">R</td><td style="text-align: center;">W</td><td style="text-align: center;">R</td><td style="text-align: center;">W</td><td style="text-align: center;">R</td><td style="text-align: center;">W</td><td style="text-align: center;">R</td><td style="text-align: center;">W</td><td style="text-align: center;">R</td><td style="text-align: center;">W</td><td style="text-align: center;">R</td><td style="text-align: center;">W</td><td style="text-align: center;">R</td><td style="text-align: center;">W</td><td style="text-align: center;">R</td><td style="text-align: center;">W</td> </tr> </table>	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																																
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 16	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.																																																																
15, 14	ESE11, ESE10	Specify the trigger of Ethernet 1 PHY interrupt signal (ETH1_GE_INT). <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ESE11</th> <th style="width: 10%;">ESE10</th> <th style="width: 80%;">Trigger of ETH1_GE_INT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Falling edge</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Rising edge (initial value)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Level detection (low level detection)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Rising and falling edges</td> </tr> </tbody> </table>	ESE11	ESE10	Trigger of ETH1_GE_INT	0	0	Falling edge	0	1	Rising edge (initial value)	1	0	Level detection (low level detection)	1	1	Rising and falling edges																																																	
ESE11	ESE10	Trigger of ETH1_GE_INT																																																																
0	0	Falling edge																																																																
0	1	Rising edge (initial value)																																																																
1	0	Level detection (low level detection)																																																																
1	1	Rising and falling edges																																																																
13, 12	ESE01, ESE00	Specify the trigger of Ethernet 0 PHY interrupt signal (ETH0_GE_INT). <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ESE01</th> <th style="width: 10%;">ESE00</th> <th style="width: 80%;">Trigger of ETH0_GE_INT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Falling edge</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Rising edge (initial value)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Level detection (low level detection)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Rising and falling edges</td> </tr> </tbody> </table>	ESE01	ESE00	Trigger of ETH0_GE_INT	0	0	Falling edge	0	1	Rising edge (initial value)	1	0	Level detection (low level detection)	1	1	Rising and falling edges																																																	
ESE01	ESE00	Trigger of ETH0_GE_INT																																																																
0	0	Falling edge																																																																
0	1	Rising edge (initial value)																																																																
1	0	Level detection (low level detection)																																																																
1	1	Rising and falling edges																																																																
1, 0	ESN01, ESN00	Specify the trigger of NMIZ. <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ES01</th> <th style="width: 10%;">ES00</th> <th style="width: 80%;">Trigger of NMIZ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Falling edge</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Rising edge</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Level detection (low level detection)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Rising and falling edges</td> </tr> </tbody> </table>	ES01	ES00	Trigger of NMIZ	0	0	Falling edge	0	1	Rising edge	1	0	Level detection (low level detection)	1	1	Rising and falling edges																																																	
ES01	ES00	Trigger of NMIZ																																																																
0	0	Falling edge																																																																
0	1	Rising edge																																																																
1	0	Level detection (low level detection)																																																																
1	1	Rising and falling edges																																																																

INTM1	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	<table border="1" style="width:100%; border-collapse: collapse;"> <tr><td>ESP151</td><td>ESP150</td><td>ESP141</td><td>ESP140</td><td>ESP131</td><td>ESP130</td><td>ESP121</td><td>ESP120</td><td>ESP111</td><td>ESP110</td><td>ESP101</td><td>ESP100</td><td>ESP91</td><td>ESP90</td><td>ESP81</td><td>ESP80</td><td>ESP71</td><td>ESP70</td><td>ESP61</td><td>ESP60</td><td>ESP51</td><td>ESP50</td><td>ESP41</td><td>ESP40</td><td>ESP31</td><td>ESP30</td><td>ESP21</td><td>ESP20</td><td>ESP11</td><td>ESP10</td><td>ESP01</td><td>ESP00</td></tr> </table>	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	Address BASE + 0714H Initial value 0000 0000H
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																																	
INTM2	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	<table border="1" style="width:100%; border-collapse: collapse;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>ESP281</td><td>ESP280</td><td>ESP271</td><td>ESP270</td><td>ESP261</td><td>ESP260</td><td>ESP251</td><td>ESP250</td><td>ESP241</td><td>ESP240</td><td>ESP231</td><td>ESP230</td><td>ESP221</td><td>ESP220</td><td>ESP211</td><td>ESP210</td><td>ESP201</td><td>ESP200</td><td>ESP191</td><td>ESP190</td><td>ESP181</td><td>ESP180</td><td>ESP171</td><td>ESP170</td><td>ESP161</td><td>ESP160</td></tr> </table>	0	0	0	0	0	0	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	Address BASE + 0718H Initial value 0000 0000H
0	0	0	0	0	0	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 0 0 R/W																																	

Bit Position	Bit Name	Description															
7 to 0	ESPn1 to ESPn0 (n = 0 to 28)	Specify the trigger of INTPZ0 to INTPZ28. <table border="1" style="width:100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width:10%;">ESn1</th> <th style="width:10%;">ESn0</th> <th style="width:80%;">Trigger of INTPZ0 to INTPZ28</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Falling edge (initial value)</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Rising edge</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Level detection (low level detection) ^{Note}</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Rising and falling edges</td> </tr> </tbody> </table>	ESn1	ESn0	Trigger of INTPZ0 to INTPZ28	0	0	Falling edge (initial value)	0	1	Rising edge	1	0	Level detection (low level detection) ^{Note}	1	1	Rising and falling edges
ESn1	ESn0	Trigger of INTPZ0 to INTPZ28															
0	0	Falling edge (initial value)															
0	1	Rising edge															
1	0	Level detection (low level detection) ^{Note}															
1	1	Rising and falling edges															

Note: If the active level (low) is input to the INTPZ0 to INTPZ28 pins, the input signal is judged as a successive pulse whose level toggles each time the internal system bus clock (HCLK) rises and an interrupt request is generated. HCLK, which is used to sample external interrupts, does not stop even in standby mode.

Remark: n = 28 to 0

21.12 Trigger-Synchronous Ports

The status of the 32-bit port pins RP00 to RP37 is updated in synchronization with an interrupt from an internal peripheral module.

Use the RPTRGMD register to specify whether to set a port to trigger-synchronous port control mode, in 1-bit units. Use the RPTFR0 to RPTFR3 registers to select the trigger.

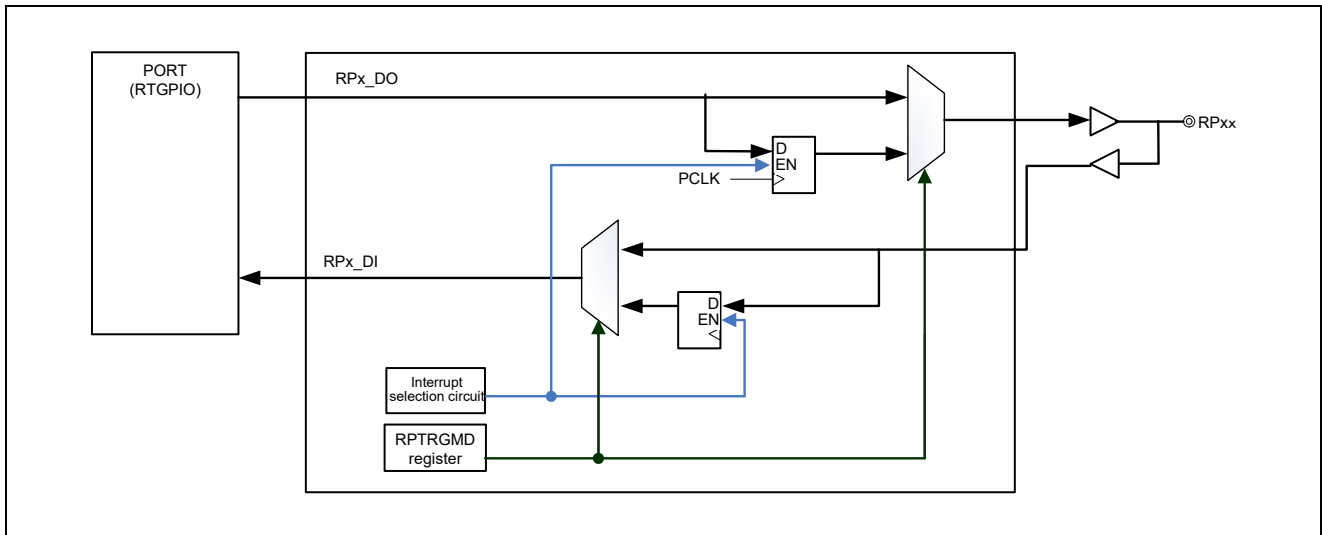
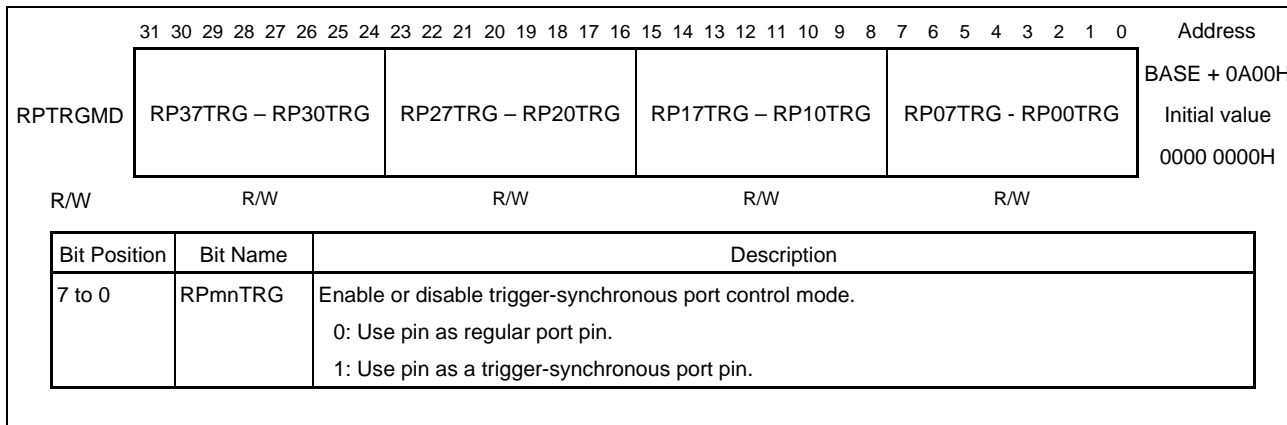


Figure 21.4 Configuration of Trigger-Synchronous Ports

21.12.1 Trigger-Synchronous Port Control Mode Register (RPTRGMD)

This register is used to select whether to set the 32-bit port pins RP00 to RP37 to trigger-synchronous port control mode, in 1-bit units.

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



Note: m = 3 to 0, n = 7 to 0

21.12.2 Trigger-Synchronous Port Source Registers (RP0TFR to RP3TFR)

A desired interrupt signal can be selected from interrupt request signals in RPTFR registers to assign it to the trigger-synchronous signal. Interrupt signals are selectable by using the trigger synchronous port source registers (RP0TFR to RP3TFR). Trigger sources can be set in units of 8-bit port of real-time ports (RP0x to RP3x).

- Access These registers can be read or written in 32-bit units.

Caution: These registers are only writable after protection has been released by a special sequence of writing to the system protection command register (SYSPCMD). For how to release protection, see the description of the system protection 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							
RPnTFR	0 0	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="width: 20px; text-align: center;">IFC6</td><td style="width: 20px; text-align: center;">IFC5</td><td style="width: 20px; text-align: center;">IFC4</td><td style="width: 20px; text-align: center;">IFC3</td><td style="width: 20px; text-align: center;">IFC2</td><td style="width: 20px; text-align: center;">IFC1</td><td style="width: 20px; text-align: center;">IFC0</td></tr> </table>	IFC6	IFC5	IFC4	IFC3	IFC2	IFC1	IFC0	BASE + 0A30H +4n Initial value 0000 0000H
IFC6	IFC5	IFC4	IFC3	IFC2	IFC1	IFC0				
R/W	0 0	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="width: 20px; text-align: center;">R/W</td><td style="width: 20px; text-align: center;">R/W</td><td style="width: 20px; text-align: center;">R/W</td><td style="width: 20px; text-align: center;">R/W</td><td style="width: 20px; text-align: center;">R/W</td><td style="width: 20px; text-align: center;">R/W</td><td style="width: 20px; text-align: center;">R/W</td></tr> </table>	R/W	R/W	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 7	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.								
6 to 0	IFC6 to IFC0	Specify the trigger source for timer-synchronous port n.								
		IFC6 to IFC0	Selection of timer count trigger source							
		00H	Timer output (TOUT0)							
		01H	Timer output (TOUT1)							
		02H	Timer output (TOUT2)							
		03H	Timer output (TOUT3)							
		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 to 3

Bit Position	Bit Name	Description																																																																						
16 to 0	IFC6 to IFC0	Specify the trigger source for timer-synchronous port n.																																																																						
		<table border="1"> <thead> <tr> <th>IFC6 to IFC0</th> <th>Selection of timer count trigger source</th> </tr> </thead> <tbody> <tr><td>0CH</td><td>CSI0 communication status interrupt</td></tr> <tr><td>0DH</td><td>CSI0 reception status interrupt</td></tr> <tr><td>0EH</td><td>CSI0 job completion interrupt</td></tr> <tr><td>0EH</td><td>CSI0 job completion interrupt</td></tr> <tr><td>0FH</td><td>CSI1 communication status interrupt</td></tr> <tr><td>10H</td><td>CSI1 reception status interrupt</td></tr> <tr><td>11H</td><td>CSI1 job completion interrupt</td></tr> <tr><td>12H</td><td>IICB0 data transmission/reception interrupt request signal</td></tr> <tr><td>13H</td><td>IICB1 data transmission/reception interrupt request signal</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/wakeup or transmission stop interrupt</td></tr> <tr><td>17H</td><td>FCN1 reception completion interrupt</td></tr> <tr><td>18H</td><td>FCN1 transmission completion interrupt</td></tr> <tr><td>19H</td><td>FCN1 sleep wakeup/transmission stop interrupt</td></tr> <tr><td>1AH</td><td>General-purpose DMAC channel 0 transfer completion interrupt</td></tr> <tr><td>1BH</td><td>General-purpose DMAC channel 1 transfer completion interrupt</td></tr> <tr><td>1CH</td><td>General-purpose DMAC channel 2 transfer completion interrupt</td></tr> <tr><td>1DH</td><td>General-purpose DMAC channel 3 transfer completion interrupt</td></tr> <tr><td>1EH</td><td>Real-time port DMAC transfer completion interrupt</td></tr> <tr><td>1FH</td><td>EtherCAT Sync0 interrupt^{Note}</td></tr> <tr><td>20H</td><td>EtherCAT Sync1 interrupt^{Note}</td></tr> <tr><td>21H</td><td>EtherCAT interrupt^{Note}</td></tr> <tr><td>22H</td><td>EtherCAT SOF interrupt^{Note}</td></tr> <tr><td>23H</td><td>EtherCAT EOF interrupt^{Note}</td></tr> <tr><td>24H</td><td>Inter-buffer DMA transfer completion</td></tr> <tr><td>25H</td><td>Ethernet PHY interrupt 0</td></tr> <tr><td>26H</td><td>Ethernet PHY interrupt 1</td></tr> <tr><td>27H</td><td>Ethernet MII management access completion interrupt</td></tr> <tr><td>28H</td><td>Ethernet pause packet transmission completion</td></tr> <tr><td>29H</td><td>Ethernet transmission completion interrupt</td></tr> <tr><td>2AH</td><td>Ethernet switch interrupt</td></tr> <tr><td>2BH</td><td>Ethernet switch DLR interrupt</td></tr> <tr><td>2CH</td><td>Ethernet switch SEC interrupt</td></tr> </tbody> </table>	IFC6 to IFC0	Selection of timer count trigger source	0CH	CSI0 communication status interrupt	0DH	CSI0 reception status interrupt	0EH	CSI0 job completion interrupt	0EH	CSI0 job completion interrupt	0FH	CSI1 communication status interrupt	10H	CSI1 reception status interrupt	11H	CSI1 job completion interrupt	12H	IICB0 data transmission/reception interrupt request signal	13H	IICB1 data transmission/reception interrupt request signal	14H	FCN0 reception completion interrupt	15H	FCN0 transmission completion interrupt	16H	FCN0 sleep/wakeup or transmission stop interrupt	17H	FCN1 reception completion interrupt	18H	FCN1 transmission completion interrupt	19H	FCN1 sleep wakeup/transmission stop interrupt	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 port DMAC transfer completion interrupt	1FH	EtherCAT Sync0 interrupt ^{Note}	20H	EtherCAT Sync1 interrupt ^{Note}	21H	EtherCAT interrupt ^{Note}	22H	EtherCAT SOF interrupt ^{Note}	23H	EtherCAT EOF interrupt ^{Note}	24H	Inter-buffer DMA transfer completion	25H	Ethernet PHY interrupt 0	26H	Ethernet PHY interrupt 1	27H	Ethernet MII management access completion interrupt	28H	Ethernet pause packet transmission completion	29H	Ethernet transmission completion interrupt	2AH	Ethernet switch interrupt	2BH	Ethernet switch DLR interrupt	2CH	Ethernet switch SEC interrupt
		IFC6 to IFC0	Selection of timer count trigger source																																																																					
		0CH	CSI0 communication status interrupt																																																																					
		0DH	CSI0 reception status interrupt																																																																					
		0EH	CSI0 job completion interrupt																																																																					
		0EH	CSI0 job completion interrupt																																																																					
		0FH	CSI1 communication status interrupt																																																																					
		10H	CSI1 reception status interrupt																																																																					
		11H	CSI1 job completion interrupt																																																																					
		12H	IICB0 data transmission/reception interrupt request signal																																																																					
		13H	IICB1 data transmission/reception interrupt request signal																																																																					
		14H	FCN0 reception completion interrupt																																																																					
		15H	FCN0 transmission completion interrupt																																																																					
		16H	FCN0 sleep/wakeup or transmission stop interrupt																																																																					
		17H	FCN1 reception completion interrupt																																																																					
		18H	FCN1 transmission completion interrupt																																																																					
		19H	FCN1 sleep wakeup/transmission stop interrupt																																																																					
		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 port DMAC transfer completion interrupt																																																																					
		1FH	EtherCAT Sync0 interrupt ^{Note}																																																																					
		20H	EtherCAT Sync1 interrupt ^{Note}																																																																					
		21H	EtherCAT interrupt ^{Note}																																																																					
		22H	EtherCAT SOF interrupt ^{Note}																																																																					
		23H	EtherCAT EOF interrupt ^{Note}																																																																					
		24H	Inter-buffer DMA transfer completion																																																																					
		25H	Ethernet PHY interrupt 0																																																																					
		26H	Ethernet PHY interrupt 1																																																																					
		27H	Ethernet MII management access completion interrupt																																																																					
28H	Ethernet pause packet transmission completion																																																																							
29H	Ethernet transmission completion interrupt																																																																							
2AH	Ethernet switch interrupt																																																																							
2BH	Ethernet switch DLR interrupt																																																																							
2CH	Ethernet switch SEC interrupt																																																																							

Note: Only supported in the R-IN32M3-EC.

Bit Position	Bit Name	Description																																																																
16 to 0	IFC6 to IFC0	Specify the trigger source for timer-synchronous port n.																																																																
		<table border="1"> <thead> <tr> <th>IFC6 to IFC0</th> <th>Selection of timer count trigger source</th> </tr> </thead> <tbody> <tr> <td>2DH, 2EH</td> <td>Reserved (setting prohibited)</td> </tr> <tr> <td>2FH</td> <td>Ethernet MACDMA reception completion</td> </tr> <tr> <td>30H</td> <td>Ethernet MACDMA transmission completion</td> </tr> <tr> <td>31H</td> <td>Reception frame normal interrupt</td> </tr> <tr> <td>32H</td> <td>Reserved (setting prohibited)</td> </tr> <tr> <td>33H</td> <td>INTPZ0 input ^{Note 1}</td> </tr> <tr> <td>34H</td> <td>INTPZ1 input ^{Note 1}</td> </tr> <tr> <td>35H</td> <td>INTPZ2 input ^{Note 1}</td> </tr> <tr> <td>36H</td> <td>INTPZ3 input ^{Note 1}</td> </tr> <tr> <td>37H</td> <td>INTPZ4 input ^{Note 1}</td> </tr> <tr> <td>38H</td> <td>INTPZ5 input ^{Note 1}</td> </tr> <tr> <td>39H</td> <td>INTPZ6 input ^{Note 1}</td> </tr> <tr> <td>3AH</td> <td>INTPZ7 input ^{Note 1}</td> </tr> <tr> <td>3BH</td> <td>INTPZ8 input ^{Note 1}</td> </tr> <tr> <td>3CH</td> <td>INTPZ9 input ^{Note 1}</td> </tr> <tr> <td>3DH</td> <td>INTPZ10 input ^{Note 1}</td> </tr> <tr> <td>3EH</td> <td>INTPZ11 input ^{Note 1}</td> </tr> <tr> <td>3FH</td> <td>INTPZ12 input ^{Note 1}</td> </tr> <tr> <td>40H</td> <td>INTPZ13 input ^{Note 1}</td> </tr> <tr> <td>41H</td> <td>INTPZ14 input ^{Note 1}</td> </tr> <tr> <td>42H</td> <td>INTPZ15 input ^{Note 1}</td> </tr> <tr> <td>43H</td> <td>INTPZ16 input ^{Note 1, Note 2}</td> </tr> <tr> <td>44H</td> <td>INTPZ17 input ^{Note 1, Note 2}</td> </tr> <tr> <td>45H</td> <td>INTPZ18 input ^{Note 1, Note 2}</td> </tr> <tr> <td>46H</td> <td>INTPZ19 input ^{Note 1, Note 2}</td> </tr> <tr> <td>47H</td> <td>INTPZ20 input ^{Note 1, Note 2}</td> </tr> <tr> <td>48H</td> <td>INTPZ21 input ^{Note 1, Note 2}</td> </tr> <tr> <td>49H</td> <td>INTPZ22 input ^{Note 1}</td> </tr> <tr> <td>4AH</td> <td>INTPZ23 input ^{Note 1}</td> </tr> <tr> <td>4BH</td> <td>INTPZ24 input ^{Note 1}</td> </tr> <tr> <td>4CH</td> <td>INTPZ25 input ^{Note 1, Note 2}</td> </tr> </tbody> </table>	IFC6 to IFC0	Selection of timer count trigger source	2DH, 2EH	Reserved (setting prohibited)	2FH	Ethernet MACDMA reception completion	30H	Ethernet MACDMA transmission completion	31H	Reception frame normal 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}	3BH	INTPZ8 input ^{Note 1}	3CH	INTPZ9 input ^{Note 1}	3DH	INTPZ10 input ^{Note 1}	3EH	INTPZ11 input ^{Note 1}	3FH	INTPZ12 input ^{Note 1}	40H	INTPZ13 input ^{Note 1}	41H	INTPZ14 input ^{Note 1}	42H	INTPZ15 input ^{Note 1}	43H	INTPZ16 input ^{Note 1, Note 2}	44H	INTPZ17 input ^{Note 1, Note 2}	45H	INTPZ18 input ^{Note 1, Note 2}	46H	INTPZ19 input ^{Note 1, Note 2}	47H	INTPZ20 input ^{Note 1, Note 2}	48H	INTPZ21 input ^{Note 1, Note 2}	49H	INTPZ22 input ^{Note 1}	4AH	INTPZ23 input ^{Note 1}	4BH	INTPZ24 input ^{Note 1}	4CH	INTPZ25 input ^{Note 1, Note 2}
IFC6 to IFC0	Selection of timer count trigger source																																																																	
2DH, 2EH	Reserved (setting prohibited)																																																																	
2FH	Ethernet MACDMA reception completion																																																																	
30H	Ethernet MACDMA transmission completion																																																																	
31H	Reception frame normal 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}																																																																	
3BH	INTPZ8 input ^{Note 1}																																																																	
3CH	INTPZ9 input ^{Note 1}																																																																	
3DH	INTPZ10 input ^{Note 1}																																																																	
3EH	INTPZ11 input ^{Note 1}																																																																	
3FH	INTPZ12 input ^{Note 1}																																																																	
40H	INTPZ13 input ^{Note 1}																																																																	
41H	INTPZ14 input ^{Note 1}																																																																	
42H	INTPZ15 input ^{Note 1}																																																																	
43H	INTPZ16 input ^{Note 1, Note 2}																																																																	
44H	INTPZ17 input ^{Note 1, Note 2}																																																																	
45H	INTPZ18 input ^{Note 1, Note 2}																																																																	
46H	INTPZ19 input ^{Note 1, Note 2}																																																																	
47H	INTPZ20 input ^{Note 1, Note 2}																																																																	
48H	INTPZ21 input ^{Note 1, Note 2}																																																																	
49H	INTPZ22 input ^{Note 1}																																																																	
4AH	INTPZ23 input ^{Note 1}																																																																	
4BH	INTPZ24 input ^{Note 1}																																																																	
4CH	INTPZ25 input ^{Note 1, Note 2}																																																																	

Notes 1. When using an external interrupt as a timer trigger source, be sure to specify edge detection (do not specify level detection).

- 2.** The INTPZ16 to INTPZ21 pin functions are multiplexed with the RP00 to RP05 pin functions. The INTPZ25 to INTPZ28 pin functions are multiplexed with the RP24 to 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.

Bit Position	Bit Name	Description																												
16 to 0	IFC6 to IFC0	Specify the trigger source for timer-synchronous port n. <table border="1" data-bbox="502 392 1369 936"> <thead> <tr> <th>IFC6 to IFC0</th> <th>Selection of timer count trigger source</th> </tr> </thead> <tbody> <tr> <td>4DH</td> <td>INTPZ26 input ^{Note 1, Note 2}</td> </tr> <tr> <td>4EH</td> <td>INTPZ27 input ^{Note 1, Note 2}</td> </tr> <tr> <td>4FH</td> <td>INTPZ28 input ^{Note 1, Note 2}</td> </tr> <tr> <td>50H to 6EH</td> <td>Reserved (setting prohibited)</td> </tr> <tr> <td>6FH</td> <td>CC-Link IE Field Network NMIZ interrupt ^{Note 3}</td> </tr> <tr> <td>70H</td> <td>CC-Link IE Field WDTZ interrupt ^{Note 3}</td> </tr> <tr> <td>71H</td> <td>CC-Link IE Field INTZ interrupt ^{Note 3}</td> </tr> <tr> <td>72H</td> <td>CC-Link IE Field Network CLKLOSSZ interrupt ^{Note 3}</td> </tr> <tr> <td>73H to 76H</td> <td>Reserved (setting prohibited)</td> </tr> <tr> <td>77H</td> <td>CC-Link INTRQ interrupt</td> </tr> <tr> <td>78H</td> <td>CC-Link REFSTB interrupt</td> </tr> <tr> <td>79H</td> <td>CC-Link MON3 interrupt</td> </tr> <tr> <td>7AH to 7FH</td> <td>Reserved (setting prohibited)</td> </tr> </tbody> </table>	IFC6 to IFC0	Selection of timer count trigger source	4DH	INTPZ26 input ^{Note 1, Note 2}	4EH	INTPZ27 input ^{Note 1, Note 2}	4FH	INTPZ28 input ^{Note 1, Note 2}	50H to 6EH	Reserved (setting prohibited)	6FH	CC-Link IE Field Network NMIZ interrupt ^{Note 3}	70H	CC-Link IE Field WDTZ interrupt ^{Note 3}	71H	CC-Link IE Field INTZ interrupt ^{Note 3}	72H	CC-Link IE Field Network CLKLOSSZ interrupt ^{Note 3}	73H to 76H	Reserved (setting prohibited)	77H	CC-Link INTRQ interrupt	78H	CC-Link REFSTB interrupt	79H	CC-Link MON3 interrupt	7AH to 7FH	Reserved (setting prohibited)
IFC6 to IFC0	Selection of timer count trigger source																													
4DH	INTPZ26 input ^{Note 1, Note 2}																													
4EH	INTPZ27 input ^{Note 1, Note 2}																													
4FH	INTPZ28 input ^{Note 1, Note 2}																													
50H to 6EH	Reserved (setting prohibited)																													
6FH	CC-Link IE Field Network NMIZ interrupt ^{Note 3}																													
70H	CC-Link IE Field WDTZ interrupt ^{Note 3}																													
71H	CC-Link IE Field INTZ interrupt ^{Note 3}																													
72H	CC-Link IE Field Network CLKLOSSZ interrupt ^{Note 3}																													
73H to 76H	Reserved (setting prohibited)																													
77H	CC-Link INTRQ interrupt																													
78H	CC-Link REFSTB interrupt																													
79H	CC-Link MON3 interrupt																													
7AH to 7FH	Reserved (setting prohibited)																													

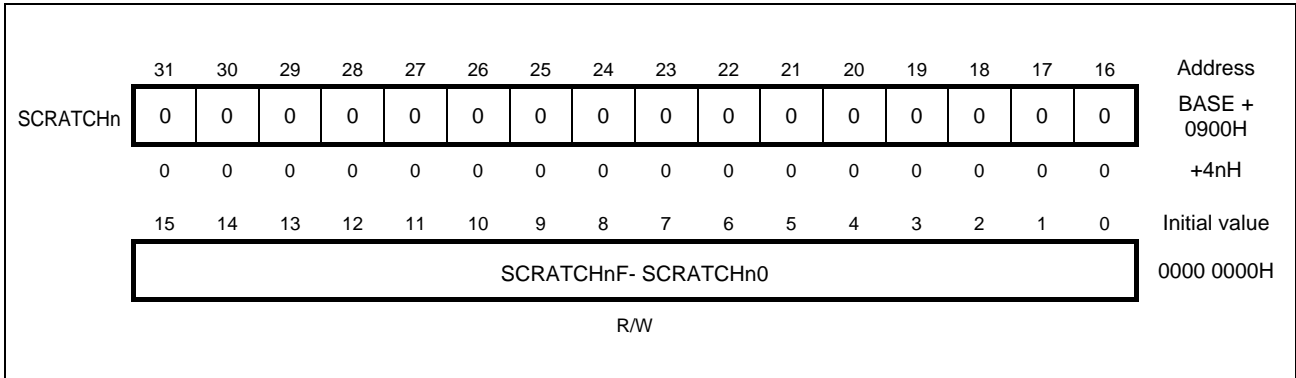
Notes 1. When using an external interrupt as a source of trigger synchronous port n, be sure to specify edge detection. (Do not specify level detection.)

2. The INTPZ16 to INTPZ21 pin functions are multiplexed with the RP00 to RP05 pin functions. The INTPZ25 to INTPZ28 pin functions are multiplexed with the RP24 to 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.

3. Only supported in the R-IN32M3-CL.

21.13 Scratch Registers (SCRATCH0 to SCRATCHD)

These are 16-bit registers that can be used for general purposes. They can also be used to transfer status information to and from the host. These registers can be read or written in 32- or 16-bit units.



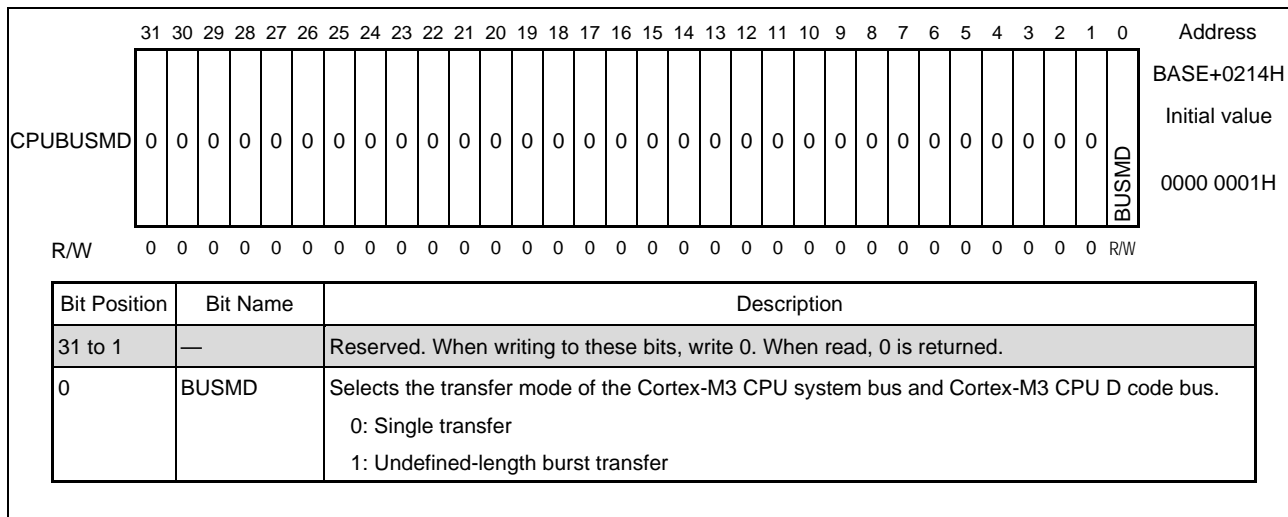
Remark: n = 0 to D

21.14 CPU Bus Operating Mode Register (CPUBUSMD)

This register is for switching the operating mode of the CPU buses between single transfer and undefined-length burst transfer.

For how to use this register, see section 4.1, Bus Occupancy by the Cortex-M3.

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



Remark: The old products do not support this register.
For the old products, see section 1.1, Type Names of R-IN32M3-Series Products.

21.15 SRAM Bridge Select Register (SRAMBRSEL)

The SRAMBRSEL register enables accesses to the CC-Link IE Field Network (direct access from the external MCU or access path from the CPU).

- Access This register can be read or written in 32- or 16-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 how to release protection, see the description of the system protection command register (SYSPCMD). No special sequence is required for reading the register.**
- 2. Write access to this register is prohibited during an access to the CC-Link IE Field Network. Set this register before the CC-Link IE Field Network function is used.**

	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	
SRAMBRSEL	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 100%;">0 0</td> </tr> </table>	0 0	BASE + 0804H
0 0			
	<table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 100%;">0 0</td> </tr> </table>	0 0	Initial value
0 0			
	R/W	Note2	

Bit Position	Bit Name	Function
31 to 1	—	Reserved. When writing to these bits, write 0. When read, 0 is returned.
0	BRCIE	This bit enables the access path to the CC-Link IE Field Network from the system bus (AHB) or from the external MCU. 0: Enables access from the external MCU. 1: Enables access from the system bus (AHB).

Notes

- 1. Only supported in the R-IN32M3-CL.**
- 2. The initial value of this register changes with the MEMIFSEL pin level.**
 0: 0000 0001H
 1: 0000 0000H

22. Debugging

The Cortex-M3 of an R-IN32M3 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-IN32M3 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-M3 of an R-IN32M3, see the CPU section in the R-IN32M3 User's Manual: Peripheral Modules.

The recommended in-circuit emulators (ICE) to be connected to an R-IN32M3 are: I-jet (without trace feature) and JTAGjet (with trace feature) from IAR Systems, and adviceLUNA II from DTS INSIGHT Corporation.

22.1 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).

The figures below are examples of connection with the ICE (In Circuit Emulator) connector: one is connection with the 20-pin half-pitch connector and the other is connection with the 20-pin full-pitch connector.

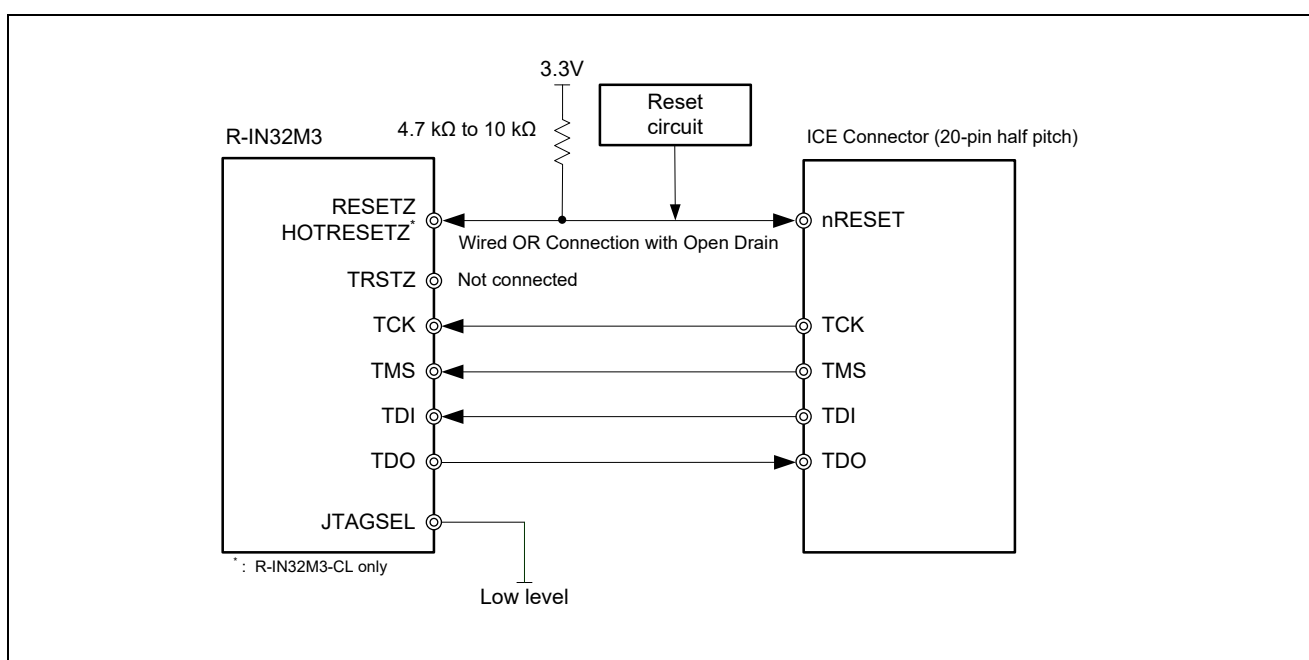


Figure 22.1 JTAG Interface Connection Example (20-Pin Half Pitch without Trace)

Note: 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.

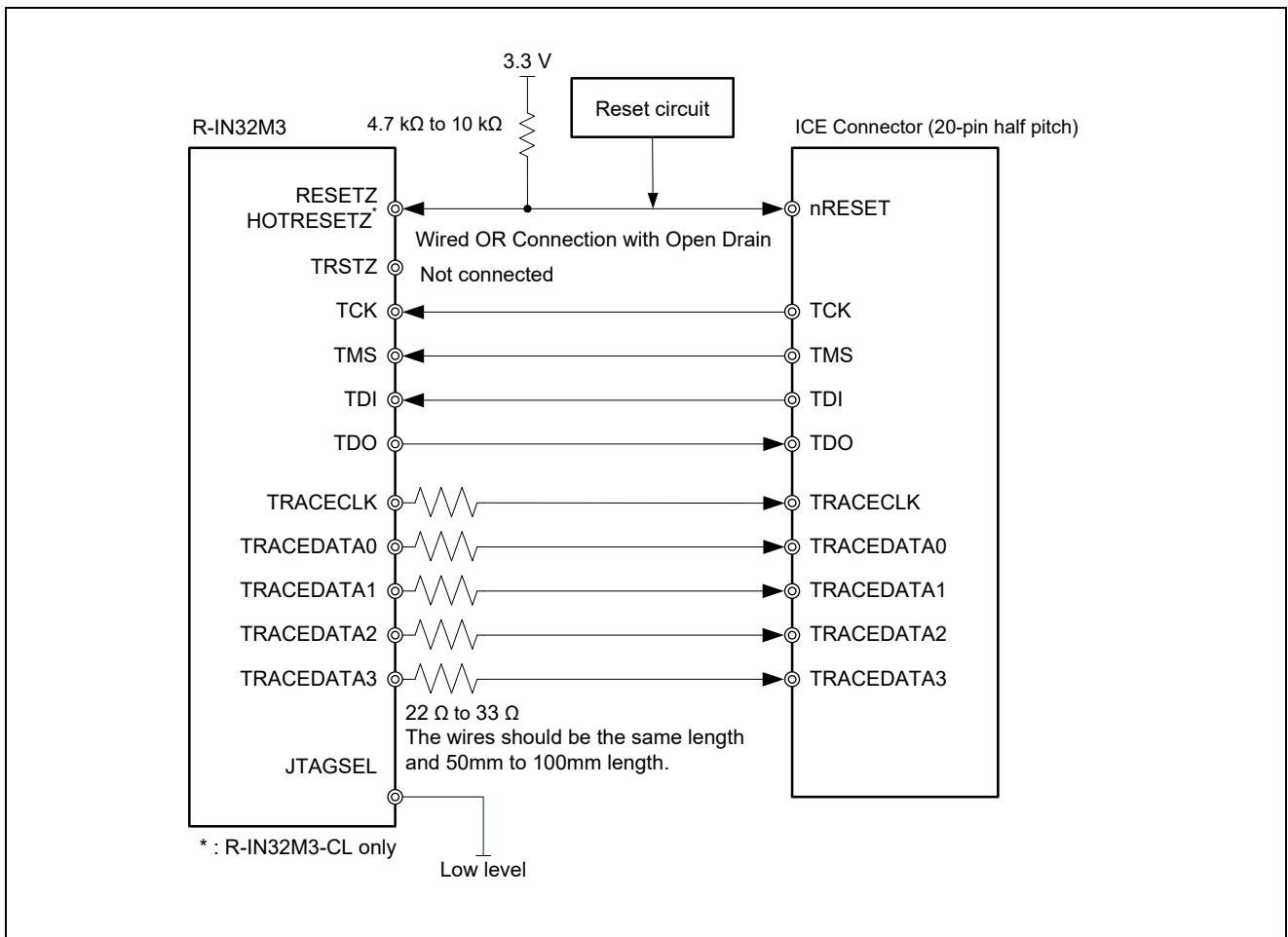


Figure 22.2 JTAG Interface Connection Example (20-Pin Half Pitch with Trace)

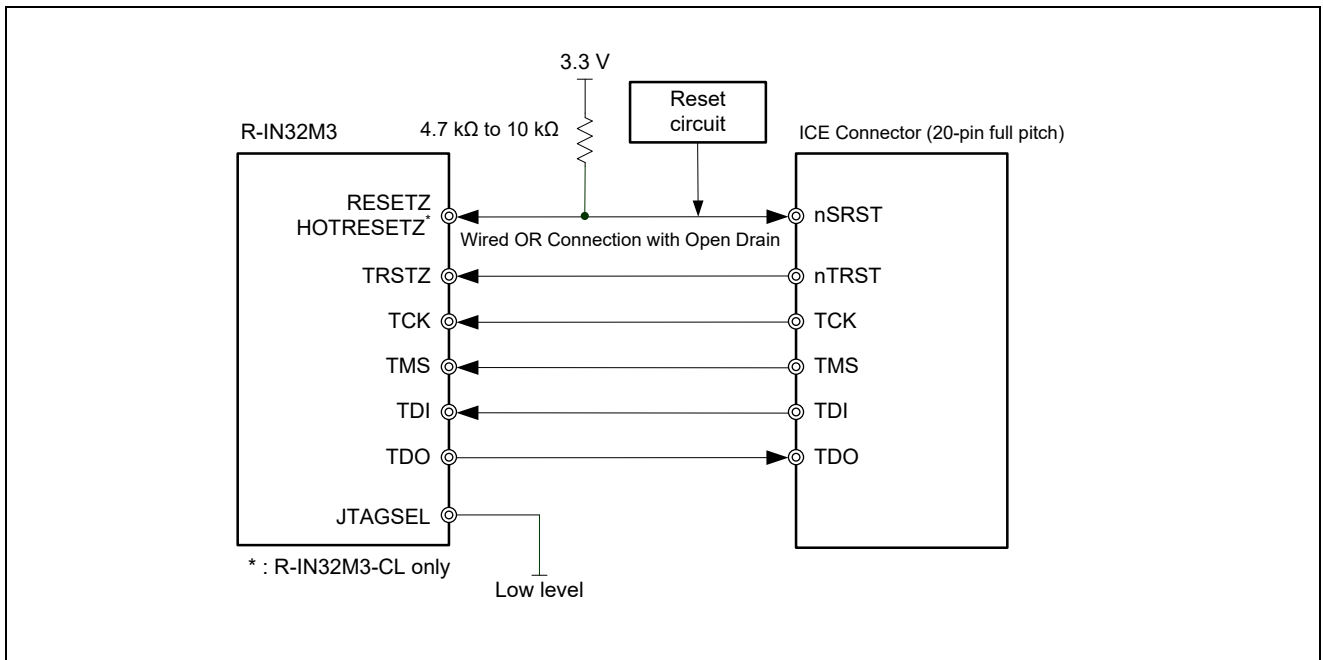


Figure 22.3 JTAG Interface Connection Example (20-Pin Full Pitch)

22.2 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)).

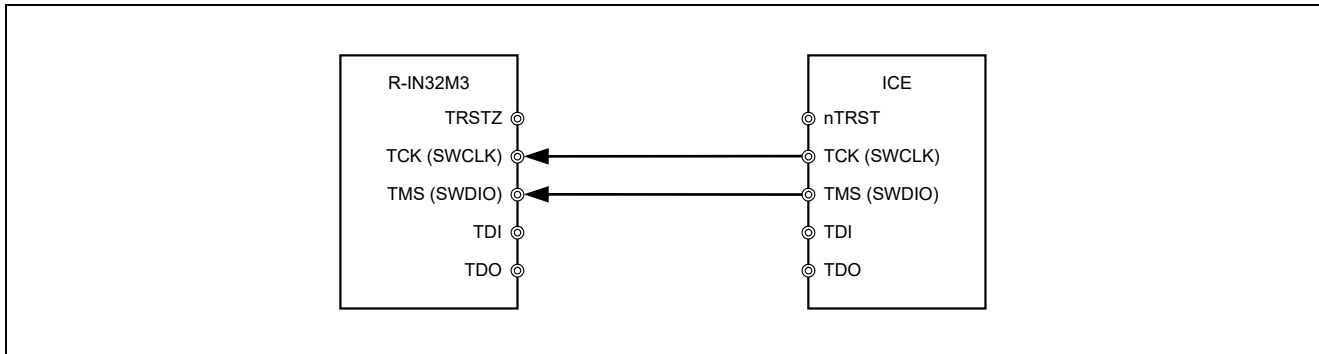


Figure 22.4 SWD Interface Connection Example

22.3 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 find branch sources and destinations. For details about trace information, see the user's manual of the ICE you are using.

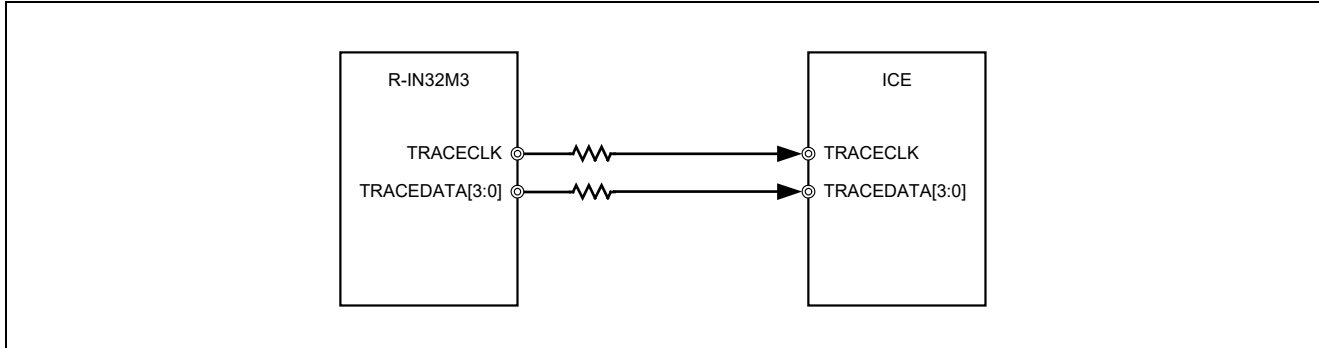


Figure 22.5 Trace Port Interface Connection Example

22.4 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.

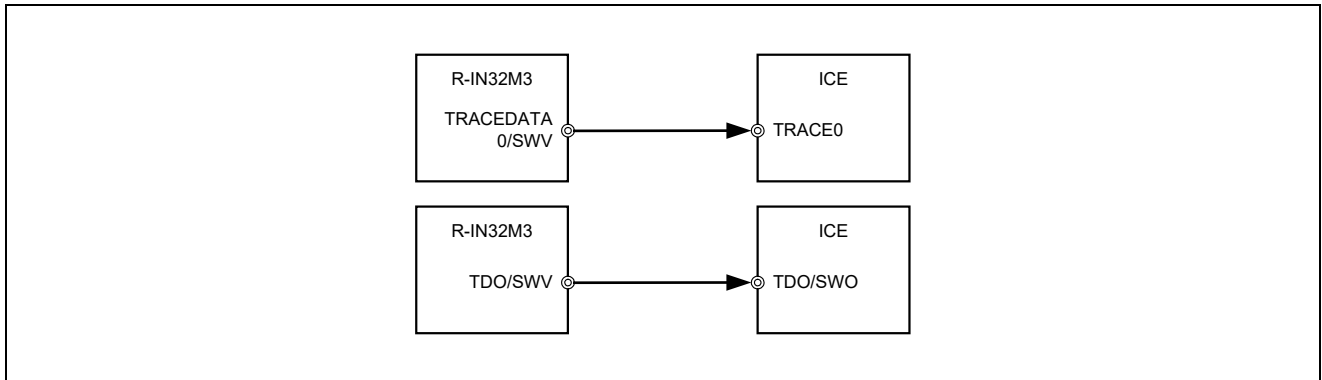


Figure 22.6 SWV Interface Connection Example

Revision History	R-IN32M3 Series User's Manual: Peripheral Modules
------------------	---

Rev.	Issue Date	Description	
		Page	Summary
Draft 1.00	Feb 08, 2013	–	Preliminary version
1.00	Mar 03, 2013	Overall	Change the description of "Timer Array Unit" "TAUJ" → "TAUJ2"
		4	Delete standby mode of 2.2 Clock Control Function
		5	Modification of CLKGTD0 register of 2.2.3 Clock control registers
		9	Addition of SYSRESET and SFTRST register of 2.3.3 Reset control registers
		19	Addition of Instruction RAM boot of 5.1 Selecting the boot mode
		43	Modification of WREN register Address of Table 9.1 The register outline of the Bus Control Functions
		52	Modification of WREN register Address of 9.3.5 Write Enable Switch Register
		73	Addition of Note 1 of 10.2.4 Synchronous burst access MEMC operation setting register
		78	Modification of 10.2.7 Synchronous burst access MEMC mode setting register
		93	Modification of T_RC value of Figure 10.4 Asynchronous SRAM, separate bus mode, read access, ADVZ enabled
		101	Modification of RDZ signal of Figure 10.12 Synchronous SRAM, multiplexed bus mode, read access, ADVZ enabled
		106	Modification of BENZ signal of Figure 10.17 Synchronous SRAM, multiplexed bus mode, burst write access (4-beat), ADVZ enabled
		108	Modification of reset release information of 11. External MPU Interface
		110	Addition of "Caution" of 11.2 Synchronization / Asynchronous SRAM Interface Mode
		112	Modification of the area's name of Figure 11.2 Bus sizing function
		113	Limitations when using CC-Link IE Field Network are added to 11.2.2 Operation
		149	Modification of SFMDTY bit of 12.2.3 Clock Control Register
		182-218	Modification of DMA Controller Register base address of 13.4.3 General DMA Controller Register Set
		198	Modification of SBE bit function of CHCFGn register of 13.4.3.2 Current Register Set
		219-252	Modification of DMA Controller Register base address of 13.4.4 Register set of DMA controller for Real-time ports
231	Modification of RTCHCTRL register name of 13.4.4.2 Current Register Set		
253	Addition of Caution 7 of 13.4.5 DMA Transfer Interface Signal Control Registers		
254,255	Modification of Address of DMAIFC0, DMAIFC1, RTDMAIFC registers of 13.4.5 DMA Transfer Interface Signal Control Registers.		
264	Modification of Interrupt output mask register of Table 13.9 DMA Controller Interrupt Output		
264	Modification of Interrupt output mask register of Table 13.10 Interrupt Output of DMA Controller for Real-time ports		

Rev.	Issue date	Description	
		Page	Points
1.00	Mar 03, 2013	295	Modification of Table13.25 Specification of the Detection Mode for each DMA Transfer Request Source
		295	Modification of DMA transfer request source of Table23.25 Specification of the Detection Mode for each DMA Transfer Request Source
		334	Addition of "Delay count function" of 14.3.1 Timer operation function
		374	Addition of "Delay count function" of 14.14.3 Delay count function
		456	Addition of "14.9 Notes"
		457	Addition of "Refer to notes" to 15 Window watchdog timer A
		457	Addition of "Connected to" of WDTA0TNMI of Table 15.3WDTA interrupts and reset outputs
		459	Modification of "the MAX value of overflow" of 15.3 Functional Overview
		469	Addition of "15.6 Notes"
		458	Delete of VAC mode of 15.3 Functional Overview
		529	Modification of CSIH0MCTL0 register of Table 17.7 CSIH0 register overview
		529	Modification of CSIH1MCTL0 register of Table 17.8 CSIH1 register overview
		542	Modification of CSIHnMCTL0 register of 17.3.1 CSIH register details
		658	Modification of Figure 18.1 Block diagram of IICBn
		784	Modification of flowchart of Figure 18.2 Continuous transfer mode setting sequence when communication reserve function is disabled
		787	Modification of Bus name of 19.1 FCN Features of R-IN32-M3
		790	Modification of Time stamp function Table 19.6 Overview of functions
		902	Addition of FCNnGMCLECCF bit of Figure19.15 Re-initialization without using the software reset function
		935	Addition of CCSMON register and CCSRUN register of 20.1.1 Register List
		935	Modification of CC-Link function of 20.1.2 CC-Link bus size control register.
		935	Modification of CC-Link function of 20.1.3 CC-Link bus bridge control register 0
		935	Modification of initial value of 20.1.3 CC-Link bus bridge control register 0
		936	Modification of initial value of 20.1.4 CC-Link bus bridge control register 1
		937	Addition of 20.1.5 CC-Link monitor register
		937	Addition of 20.1.6 CC-Link Slave RUN LED control register
		938	Modification of CC-Link function of 20.1.7 CC-Link reset register
		940	Modification of Access bit unit of IDCODE register of 21.1 Registers
		940	The following registers are deleted of 21.1 Registers REMAP, WREN, SMADSEL0-3, BCLKSEL, SMC352MD, CLKGTD0-1, PSAV, DRCTL*
		943	Modification of Address of 21.2 Operating mode monitor register
		943	Addition of ADMUXMODE bit and MEMCSEL bit of 21.2 Operating mode monitor register
943	Addition of Mode bit of 21.2 Operating mode monitor register		
944	Modification of Address of 21.4 Version register		

Rev.	Issue date	Description	
		Page	Points
1.00	Mar 03, 2013	946	Modification of Function of 21.6 CPURESET register
		962	Modification of initial value of INTM1 and INTM2 register of 21.9.3 External interrupt mode registers
		964	Modification of Address of RPTRGMD register of 21.10.1 Trigger-synchronous port control mode register
		964	Addition of Note of 21.10.1 Trigger-synchronous port control mode register
2.00	May 10,2013	2	Addition of note1 and note2 of Description of internal clocks
		3	Modification of BUSCLK of Figure2.1 Clock configuration diagram
		31	Addition of note2 and note3 of 7.1.1 MAC select register
		72	Addition of Duty ratio of 10.2.3 BUSCLK division setting register
		76	Addition of note of 10.2.6 Cycle setting register
		108	Addition of the following matters of 11. External MPU Interface Access over the 16-byte boundary of PageROM read-out is forbidden Synchronous SRAM burst access is forbidden
		328	Delete of Gate Count mode of 14. Timer Array Unit Overflow Function of 14. Timer Array Unit
		447	Delete of WDTA Start-up Options
		654	Modification of IICBnLGDF bit function of 18.3 IICBn control register 1
		658	Modification of Data hold time of Table18.5 Conditions for generating serial output timing
3.00	JUN 28,2013	32	Addition of note 7.1.2 MAC clock selection register
		325-425	Modification of All of 14. Timer Array Unit
		491	Addition of Example of Baud rate of Table 16.9 Example of baud rate generator settings
3.01	DEC 9,2013	902-906	Modification of supported station of 20. CC-link Function
4.00	FEB 17,2014	10	Add chart of Rest Behavior
		46	Delete cash description of 9.3.3 Static Memory Control Registers
		52	ADD write protect description of 9.3.5Write Enable Switch Register
		294	Modification of Chanel number of 13.8.3DMA Transfer Request
		438	Addition of Baud rate setting of 16.1 R-IN32M3 UARTJn Features
		916	Addition of 21.8 HW-RTOS reset register
		939-941	Addition of Figure 22.1,22.2,22.3 for JTAG interface
5.00	APR 18,2014	326	Modification of Table14.2TAUJ2 I/O signals
		501	Modification of CSIHnCTL1 - CSIH control register 1
		930	Modification of 21.10.3 External interrupt mode register
		945	Modification of Figure 22.4 SWV interface connection example
		32-69	Modification and Addition of 7.Gigabit Ethernet MAC

Rev.	Issue date	Description	
		Page	Points
5.00	Apr 18, 2014	1	Add description of base addresses to "1. Introduction"
		5	Modify register address of "2.2.2 Clock control registers (CLKGTD0, CLKGTD1)"
		78	Modify initial value of "9.3.2 Bus Size Control Register (BSC)"
		104,105	Modify register address of "10.2.2 Synchronous burst access MEMC control registers (SMADSEL0 to SMADSEL3)"
		311	Modify register address of "DMAC error status register (RTDSTER)"
		556-560	Modify register address of 17.3 CSIH Control Registers
		819	Modify initial value of FCNnDNBMRXk
		943	Modify register name of "21.1 Registers"
		937-942	Modify register address and name of "20.1.1 Register List"
		836	Modify "Figure 19.2 FCN module clock"
		473	Modify "Table 16.3 UARTJn I/O signals"
6.00	May 30, 2014	36	Add R1 Register of "7.3.1 Register list"
		56	Add R1 Register of "7.3.5.3 Hardware function return value register"
		57	Add R1 Register of "Figure 7.3 Hardware function call issue processing flow"
		58	Modify R0,R1 Register of "7.4.1.2 Get the buffer for transmit"
		62	Add R1 Register of "7.4.1.6 Completion of transmit"
		63	Add R1 Register of "7.4.2.3 Wake up Rx DMA"
		123	Add the detail description of "8. Ethernet Switch"
6.01	Dec 25, 2014	7	2.3.2 "Features" Modify The low-level width of each reset to 1ms. Add the notice of in case of ICE half pitch connector
		21	Modify the description for BOOT pin setting in figure at "5.3 Memory map in each boot mode"
		36, 53, 63, 64	Modify Register name and description of "7.3.4.18 Receive Buffer information register "
		52	Modify the initial value to 0000 080FH at 7.3.4.17 GMAC_LPI_TIMING
		230	Modify the area name and address of " Figure 11.1 External MPU interface area"
		984	Add a condition for CAN baud rate in "19.13.1 Baud rate setting conditions"
		1030	Modify resister description of "21.2 Operating mode monitor register (MDMNT)"
		1035	Modify resister description and initial value of "21.6 CPURESET register"
		1064	Add setting of JTAGSEL pin of "22.1 JTAG interface"
7.00	Aug 31, 2015	228	Cautions added as a supplementary explanation
		233	Description of the synchronous mode, divided into separately for reading and writing
		235-237, 239-240	According to the change of AC characteristics in the datasheet, timing charts were changed and added.
8.00	Jan 15, 2016	1-1	1. Introduction, the second and subsequent paragraphs moved and modified to "1.2 Base Addresses of the System Registers Area"
		1-1	1.1 Type Names of R-IN32M3-Series Products, added
		2-5	2.2.2 Clock Control Registers (CLKGTD0, CLKGTD1), initial value of CLKGTD1 modified, Note 3 added

Rev.	Issue date	Description	
		Page	Points
8.00	Jan 15, 2016	2-6	2.3.1 Overview, Note added to HOTRESETZ
			2.3.2 (1) Reset by signal input from a pin, part of description deleted
			2.3.2 (3) Reset by using software, RESETZ pin corrected to HOTRESETZ pin
		2-7	Table 2.1 Reset Source and Targets to be Reset, Note 2 added to CATRESET register
		2-8	2.3.3 (1) System Reset Register (SYSRESET), HOTRESETZ input pin in the text corrected to HOTRESETZ input pin
		2-9	Figure 2.2 Timing of Reset at Power On, RSTOUTZ waveform added
		4-2	4.1 Bus Occupancy by the Cortex-M3, added
		8-3	8.3.1 (1) Management registers, Ethernet switch 10-Mbps half-duplex mode setting register added
		8-9	8.3.2.4 Ethernet Switch 10-Mbps Half-Duplex Mode Setting Register, added
		8-73	Table 8.7 Typical HUB MAC Filter Setup, entries under "Notes" corrected: HUB_FLT_MACnlo → HUB_FLT_MAC6lo HUB_FLT_MACnhi → HUB_FLT_MAC6hi
		11-1	Table 11.1 Mode of the External MCU Interface Selected by the Level on the Operating Mode Setting Pin, added
			Note added below Table 11.1
		11-3	Figure 11.1 External MCU interface Space, modified
			Notes 1 to 3 added below Figure 11.1
		14-2	Table 14.3 TAUJ2 Interrupt Signals, modified
		14-3	Table 14.4 TAUJ2 operation functions, External Event Count Function added
		14-15... 14-18, 14-74	14.3.3 (3) TAUJ2CMORM - TAUJ2 channel mode OS register, entries under "Function" column for Bit[13:12] in the table modified, description of event count mode added
		15-4	15.3.2 (2) WDTA Mode Register (WDTAnMD), bit names of WDTAn*** unified to WDTA0***
		17-2	Table 17.5 CSIHn interrupt and DMA/DTS requests, entry added to the table
		18-120	Figure 18.16 Master operate setting sequence during continuous transfer mode (single master environment), modified
		19-2	Table 19.4 FCNn interrupts and DMA requests, modified
19-114	Table 19.20 Representative examples of baud rate settings (fCANMOD = 20 MHz), title and entries in the table modified		
21-2	21.1 Registers, modified		
	21.1 Registers, CPU bus mode register, added		
21-4	21.4 Version register (RINVER), modified, remark added		
21-6	21.6 CPURESET register (CPURESET), symbol CPURESET added, description modified		
21-31	21.12 CPU Bus Operating Mode Register (CPUBUSMD), added		

Rev.	Issue date	Description	
		Page	Points
9.00	May 20, 2016	3-3	3.4.2 Read Buffer , 2-bit ECC error handling, modified
		4-1	4. Bus Architecture, Table 4.1 , CC-Link as slave, added
		7-5	7.3.1(4) Hardware function call registers , CMD, CNTX_TYPE0 and CNTX_STAT0 register added
		7-6	7.3.2.1 MAC Select Register (MACSEL) , MAC2 to MAC0 setting value (010→011), modified
		7-13	7.3.4.6 TX Mode Register (GMAC_TXMODE) , Note for LPTXEN bit, added
		7-25 to 7-29	7.3.5 Hardware Function Call Register , description for existence register, complemented and new register, added
		7-30 to 7-54	7.4.1 Hardware Functions , Newly added
		7-55 to 7-56	7.4.2 Interrupts , Newly added
		7-65	7.4.4.5 Rx Data Format, Figure 7.16 , Newly added
		9-2	9.2(1)(a) SRAM and external I/O connection , cycles for idle wait, modified
			9.2(1)(b) Page ROM connection , cycles for idle wait, modified
		10-42	10.4.2 Synchronous Access Timing , Caution added
		11-1	11. External MCU Interface , Caution added
		11-29	11.3.4(2) HOSTIF synchronous SRAM control register 0 (HIFEXT0) , Caution for Reserved bit, added
		11-30	11.3.4(3) HOSTIF synchronous SRAM control register 1 (HIFEXT1) , Caution for Reserved bit, added
		12-25	12.4.6(3) SPI Bus Cycle Generation in Direct Communications Mode , Caution for usage of non-support mode, added
		13-4	13.1.1 Overview, Table 13.3 , Caution for transfer size, added
		16-9	16.4(2) UARTJn control register 1 (URTJnCTL1) , URTJnBLG0 setting value of BF length, modified
		16-15	16.4(5) UARTJn status register 0 (URTJnSTR0) , Note for each bit setting disabled, added
		16-16 to 16-17	16.4(6) UARTJn status register 1 (URTJnSTR1) , Note for each bit setting disabled, added
		16-31	16.5.3 Status Interrupt Request INTUAJnTIS, Figure 16.4 , Overrun error judgement flow, modified
		16-46	16.6.6(2) Reception start and stop, Figure 16.15 , Pointer value setting, complemented
		16-53 to 16-54	16.7 Bit-Rate Generator , Calculation method of error in the bit rate, added
		18-14	18.3(6)(a) Setting transfer clock by using IICBnWL and IICBnWH registers , Calculation method of transfer clock, modified
18-117	18.9.1(1) Master operation setting procedure during single transfer mode, Figure 18.14 , Branch after the final data transmission completed, modified		
18-121 to 18-122	18.9.2(1) Single transfer mode setting procedure when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0), Figure 18.18 , Branch of reference symbol in flowchart, modified		

Rev.	Issue date	Description	
		Page	Points
9.00	May 20, 2016	18-123	18.9.2(2) Single transfer mode setting procedure when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1), Figure 18.19 , Branch after start communication, modified
		18-125	18.9.2(3) Continuous transfer mode setting procedure when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0), Figure 18.20(1/2) , Branch after start communication, modified
		18-126	18.9.2(3) Continuous transfer mode setting procedure when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0), Figure 18.20(2/2) , Branch after interrupt generation, modified
		18-127	18.9.2(4) Continuous transfer mode setting procedure when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1), Figure 18.21(1/2) , Branch after start communication, modified
		18-128	18.9.2(4) Continuous transfer mode setting procedure when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1), Figure 18.21(2/2) , Processing after reference symbol E, deleted
		19-106	19.14.1 Initialization, Figure 19.14 , Insufficient processing, added
		19-122	19.14.2 Message Transmission, Figure 19.28 , Insufficient processing, added
		21-2	21.1 List of Registers , SRAM bridge select register, added
		21-32	21.15 SRAM Bridge Select Register (SRAMBRSEL) , Newly added
10.00	Feb 28, 2017	2-2	2.1.2 Clock Configuration Diagram Register symbol corrected WDTATCKI pin added
		3-3	3.4.1 Outline of Features , description and list of ECC error interrupts added
		3-4	3.5.1 Outline of Features , description and list of ECC error interrupts added
		3-5	3.6.1 Outline of Features , description and list of ECC error interrupts added
		7-9	7.3.4.1 MIIM Register (GMAC_MIIM) Description of the RWDV bit of the MIIM register added
		7-11	7.3.4.3 TX Result Register (GMAC_TXRESULT) Description of the GMAC_TXRESULT register added
		7-12, 7-13	7.3.4.5 RX Mode Register (GMAC_RXMODE) Description of the GMAC_RXMODE register corrected
		7-14, 7-15	7.3.4.6 TX Mode Register (GMAC_TXMODE) Description of the GMAC_TXMODE register corrected
		7-16	7.3.4.7 Reset Register (GMAC_RESET) Description of the GMAC_RESET register corrected
		7-18	7.3.4.9 RX Flow Control Register (GMAC_FLWCTL) Description of the GMAC_FLWCTL register corrected
		7-19	7.3.4.10 Pause Packet Register (GMAC_PAUSPKT) Description of the GMAC_PAUSPKT register modified
		7-21	7.3.4.12 RX FIFO Status Register (GMAC_RXFIFO) Description of the RRT bit of the GMAC_RXFIFO register corrected
		7-22	7.3.4.13 TX FIFO Status Register (GMAC_TXFIFO) Description of the GMAC_TXFIFO register modified

Rev.	Issue date	Description	
		Page	Points
10.00	Feb 28, 2017	7-23	7.3.4.14 TCPIPACC Register (GMAC_ACC) Description of the RTCPEN bit of the GMAC_ACC register modified
		7-24	7.3.4.16 LPI mode control register (GMAC_LPI_MODE) Description of the GMAC_LPI_MODE register added
		7-25	7.3.4.18 Receive Buffer Information Register (BUFID) Description of the BUFID register added Method of calculating the start address of the received frame information, description modified
		7-31	Figure 7.3 Schematic Block Diagram of the Hardware Functions AHB2DMA bus bridge added
		7-32	7.4.1.1 Initial Settings Step added to the flow of initial settings
		7-33	7.4.1.3 (1) Functional Overview Operation when an unsecured buffer area is accessed added
		7-36	7.4.1.3 (2) (e) List of hardware function calls Error source of a hardware function call of the buffer allocator added
		7-39	7.4.1.3 (2) (e) Table 7.5 HWFNC_Buffer_Return Description of return values of HWFNC_Buffer_Return modified
		7-41	7.4.1.4 (2) DMA for the Reception MAC The maximum pieces of Rx information storable in BUFID corrected
		7-42	7.4.1.4 (2) (a) Description of the Individual functions of the MAC DMA controller Description of the individual functions of the Rx MAC DMA controller modified
		7-43	7.4.1.4 (2) (a) Figure 7.10 Conceptual Diagram of Judging Whether a Received Frame is Valid or Invalid RX Frame Control corrected to RX Frame Information and unused bits corrected to Reserved
		7-44	7.4.1.4(2)(b) Usage , Bit name corrected
		7-45	7.4.1.4(2)(c) List of hardware function calls Description of R7 of HWFNC_MACDMA_RX_Enable corrected
		7-46	7.4.1.4(2)(c) List of hardware function calls Description of R7 of HWFNC_MACDMA_RX_Disable corrected
		7-47	7.4.1.4 (2) (c) Table 7.9 HWFNC_MACDMA_RX_Errstat Description of return values of HWFNC_MACDMA_RX_Errstat corrected
		7-50	7.4.1.4 (3) (d) Table 7.10 HWFNC_MACDMA_TX_Start The maximum transmission size of HWFNC_MACDMA_TX_Start corrected
		7-51	7.4.1.5 (2) (a) Transfer between the buffer RAM and the data RAM Description of transfer between the buffer RAM and the data RAM corrected
		7-51	7.4.1.5 (2) (b) Replacing data in the buffer RAM or data RAM Description of data replacement in the buffer RAM or data RAM added
		7-51	7.4.1.5 (2) (c) Transfer between the buffer RAMs Description of transfer between the buffer RAMs added
		7-52	7.4.1.5(2)(d) List of hardware function calls Hardware Function Call name corrected

Rev.	Issue date	Description	
		Page	Points
10.00	Feb 28, 2017	7-53	7.4.1.5(2)(d) List of hardware function calls Description of HWFNC_Direct_Memory_Replace added
		7-56	7.4.2 Table 7.16 Interrupts Related to Operations for Transmission Description of the TX-FIFO error interrupt corrected
		7-58	7.4.2 Table 7.18 Interrupts Related to Other Operations Description of interrupts corrected
		7-60	7.4.3.1 Acquiring a Transmit Buffer Description of return values of R0 corrected
		7-61	7.4.3.2 Creating TX Data Allocation of Tx frame control information and Ethernet frame data shown in figure
		7-63	7.4.3.2 (1) Tx frame control information ICRC and APAD of Tx frame control information modified Note2 added in TCPIP ACC OFF
		7-63	7.4.3.2 (1) Tx frame control information The formula for the transmission size of Tx frame control information corrected
		7-64	7.4.3.2 (2) Ethernet frame The transmission Ethernet frame data format modified
		7-65, 7-66	7.4.3.2 (2) Ethernet frame Patterns of the transmission Ethernet frame data format added
		7-67	7.4.3.3 Creating TX Descriptors Restrictions on Tx descriptors deleted
		7-68	7.4.3.5 Completion of Transmission Description of interrupt generation on the completion of transmission added
		7-69	7.4.4 Receiving Ethernet Frames Reference number corrected
		7-70	7.4.4.5 Rx Data Format Description of alignment of the Rx data format modified
		7-70	7.4.4.5 Rx Data Format Allocation of Ethernet frame data and Rx frame information shown in figure
		7-71, 7-72	7.4.4.5 (1) Rx frame information Name of the FIFOFULL field corrected to FIFOOVF Description of the fields of Rx frame information modified Note2 added
		7-72	7.4.4.5 (1) Rx frame information Note on the number of received bytes of Rx frame information modified
		7-73	7.4.4.5 (2) Rx Ethernet frame Description of the Rx Ethernet frame format modified
		7-74	7.4.4.5 (2) Rx Ethernet frame Caution on recovery of the destination MAC address of the frame received while the management tag is enabled added
		7-75 to 7-77	7.4.4.5 (2) Rx Ethernet frame Patterns of the Rx Ethernet frame data format added

Rev.	Issue date	Description	
		Page	Points
10.00	Feb 28, 2017	7-78, 7-79	7.4.5 TCPIP accelerator function Description of the TCPIP accelerator function newly added
		7-80	7.5.1 Appending Padding to the MAC Header Section within the TX Frame Padding to the MAC header section within the Tx frame modified
		7-80	7.5.2 Erroneous Judgment about Checksum Validation at Specific Packet Reception Precaution on the Rx TCPIP accelerator added
		7-80 to 7-84	7.5.3 Error of Rx Frame Information at RX FIFO Overflow Precaution and workaround on Rx FIFO Overflow added
		7-84, 7-85	7.5.4 Error of Rx Frame Information at Reception of the Frame more than 64 bytes with padding Precaution and workaround on receiving the Frame more than 64 bytes with padding added
		8-2	8.2 Characteristics Interrupt and I/O signals of Ethernet Switch added
		8-6	8.3.2.1 Ethernet PHY LINK Mode Register (ETHPHYLNK) Positions of 0 and 1 for description of the bits of the ETHPHYLNK register corrected
		10-1, 10-2	10.1 Features Notations of the pins unified Duplicate description deleted
		10-3	10.2 Control Registers Register names and symbols corrected
		10-4, 10-5	10.2.1 Wait Signals Selection Register (WAITZSEL) Register name modified Notations of the pins unified Description of the WSEL0n to WSEL3n bits corrected
		10-6, 10-7	10.2.2 Synchronous Burst Access Memory Controller Area Select Registers (SMADSEL0 to SMADSEL3) Notations of the pins unified Description in cautions modified Remark 2 added
		10-8	10.2.3 Bus Clock Division Setting Register (BCLKSEL) Register name modified Description modified Description in caution 2 modified
10-9	10.2.4 Synchronous Burst Access Memory Controller Operation Mode Setting Register (SMCMD) Section title and register symbol corrected Description of the SMCLKTH bit corrected Description in caution 2 modified Notations of the pins unified		

Rev.	Issue date	Description	
		Page	Points
10.00	Feb 28, 2017	10-10	<p>10.2.5 Synchronous Burst Access Memory Controller Direct Command Register (DIRECTCMD)</p> <p>Section title, register name, and register symbol corrected</p> <p>Remark added with correction of the register symbol</p> <p>Notations of the pins unified</p>
		10-11	<p>10.2.6 Synchronous Burst Access Memory Controller Cycle Setting Register (SETCYCLES)</p> <p>Section title, register name, and register symbol corrected</p> <p>Notations of the pins unified</p>
		10-12	<p>10.2.6 Synchronous Burst Access Memory Controller Cycle Setting Register (SETCYCLES)</p> <p>Caution on the T_WC and T_RC bits moved to Note 2</p> <p>Description of the T_CEOE, T_WC, and T_RC bits modified</p> <p>Notations of the pins unified</p>
		10-13, 10-14	<p>10.2.7 Synchronous Burst Access Memory Controller Mode Setting Register (SETOPMODE)</p> <p>Section title, register name, and register symbol corrected</p> <p>Description of the ADV bit corrected</p> <p>Notations of the pins unified</p> <p>A point to note on the WR_BL and RD_BL bits moved below the table as Note.</p>
		10-15	<p>10.2.8 Synchronous Burst Access Memory Controller Refresh Setting Register (REFRESH0)</p> <p>Section title, register name, and register symbol corrected</p>
		10-15	<p>10.2.9 Synchronous Burst Access Memory Controller CSZn Cycle Setting Registers (SRAM_CYCLES0_n)</p> <p>Notations of the pins unified</p> <p>Register name and register symbol corrected</p>
		10-16	<p>10.2.10 Synchronous Burst Access Memory Controller CSZn Mode Registers (OPMODE0_n)</p> <p>Register name and register symbol corrected</p> <p>Notations of the pins unified</p>
		10-17	<p>10.2.11 Register Setup Procedure</p> <p>Register symbols corrected</p> <p>Unsupported register (DMCBUFMD) deleted</p> <p>Notations of the pins unified</p>
		10-18	<p>10.3.1 Bus Clock Control</p> <p>Section title and structure changed</p> <p>Register symbol corrected</p> <p>Figure illustrating operation for bus clock masking divided</p> <p>Remark added</p>
		10-19	<p>10.3.2 Address Output</p> <p>External address pin names and address space size corrected</p>
		10-19	<p>10.3.3 Address/Data Multiplexing Feature</p> <p>Table describing the address/data multiplexing feature added</p>

Rev.	Issue date	Description	
		Page	Points
10.00	Feb 28, 2017	10-20	10.3.4 Write Enable Signal (WRZn) Cycle Extension Register symbol corrected Remark added
		10-21	10.3.5 Controlling the Data Read Timing Register symbol corrected Remark added and corrected
		10-22 to 10-24	10.3.6 Wait Signals Control Notations of the pins unified Remark added
		10-26	10.3.8 Switching External Memory Area Mapping Notations of the pins unified Description in caution 4 modified
		10-27	10.4 Memory Access Timing Example , Figure 10.23 added to the table
		10-28 to 10-35	10.4.1 Asynchronous Access Timing Supplementary explanation and remark added to figures 10.7 to 10.14
		10-36 to 10-42	10.4.2 Synchronous Access Timing Supplementary explanation and remark added to figures 10.15 to 10.21
		10-43 to 10-44	10.4.3 Wait Timing Supplementary explanation and remark added to figure 10.22 Figure 10.23 newly added
		12-26 to 12-34	12.5 Example of Configuration Serial flash ROM memory controller setup examples newly added
		13-136 13-137	13.9.1 Setting Example 1 (Register Mode, Single Transfer Mode, and Hardware Trigger) Interrupt symbol corrected
		13-138	13.9.2 Setting Example 2 (Register Mode, Block Transfer Mode, and Software Trigger) CHCFG2 register setting corrected
		13-139	13.9.2 Setting Example 2 (Register Mode, Block Transfer Mode, and Software Trigger) "R/W" in the setting value space corrected to "Setting" Interrupt symbol corrected
		13-140	13.9.2 Setting Example 2 (Register Mode, Block Transfer Mode, and Software Trigger) Interrupt symbol corrected
		13-142	13.9.3 Setting Example 3 (Register Mode: Continuous Execution, Block Transfer Mode, and Software Trigger) "R/W" in the setting value space corrected to "Setting" Interrupt symbol corrected
		13-143	13.9.3 Setting Example 3 (Register Mode: Continuous Execution, Block Transfer Mode, and Software Trigger) Interrupt symbol corrected

Rev.	Issue date	Description	
		Page	Points
10.00	Feb 28, 2017	13-146	13.9.4 Setting Example 4 (Link Mode, Block Transfer Mode, and Software Trigger) Register symbols corrected Unsupported register (DMAESEL) deleted
		13-146	13.9.4 Setting Example 4 (Link Mode, Block Transfer Mode, and Software Trigger) Interrupt symbol corrected
		18-118	18.9.1(2) Slave operation setting procedure during single transfer mode Figure 18.15 Slave Operation Setting Procedure during Single Transfer Mode (Single Master Environment) corrected
		20-2	20.1.3 CC-Link Bus Bridge Control Register 0 (CCSMC0) Bit name "CCSMC" corrected to "CCSMC0"
11.00	Dec. 28, 2018	2-9	Figure 2.2 Timing of Reset at Power On Figure 2.3 Timing of Reset at System Reset Timing charts modified
		5-1	Table 5.1 Selecting the Boot Mode Signal name corrected (STCSZ0 → CSZ0)
		5-1	(1) External memory boot mode Signal name corrected (STCSZ0 → CSZ0)
		7-9	7.3.4.1 MIIM Register (GMAC_MIIM) Caution modified
		7-13	7.3.4.5 RX Mode Register (GMAC_RXMODE) Description of SFRXFIFO (b29) modified
		7-14	7.3.4.6 TX Mode Register (GMAC_TXMODE) FSTTH (b15-14) changed to reserved bits SF (b29): Setting of 0 changed to prohibited setting: Note 2, added
		7-33	7.4.1.2 Flow of Processing for Issuing the Hardware Function Call Figure 7.4 Flow of Processing for Issuing the Hardware Function, modified
		7-34	7.4.1.3 Buffer Allocator, (1) Functional Overview The description on an generation of an exception, modified
		7-37	7.4.1.3 Buffer Allocator, (2) Buffer Control Operation, (e) List of hardware function calls Table 7.2 HWFNC_LongBuffer_Get Return value registers, modified
		7-38	7.4.1.3 Buffer Allocator, (2) Buffer Control Operation, (e) List of hardware function calls Table 7.3 HWFNC_ShortBuffer_Get Return value registers, modified
		7-45	7.4.1.4 MAC DMA Controller, (2) DMA for the Reception MAC, (b) Usage • Procedure for reading and releasing buffers Example modified
		7-46	7.4.1.4 MAC DMA Controller, (2) DMA for the Reception MAC, (c) List of hardware function calls The description modified

Rev.	Issue date	Description	
		Page	Points
11.00	Dec. 28, 2018	7-51	7.4.1.4 MAC DMA Controller, (3) DMA for the Transmission MAC, (d) List of hardware function calls The description modified
		7-52	7.4.1.4 MAC DMA Controller, (3) DMA for the Transmission MAC, (d) List of hardware function calls Table 7.11 HWFNC_MACDMA_TX_Errstat The description on the return value register R0, modified
		7-54	7.4.1.5 Buffer RAM DMA Controller, (2) DMA Transfer, (d) List of hardware function calls The description modified
		7-54	7.4.1.5 Buffer RAM DMA Controller, (2) DMA Transfer, (d) List of hardware function calls Table 7.12 HWFNC_Direct_Memory_Transfer The description on argument registers, modified
		7-58	7.4.2 Interrupts Table 7.16 Interrupts Related to Operations for Transmission The condition to generate an MACDMA transmission error interrupt, that is related to operations for transmission, modified
		7-60	7.4.2 Interrupts Table 7.18 Interrupts Related to Other Operations The buffer RAM area access error was added to interrupts related to other operations
		7-87	7.5.5 Transmitting Data in Cut-Through Mode 7.5.6 Jumbo Frames Above two sections added to section 7.5, Notes
		8-4	8.3.1 (4) MAC Port Registers Address of the transmit IPG length register n (shared), corrected (4007 005CH + 2000H*n → 4007 805CH + 2000H*n)
		8-4	8.3.1 (5) Timer Module Registers Address of the interrupt status/ACK register, corrected (4007 0008H → 4007 C008H) Address of the port timestamp register n, corrected (4007 0024H + 0008H*n → 4007 C024H + 0008H*n)
		8-5	8.3.1 (6) DLR Module Registers Address of the DLR local MAC address high register, corrected (4007 0018H → 4007 E018H)
		8-14	8.3.3.5 Input Learning Blocking Register (INPUT_LEARN_BLOCK) The settings of bits 18 to 16, corrected
		9-17	Figure 9.8 SRAM Read Cycles Signal name corrected (STCSZn → CSZn)
		9-18	Figure 9.9 SRAM Read Cycles (with Wait Settings) Signal name corrected (STCSZn → CSZn)
		9-19	Figure 9.10 SRAM Read Cycles (External Wait Insertion) Signal name corrected (STCSZn → CSZn)

Rev.	Issue date	Description	
		Page	Points
11.00	Dec. 28, 2018	9-20	Figure 9.11 SRAM Write Cycles (with No Wait) ACn3-ACn0=000B/0001B in the description above the chart modified ("no wait" → "(1 wait cycle)")
		9-20	Figure 9.11 SRAM Write Cycles (with No Wait) Signal name corrected (STCSZn → CSZn)
		9-21	Figure 9.12 SRAM Write Cycles (with Wait States) Signal name corrected (STCSZn → CSZn)
		9-22	Figure 9.13 SRAM Write Cycles (External Wait Insertion) Signal name corrected (STCSZn → CSZn)
		9-23	Figure 9.14 Page ROM Read Cycles (Single Transfer) Signal name corrected (STCSZn → CSZn)
		9-24	Figure 9.15 Page ROM Read Cycles (Four Burst Transfer) Signal name corrected (STCSZn → CSZn)
		11-3, 11-4	11.1 Memory Map The instruction RAM area was modified to the instruction RAM mirror area. Note 4 for instruction RAM mirror area, added
		11-17	11.2.5 Control Registers, (2) HOSTIF Bus Control Register (HIFBCC) The instruction RAM area was modified to the instruction RAM mirror area.
		11-18	11.2.5 Control Registers, (2) HOSTIF Bus Control Register (HIFBCC) Table 11.7 Address Range for which Advance Reading and Page ROM Reading are Selectable The instruction RAM area was modified to the instruction RAM mirror area. Caution regarding access to the instruction RAM mirror area while advance reading is enabled, added
		11-20	11.2.5 Control Registers, (4) HOSTIF page ROM control register (HIFPRC) The instruction RAM area was modified to the instruction RAM mirror area.
		12-12	12.3 Connection with Serial Flash ROM The names of R-IN pins in Figure 12.1 were modified
		13-2	13.1.1 Overview "• Skipping" "separation access size" in the description changed to "skip space size"
		13-85	13.4.6 DMA Trigger Source Select Registers (DTFRn, RTDTFR) A note regarding external DMA transfer request inputs that are selected as DMA transfer trigger sources, added
		13-89	Table 13.9 General DMA Controller Interrupt Output Table 13.10 Interrupt Output of DMA Controller for Real-Time Ports The column "Switch between Pulse Output and Interrupt Output", deleted.
		17-38	17.4.2 Master/Slave Connections (1) One master and one slave Figure 17.4 Direct Master/Slave Connection Diagram of connections between one master and one slave, modified

Rev.	Issue date	Description	
		Page	Points
11.00	Dec. 28, 2018	17-38, 17-39	17.4.2 Master/Slave Connections (2) One master and two slaves, (3) CSISOn output control Figure 17.5 Connection between One Master and Two Slaves In the diagram, CSIH pin names were changed and the CSIHnTSSI pin was deleted. CSIH pin names in the description were also changed.
		18-13	18.3 (6) IICBn high-level width setting register (IICBnWH) Table 18.4 Conditions for Generating Serial Output Timing Generation timing of t _{SU:STA} modified; t _{HD:DAT} added in the timing chart
		18-41	18.6.1 Single Transfer Mode (3) Example of communications in single transfer mode (slave reception), <5> Data reception completion processing Unnecessary bit IICBnCTL0.IICBnSLWT was deleted from the description.
		21-7	21.7 System Protect Command Register (SYSPCMD) Caution 2, modified ("to clear this bit to 0" indicates that "0 is set for protection")
		22-1	22. Debugging The recommended in-circuit emulator (ICE), modified
		—	Error corrected, description modified, and contents and expressions adjusted
12.00	May 31, 2024	2-6	2.3.2 Types of Reset Name of the target reset signal in (1) Reset Control by Hardware corrected
		10-19	10.3.2 Address Output External address pin assignment corrected
		10-19	10.3.3 Address/Data Multiplexing Feature External SRAM pin assignment corrected
		10-28 to 10-44	10.4.1 Asynchronous Access Timing to 10.4.3 Wait Timing Address signals in the timing chart corrected
		18-1 to 18-128	18. I2C BUS (IICB) Interrupt signals, IICBTIA _n and IICBTIS _n , renamed to INTIICBnITA and INTIICBnTIS

R-IN32M3 Series User's Manual
Peripheral Modules

Publication Date: Rev.1.00 Feb 08, 2013 (Draft)
Rev.12.00 May 31, 2024

Published by: Renesas Electronics Corporation

R-IN32M3 Series User's Manual Peripheral Modules