REJ09B0067-0110

Everywhere you imagine. RENESAS

32

32176 Group

Hardware Manual

RENESAS 32-BIT RISC SINGLE-CHIP MICROCOMPUTER

M32R FAMILY / M32R/ECU SERIES

Hardware Manual

Before using this material, please visit our website to confirm that this is the most current document available.

Rev. 1.10
Revision date: Jun 20, 2006

RenesasTechnology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Rev.	Date	Description	
		Page	Summary
1.01	Oct 31, 2003	–	First edition issued
1.10	Jun 20, 2006		Add “Before Use The optional Specification for the product of 32176 Group”
		Common in all chapters	Incorrect) A-D Converter Correct) A/D Converter Incorrect) A-D Conversion Correct) A/D Conversion Incorrect) D-A Converter Correct) D/A Converter Incorrect) D-A Conversion Correct) D/A Conversion Incorrect) Serial I/O Correct) Serial Interface
		1-2,4	Remove the Table from 1.1.2 to replace 1.1.1 and add Note 1
		1-17	Correct Pin No. 135 and 136 lines in the table 1.4.1
		2-4	Add Note 1 to IE bit in the PWS register
		3-4 to 3-6	Combine Figure 3.1.1 to Figure3.1.3 with Figure 3.2.1 to Figure 3.2.3
		3-19	Add H’0080 0600 to H’0080 0603 Dummy access area (note 1)
		3-31	Add Note 1
		3-35	Add descriptions for Dummy access areas
		5-6	Correct notes for IMASK register
		5-7	Add a note to SBICR register
		5-12,14	Combine Table 5.4.1 with Table 5.5.1
		5-15	Add a note to [6] Enabling multiple interrupts
		5-17	Correct Note 2 and Note 5 in the Figure 5.5.2
		5-18	Add a sentence to 5.6.1
		6-7	Add a description to FAENS bit in FMOD register
		6-9	Correct descriptions in (1) FENTRY bit
		6-13	Add Note 1 to Flash Control Register 4
		6-17	Add precautions to be observed after boot mode start
		6-19,21	Add Note 1 to Figure 6.5.2 and Figure 6.5.4
		6-26	Add “Procedure for switching to nomal mode” to Figure 6.5.7
		6-28	Add Note 2 to Figure 6.5.9
		6-30	Add Note 2 to Figure 6.5.10
		6-31	Add Note 2 to FIgure 6.5.11
		6-48	Correct Table 6.7.1: Incorrect) pull low Correct) pull low (0 –100kΩ)
		6-49	Add SBI# pin and Note 1 to Figure 6.7.1
		6-50,51	Add pages 6.8 Connecting to a Serial Programmer (UART Mode)

Rev.	Date	Description	
		Page	Summary
1.10	Jun 20, 2006	6-52	Add descriptions to (2) Protection by FP pin
		6-53	Add 6.10 Notes on the Internal RAM
		8-9	Add Note 2 to P70MOD bit in the P7MOD register
		8-15	Add precaution to 3) Method for using XSTAT to detect XIN oscillation stoppage
		8-16	Add Note to Figure 8.3.1
		8-18,19	Replace Figure 8.4.1 Correct PG67LEV register Incorrect) VT26EL0 Correct) VT6SEL0 Incorrect) VT26EL1 Correct) VT6SEL1
		8-20 to 8-23	Add Note 2 to Port Peripheral Circuit Diagrams
		8-20	Add P130 – P137 (TIN16 – TIN23) to FIgure 8.5.1
		8-25	Add “About the peripheral function input when it is set to the general purpose port”
		9-11	Correct descriptions of (2) TREQFn bit and (4) TENLn bit
		9-31	Add “DMA interrupt related registers” to Table 9.4.1
		Chap. 10	Replace diagrams of Timing Describe reload timing of reload registers
		10-6	Replace Figure 10.1.3
		10-96	Change the descriptions of Reload register updates in TIO PWM output mode Replace Figure 10.4.10
		10-96,97	Add Figure 10.4.11 Update timing of PWM period and descriptions
		12-8	Change the description of (4) Notes on using transmit interrupts
		12-22	Add Table 12.2.1 and Table 12.2.2
		12-40	Add the precaution of switching from general-purpose to serial interface pin
		12-55	Replace Figure 12.7.5
		12-56	Replace Figure 12.8.1
		12-57	Add the precaution of switching from general-purpose to serial interface pin

Rev.	Date	Description	
		Page	Summary
1.10	Jun 20, 2006	13-17	Add descriptions to notes of LBM bit Add descriptions to RST bit
		13-20	Add descriptions to CRS bit
		13-75	Add Note 1 to Figure 13.3.4
		13-81, 84 90,95	Replace Figure 13.5.2, 13.6.2, 13.7.2, and 13.8.2
		13-85	Correct the note of clearing TRFIN bit
		15-3	Correct descriptions of (9) Hold control
		15-4	Add Note 2 to P70MOD bit in P7Mod register
		15-5	Add Note 1 to BUSMOD bit in BUSMODC register
		17-2	Add Figure 17.1.1 RAM Backup Area
		18.2	Replace Figure 18.1.1
		18-3	Change the descriptions of 18.1.2 and replace Figure 18.1.2
		18-4	Add a centence to 3) of XSTAT bit Add Note to Figure 18.1.3
		18-6	Add Note 1 to Figure 18.1.4
		20-2	Add Note 1 to Figure 20.1.1
		20-3 to 20-7	Correct Notes in Figure 20.2.1 to Figure 20.3.3
		20-8	Add Note 1 to Figure 20.3.4 and Figure 20.3.5
		21-20 to 21-53	Change the configuration of A.C. Characteristics
		21-20	Add tc(XIN)[119], tw(RESET)[124], tw(XINH)[120], tw(XINL)[121], tr(XINL)[122], and tr(XINL)[123] Add Figure 21.8.1 Clock and Reset Timing
		21-27	Correct tw(BLWL) tw(BHWL)[51] Add td(CSL-RDL)[93]
		21-30	td(CSL-RDL)[93], td(BLEL-RDL) td(BHEL-RDL)[136], and tv(RDH-BLEL) tv(RDH-BHEL)[137]
		21-31	Add Figure 21.8.12 Read Timing (Byte Enable Mode)
		21-32	Add td(CSL-WRL)[96]

Rev.	Date	Description	
		Page	Summary
1.10	Jun 20, 2006	21-37	Add tc(XIN)[119], tw(RESET)[124], tw(XINH)[120], tw(XINL)[122], and tr(XINL)[123] Add Figure 21.9.1 Clock and Reset Timing
		21-44	Correct tw(BLWL) tw(BHWL)[51] Add td(CSL-RDL)[93]
		21-47	td(CSL-RDL)[93], td(BLEL-RDL) td(BHEL-RDL)[136], and tv(RDH-BLEL) tv(RDH-BHEL)[137]
		21-48	Add Figure 21.9.12 Read Timing (Byte Enable Mode)
		21-49	Add td(CSL-WRL)[96]
		App.1-2	Replace Dimensional Outline Drawing
		App.4-3	Add descriptions for dummy access areas Add Notes on the Internal RAM
		App.4-4	Add “About the peripheral function input when it is set to the general
			purpose port”
		App.4-5	Add “DMA interrupt related registers” to Appendix Table 4.8.1
		App.4-7	Replace Appendix Figure 4.9.1
		App.4-8	Replace Appendix Figure 4.9.2
		App.4-16	Add the precaution of switching from general-purpose to serial interface pin

Before Use

• Guide to Understanding the Register Table

(1) Bit number: Indicates a register's bit number.

(2) Register border: The registers enclosed with thick border lines must be accessed in halfwords or words.

(3) Status after reset: The initial state of each register after reset is indicated in hexadecimal or binary.

(4) Status after reset: The initial state of each register after reset is indicated bitwise.

0: This bit is "0" after reset.
1: This bit is "1" after reset.
?: This bit is undefined after reset.

(5) The shaded bits mean that they have no functions assigned.

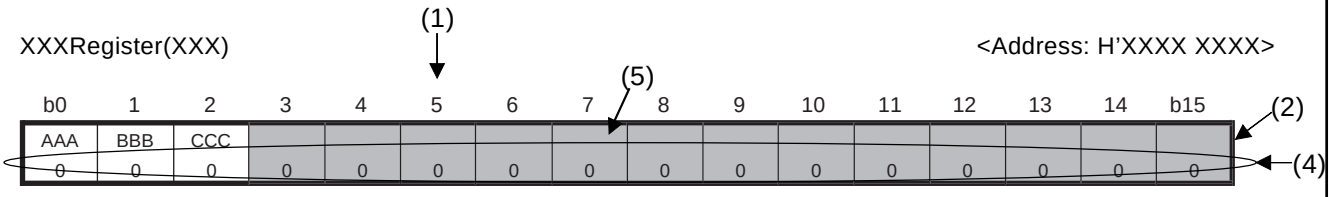
(6) Read conditions:

R: This bit can be accessed for read.
?: The value read from this bit is undefined. (Reading this bit has no effect.)
0: The value read from this bit is always "0".
1: The value read from this bit is always "1".

(7) Write conditions:

W: This bit can be accessed for write.
N: This bit is write protected.
0: To write to this bit, always write "0".
1: To write to this bit, always write "1".
–: Writing to this bit has no effect. (It does not matter whether this bit is set to "0" or "1" by writing in software.)
Note: Care must be taken when writing to this bit. See Note in each register table.

XXXRegister(XXX) <Address: H'XXXX XXXX>



(3) <After reset: H'0000>

b	Bit name	Function	R	W
0	AAA . . . . . bit	0 : . . . . . bit 1 : . . . . . bit	R	W
1	BBB . . . . . bit	0 : . . . . . bit 1 : . . . . . bit	R	W
2	CCC . . . . . bit	0 : . . . . . bit 1 : . . . . . bit	R	(Note 1)
3–15	No function assigned. Fix to "0".		0	0

Note 1: Only writing "0" is effective. Writing "1" has no effect, in which case the bit retains the value it had before the write.

(6) (7)

• Notation of "L" (signals)

The symbol "#" suffixed to the pin (or signal) names means that the pins (or signals) are "L" .

Before Use

The optional specification for the product of 32176Group

[Contents]

The optional specification for the product of M32176F4VFP,M32176F3VFP,M32176F2VFP is shown below.

(1) Flash 10000(10k) times rewritable product (4Kbyte block x 2 only)
(2) 40MHz operation product
(3) Lead-Free product

However it is not able to select both (1) and (2) at same time.
Note. There is no (1) optional specification for M32176F4TFP, M32176F3TFP, M32176F2TFP.

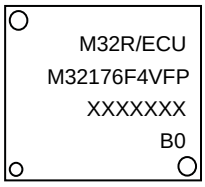
The product cord is allocated for each optional specification.
The list of product cord and the figure of marking are shown below.

[The list of products specification]

Model name	Product cord		Contents of specification
M32176F4VFP	B	0	32MHz/125°C, Flash is rewritable for 100 times
M32176F3VFP	or	1	32MHz/125°C, Flash is rewritable for 10000(10k) times (4Kbyte block x 2 only)
M32176F2VFP	U	2	40MHz/125°C, Flash is rewritable for 100 times
M32176F4TFP	B		
M32176F3TFP	or	0	40MHz/85°C, Flash is rewritable for 100 times
M32176F2TFP	U		

*B:Non-Lead-Free product, U:Lead-Free product

[The figure of marking]



: Show the family name.
: Show the product name M32176F4VFP.
: Show the administration number of the date cord (7 figure) in manufacturing.
: Show the product cord.

Please contact Renesas Technology Corp. for further details on these optional specifications or price.
When samples are requred, specify products cord (B0~B2,U0~U2). If not, the standard specification (B0) is requred.

查询"32176"供应商

This page is blank for reasons of layout.

Table of contents

APPENDIX 3 PROCESSING OF UNUSED PINS

APPENDIX 4 SUMMARY OF PRECAUTIONS

CHAPTER 1

OVERVIEW

1.1 Outline of the 32176 Group

Table 1.1.1 Product List

Type Name	ROM capacity	RAM capacity	Frequency	Power supply voltage	Temperature Range (Note 1)
M32176F4VFP	512 Kbytes	24 Kbytes	32 MHz (Note 2)	5 V or 3.3 V	–40°C to +125°C
M32176F4TFP	512 Kbytes	24 Kbytes	40 MHz	5 V or 3.3 V	–40°C to +85°C
M32176F3VFP	384 Kbytes	24 Kbytes	32 MHz (Note 2)	5 V or 3.3 V	–40°C to +125°C
M32176F3TFP	384 Kbytes	24 Kbytes	40 MHz	5 V or 3.3 V	–40°C to +85°C
M32176F2VFP	256 Kbytes	24 Kbytes	32 MHz (Note 2)	5 V or 3.3 V	–40°C to +125°C
M32176F2TFP	256 Kbytes	24 Kbytes	40 MHz	5 V or 3.3 V	–40°C to +85°C

Note 1: This does not guarantee continuous operation and there is a limitation on the length of use (temperature profile).

Note 2: There is a 40 MHz operation product optional specification for the product of M32176F4VFP, M32176F3VFP, and M32176F2VFP. Please contact Renesas Technology Corp. for further details on those optional specifications or price.

1.1.1 M32R Family CPU Core

(1) Based on a RISC architecture

- The 32176 is a 32-bit RISC single-chip microcomputer which is built around the M32R family CPU core (hereinafter referred to as the M32R) and incorporates flash memory, RAM and various other peripheral functions-all integrated into a single chip.
- The M32R is based on a RISC architecture. Memory is accessed using load/store instructions, and various arithmetic operations are executed using register-to-register operation instructions. The M32R internally contains sixteen 32-bit general-purpose registers and has 83 instructions.
- The M32R supports compound instructions such as Load & Address Update and Store & Address Update, in addition to ordinary load and store instructions. These instructions help to speed up data transfers.

(2) Five-stage pipelined processing

- The M32R supports five-stage pipelined instruction processing consisting of Instruction Fetch, Decode, Execute, Memory Access and Write Back. Not just load/store instructions and register-to-register operation instructions, compound instructions such as Load & Address Update and Store & Address Update are executed in one CPUCLK period (which is equivalent to 25 ns when f(CPUCLK) = 40 MHz).
- Although instructions are supplied to the execution stage in the order in which they were fetched, it is possible that if the load/store instruction supplied first is extended by wait cycles inserted in memory access, the subsequent register-to-register operation instruction will be executed before that instruction. Using such a facility, which is known as the “out-of-order-completion” mechanism, the M32R is able to control instruction execution without wasting clock cycles.

(3) Compact instruction code

- The M32R supports two instruction formats: one 16 bits long, and one 32 bits long. Use of the 16-bit instruction format especially helps to suppress the code size of a program.
- Moreover, the availability of 32-bit instructions makes programming easier and provides higher performance at the same clock speed than in architectures where the address space is segmented. For example, some 32-bit instructions allow control to jump to an address 32 Mbytes forward or backward from the currently executed address in one instruction, making programming easy.

1.1.2 Built-in Multiplier/Accumulator

(1) Built-in high-speed multiplier

- The M32R contains a 32 bits × 16 bits high-speed multiplier which enables the M32R to execute a 32 bits × 32 bits integral multiplication instruction in three CPUCLK periods.

(2) DSP-comparable multiply-accumulate instructions

- The M32R supports the following four types of multiply-accumulate instructions (or multiplication instructions) which each can be executed in one CPUCLK period using a 56-bit accumulator.

  (1) 16 high-order bits of register × 16 high-order bits of register
  (2) 16 low-order bits of register × 16 low-order bits of register
  (3) Whole 32 bits of register × 16 high-order bits of register
  (4) Whole 32 bits of register × 16 low-order bits of register

- The M32R has some special instructions to round the value stored in the accumulator to 16 or 32 bits or shift the accumulator value before storing in a register to have its digits adjusted. Because these instructions are also executed in one CPUCLK period, when used in combination with high-speed data transfer instructions such as Load & Address Update or Store & Address Update, they enable the M32R to exhibit data processing capability comparable to that of a DSP.

1.1.3 Built-in Flash Memory and RAM

- The 32176 contains a RAM that can be accessed with zero wait state, allowing to design a high-speed embedded system.
- The internal flash memory can be written to while mounted on a printed circuit board (on-board writing). Use of flash memory facilitates development work, because the chip used at the development stage can be used directly in mass-production, allowing for a smooth transition from prototype to mass-production without the need to change the printed circuit board.
- The internal flash memory has a virtual flash emulation function, allowing the internal RAM to be virtually mapped into part of the internal flash memory. When combined with the internal Real-Time Debugger (RTD), this function makes the ROM table data tuning easy.
- The internal RAM can be accessed for reading or rewriting data from an external device independently of the M32R by using the Real-Time Debugger. The external device is communicated using the Real-Time Debugger's exclusive clock-synchronized serial interface.

1.1.4 Built-in Clock Frequency Multiplier

• The 32176 contains a clock frequency multiplier, which is schematically shown in Figure 1.1.1 below.

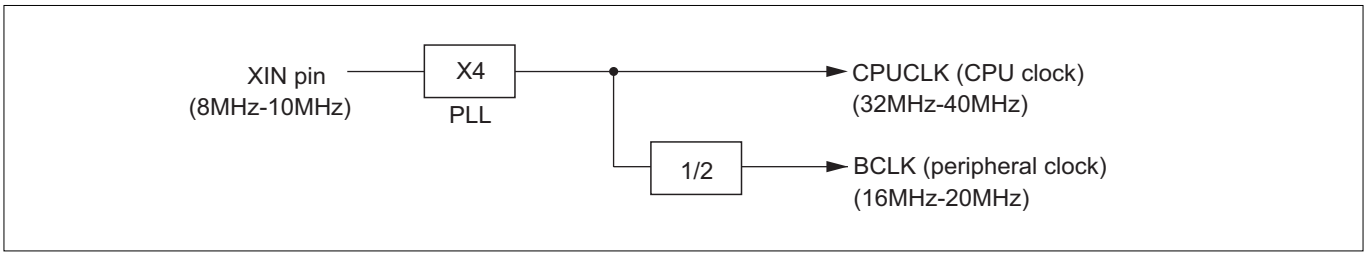


Figure 1.1.1 Conceptual Diagram of the Clock Frequency Multiplier

Table 1.1.2 Clock

Functional Block	Features
CPUCLK	• CPU clock: Defined as f(CPUCLK) when it indicates the operating clock frequency for the M32R core, internal flash memory and internal RAM.
BCLK	• Peripheral clock: Defined as f(BCLK) when it indicates the operating clock frequency for the internal peripheral I/O and external data bus.
Clock output (BCLK pin output)	• A clock with the same frequency as f(BCLK) is output from this pin.

1.1.5 Powerful Built-in Peripheral Functions

(1) 8-level interrupt controller (ICU)
(2) 10-channel DMAC
(3) 37-channel Multijunction timer (MJT)
(4) 16-channel A/D converter (ADC)
(5) 4-channel high-speed serial interface (SIO)
(6) 2-channel Full-CAN
(7) Real-time debugger (RTD)
(8) Wait controller
(9) M32R family's common debug function (Scalable Debug Interface or SDI)

1.2 Block Diagram

Figure 1.2.1 shows a block diagram of the 32176. The features of each block are described in Table 1.2.1.

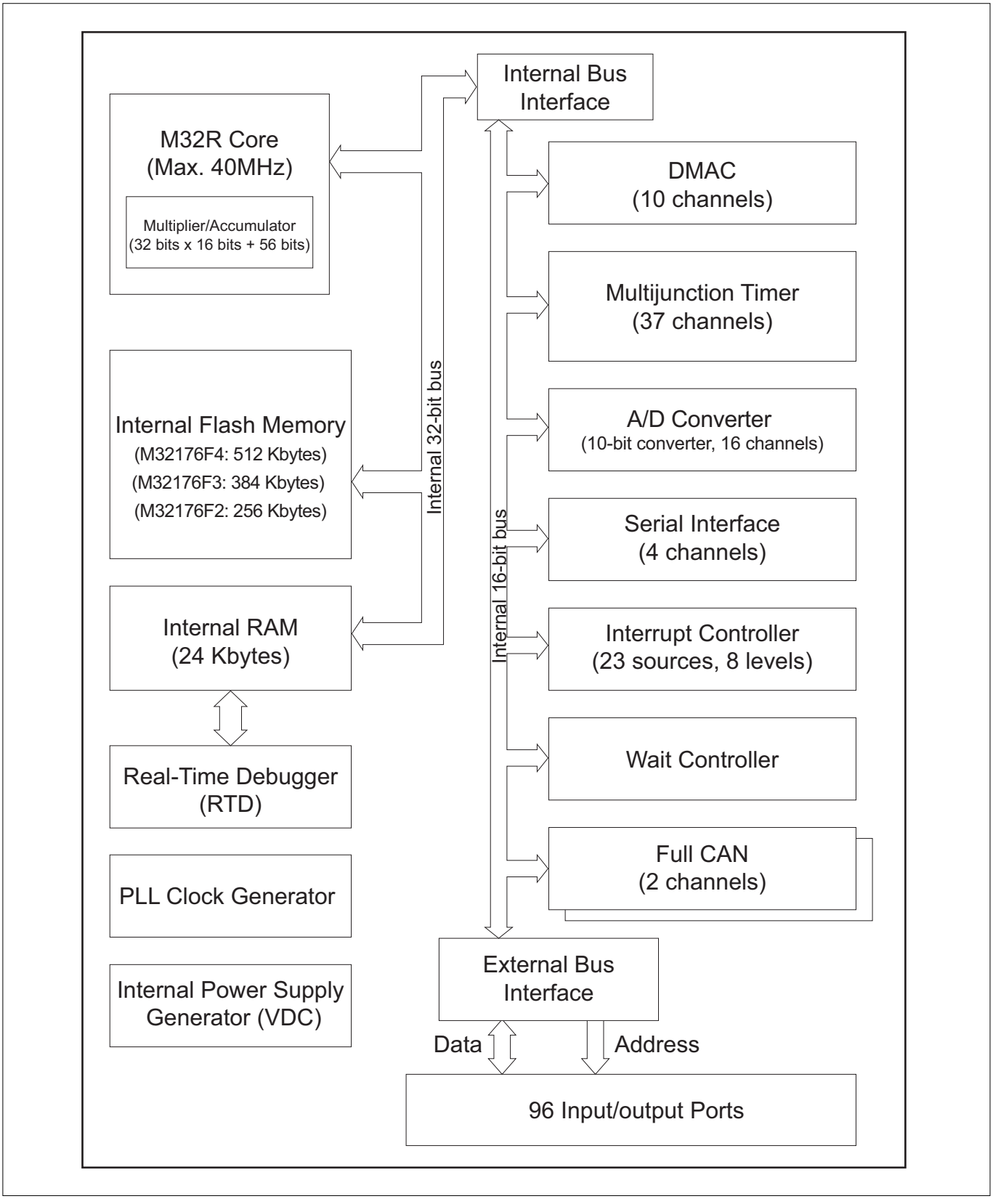


Figure 1.2.1 Block Diagram of the 32176

Table 1.2.1 Features of the 32176 (1/2)

Functional Block	Features
M32R CPU core	• Implementation: Five-stage pipelined instruction processing • Internal 32-bit structure of the core • Register configuration General-purpose registers: 32 bits × 16 registers Control registers: 32 bits × 5 registers • Instruction set 16 and 32-bit instruction formats 83 instructions and six addressing modes • Internal multiplier/accumulator (32 bits × 16 bits + 56 bits)
RAM	• Capacity: 24 Kbytes • Zero-wait access • The internal RAM can be accessed for reading or rewriting data from the outside independently of the M32R by using the Real-Time Debugger, without ever causing the CPU performance to decrease. • The internal RAM can be backed up by using RAM back up mode when turn off the power supply.
Flash memory	• Capacity: M32176F2: 256 Kbytes, M32176F3: 384 Kbytes, M32176F4: 512 Kbytes • Zero-wait access • Durability: Standard product : 100 times 10000 (10k) times rewritable -product (Note 1) : 4-Kbyte block (Note 2) : 10000 (10k) times : Other blocks : 1000 (1k) times
Bus specification	• Fundamental bus cycle : 25 ns (when f(CPUCLK = 40 MHz) • Logical address space : 4 Gbytes linear • Internal bus specification : Internal 32-bit data bus (for CPU <-> internal flash memory and RAM access) : Internal 16-bit data bus (for internal peripheral I/O access) • External area: Maximum 2 Mbytes (during processor mode) • External extension area: Maximum 2 Mbytes • External data address bus: 19-bit address • External data bus: 16-bit data bus • Shortest external bus access: 2 BCLK periods during read, 2 BCLK periods during write
DMAC	• Number of channels: 10 • Transfers between internal peripheral I/O's or internal RAM's or between internal peripheral I/O and internal RAM are supported. • Capable of advanced DMA transfers when used in combination with internal peripheral I/O • Transfer request: Software or internal peripheral I/O (A/D converter, MJT, serial interface or CAN) • DMA channels can be cascaded. (DMA transfer on a channel can be started by completion of a transfer on another channel.) • Interrupt request: DMA transfer counter register underflow
Multijunction timer (MJT)	• 37-channel multi-functional timer 16-bit output related timer × 11 channels, 16-bit input/output related timer × 10 channels, 16-bit input related timer × 8 channels, 32-bit input related timer × 8 channels • Flexible timer configuration is possible by interconnecting these timer channels. • Interrupt request: Counter underflow or overflow and rising, falling or both edges or "H" or "L"level from the TIN pin (These can be used as external interrupt inputs irrespective of timer operation.) • DMA transfer request: Counter underflow or overflow and rising, falling or both edges or "H" or "L" level from the TIN pin (These can be used as external DMA transfer request inputs irrespective of timer operation.)

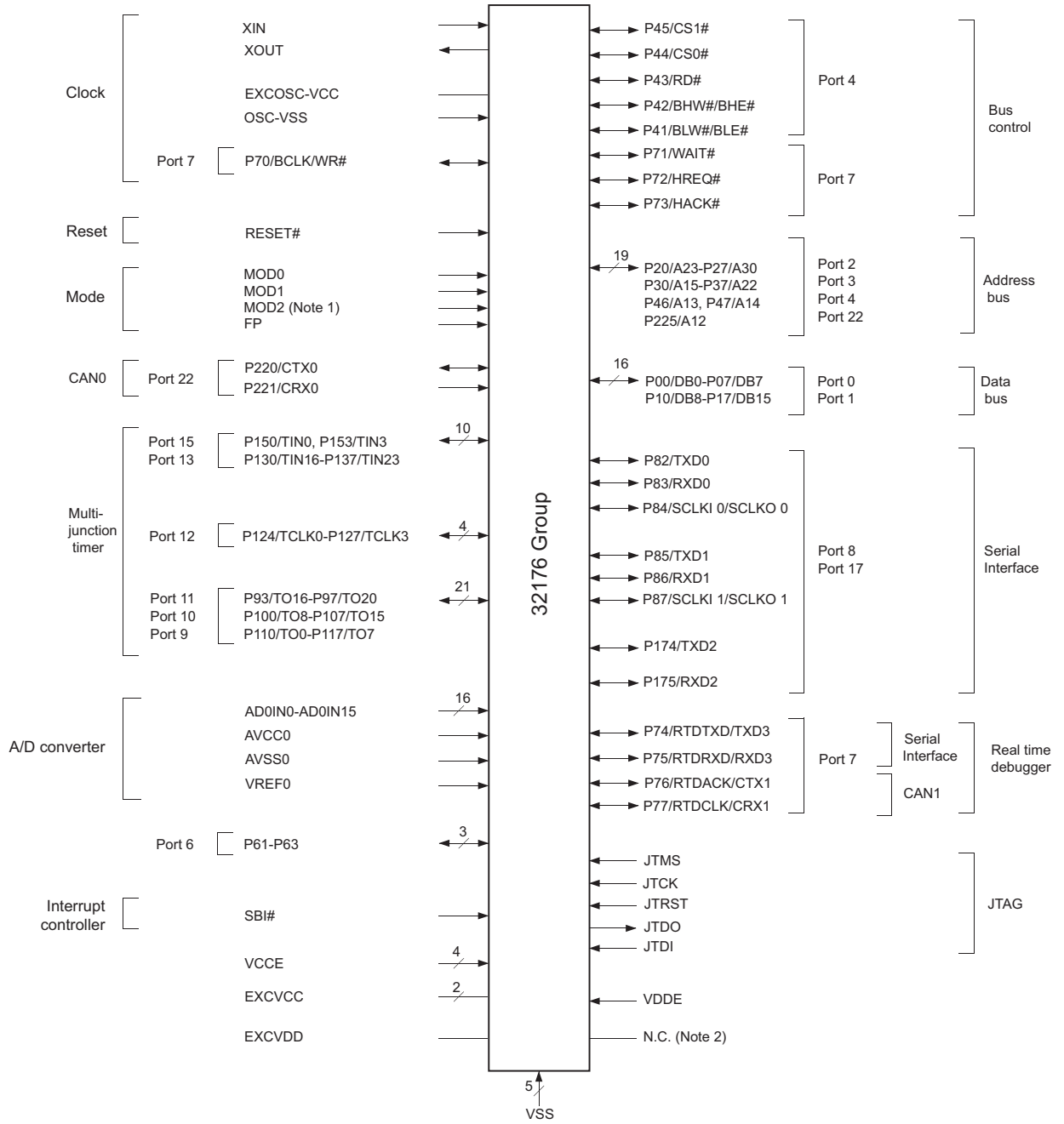
Table 1.2.1 Features of the 32176 (2/2)

Functional Block	Features
A/D converter (ADC)	• 16 channels: 10-bit resolution A/D converter • Conversion modes: Ordinary conversion modes plus comparator mode are built-in • Operation modes: Single conversion mode and n-channel scan mode (n = 1–16) • Sample-and-hold function: Sample-and-hold function can be enabled or disabled as necessary. • A/D disconnection detection assist function: Influences of the analog input voltage leakage from any preceding channel during scan mode operation are suppressed. • An inflow current bypass circuit is built-in. • Can generate an interrupt or start DMA transfer upon completion of A/D conversion. • Either 8 or 10-bit conversion results can be read out. • Interrupt request: Completion of A/D conversion • DMA transfer request: Completion of A/D conversion
Serial Interface (SIO)	• 4-channel serial interface • Can be chosen to be clock-synchronized serial interface or clock-asynchronous serial interface. • Data can be transferred at high speed (2 Mbits per second during clock-synchronized mode or 156 Kbits per second during UART mode when f(BCLK) = 20 MHz). • Interrupt request: Reception completed, receive error, transmit buffer empty or transmission completed • DMA transfer request: Reception completed or transmit buffer empty
CAN	• 16 message slots × 2 blocks • Compliant with CAN specification 2.0B active. • Interrupt request: Transmission completed, reception completed, bus error, error-passive, bus-off or single shot • DMA transfer request: Transmission failed, transmission completed or reception completed
Real-Time Debugger (RTD)	• Internal RAM can be rewritten or monitored independently of the CPU by entering a command from the outside. • Comes with exclusive clock-synchronized serial ports. • Interrupt request: RTD interrupt command input
Interrupt Controller (ICU)	• Controls interrupt requests from the internal peripheral I/O. • Supports 8-level interrupt priority including an interrupt disabled state. • External interrupt: 11 sources (SBI#, TIN0,TIN3, TIN16-TIN23) • TIN pin input sensing: Rising, falling or both edges or “H” or “L” level
Wait Controller	• Controls wait states for access to the external extension area. • Insertion of 1-4 wait states by setting up in software + wait state extension by entering WAIT# signal
PLL	• A multiply-by-4 clock generating circuit
Clock	• Maximum external input clock frequency (XIN) is 10.0 MHz. (Note 3) • CPUCLK: Operating clock for the M32R-CPU core, internal flash memory and internal RAM The maximum CPU clock is 40 MHz (when f(XIN) = 10 MHz). • BCLK: Operating clock for the internal peripheral I/O and external data bus The maximum peripheral clock is 20 MHz (peripheral module access when f(XIN) = 10 MHz). • Clock output (BCLK pin output): A clock with the same frequency as BCLK is output from this pin.
JTAG	• Boundary scan function
VDC	• Internal power supply generating circuit: Generates the internal power supply (2.5 V) from an external single power supply (5 or 3.3 V).
Ports	• Input/output pins: 96 pins • The port input threshold can be set in a program to one of three levels individually for each port group (with or without Schmitt circuit, selectable).

Note 1: The 10000 (10k) times rewritable product is offered as an optional item. For details about it, please contact your nearest office of Renesas or its distributor.
Note 2: Block 1: H’0000 2000 to H’0000 2FFFF
Block 2: H’0000 3000 to H’0000 3FFFF
Note 3: Maximum external input clock frequency (XIN) for the M32176F2VFP, M32176F3VFP and M32176F4VFP is 8.0 MHz.

1.3 Pin Functions

Figure 1.3.1 shows the 32176's pin function diagram. Pin functions are described in Table 1.3.1.



Note 1: MOD2 must be connected to the ground.
Note 2: N.C. indicates non-connected pin. Connect the pin to the power supply, ground, or the like that has no voltage change.
Note: • The symbol "#" suffixed to the pin (or signal) name means that the pins (or signals) are "L."

Figure 1.3.1 Pin Function Diagram

Table 1.3.1 Description of Pin Functions (1/4)

Type	Pin Name	Signal Name	Input/Output	Description
Power supply	VCCE	Main power supply	–	Power supply for the device (5.0 V ± 0.5 V or 3.3 V ± 0.3 V).
	EXCVCC	Internal power supply	–	This pin connects an external capacitor.
	VDDE	RAM power supply	–	Backup power supply for the internal RAM (5.0 V ± 0.5 V or 3.3 V ± 0.3 V).
	EXCVDD	Internal power supply of RAM	–	This pin connects an external capacitor for the internal power supply of the internal RAM.
	VSS	Ground	–	Connect all VSS pins to ground (GND).
Clock	XIN, XOUT	Clock input Clock output	Input Output	These are clock input/output pins. A PLL-based ×4 frequency multiplier is included, which accepts as input a clock whose frequency is 1/4 of the internal CPU clock frequency. (XIN input is 10 MHz when f(CPUCLK) = 40 MHz.)
	BCLK	System clock	Output	This pin outputs a clock whose frequency is twice that of the external input clock (XIN). (BCLK output is 20 MHz when f(CPUCLK) = 40 MHz.) Use this clock to synchronize the operation of external devices.
	EXCOSC-VCC	Clock power supply	–	This pin connects an external capacitor for the oscillator circuit.
	OSC-VSS	Clock ground	–	Connect OSC-VSS to ground.
Reset	RESET#	Reset	Input	Reset input pin for the internal circuit.
Mode	MOD0–MOD2	Mode	Input	Set the microcomputer's operation mode.
Flash	FP	Flash protect	Input	This special pin protects the flash memory against rewrites in hardware.
Address bus	A12–A30	Address bus	Output	To allow two areas of up to 1 Mbyte memory space to be connected external to the chip, the device has 19 address lines (A12–A30). A31 is not output.

MOD0	MOD1	MOD2	Mode
L	L	L	Single-chip mode
L	H	L	External extension mode
H	L	L	Processor mode (boot mode) (Note1)
H	H	L	(Settings inhibited)
X	X	H	(Settings inhibited)

X: Don't care

Note 1: Boot mode requires that the FP pin should be at "H" level. For details about boot mode, see Chapter 6, "Internal Memory."

Table 1.3.1 Description of Pin Functions (2/4)

Type	Pin Name	Signal Name	Input/Output	Description
Data bus	DB0–DB15	Data bus	Input/output	This 16-bit data bus is used to connect external devices. When writing in byte units during a write cycle, the output data at the invalid byte position is undefined. During a read cycle, data on the entire 16-bit data bus is always read in. However, only the data at the valid byte position is transferred into the internal circuit.
Bus control	CS0#,CS1#	Chip select	Output	These are chip select signals for external devices.
	RD#	Read	Output	This signal is output when reading an external device.
	WR#	Write	Output	This signal is output when writing to an external device.
	BHW#,BLW#	Byte high write/ Byte low write	Output	When writing to an external device, this signal indicates the valid byte position to which data is transferred. BHW# and BLW# correspond to the upper address side (bits 0–7 are valid) and the lower address side (bits 8–15 are valid), respectively.
	BHE#	Byte high enable	Output	During an external device access, this signal indicates that the high-order data (bits 0–7) is valid.
	BLE#	Byte low enable	Output	During an external device access, this signal indicates that the low-order data (bits 8–15) is valid.
	WAIT#	Wait	Input	When accessing an external device, a “L” signal input on WAIT# pin extends the wait cycle.
	HREQ#	Hold request	Input	This input is used by an external device to request control of the external bus. A “L” level input on HREQ# pin places the CPU in a hold state.
	HACK#	Hold acknowledge	Output	This signal notifies that the CPU has entered a hold state and relinquished control of the external bus.
Multijunction timer	TIN0, TIN3, TIN16–TIN23	Timer input	Input	Input pins for the multijunction timer.
	TO0–TO20	Timer output	Output	Output pins for the multijunction timer.
	TCLK0 –TCLK3	Timer clock	Input	Clock input pins for the multijunction timer.
A/D converter	AVCC0	Analog power supply	–	AVCC0 is the power supply for the A/D0 converter. Connect AVCC0 to the power supply rail.
	AVSS0	Analog ground	–	AVSS0 is the analog ground for the A/D0 converter. Connect AVSS0 to ground.
	AD0IN0 –AD0IN15	Analog input	Input	16-channel analog input pins for the A/D0 converter.
	VREF0	Reference voltage input	Input	VREF0 is the reference voltage input pin for the A/D0 converter.
Interrupt controller	SBI#	System break interrupt	Input	This is the system break interrupt (SBI) input pin for the interrupt controller.

Table 1.3.1 Description of Pin Functions (3/4)

Type	Pin Name	Signal Name	Input/Output	Description
Serial Interface	SCLKI0/ SCLKO0	UART transmit/receive clock output or CSIO transmit/receive clock input/output	Input/output	When channel 0 is in UART mode: This pin outputs a clock derived from BRG output by dividing it by 2. When channel 0 is in CSIO mode: This pin accepts as input a transmit/receive clock when external clock is selected or outputs a transmit/receive clock when internal clock is selected.
	SCLKI1/ SCLKO1	UART transmit/receive clock output or CSIO transmit/receive clock input/output	Input/output	When channel 1 is in UART mode: This pin outputs a clock derived from BRG output by dividing it by 2. When channel 1 is in CSIO mode: This pin accepts as input a transmit/receive clock when external clock is selected or outputs a transmit/receive clock when internal clock is selected.
	TXD0–TXD3	Transmit data	Output	Transmit data output pin for serial interface.
	RXD0–RXD3	Received data	Input	Received data input pin for serial interface.
Real-time debugger (RTD)	RTDTXD	Transmit data	Output	Serial data output pin for the real-time debugger.
	RTDRXD	Received data	Input	Serial data input pin for the real-time debugger.
	RTDCLK	Clock input	Input	Serial data transmit/receive clock input pin for the real-time debugger.
	RTDACK	Acknowledge	Output	A ”L” pulse is output from this pin synchronously with the start clock for the real-time debugger’s serial data output word. The “L” pulse width indicates the type of command/data received by the real-time debugger.
CAN	CTX0, CTX1	Data output	Output	This pin outputs data from the CAN module.
	CRX0, CRX1	Data input	Input	This pin inputs the data to the CAN module.
JTAG	JTMS	Test mode	Input	Test mode select input to control the state transition of the test circuit.
	JTCK	Clock	Input	Clock input for the debug module and test circuit.
	JTRST	Test reset	Input	Test reset input to initialize the test circuit asynchronously with device operation.
	JTDI	Serial input	Input	This pin inputs the test instruction code or test data that is serially received.
	JTDO	Serial output	Output	This pin outputs the test instruction code or test data serially.

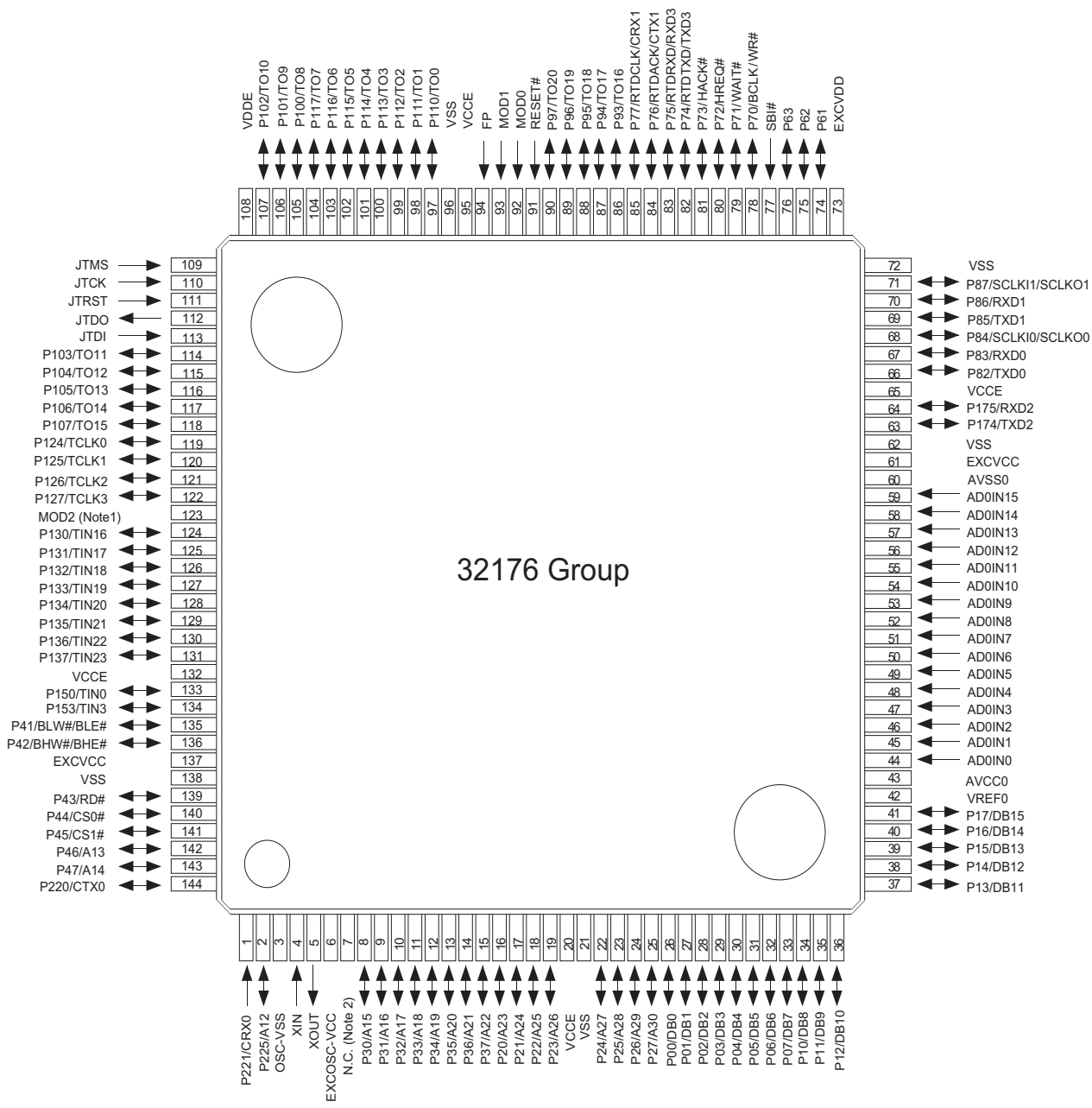
Table 1.3.1 Description of Pin Functions (4/4)

Type	Pin Name	Signal Name	Input/Output	Description
Input/output ports (Note 1)	P00–P07	Input/output port 0	Input/output	Programmable input/output port.
	P10–P17	Input/output port 1	Input/output	
	P20–P27	Input/output port 2	Input/output	
	P30–P37	Input/output port 3	Input/output	
	P41–P47	Input/output port 4	Input/output	
	P61–P63	Input/output port 6	Input/output	
	P70–P77	Input/output port 7	Input/output	
	P82–P87	Input/output port 8	Input/output	
	P93–P97	Input/output port 9	Input/output	
	P100–P107	Input/output port 10	Input/output	
	P110–P117	Input/output port 11	Input/output	
	P124–P127	Input/output port 12	Input/output	
	P130–P137	Input/output port 13	Input/output	
	P150, P153	Input/output port 15	Input/output	
	P174, P175	Input/output port 17	Input/output	
	P220, P221(Note 2), P225,	Input/output port 22	Input/output	

Note 1: Input/output port 5 is reserved for future use. Also, input/output ports 14, 16 and 18-21 are nonexistent.
Note 2: P221 is input-only port.

1.4 Pin Assignments

Figure 1.4.1 shows the 32176's pin assignment diagram. A pin assignment table is shown in Table 1.4.1.



Package: 144pin LQFP (PLQP0144KA-A)

Note 1: MOD2 must be connected to the ground.
Note 2: N.C. indicates non-connected pin. Connect the pin to the power supply, ground, or the like that has no voltage change.
Note: • The symbol "#" suffixed to the pin (or signal) name means that the pins (or signals) are "L."

Figure 1.4.1 Pin Assignment Diagram (Top View)

The pins directed for input go to a high-impedance state (Hi-Z) when reset. The term “when reset” refers to the period when input on RESET# pin is held “L” (the device remains reset), as well as when the RESET# pin is released back “H” (the device comes out of reset).

Table 1.4.1 Pin Assignments of the 32176 (1/4)

Pin No.	Symbol	Function			Type	Condition	Pin State When Reset			
		Port	Other than port	Other than port			Function	Type	State during reset	State upon exiting reset
1	P221/CRX0	P221	CRX0	-	Input		P221	Input	Hi-Z	Hi-Z
2	P225/A12	P225	A12	-	Input/output	During single-chip mode	P225	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A12	Output	Hi-Z	Undefined
3	OSC-VSS	-	OSC-VSS	-	-		OSC-VSS	-	-	-
4	XIN	-	XIN	-	Input		XIN	Input	-	-
5	XOUT	-	XOUT	-	Output		XOUT	Output	XOUT	XOUT
6	EXCOSC-VCC	-	EXCOSC-VCC	-	-		EXCOSC-VCC	-	-	-
7	N.C.	-	-	-	-		-	-	-	-
8	P30/A15	P30	A15	-	Input/output	During single-chip mode	P30	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A15	Output	Hi-Z	Undefined
9	P31/A16	P31	A16	-	Input/output	During single-chip mode	P31	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A16	Output	Hi-Z	Undefined
10	P32/A17	P32	A17	-	Input/output	During single-chip mode	P32	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A17	Output	Hi-Z	Undefined
11	P33/A18	P33	A18	-	Input/output	During single-chip mode	P33	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A18	Output	Hi-Z	Undefined
12	P34/A19	P34	A19	-	Input/output	During single-chip mode	P34	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A19	Output	Hi-Z	Undefined
13	P35/A20	P35	A20	-	Input/output	During single-chip mode	P35	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A20	Output	Hi-Z	Undefined
14	P36/A21	P36	A21	-	Input/output	During single-chip mode	P36	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A21	Output	Hi-Z	Undefined
15	P37/A22	P37	A22	-	Input/output	During single-chip mode	P37	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A22	Output	Hi-Z	Undefined
16	P20/A23	P20	A23	-	Input/output	During single-chip mode	P20	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A23	Output	Hi-Z	Undefined
17	P21/A24	P21	A24	-	Input/output	During single-chip mode	P21	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A24	Output	Hi-Z	Undefined
18	P22/A25	P22	A25	-	Input/output	During single-chip mode	P22	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A25	Output	Hi-Z	Undefined
19	P23/A26	P23	A26	-	Input/output	During single-chip mode	P23	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A26	Output	Hi-Z	Undefined
20	VCCE	-	VCCE	-	-		VCCE	-	-	-
21	VSS	-	VSS	-	-		VSS	-	-	-
22	P24/A27	P24	A27	-	Input/output	During single-chip mode	P24	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A27	Output	Hi-Z	Undefined
23	P25/A28	P25	A28	-	Input/output	During single-chip mode	P25	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A28	Output	Hi-Z	Undefined
24	P26/A29	P26	A29	-	Input/output	During single-chip mode	P26	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A29	Output	Hi-Z	Undefined
25	P27/A30	P27	A30	-	Input/output	During single-chip mode	P27	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A30	Output	Hi-Z	Undefined
26	P00/DB0	P00	DB0	-	Input/output	During single-chip mode	P00	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB0	Input/output	Hi-Z	Hi-Z

Table 1.4.1 Pin Assignments of the 32176 (2/4)

Pin No.	Symbol	Function			Type	Condition	Pin State When Reset			
		Port	Other than port	Other than port			Function	Type	State during reset	State upon exiting reset
27	P01/DB1	P01	DB1	-	Input/output	During single-chip mode	P01	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB1	Input/output	Hi-Z	Hi-Z
28	P02/DB2	P02	DB2	-	Input/output	During single-chip mode	P02	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB2	Input/output	Hi-Z	Hi-Z
29	P03/DB3	P03	DB3	-	Input/output	During single-chip mode	P03	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB3	Input/output	Hi-Z	Hi-Z
30	P04/DB4	P04	DB4	-	Input/output	During single-chip mode	P04	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB4	Input/output	Hi-Z	Hi-Z
31	P05/DB5	P05	DB5	-	Input/output	During single-chip mode	P05	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB5	Input/output	Hi-Z	Hi-Z
32	P06/DB6	P06	DB6	-	Input/output	During single-chip mode	P06	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB6	Input/output	Hi-Z	Hi-Z
33	P07/DB7	P07	DB7	-	Input/output	During single-chip mode	P07	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB7	Input/output	Hi-Z	Hi-Z
34	P10/DB8	P10	DB8	-	Input/output	During single-chip mode	P10	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB8	Input/output	Hi-Z	Hi-Z
35	P11/DB9	P11	DB9	-	Input/output	During single-chip mode	P11	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB9	Input/output	Hi-Z	Hi-Z
36	P12/DB10	P12	DB10	-	Input/output	During single-chip mode	P12	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB10	Input/output	Hi-Z	Hi-Z
37	P13/DB11	P13	DB11	-	Input/output	During single-chip mode	P13	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB11	Input/output	Hi-Z	Hi-Z
38	P14/DB12	P14	DB12	-	Input/output	During single-chip mode	P14	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB12	Input/output	Hi-Z	Hi-Z
39	P15/DB13	P15	DB13	-	Input/output	During single-chip mode	P15	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB13	Input/output	Hi-Z	Hi-Z
40	P16/DB14	P16	DB14	-	Input/output	During single-chip mode	P16	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB14	Input/output	Hi-Z	Hi-Z
41	P17/DB15	P17	DB15	-	Input/output	During single-chip mode	P17	Input	Hi-Z	Hi-Z
						During external extension and processor modes	DB15	Input/output	Hi-Z	Hi-Z
42	VREF0	-	VREF0	-	-		VREF0	-	-	-
43	AVCC0	-	AVCC0	-	-		AVCC0	-	-	-
44	AD0IN0	-	AD0IN0	-	Input		AD0IN0	Input	Hi-Z	Hi-Z
45	AD0IN1	-	AD0IN1	-	Input		AD0IN1	Input	Hi-Z	Hi-Z
46	AD0IN2	-	AD0IN2	-	Input		AD0IN2	Input	Hi-Z	Hi-Z
47	AD0IN3	-	AD0IN3	-	Input		AD0IN3	Input	Hi-Z	Hi-Z
48	AD0IN4	-	AD0IN4	-	Input		AD0IN4	Input	Hi-Z	Hi-Z
49	AD0IN5	-	AD0IN5	-	Input		AD0IN5	Input	Hi-Z	Hi-Z
50	AD0IN6	-	AD0IN6	-	Input		AD0IN6	Input	Hi-Z	Hi-Z
51	AD0IN7	-	AD0IN7	-	Input		AD0IN7	Input	Hi-Z	Hi-Z
52	AD0IN8	-	AD0IN8	-	Input		AD0IN8	Input	Hi-Z	Hi-Z
53	AD0IN9	-	AD0IN9	-	Input		AD0IN9	Input	Hi-Z	Hi-Z
54	AD0IN10	-	AD0IN10	-	Input		AD0IN10	Input	Hi-Z	Hi-Z
55	AD0IN11	-	AD0IN11	-	Input		AD0IN11	Input	Hi-Z	Hi-Z
56	AD0IN12	-	AD0IN12	-	Input		AD0IN12	Input	Hi-Z	Hi-Z
57	AD0IN13	-	AD0IN13	-	Input		AD0IN13	Input	Hi-Z	Hi-Z
58	AD0IN14	-	AD0IN14	-	Input		AD0IN14	Input	Hi-Z	Hi-Z
59	AD0IN15	-	AD0IN15	-	Input		AD0IN15	Input	Hi-Z	Hi-Z

Table 1.4.1 Pin Assignments of the 32176 (3/4)

Pin No.	Symbol	Function			Type	Condition	Pin State When Reset			
		Port	Other than port	Other than port			Function	Type	State during reset	State upon exiting reset
60	AVSS0	-	AVSS0	-	-		AVSS0	-	-	-
61	EXCVCC	-	EXCVCC	-	-		EXCVCC	-	-	-
62	VSS	-	VSS	-	-		VSS	-	-	-
63	P174/TXD2	P174	TXD2	-	Input/output		P174	Input	Hi-Z	Hi-Z
64	P175/RXD2	P175	RXD2	-	Input/output		P175	Input	Hi-Z	Hi-Z
65	VCCE	-	VCCE	-	Input/output		VCCE	-	-	-
66	P82/TXD0	P82	TXD0	-	Input/output		P82	Input	Hi-Z	Hi-Z
67	P83/RXD0	P83	RXD0	-	Input/output		P83	Input	Hi-Z	Hi-Z
68	P84/SCLKI0/SCLKO0	P84	SCLKI0	SCLKO0	Input/output		P84	Input	Hi-Z	Hi-Z
69	P85/TXD1	P85	TXD1	-	Input/output		P85	Input	Hi-Z	Hi-Z
70	P86/RXD1	P86	RXD1	-	Input/output		P86	Input	Hi-Z	Hi-Z
71	P87/SCLKI1/SCLKO1	P87	SCLKI1	SCLKO1	Input/output		P87	Input	Hi-Z	Hi-Z
72	VSS	-	VSS	-	-		VSS	-	-	-
73	EXCVDD	-	EXCVDD	-	-		EXCVDD	-	-	-
74	P61	P61	-	-	Input/output		P61	Input	Hi-Z	Hi-Z
75	P62	P62	-	-	Input/output		P62	Input	Hi-Z	Hi-Z
76	P63	P63	-	-	Input/output		P63	Input	Hi-Z	Hi-Z
77	SBI#		SBI#	-	Input		SBI#	Input	Hi-Z	Hi-Z
78	P70/BCLK/WR#	P70	BCLK	WR#	Input/output		P70	Input	Hi-Z	Hi-Z
79	P71/WAIT#	P71	WAIT#	-	Input/output		P71	Input	Hi-Z	Hi-Z
80	P72/HREQ#	P72	HREQ#	-	Input/output		P72	Input	Hi-Z	Hi-Z
81	P73/HACK#	P73	HACK#	-	Input/output		P73	Input	Hi-Z	Hi-Z
82	P74/RTDTXD/TXD3	P74	RTDTXD	TXD3	Input/output		P74	Input	Hi-Z	Hi-Z
83	P75/RTDRXD/RXD3	P75	RTDRXD	RXD3	Input/output		P75	Input	Hi-Z	Hi-Z
84	P76/RTDACK/CTX1	P76	RTDACK	CTX1	Input/output		P76	Input	Hi-Z	Hi-Z
85	P77/RTDCLK/CRX1	P77	RTDCLK	CRX1	Input/output		P77	Input	Hi-Z	Hi-Z
86	P93/TO16	P93	TO16	-	Input/output		P93	Input	Hi-Z	Hi-Z
87	P94/TO17	P94	TO17	-	Input/output		P94	Input	Hi-Z	Hi-Z
88	P95/TO18	P95	TO18	-	Input/output		P95	Input	Hi-Z	Hi-Z
89	P96/TO19	P96	TO19	-	Input/output		P96	Input	Hi-Z	Hi-Z
90	P97/TO20	P97	TO20	-	Input/output		P97	Input	Hi-Z	Hi-Z
91	RESET#	-	RESET#	-	Input		RESET#	Input	Hi-Z	Hi-Z
92	MOD0	-	MOD0	-	Input		MOD0	Input	Hi-Z	Hi-Z
93	MOD1	-	MOD1	-	Input		MOD1	Input	Hi-Z	Hi-Z
94	FP	-	FP	-	Input		FP	Input	Hi-Z	Hi-Z
95	VCCE	-	VCCE	-	-		VCCE	-	-	-
96	VSS	-	VSS	-	-		VSS	-	-	-
97	P110/TO0	P110	TO0	-	Input/output		P110	Input	Hi-Z	Hi-Z
98	P111/TO1	P111	TO1	-	Input/output		P111	Input	Hi-Z	Hi-Z
99	P112/TO2	P112	TO2	-	Input/output		P112	Input	Hi-Z	Hi-Z
100	P113/TO3	P113	TO3	-	Input/output		P113	Input	Hi-Z	Hi-Z
101	P114/TO4	P114	TO4	-	Input/output		P114	Input	Hi-Z	Hi-Z
102	P115/TO5	P115	TO5	-	Input/output		P115	Input	Hi-Z	Hi-Z
103	P116/TO6	P116	TO6	-	Input/output		P116	Input	Hi-Z	Hi-Z
104	P117/TO7	P117	TO7	-	Input/output		P117	Input	Hi-Z	Hi-Z
105	P100/TO8	P100	TO8	-	Input/output		P100	Input	Hi-Z	Hi-Z
106	P101/TO9	P101	TO9	-	Input/output		P101	Input	Hi-Z	Hi-Z
107	P102/TO10	P102	TO10	-	Input/output		P102	Input	Hi-Z	Hi-Z
108	VDDE	-	VDDE	-	-		VDDE	-	-	-
109	JTMS (Note 1)	-	JTMS	-	Input		JTMS	Input	Hi-Z	Hi-Z
110	JTCK (Note 1)	-	JTCK	-	Input		JTCK	Input	Hi-Z	Hi-Z
111	JTRST (Note 1)	-	JTRST	-	Input		JTRST	Input	Hi-Z	Hi-Z
112	JTDO (Note 1)	-	JTDO	-	Output		JTDO	Output	Hi-Z	Hi-Z
113	JTDI (Note 1)	-	JTDI	-	Input		JTDI	Input	Hi-Z	Hi-Z
114	P103/TO11	P103	TO11	-	Input/output		P103	Input	Hi-Z	Hi-Z
115	P104/TO12	P104	TO12	-	Input/output		P104	Input	Hi-Z	Hi-Z
116	P105/TO13	P105	TO13	-	Input/output		P105	Input	Hi-Z	Hi-Z
117	P106/TO14	P106	TO14	-	Input/output		P106	Input	Hi-Z	Hi-Z

Note 1: The JTCK, JTDI, JTDO and JTMS pins are reset by input from the JTRST pin, and not reset from the RESET# pin. When a “L” level is applied to the JTRST pin, the JTCK, JTDI, JTDO and JTMS pins are in the high impedance state.

Table 1.4.1 Pin Assignments of the 32176 (4/4)

Pin No.	Symbol	Function			Type	Condition	Pin State When Reset			
		Port	Other than port	Other than port			Function	Type	State during reset	State upon exiting reset
118	P107/TO15	P107	TO15	-	Input/output		P107	Input	Hi-Z	Hi-Z
119	P124/TCLK0	P124	TCLK0	-	Input/output		P124	Input	Hi-Z	Hi-Z
120	P125/TCLK1	P125	TCLK1	-	Input/output		P125	Input	Hi-Z	Hi-Z
121	P126/TCLK2	P126	TCLK2	-	Input/output		P126	Input	Hi-Z	Hi-Z
122	P127/TCLK3	P127	TCLK3	-	Input/output		P127	Input	Hi-Z	Hi-Z
123	MOD2	-	MOD2	-	-		MOD2	-	-	-
124	P130/TIN16	P130	TIN16	-	Input/output		P130	Input	Hi-Z	Hi-Z
125	P131/TIN17	P131	TIN17	-	Input/output		P131	Input	Hi-Z	Hi-Z
126	P132/TIN18	P132	TIN18	-	Input/output		P132	Input	Hi-Z	Hi-Z
127	P133/TIN19	P133	TIN19	-	Input/output		P133	Input	Hi-Z	Hi-Z
128	P134/TIN20	P134	TIN20	-	Input/output		P134	Input	Hi-Z	Hi-Z
129	P135/TIN21	P135	TIN21	-	Input/output		P135	Input	Hi-Z	Hi-Z
130	P136/TIN22	P136	TIN22	-	Input/output		P136	Input	Hi-Z	Hi-Z
131	P137/TIN23	P137	TIN23	-	Input/output		P137	Input	Hi-Z	Hi-Z
132	VCCE	-	VCCE	-	-		VCCE	-	-	-
133	P150/TIN0	P150	TIN0	-	Input/output		P150	Input	Hi-Z	Hi-Z
134	P153/TIN3	P153	TIN3	-	Input/output		P153	Input	Hi-Z	Hi-Z
135	P41/BLW#/BLE#	P41	BLW#	BLE#	Input/output	During single-chip mode	P41	Input	Hi-Z	Hi-Z
						During external extension and processor modes	BLW#	Output	Hi-Z	"H" level
136	P42/BHW#/BHE#	P42	BHW#	BHE#	Input/output	During single-chip mode	P42	Input	Hi-Z	Hi-Z
						During external extension and processor modes	BHW#	Output	Hi-Z	"H" level
137	EXCVCC	-	EXCVCC	-	-		EXCVCC	-	-	-
138	VSS	-	VSS	-	-		VSS	-	-	-
139	P43/RD#	P43	RD#	-	Input/output	During single-chip mode	P43	Input	Hi-Z	Hi-Z
						During external extension and processor modes	RD#	Output	Hi-Z	"H" level
140	P44/CS0#	P44	CS0#	-	Input/output	During single-chip mode	P44	Input	Hi-Z	Hi-Z
						During external extension and processor modes	CS0#	Output	Hi-Z	"H" level
141	P45/CS1#	P45	CS1#	-	Input/output	During single-chip mode	P45	Input	Hi-Z	Hi-Z
						During external extension and processor modes	CS1#	Output	Hi-Z	"H" level
142	P46/A13	P46	A13	-	Input/output	During single-chip mode	P46	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A13	Output	Hi-Z	Undefined
143	P47/A14	P47	A14	-	Input/output	During single-chip mode	P47	Input	Hi-Z	Hi-Z
						During external extension and processor modes	A14	Output	Hi-Z	Undefined
144	P220/CTX0	P220	CTX0	-	Input/output		P220	Input	Hi-Z	Hi-Z

This page is blank for reasons of layout.

CHAPTER 2

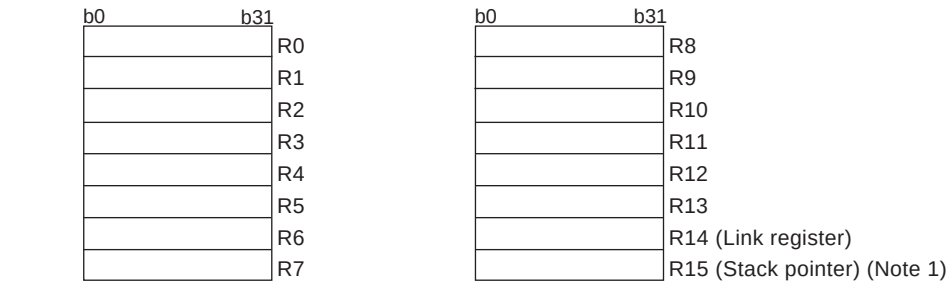
CPU

2.1 CPU Registers

The M32R contains 16 general-purpose registers, five control registers, an accumulator and a program counter. The accumulator is configured with 56 bits, and all other registers are 32 bits wide.

2.2 General-purpose Registers

The 16 general-purpose registers (R0–R15) are of 32-bit width and are used to retain data, base address, etc. R14 is used as the link register and R15 as the stack pointer. The link register is used to store the return address when executing a subroutine call instruction. The Interrupt Stack Pointer (SPI) and the User Stack Pointer (SPU) are alternately represented by R15 depending on the value of the Stack Mode (SM) bit in the Processor Status Word Register (PSW).
Upon exiting the reset state, the value of the general-purpose registers is undefined.

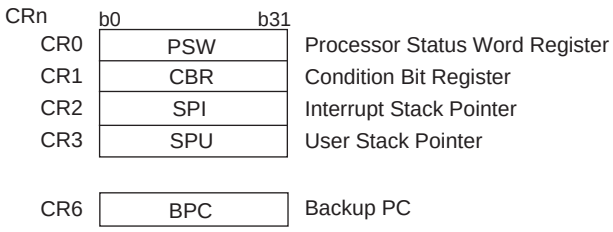


Note 1: The stack pointer functions as either the SPI or the SPU depending on the value of the SM bit in the PSW.

Figure 2.2.1 General-purpose Registers

2.3 Control Registers

There are 5 control registers which are the Processor Status Word Register (PSW), the Condition Bit Register (CBR), the Interrupt Stack Pointer (SPI), the User Stack Pointer (SPU) and the Backup PC (BPC).
The dedicated MVTC and MVFC instructions are used for writing and reading these control registers.

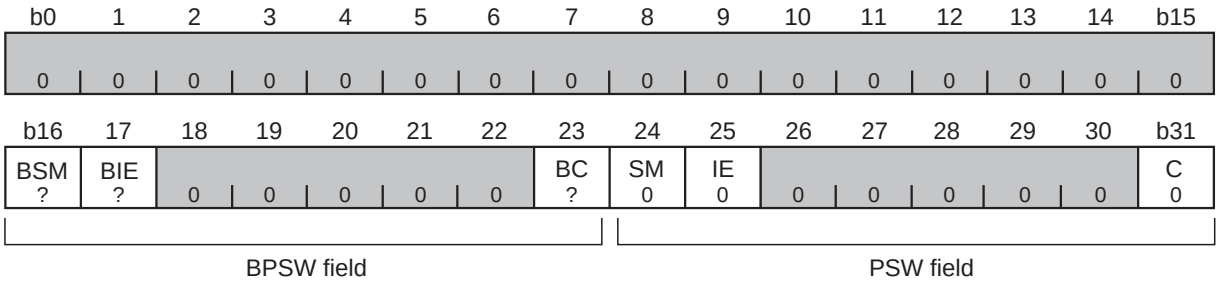


Notes: • CRn (n = 0-3 and 6) denotes the control register number.
• The dedicated MVTC and MVFC instructions are used for writing and reading these control registers.

Figure 2.3.1 Control Registers

2.3.1 Processor Status Word Register: PSW (CR0)

The Processor Status Word Register (PSW) indicates the M32R status. It consists of the PSW field which is regularly used, and the BPSW field where a copy of the PSW field is saved when an EIT occurs.
The PSW field consists of the Stack Mode (SM) bit, the Interrupt Enable (IE) bit and the Condition (C) bit.
The BPSW field consists of the Backup Stack Mode (BSM) bit, the Backup Interrupt Enable (BIE) bit and the Backup Condition (BC) bit.
Upon exiting the reset state, BSM, BIE and BC are undefined. All other bits are "0".



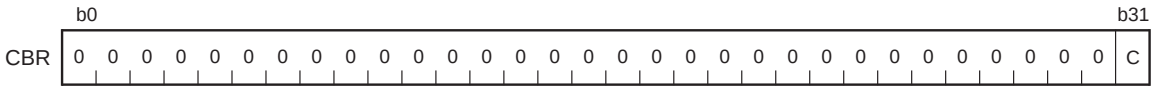
<Upon exiting reset: B'0000 0000 0000 0000 ??00 000? 0000 0000>

b	Bit Name	Function	R	W
0–15	No function assigned. Fix to "0".		0	0
16	BSM Backup SM Bit	Saves value of SM bit when EIT occurs	R	W
17	BIE Backup IE Bit	Saves value of IE bit when EIT occurs	R	W
18–22	No function assigned. Fix to "0".		0	0
23	BC Backup C Bit	Saves value of C bit when EIT occurs	R	W
24	SM Stack Mode Bit	0: Uses R15 as the interrupt stack pointer 1: Uses R15 as the user stack pointer	R	W
25	IE Interrupt Enable Bit (Note 1)	0: Does not accept interrupt 1: Accepts interrupt	R	W
26–30	No function assigned. Fix to "0".		0	0
31	C Condition Bit	Indicates carry, borrow or overflow resulting from operations (instruction dependent)	R	W

Note 1: Interruput which is controllable is External Interrupt (EI). Reserved Instruction Exception (RIE), Address Except (AE), Reset Interrupt (RI), System Break Interrupt (SBI) and Trap are not controlled.

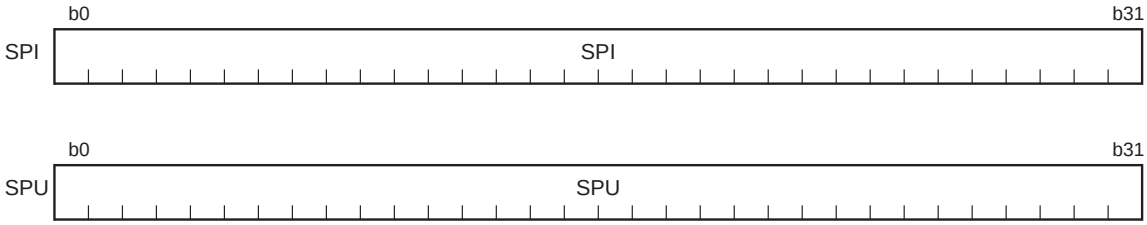
2.3.2 Condition Bit Register: CBR (CR1)

The Condition Bit Register (CBR) is derived from the PSW register by extracting its Condition (C) bit. The value written to the PSW register's C bit is reflected in this register. The register can only be read. (Writing to the register with the MVTC instruction is ignored.)
Upon exiting the reset state, the value of CBR is H'0000 0000.



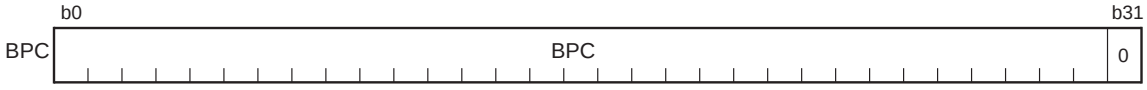
2.3.3 Interrupt Stack Pointer: SPI (CR2) and User Stack Pointer: SPU (CR3)

The Interrupt Stack Pointer (SPI) and the User Stack Pointer (SPU) retain the address of the current stack pointer. These registers can be accessed as the general-purpose register R15. R15 switches between representing the SPI and SPU depending on the value of the Stack Mode (SM) bit in the PSW.
Upon exiting the reset state, the values of the SPI and SPU are undefined.



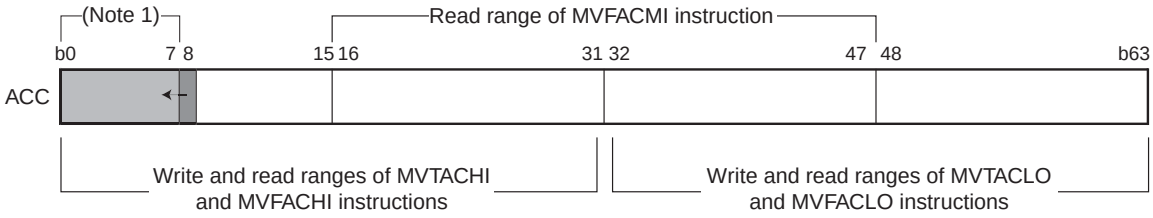
2.3.4 Backup PC: BPC (CR6)

The Backup PC (BPC) is used to save the value of the Program Counter (PC) when an EIT occurs. Bit 31 is fixed to "0".
When an EIT occurs, the register sets either the PC value immediately before the EIT occurred or the PC value for the next instruction. The BPC value is loaded to the PC when the RTE instruction is executed. However, the values of the lower 2 bits of the PC are always "00" when returned. (PC always returns to the word-aligned address.)
Upon exiting the reset state, the value of the BPC is undefined.



2.4 Accumulator

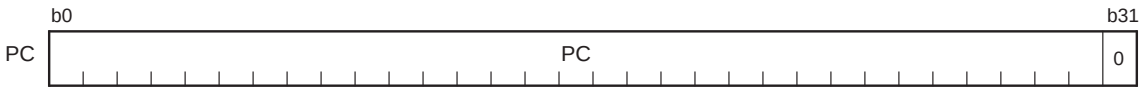
The Accumulator (ACC) is a 56-bit register used for DSP function instructions.
The accumulator is handled as a 64-bit register when accessed for read or write. When reading data from the accumulator, the value of bit 8 is sign-extended. When writing data to the accumulator, bits 0 to 7 are ignored. The accumulator is also used for the multiply instruction “MUL,” in which case the accumulator value is destroyed by instruction execution.
Use the MVTACHI and MVTACLO instructions for writing to the accumulator. The MVTACHI and MVTACLO instructions write data to the high-order 32 bits (bits 0–31) and the low-order 32 bits (bits 32–63), respectively.
Use the MVFACHI, MVFACLO and MVFACMI instructions for reading data from the accumulator. The MVFACHI, MVFACLO and MVFACMI instructions read data from the high-order 32 bits (bits 0–31), the low-order 32 bits (bits 32–63) and the middle 32 bits (bits 16–47), respectively.
Upon exiting the reset state, the value of accumulator is undefined.



Note 1: When read, bits 0 to 7 always show the sign-extended value of the value of bit 8. Writing to this bit field is ignored.

2.5 Program Counter

The Program Counter (PC) is a 32-bit counter that retains the address of the instruction being executed. Since the M32R instruction starts with even-numbered addresses, the LSB (bit 31) is always "0".
Upon exiting the reset state, the value of PC is H’0000 0000.



2.6 Data Formats

2.6.1 Data Types

The data types that can be handled by the M32R instruction set are signed or unsigned 8, 16 and 32-bit integers. The signed integers are represented by 2's complements.

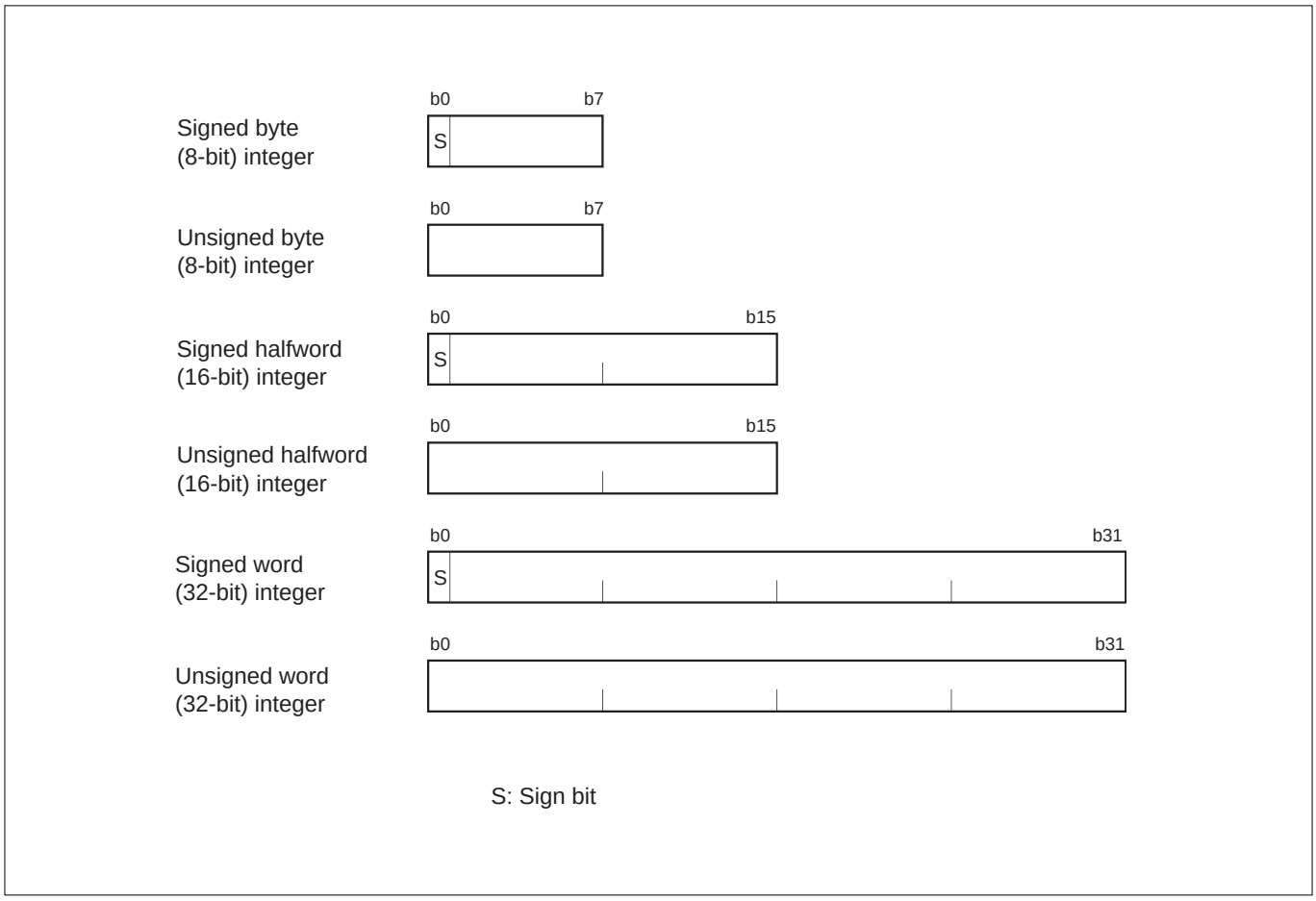


Figure 2.6.1 Data Types

2.6.2 Data Formats

(1) Data formats in registers

The data sizes in the M32R registers are always words (32 bits).
When loading byte (8-bit) or halfword (16-bit) data from memory into a register, the data is sign-extended (LDB, LDH instructions) or zero-extended (LDUB, LDUH instructions) to a word (32-bit) quantity before being loaded in the register.
When storing data from a register into a memory, the 32-bit data, the 16-bit data on the LSB side and the 8-bit data on the LSB side of the register are stored into memory by the ST, STH and STB instructions, respectively.

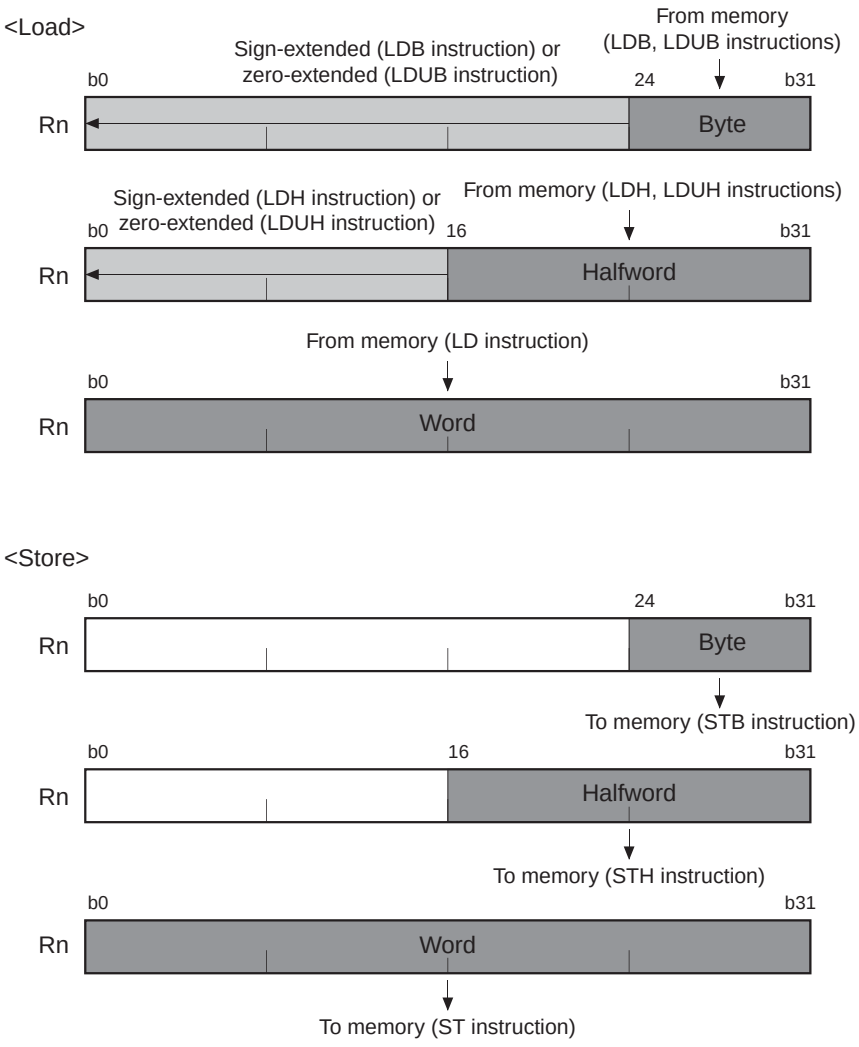


Figure 2.6.2 Data Formats in Registers

(2) Data formats in memory

The data sizes in memory can be byte (8 bits), halfword (16 bits) or word (32 bits). Although byte data can be located at any address, halfword and word data must be located at the addresses aligned with a halfword boundary (least significant address bit = "0") or a word boundary (two low-order address bits = "00"), respectively. If an attempt is made to access memory data that overlaps the halfword or word boundary, an address exception occurs.

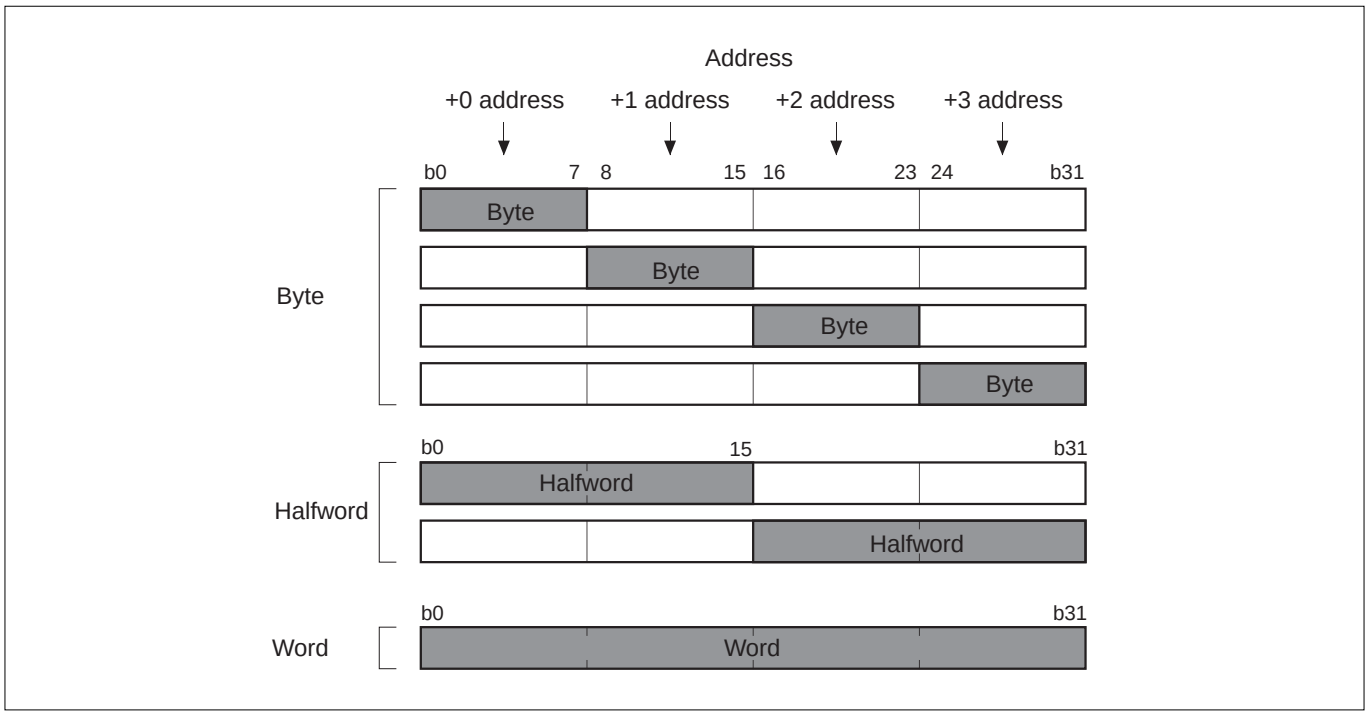


Figure 2.6.3 Data Formats in Memory

(3) Endian

The diagrams below show a general endian system and the endian adopted for the M32R family micro-computers.

	Bit endian (H'01)	Byte endian (H'01234567)
Big endian	B'0000001 b0 b7	H'01 H'23 H'45 H'67 HH HL LH LL
Little endian	B'0000001 b7 b0	H'67 H'45 H'23 H'01 LL LH HL HH

Note: • Even when bits are arranged in big endian, H'01 is not B'10000000.

Figure 2.6.4 General Endian System

Renesas microcomputer family name	7700 and M16C families		M32R family
Endian (bit/byte)	Little/little	Little/big	Big/big
Address	+0 +1 +2 +3	+0 +1 +2 +3	+0 +1 +2 +3
Data arrangement	LL LH HL HH	HH HL LH LL	HH HL LH LL
Bit number	7–0 15–8 23–16 31–24	31–24 23–16 15–8 7–0	0–7 8–15 16–23 24–31
Example: 0x01234567	.byte 67,45,23,01	.byte 01,23,45,67	.byte 01,23,45,67

Note: • The M32R family uses the big endian for both bits and bytes.

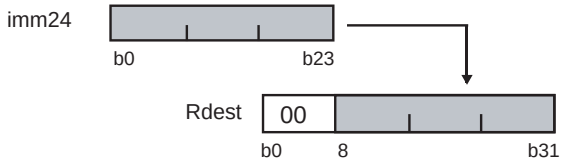
Figure 2.6.5 Endian Adopted for the M32R Family

(4) Transfer instructions

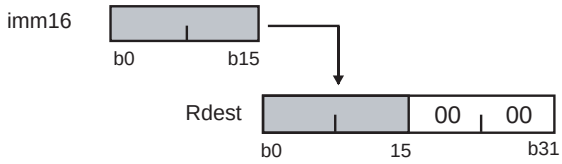
• Constant transfer

LD24 Rdest, #imm24
LDI Rdest, #imm16
LDI Rdest, #imm8
SETH Rdest, #imm16

LD24 Rdest, #imm24



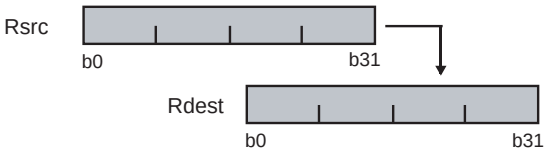
SETH Rdest, #imm16



• Register to register transfer

MV Rdest, Rsrc

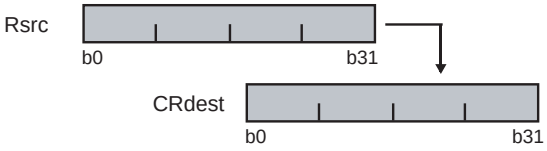
MV Rdest, Rsrc



• Control register transfer

MVFC Rdest, CRsrc
MVTC Rsrc, CRdest

MVTC Rsrc, CRdest



Note: • The condition bit C changes state when data is written to CR0 (PSW) using the MVTC instruction.

Figure 2.6.6 Transfer Instructions

(5) Transfer from memory (signed) to registers

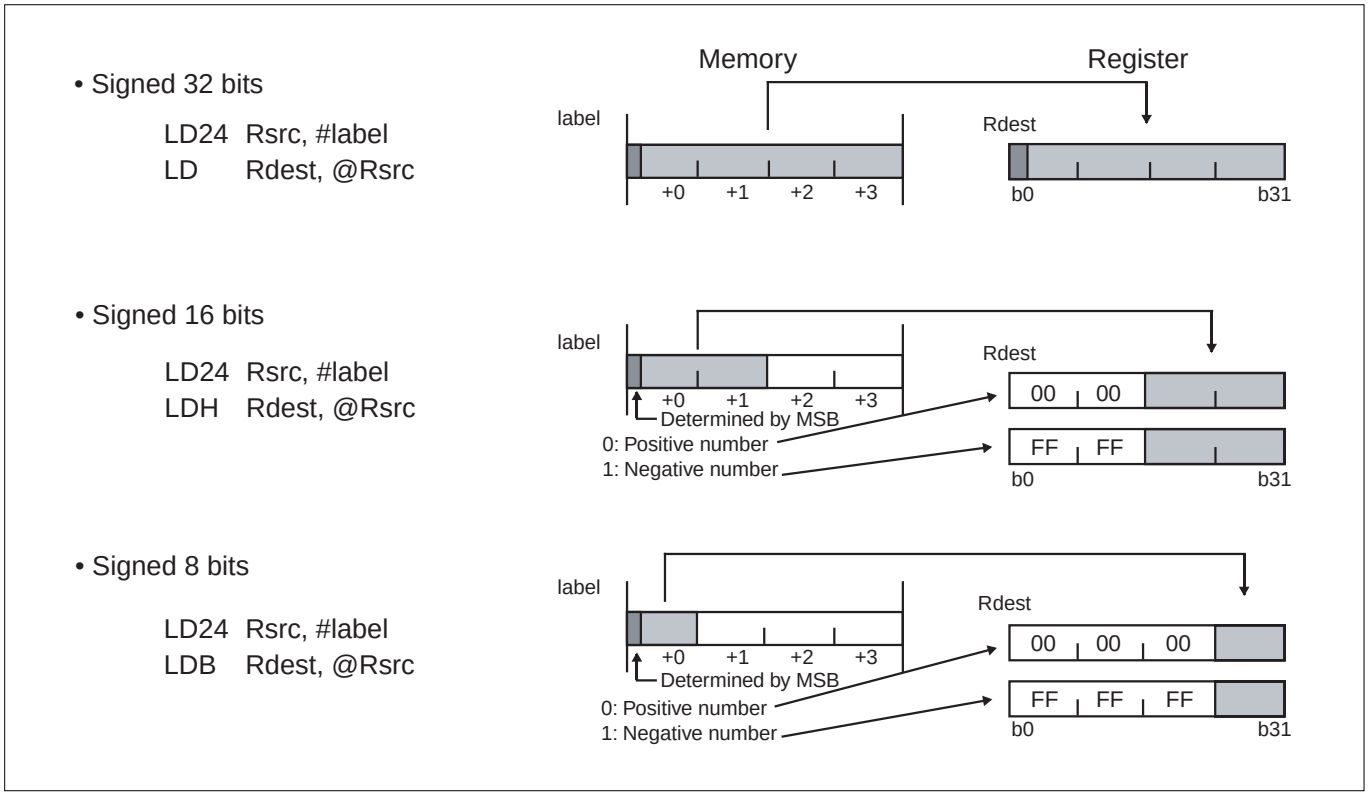


Figure 2.6.7 Transfer from Memory (Signed) to Registers

(6) Transfer from memory (unsigned) to registers

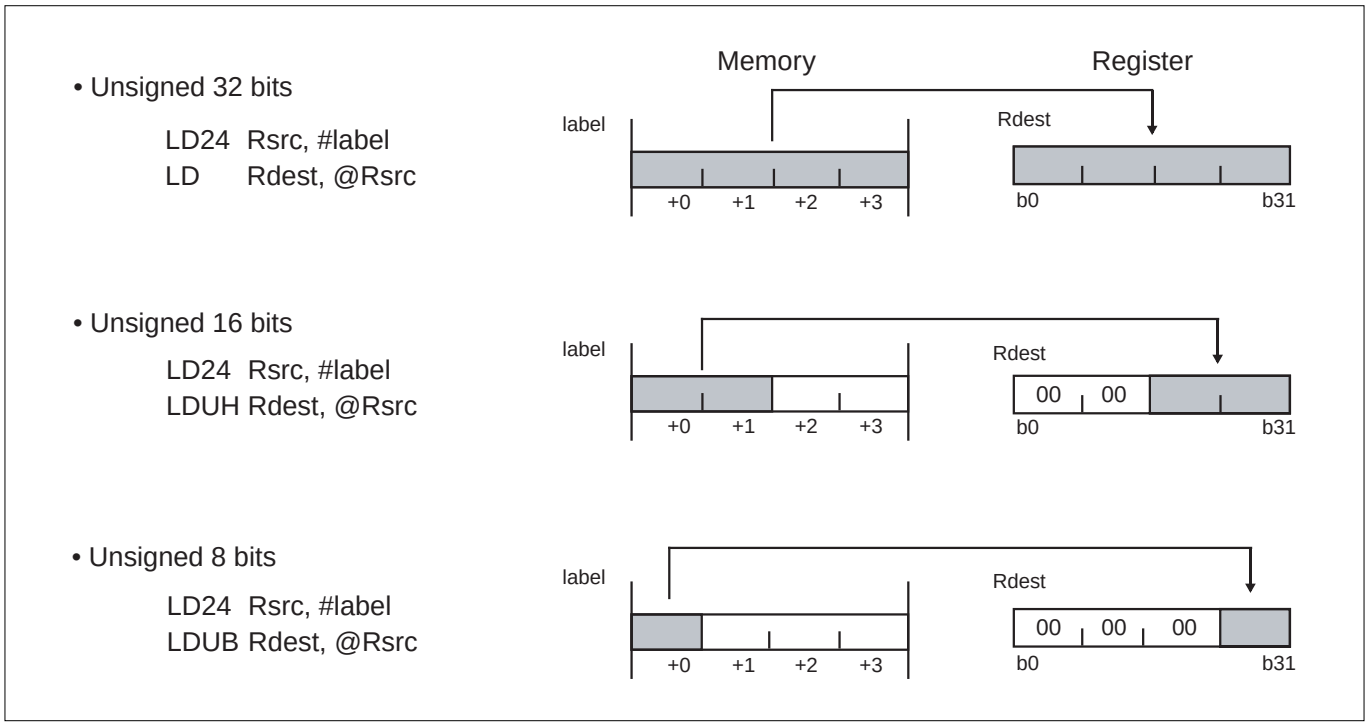


Figure 2.6.8 Transfer from Memory (Unsigned) to Registers

(7) Notes on data transfer

When transferring data, be aware that data arrangements in registers and memory are different.

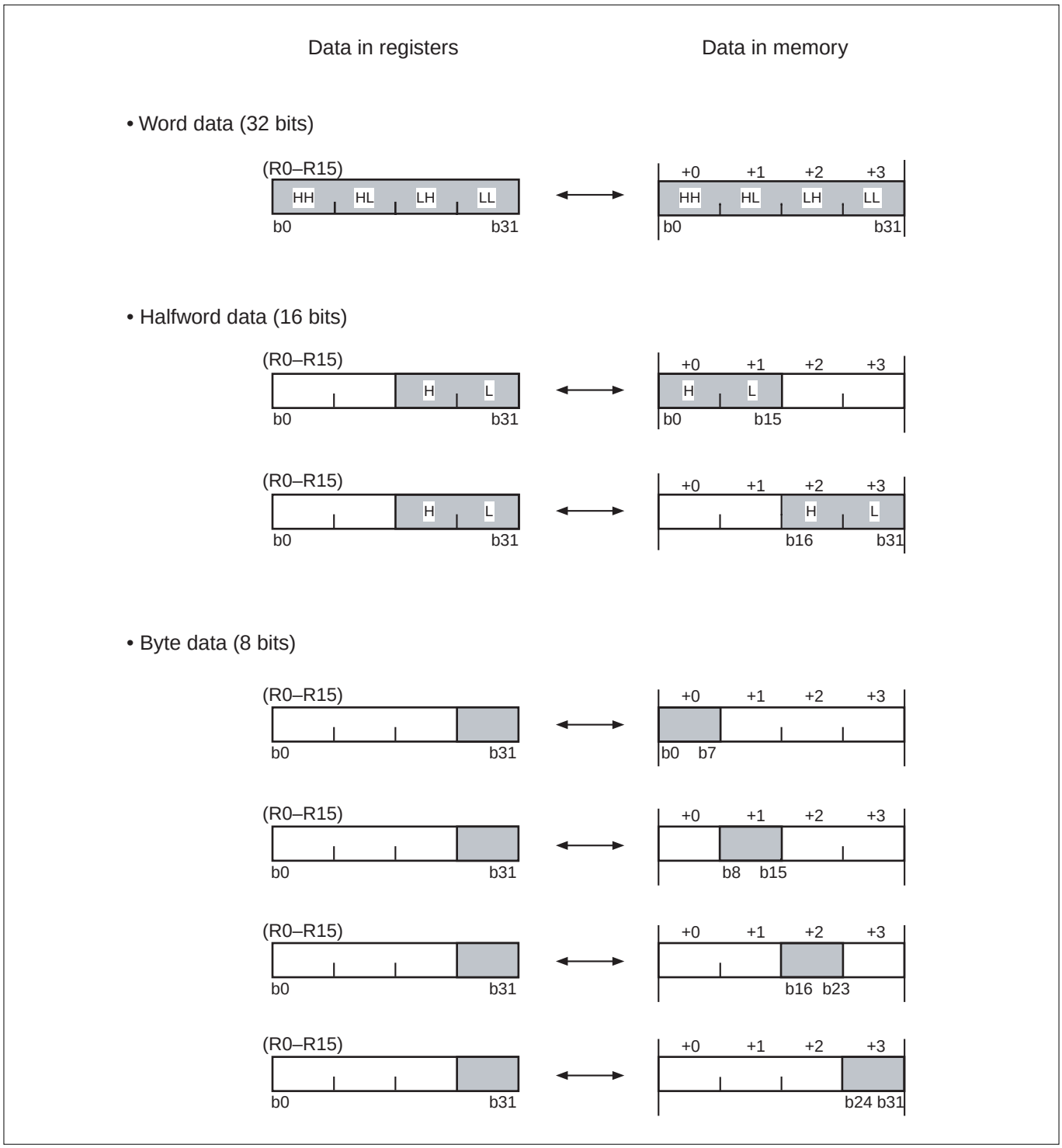


Figure 2.6.9 Difference in Data Arrangements

2.7 Supplementary Explanation for LOCK and UNLOCK Instruction Execution

The LOCK instruction sets the LOCK bit, as well as performs an ordinary load operation. The UNLOCK instruction is used to clear the LOCK bit.

The LOCK bit is located inside the CPU, and cannot directly be accessed for read or write by users. This bit controls granting of bus control requested by devices other than the CPU.

- When LOCK bit = "0"
  Control of the bus requested by devices other than the CPU is granted
- When LOCK bit = "1"
  Control of the bus requested by devices other than the CPU is denied

Control of the bus may be requested by devices other than the CPU in the following two cases:

- When DMA transfer is requested by the internal DMAC
- When HREQ# input is pulled low to request that the CPU be placed in a hold state

CHAPTER 3

ADDRESS SPACE

3.1 Outline of the Address Space

The logical addresses of the M32R are always handled in 32 bits, providing a linear address space of up to 4 Gbytes. The address space of the M32R consists of the following:

(1) User space

- Internal ROM area
- External extension area
- Internal RAM area
- SFR (Special Function Register) area

The 2 Gbytes from the address H'0000 0000 to the address H'7FFF FFFF comprise the user space. Located in this space are the internal ROM area, an external extension area, the internal RAM area and the SFR (Special Function Register) area (in which a set of internal peripheral I/O registers exist). Of these, the internal ROM and external extension areas are located differently depending on mode settings as will be described later.

(2) System space

The 2 Gbytes from the address H'8000 0000 to the address H'FFFF FFFF comprise the system space. This space is reserved for use by development tools such as an in-circuit emulator and debug monitor, and cannot be used by the user.

3.2 Operation Modes

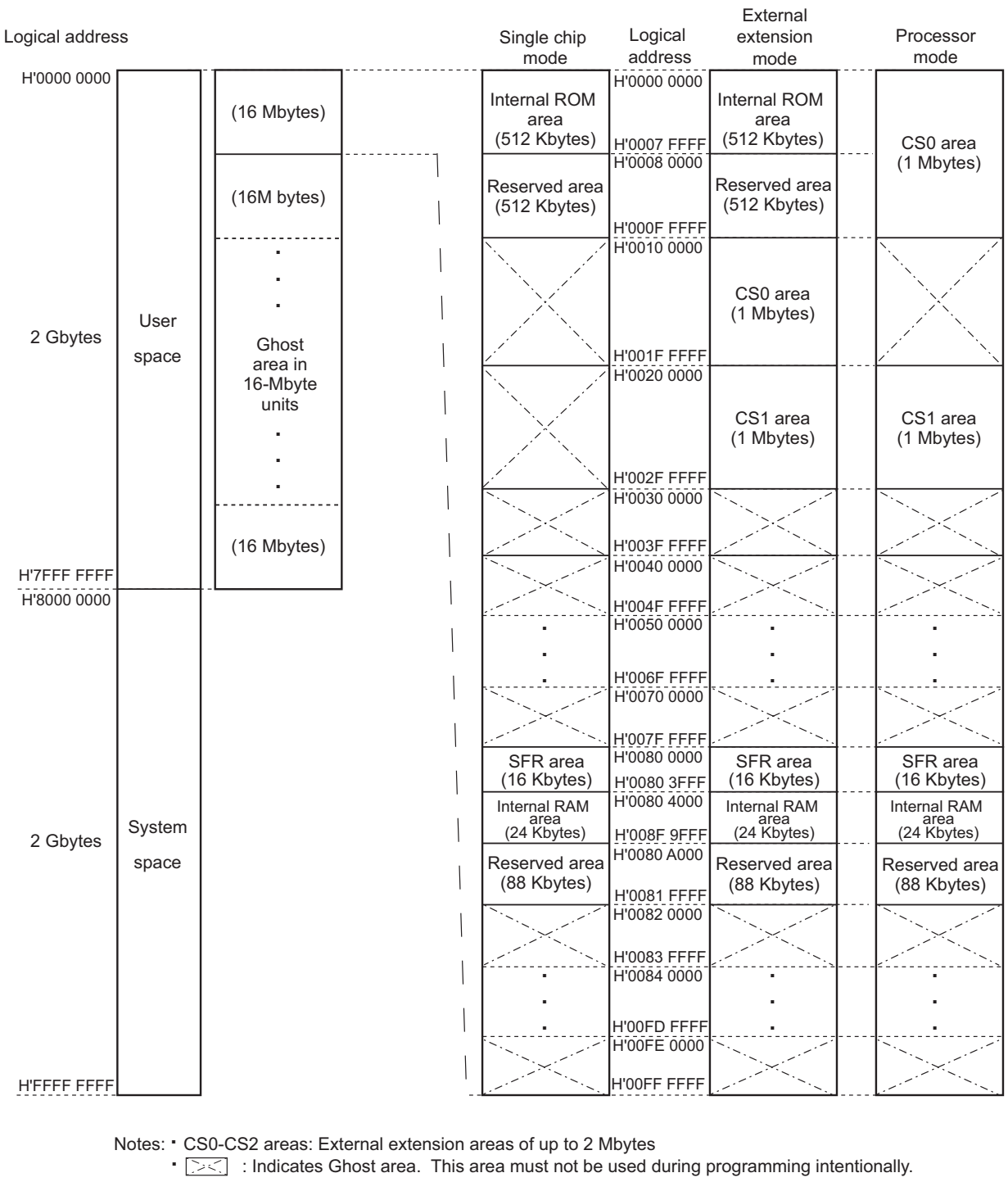
The microcomputer is placed in one of the following modes depending on how CPU operation mode is set by MOD0 and MOD1 pins. The operation mode used for rewriting the internal flash memory is described separately in Section 6.5, “Programming the Internal Flash Memory.”

Table 3.2.1 Operation Mode Settings

MOD0	MOD1	MOD2 (Note 1)	Operation mode (Note 2)
VSS	VSS	VSS	Single-chip mode
VSS	VCCE	VSS	External extension mode
VCCE	VSS	VSS	Processor mode (FP = VSS)
VCCE	VCCE	VSS	Reserved (use inhibited)
-	-	VCCE	Reserved (use inhibited)

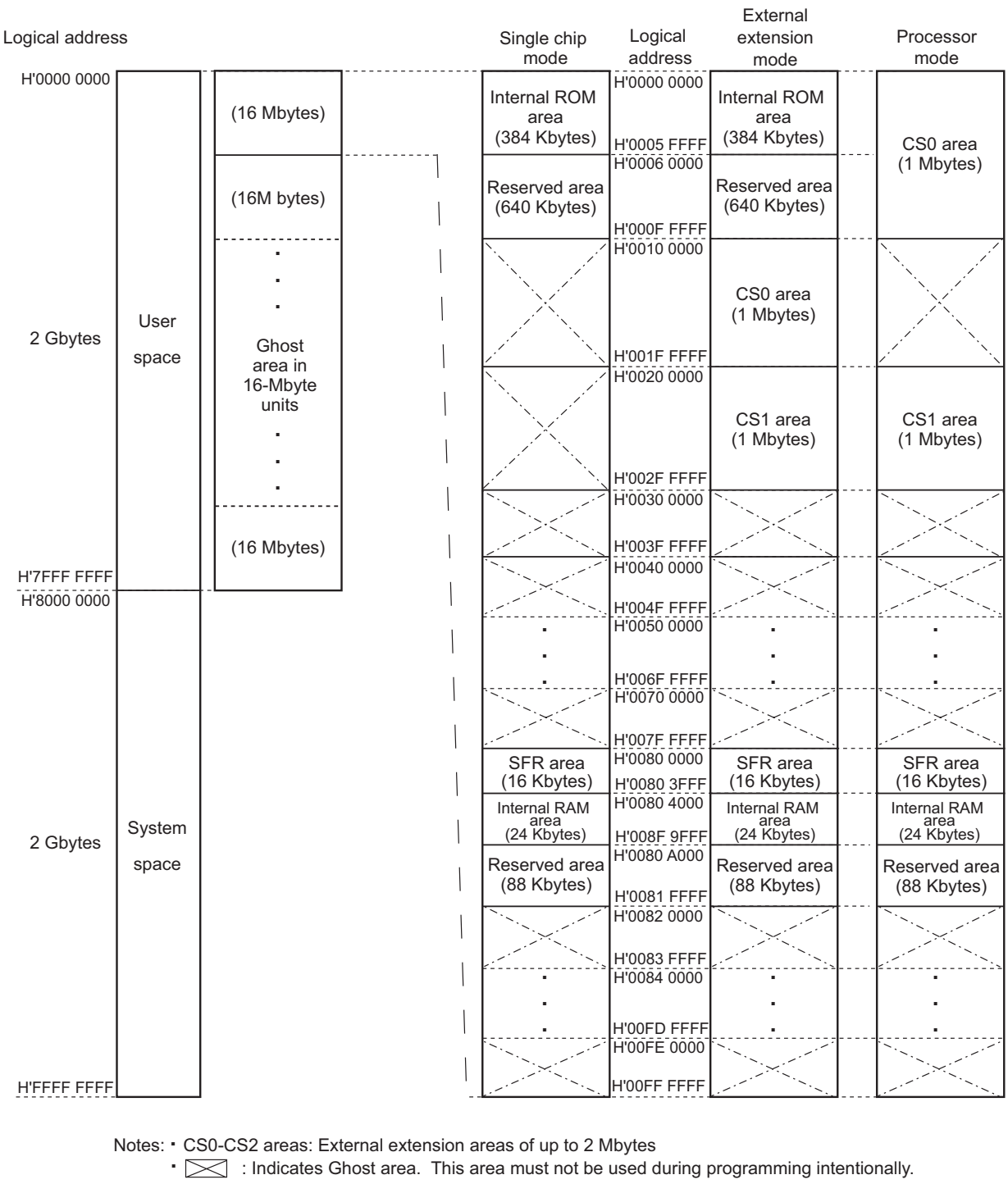
Note 1: Connect VCCE and VSS to the VCCE input power supply and ground, respectively.
Note 2: For the operation mode used to rewrite the internal flash memory (FP = VCCE) which is not shown in the above table, see Section 6.5, “Programming the Internal Flash Memory.”

The internal ROM and external extension areas are located differently depending on how operation mode is set. (All other areas in the address space are located the same way.) The diagram below shows how the internal ROM and external extension areas are mapped into the address space in each operation mode. (For flash rewrite mode, see Section 6.5, “Programming the Internal Flash Memory.”)



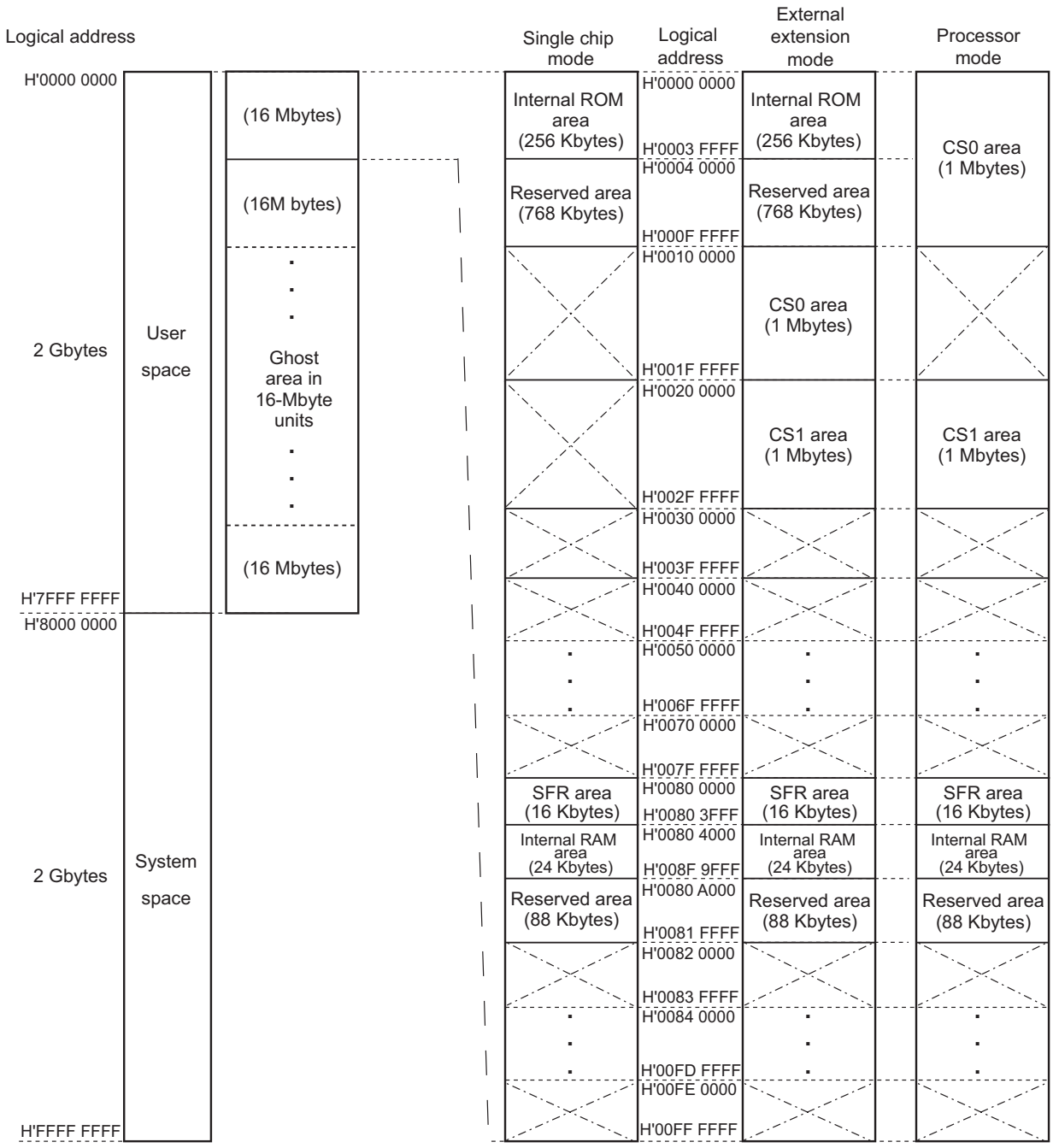
Notes: • CS0-CS2 areas: External extension areas of up to 2 Mbytes
• >< : Indicates Ghost area. This area must not be used during programming intentionally.

Figure 3.2.1 Address Space of the M32176F4



Notes: • CS0-CS2 areas: External extension areas of up to 2 Mbytes
• ⊠ : Indicates Ghost area. This area must not be used during programming intentionally.

Figure 3.2.2 Address Space of the M32176F3



Notes: • CS0-CS2 areas: External extension areas of up to 2 Mbytes
• : Indicates Ghost area. This area must not be used during programming intentionally.

Figure 3.2.3 Address Space of the M32176F2

3.3 Internal ROM and External Extension Areas

The 8-Mbyte area in the user space from the address H'0000 0000 to the address H'007F FFFF comprise the internal ROM and external extension areas. For the address mapping of these areas that differs with each operation mode, see Section 3.2, "Operation Modes."

3.3.1 Internal ROM Area

The internal ROM is allocated to the addresses shown below. Located at the beginning of this area is the EIT vector entry (and the ICU vector table).

Table 3.3.1 Internal ROM Allocation Address

Type Name	Size	Allocation Address
M32176F4	512 Kbytes	H'0000 0000 to H'0007 FFFF
M32176F3	384 kbytes	H'0000 0000 to H'0005 FFFF
M32176F2	256 Kbytes	H'0000 0000 to H'0003 FFFF

3.3.2 External Extension Area

The external extension area is only available when external extension or processor mode is selected by operation mode settings. When accessing the external extension area, the control signals necessary to access external devices are output.

The CS0# and CS1# signals are output corresponding to the address mapping of the external extension area. The CS0# and CS1# signals are output for the CS0 and CS1 areas, respectively.

Table 3.3.2 Address Mapping of the External Extension Area in Each Operation Mode

Operation Mode	Address Mapping of External Extension Area	
Single-chip mode	None	
External extension mode	H'0010 0000 to H'001F FFFF (CS0 area: 1 Mbyte)	
	H'0020 0000 to H'002F FFFF (CS1 area: 1 Mbyte)	(Note 1)
Processor mode	H'0000 0000 to H'000F FFFF (CS0 area: 1 Mbyte)	(Note 2)
	H'0020 0000 to H'002F FFFF (CS1 area: 1 Mbyte)	(Note 2)

Note 1: During external extension mode, a ghost (1 Mbyte) of the CS1 area appears in an area of H'0030 0000 through H'003F FFFF.

Note 2: During processor mode, a ghost (1 Mbyte) of the CS0 area appears in an area of H'0010 0000 through H'001F FFFF and a ghost (1 Mbyte) of the CS1 area appears in an area of H'0030 0000 through H'003F FFFF.

3.4 Internal RAM and SFR Areas

The 8-Mbyte area from the address H'0080 0000 to the address H'00FF FFFF comprise the internal RAM and SFR (Special Function Register) areas. Of these, the space that the user can actually use is a 128-Kbyte area from the address H'0080 0000 to the address H'0081 FFFF. The other areas here are ghosts in 128-Kbyte units. (Do not use the ghost area intentionally during programming.)

3.4.1 Internal RAM Area

The internal RAM area is allocated to the addresses shown below.

Table 3.4.1 Internal RAM Allocation Address

Type Name	Size	Allocation Address
M32176F4	24 Kbytes	H'0080 4000 to H'0080 9FFF
M32176F3	24 Kbytes	H'0080 4000 to H'0080 9FFF
M32176F2	24 Kbytes	H'0080 4000 to H'0080 9FFF

3.4.2 SFR (Special Function Register) Area

The addresses H'0080 0000 to H'0080 3FFFF comprise the SFR (Special Function Register) area. Located in this area are the internal peripheral I/O registers.

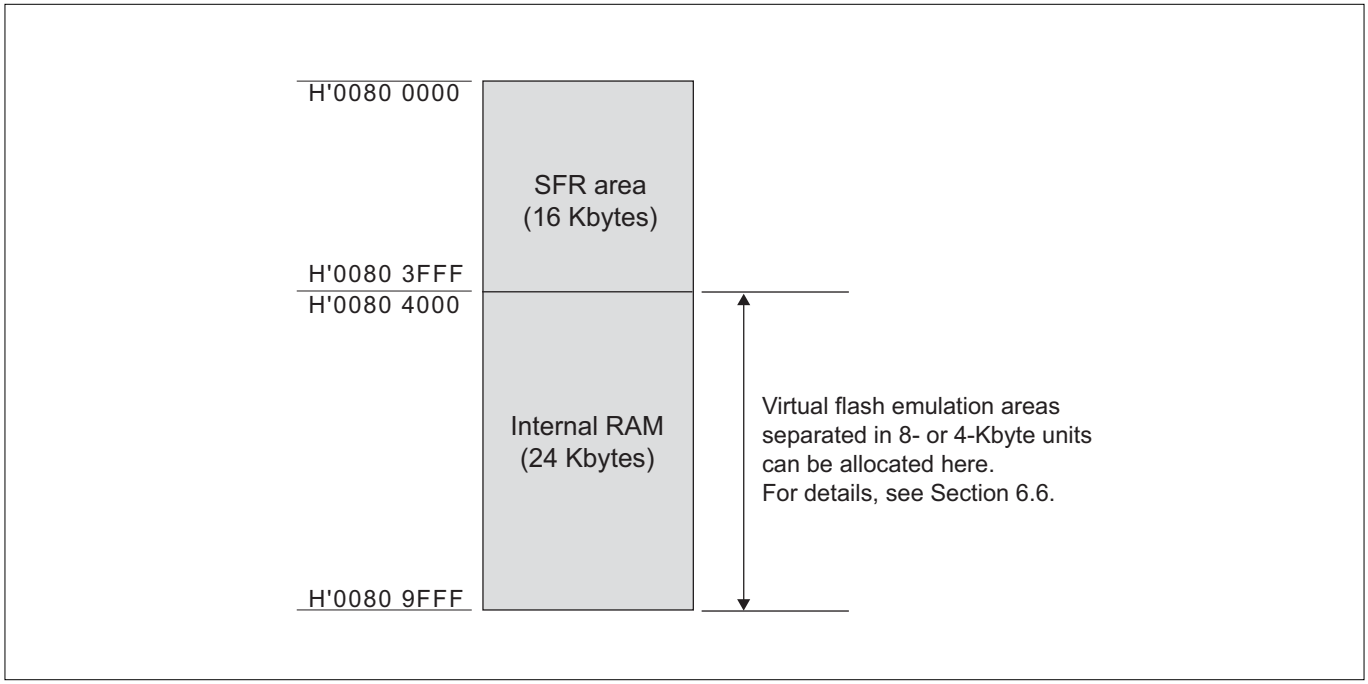
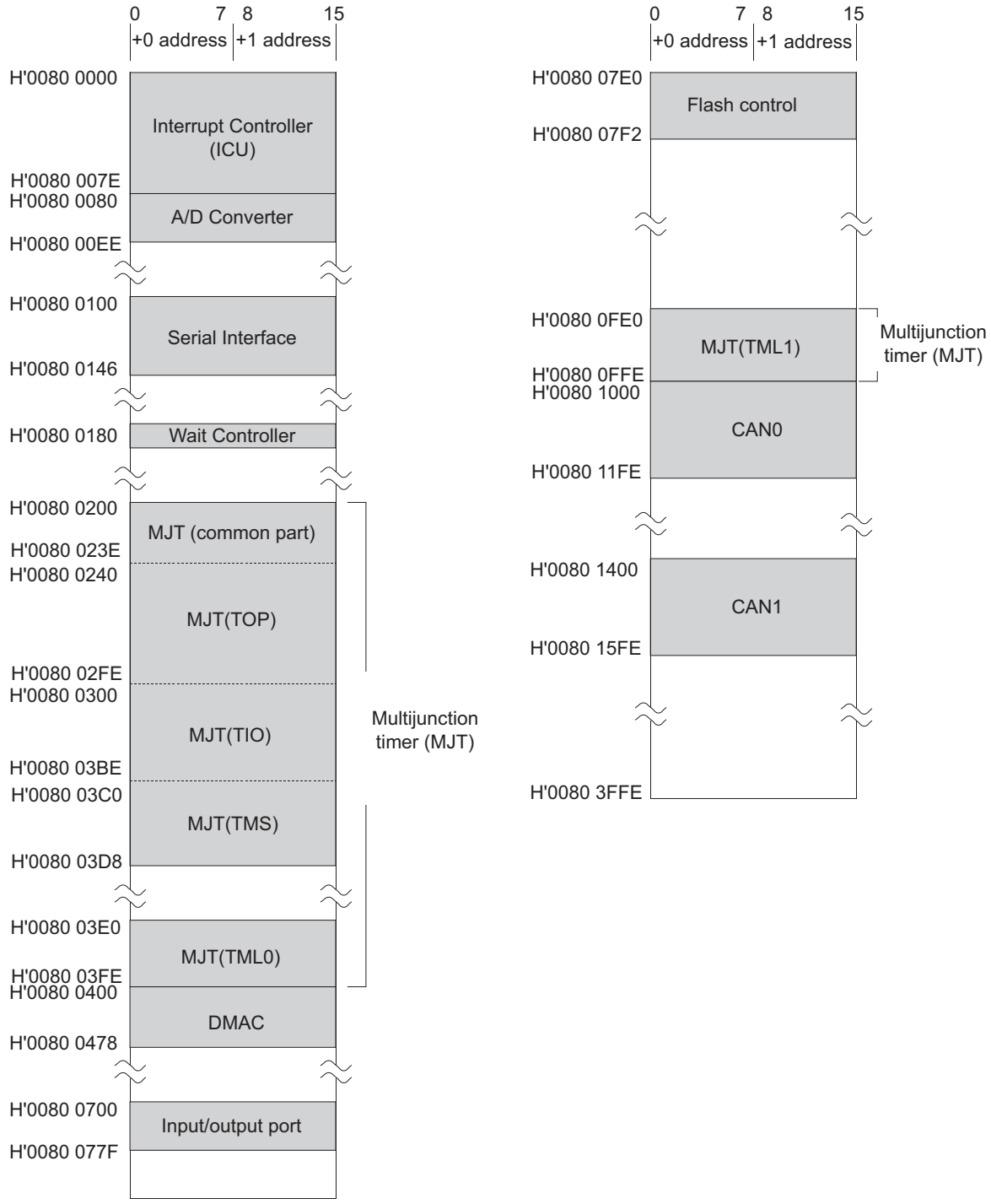


Figure 3.4.1 Internal RAM and SFR (Special Function Register) Areas



Note: • The Real-time Debugger (RTD) is an independent module that is operated from the outside, and is transparent to the CPU.

Figure 3.4.2 Outline Mapping of the SFR Area

SFR Area Register Map (1/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0000	Interrupt Vector Register (IVECT)		5-5
H'0080 0002	(Use inhibited area)		
H'0080 0004	Interrupt Request Mask Register (IMASK)	(Use inhibited area)	5-6
H'0080 0006	SBI Control Register (SBICR)	(Use inhibited area)	5-7
	(Use inhibited area)		
H'0080 0060	CAN0 Transmit/Receive & Error Interrupt Control Register (ICAN0CR)	(Use inhibited area)	5-8
H'0080 0062	(Use inhibited area)		
H'0080 0064	(Use inhibited area)		
H'0080 0066	(Use inhibited area)	RTD Interrupt Control Register (IRTDCR)	5-8
H'0080 0068	SIO2, 3 Transmit/Receive Interrupt Control Register (ISIO23CR)	DMA5–9 Interrupt Control Register (IDMA59CR)	5-8
H'0080 006A	(Use inhibited area)		
H'0080 006C	A/D0 Conversion Interrupt Control Register (IAD0CCR)	SIO0 Transmit Interrupt Control Register (ISIO0TXCR)	5-8
H'0080 006E	SIO0 Receive Interrupt Control Register (ISIO0RXCR)	SIO1 Transmit Interrupt Control Register (ISIO1TXCR)	5-8
H'0080 0070	SIO1 Receive Interrupt Control Register (ISIO1RXCR)	DMA0–4 Interrupt Control Register (IDMA04CR)	5-8
H'0080 0072	MJT Output Interrupt Control Register 0 (IMJTOCR0)	MJT Output Interrupt Control Register 1 (IMJTOCR1)	5-8
H'0080 0074	MJT Output Interrupt Control Register 2 (IMJTOCR2)	MJT Output Interrupt Control Register 3 (IMJTOCR3)	5-8
H'0080 0076	MJT Output Interrupt Control Register 4 (IMJTOCR4)	MJT Output Interrupt Control Register 5 (IMJTOCR5)	5-8
H'0080 0078	MJT Output Interrupt Control Register 6 (IMJTOCR6)	MJT Output Interrupt Control Register 7 (IMJTOCR7)	5-8
H'0080 007A	(Use inhibited area)	MJT Input Interrupt Control Register 1 (IMJTICR1)	5-8
H'0080 007C	MJT Input Interrupt Control Register 2 (iIMJTICR2)	MJT Input Interrupt Control Register 3 (IMJTICR3)	5-8
H'0080 007E	MJT Input Interrupt Control Register 4 (IMJTICR4)	CAN1 Transmit/Receive & Error Interrupt Control Register (ICAN1CR)	5-8
H'0080 0080	A/D0 Single Mode Register 0 (AD0SIM0)	A/D0 Single Mode Register 1 (AD0SIM1)	11-14 11-16
H'0080 0082	(Use inhibited area)		
H'0080 0084	A/D0 Scan Mode Register 0 (AD0SCM0)	A/D0 Scan Mode Register 1 (AD0SCM1)	11-18 11-20
H'0080 0086	A/D0 Disconnection Detection Assist Function Control Register (AD0DDACR)	A/D0 Conversion Speed Control Register (AD0CVSCR)	11-23 11-22
H'0080 0088	A/D0 Successive Approximation Register (AD0SAR)		11-27
H'0080 008A	A/D0 Disconnection Detection Assist Method Select Register (AD0DDASEL)		11-24
H'0080 008C	A/D0 Comparate Data Register (AD0CMP)		11-28
H'0080 008E	(Use inhibited area)		
H'0080 0090	10-bit A/D0 Data Register 0 (AD0DT0)		11-29
H'0080 0092	10-bit A/D0 Data Register 1 (AD0DT1)		11-29
H'0080 0094	10-bit A/D0 Data Register 2 (AD0DT2)		11-29
H'0080 0096	10-bit A/D0 Data Register 3 (AD0DT3)		11-29
H'0080 0098	10-bit A/D0 Data Register 4 (AD0DT4)		11-29
H'0080 009A	10-bit A/D0 Data Register 5 (AD0DT5)		11-29

SFR Area Register Map (2/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 009C	10-bit A/D0 Data Register 6 (AD0DT6)		11-29
H'0080 009E	10-bit A/D0 Data Register 7 (AD0DT7)		11-29
H'0080 00A0	10-bit A/D0 Data Register 8 (AD0DT8)		11-29
H'0080 00A2	10-bit A/D0 Data Register 9 (AD0DT9)		11-29
H'0080 00A4	10-bit A/D0 Data Register 10 (AD0DT10)		11-29
H'0080 00A6	10-bit A/D0 Data Register 11 (AD0DT11)		11-29
H'0080 00A8	10-bit A/D0 Data Register 12 (AD0DT12)		11-29
H'0080 00AA	10-bit A/D0 Data Register 13 (AD0DT13)		11-29
H'0080 00AC	10-bit A/D0 Data Register 14 (AD0DT14)		11-29
H'0080 00AE	10-bit A/D0 Data Register 15 (AD0DT15)		11-29
H'0080 00D0	(Use inhibited area)	8-bit A/D0 Data Register 0 (AD08DT0)	11-30
H'0080 00D2	(Use inhibited area)	8-bit A/D0 Data Register 1 (AD08DT1)	11-30
H'0080 00D4	(Use inhibited area)	8-bit A/D0 Data Register 2 (AD08DT2)	11-30
H'0080 00D6	(Use inhibited area)	8-bit A/D0 Data Register 3 (AD08DT3)	11-30
H'0080 00D8	(Use inhibited area)	8-bit A/D0 Data Register 4 (AD08DT4)	11-30
H'0080 00DA	(Use inhibited area)	8-bit A/D0 Data Register 5 (AD08DT5)	11-30
H'0080 00DC	(Use inhibited area)	8-bit A/D0 Data Register 6 (AD08DT6)	11-30
H'0080 00DE	(Use inhibited area)	8-bit A/D0 Data Register 7 (AD08DT7)	11-30
H'0080 00E0	(Use inhibited area)	8-bit A/D0 Data Register 8 (AD08DT8)	11-30
H'0080 00E2	(Use inhibited area)	8-bit A/D0 Data Register 9 (AD08DT9)	11-30
H'0080 00E4	(Use inhibited area)	8-bit A/D0 Data Register 10 (AD08DT10)	11-30
H'0080 00E6	(Use inhibited area)	8-bit A/D0 Data Register 11 (AD08DT11)	11-30
H'0080 00E8	(Use inhibited area)	8-bit A/D0 Data Register 12 (AD08DT12)	11-30
H'0080 00EA	(Use inhibited area)	8-bit A/D0 Data Register 13 (AD08DT13)	11-30
H'0080 00EC	(Use inhibited area)	8-bit A/D0 Data Register 14 (AD08DT14)	11-30
H'0080 00EE	(Use inhibited area)	8-bit A/D0 Data Register 15 (AD08DT15)	11-30
	(Use inhibited area)		
H'0080 0100	SIO23 Interrupt Request Status Register (SI23STAT)	SIO03 Interrupt Request Mask Register (SI03MASK)	12-9 12-10
H'0080 0102	SIO03 Interrupt Request Source Select Register (SI03SEL)	(Use inhibited area)	12-11
	(Use inhibited area)		
H'0080 0110	SIO0 Transmit Control Register (S0TCNT)	SIO0 Transmit/Receive Mode Register (S0MOD)	12-13 12-14
H'0080 0112	SIO0 Transmit Buffer Register (S0TXB)		12-17
H'0080 0114	SIO0 Receive Buffer Register (S0RXB)		12-18
H'0080 0116	SIO0 Receive Control Register (S0RCNT)	SIO0 Baud Rate Register (S0BAUR)	12-19 12-21

SFR Area Register Map (3/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0118	SIO0 Special Mode Register (S0SMOD)	(Use inhibited area)	12-23
	(Use inhibited area)		
H'0080 0120	SIO1 Transmit Control Register (S1TCNT)	SIO1 Transmit/Receive Mode Register (S1MOD)	12-13 12-14
H'0080 0122	SIO1 Transmit Buffer Register (S1TXB)		12-17
H'0080 0124	SIO1 Receive Buffer Register (S1RXB)		12-18
H'0080 0126	SIO1 Receive Control Register (S1RCNT)	SIO1 Baud Rate Register (S1BAUR)	12-19 12-21
H'0080 0128	SIO1 Special Mode Register (S1SMOD)	(Use inhibited area)	12-23
	(Use inhibited area)		
H'0080 0130	SIO2 Transmit Control Register (S2TCNT)	SIO2 Transmit/Receive Mode Register (S2MOD)	12-13 12-14
H'0080 0132	SIO2 Transmit Buffer Register (S2TXB)		12-17
H'0080 0134	SIO2 Receive Buffer Register (S2RXB)		12-18
H'0080 0136	SIO2 Receive Control Register (S2RCNT)	SIO2 Baud Rate Register (S2BAUR)	12-19 12-21
	(Use inhibited area)		
H'0080 0140	SIO3 Transmit Control Register (S3TCNT)	SIO3 Transmit/Receive Mode Register (S3MOD)	12-13 12-14
H'0080 0142	SIO3 Transmit Buffer Register (S3TXB)		12-17
H'0080 0144	SIO3 Receive Buffer Register (S3RXB)		12-18
H'0080 0146	SIO3 Receive Control Register (S3RCNT)	SIO3 Baud Rate Register (S3BAUR)	12-19 12-21
	(Use inhibited area)		
H'0080 0180	Wait Cycles Control Register (WTCCR)	(Use inhibited area)	16-4
	(Use inhibited area)		
H'0080 0200	(Use inhibited area)	Clock Bus & Input Event Bus Control Register (CKIEBCR)	10-13
H'0080 0202	Prescaler Register 0 (PRS0)	Prescaler Register 1 (PRS1)	10-9
H'0080 0204	Prescaler Register 2 (PRS2)	Output Event Bus Control Register (OEBCR)	10-9 10-14
	(Use inhibited area)		
H'0080 0210	TCLK Input Processing Control Register (TCLKCR)		10-17
H'0080 0212	TIN Input Processing Control Register 0 (TINCR0)		10-18
H'0080 0214	(Use inhibited area)		
H'0080 0216	(Use inhibited area)		
H'0080 0218	TIN Input Processing Control Register 3 (TINCR3)		10-19
H'0080 021A	TIN Input Processing Control Register 4 (TINCR4)		10-19
H'0080 021C	(Use inhibited area)		
H'0080 021E	(Use inhibited area)		
H'0080 0220	F/F Source Select Register 0 (FFS0)		10-21
H'0080 0222	(Use inhibited area)	F/F Source Select Register 1 (FFS1)	10-22
H'0080 0224	F/F Protect Register 0 (FFP0)		10-23

SFR Area Register Map (4/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0226	F/F Data Register 0 (FFD0)		10-24
H'0080 0228	(Use inhibited area)	F/F Protect Register 1 (FFP1)	10-23
H'0080 022A	(Use inhibited area)	F/F Data Register 1 (FFD1)	10-24
|	(Use inhibited area)		
H'0080 0230	TOP Interrupt Control Register 0 (TOPIR0)	TOP Interrupt Control Register 1 (TOPIR1)	10-29
H'0080 0232	TOP Interrupt Control Register 2 (TOPIR2)	TOP Interrupt Control Register 3 (TOPIR3)	10-31 10-32
H'0080 0234	TIO Interrupt Control Register 0 (TIOIR0)	TIO Interrupt Control Register 1 (TIOIR1)	10-33 10-34
H'0080 0236	TIO Interrupt Control Register 2 (TIOIR2)	TMS Interrupt Control Register (TMSIR)	10-35 10-36
H'0080 0238	TIN Interrupt Control Register 0 (TINIR0)	TIN Interrupt Control Register 1 (TINIR1)	10-37 10-38
H'0080 023A	(Use inhibited area)		
H'0080 023C	TIN Interrupt Control Register 4 (TINIR4)	TIN Interrupt Control Register 5 (TINIR5)	10-39
H'0080 023E	TIN Interrupt Control Register 6 (TINIR6)	(Use inhibited area)	10-41
H'0080 0240	TOP0 Counter (TOP0CT)		10-53
H'0080 0242	TOP0 Reload Register (TOP0RL)		10-54
H'0080 0244	(Use inhibited area)		
H'0080 0246	TOP0 Correction Register (TOP0CC)		10-55
|	(Use inhibited area)		
H'0080 0250	TOP1 Counter (TOP1CT)		10-53
H'0080 0252	TOP1 Reload Register (TOP1RL)		10-54
H'0080 0254	(Use inhibited area)		
H'0080 0256	TOP1 Correction Register (TOP1CC)		10-55
|	(Use inhibited area)		
H'0080 0260	TOP2 Counter (TOP2CT)		10-53
H'0080 0262	TOP2 Reload Register (TOP2RL)		10-54
H'0080 0264	(Use inhibited area)		
H'0080 0266	TOP2 Correction Register (TOP2CC)		10-55
|	(Use inhibited area)		
H'0080 0270	TOP3 Counter (TOP3CT)		10-53
H'0080 0272	TOP3 Reload Register (TOP3RL)		10-54
H'0080 0274	(Use inhibited area)		
H'0080 0276	TOP3 Correction Register (TOP3CC)		10-55
|	(Use inhibited area)		
H'0080 0280	TOP4 Counter (TOP4CT)		10-53
H'0080 0282	TOP4 Reload Register (TOP4RL)		10-54
H'0080 0284	(Use inhibited area)		

SFR Area Register Map (5/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0286	TOP4 Correction Register (TOP4CC)		10-55
	(Use inhibited area)		
H'0080 0290	TOP5 Counter (TOP5CT)		10-53
H'0080 0292	TOP5 Reload Register (TOP5RL)		10-54
H'0080 0294	(Use inhibited area)		
H'0080 0296	TOP5 Correction Register (TOP5CC)		10-55
H'0080 0298	(Use inhibited area)		
H'0080 029A	TOP0–5 Control Register 0 (TOP05CR0)		10-49
H'0080 029C	(Use inhibited area)	TOP0–5 Control Register 1 (TOP05CR1)	10-49
H'0080 029E	(Use inhibited area)		
H'0080 02A0	TOP6 Counter (TOP6CT)		10-53
H'0080 02A2	TOP6 Reload Register (TOP6RL)		10-54
H'0080 02A4	(Use inhibited area)		
H'0080 02A6	TOP6 Correction Register (TOP6CC)		10-55
H'0080 02A8	(Use inhibited area)		
H'0080 02AA	TOP6, 7 Control Register (TOP67CR)		10-51
	(Use inhibited area)		
H'0080 02B0	TOP7 Counter (TOP7CT)		10-53
H'0080 02B2	TOP7 Reload Register (TOP7RL)		10-54
H'0080 02B4	(Use inhibited area)		
H'0080 02B6	TOP7 Correction Register (TOP7CC)		10-55
	(Use inhibited area)		
H'0080 02C0	TOP8 Counter (TOP8CT)		10-53
H'0080 02C2	TOP8 Reload Register (TOP8RL)		10-54
H'0080 02C4	(Use inhibited area)		
H'0080 02C6	TOP8 Correction Register (TOP8CC)		10-55
	(Use inhibited area)		
H'0080 02D0	TOP9 Counter (TOP9CT)		10-53
H'0080 02D2	TOP9 Reload Register (TOP9RL)		10-54
H'0080 02D4	(Use inhibited area)		
H'0080 02D6	TOP9 Correction Register (TOP9CC)		10-55
	(Use inhibited area)		
H'0080 02E0	TOP10 Counter (TOP10CT)		10-53
H'0080 02E2	TOP10 Reload Register (TOP10RL)		10-54
H'0080 02E4	(Use inhibited area)		

SFR Area Register Map (6/22)

Address	+0 address b0 b7	+1address b8 b15	See pages
H'0080 02E6	TOP10 Correction Register (TOP10CC)		10-55
H'0080 02E8	(Use inhibited area)		
H'0080 02EA	TOP8–10 Control Register (TOP810CR)		10-52
	(Use inhibited area)		
H'0080 02FA	TOP0-10 External Enable Permit Register (TOPEEN)		10-56
H'0080 02FC	TOP0-10 Enable Protect Register (TOPPRO)		10-56
H'0080 02FE	TOP0-10 Count Enable Register (TOPCEN)		10-57
H'0080 0300	TIO0 Counter (TIO0CT)		10-87
H'0080 0302	(Use inhibited area)		
H'0080 0304	TIO0 Reload 1 Register (TIO0RL1)		10-89
H'0080 0306	TIO0 Reload 0/ Measure Register (TIO0RL0)		10-88
	(Use inhibited area)		
H'0080 0310	TIO1 Counter (TIO1CT)		10-87
H'0080 0312	(Use inhibited area)		
H'0080 0314	TIO1 Reload 1 Register (TIO1RL1)		10-89
H'0080 0316	TIO1 Reload 0/ Measure Register (TIO1RL0)		10-88
H'0080 0318	(Use inhibited area)		
H'0080 031A	TIO0–3 Control Register 0 (TIO03CR0)		10-80
H'0080 031C	(Use inhibited area)	TIO0–3 Control Register 1 (TIO03CR1)	10-81
H'0080 031E	(Use inhibited area)		
H'0080 0320	TIO2 Counter (TIO2CT)		10-87
H'0080 0322	(Use inhibited area)		
H'0080 0324	TIO2 Reload 1 Register (TIO2RL1)		10-89
H'0080 0326	TIO2 Reload 0/ Measure Register (TIO2RL0)		10-88
	(Use inhibited area)		
H'0080 0330	TIO3 Counter (TIO3CT)		10-87
H'0080 0332	(Use inhibited area)		
H'0080 0334	TIO3 Reload 1 Register (TIO3RL1)		10-89
H'0080 0336	TIO3 Reload 0/ Measure Register (TIO3RL0)		10-88
	(Use inhibited area)		
H'0080 0340	TIO4 Counter (TIO4CT)		10-87
H'0080 0342	(Use inhibited area)		
H'0080 0344	TIO4 Reload 1 Register (TIO4RL1)		10-89
H'0080 0346	TIO4 Reload 0/ Measure Register (TIO4RL0)		10-88
H'0080 0348	(Use inhibited area)		

SFR Area Register Map (7/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 034A	TIO4 Control Register (TIO4CR)	TIO5 Control Register (TIO5CR)	10-82 10-84
	(Use inhibited area)		
H'0080 0350	TIO5 Counter (TIO5CT)		10-87
H'0080 0352	(Use inhibited area)		
H'0080 0354	TIO5 Reload 1 Register (TIO5RL1)		10-89
H'0080 0356	TIO5 Reload 0/ Measure Register (TIO5RL0)		10-88
	(Use inhibited area)		
H'0080 0360	TIO6 Counter (TIO6CT)		10-87
H'0080 0362	(Use inhibited area)		
H'0080 0364	TIO6 Reload 1 Register (TIO6RL1)		10-89
H'0080 0366	TIO6 Reload 0/ Measure Register (TIO6RL0)		10-88
H'0080 0368	(Use inhibited area)		
H'0080 036A	TIO6 Control Register (TIO6CR)	TIO7 Control Register (TIO7CR)	10-85 10-86
	(Use inhibited area)		
H'0080 0370	TIO7 Counter (TIO7CT)		10-87
H'0080 0372	(Use inhibited area)		
H'0080 0374	TIO7 Reload 1 Register (TIO7RL1)		10-89
H'0080 0376	TIO7 Reload 0/ Measure Register (TIO7RL0)		10-88
	(Use inhibited area)		
H'0080 0380	TIO8 Counter (TIO8CT)		10-87
H'0080 0382	(Use inhibited area)		
H'0080 0384	TIO8 Reload 1 Register (TIO8RL1)		10-89
H'0080 0386	TIO8 Reload 0/ Measure Register (TIO8RL0)		10-88
H'0080 0388	(Use inhibited area)		
H'0080 038A	TIO8 Control Register (TIO8CR)	TIO9 Control Register (TIO9CR)	10-86 10-87
	(Use inhibited area)		
H'0080 0390	TIO9 Counter (TIO9CT)		10-87
H'0080 0392	(Use inhibited area)		
H'0080 0394	TIO9 Reload 1 Register (TIO9RL1)		10-89
H'0080 0396	TIO9 Reload 0/ Measure Register (TIO9RL0)		10-88
	(Use inhibited area)		
H'0080 03BC	TIO0-9 Enable Protect Register (TIOPRO)		10-90
H'0080 03BE	TIO0-9 Count Enable Register (TIOCEN)		10-91
H'0080 03C0	TMS0 Counter (TMS0CT)		10-109
H'0080 03C2	TMS0 Measure 3 Register (TMS0MR3)		10-109

SFR Area Register Map (8/22)

Address	+0 address b0 b7	+1address b8 b15	See pages
H'0080 03C4	TMS0 Measure 2 Register (TMS0MR2)		10-109
H'0080 03C6	TMS0 Measure 1 Register (TMS0MR1)		10-109
H'0080 03C8	TMS0 Measure 0 Register (TMS0MR0)		10-109
H'0080 03CA	TMS0 Control Register (TMS0CR)	TMS1 Control Register (TMS1CR)	10-108
	(Use inhibited area)		
H'0080 03D0	TMS1 Counter (TMS1CT)		10-109
H'0080 03D2	TMS1 Measure 3 Register (TMS1MR3)		10-109
H'0080 03D4	TMS1 Measure 2 Register (TMS1MR2)		10-109
H'0080 03D6	TMS1 Measure 1 Register (TMS1MR1)		10-109
H'0080 03D8	TMS1 Measure 0 Register (TMS1MR0)		10-109
	(Use inhibited area)		
H'0080 03E0	TML0 Counter (TML0CTH)	(Upper)	10-114
H'0080 03E2	TML0 Counter (TML0CTL)	(Lower)	10-114
	(Use inhibited area)		
H'0080 03EA	(Use inhibited area)	TML0 Control Register (TML0CR)	10-113
	(Use inhibited area)		
H'0080 03F0	TML0 Measure 3 Register (TML0MR3H)	(Upper)	10-115
H'0080 03F2	TML0 Measure 3 Register (TML0MR3L)	(Lower)	10-115
H'0080 03F4	TML0 Measure 2 Register (TML0MR2H)	(Upper)	10-115
H'0080 03F6	TML0 Measure 2 Register (TML0MR2L)	(Lower)	10-115
H'0080 03F8	TML0 Measure 1 Register (TML0MR1H)	(Upper)	10-115
H'0080 03FA	TML0 Measure 1 Register (TML0MR1L)	(Lower)	10-115
H'0080 03FC	TML0 Measure 0 Register (TML0MR0H)	(Upper)	10-115
H'0080 03FE	TML0 Measure 0 Register (TML0MR0L)	(Lower)	10-115
H'0080 0400	DMA0–4 Interrupt Request Status Register (DM04ITST)	DMA0–4 Interrupt Request Mask Register (DM04ITMK)	9-18 9-19
	(Use inhibited area)		
H'0080 0408	DMA5–9 Interrupt Request Status Register (DM59ITST)	DMA5–9 Interrupt Request Mask Register (DM59ITMK)	9-18 9-19
	(Use inhibited area)		
H'0080 0410	DMA0 Channel Control Register (DM0CNT)	DMA0 Transfer Count Register (DM0TCT)	9-6 9-15
H'0080 0412	DMA0 Source Address Register (DM0SA)		9-13
H'0080 0414	DMA0 Destination Address Register (DM0DA)		9-14
H'0080 0416	(Use inhibited area)		
H'0080 0418	DMA5 Channel Control Register (DM5CNT)	DMA5 Transfer Count Register (DM5TCT)	9-8 9-15
H'0080 041A	DMA5 Source Address Register (DM5SA)		9-13
H'0080 041C	DMA5 Destination Address Register (DM5DA)		9-14

SFR Area Register Map (9/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 041E	(Use inhibited area)		
H'0080 0420	DMA1 Channel Control Register (DM1CNT)	DMA1 Transfer Count Register (DM1TCT)	9-6 9-15
H'0080 0422	DMA1 Source Address Register (DM1SA)		9-13
H'0080 0424	DMA1 Destination Address Register (DM1DA)		9-14
H'0080 0426	(Use inhibited area)		
H'0080 0428	DMA6 Channel Control Register (DM6CNT)	DMA6 Transfer Count Register (DM6TCT)	9-9 9-15
H'0080 042A	DMA6 Source Address Register (DM6SA)		9-13
H'0080 042C	DMA6 Destination Address Register (DM6DA)		9-14
H'0080 042E	(Use inhibited area)		
H'0080 0430	DMA2 Channel Control Register (DM2CNT)	DMA2 Transfer Count Register (DM2TCT)	9-7 9-15
H'0080 0432	DMA2 Source Address Register (DM2SA)		9-13
H'0080 0434	DMA2 Destination Address Register (DM2DA)		9-14
H'0080 0436	(Use inhibited area)		
H'0080 0438	DMA7 Channel Control Register (DM7CNT)	DMA7 Transfer Count Register (DM7TCT)	9-9 9-15
H'0080 043A	DMA7 Source Address Register (DM7SA)		9-13
H'0080 043C	DMA7 Destination Address Register (DM7DA)		9-14
H'0080 043E	(Use inhibited area)		
H'0080 0440	DMA3 Channel Control Register (DM3CNT)	DMA3 Transfer Count Register (DM3TCT)	9-7 9-15
H'0080 0442	DMA3 Source Address Register (DM3SA)		9-13
H'0080 0444	DMA3 Destination Address Register (DM3DA)		9-14
H'0080 0446	(Use inhibited area)		
H'0080 0448	DMA8 Channel Control Register (DM8CNT)	DMA8 Transfer Count Register (DM8TCT)	9-10 9-15
H'0080 044A	DMA8 Source Address Register (DM8SA)		9-13
H'0080 044C	DMA8 Destination Address Register (DM8DA)		9-14
H'0080 044E	(Use inhibited area)		
H'0080 0450	DMA4 Channel Control Register (DM4CNT)	DMA4 Transfer Count Register (DM4TCT)	9-8 9-15
H'0080 0452	DMA4 Source Address Register (DM4SA)		9-13
H'0080 0454	DMA4 Destination Address Register (DM4DA)		9-14
H'0080 0456	(Use inhibited area)		
H'0080 0458	DMA9 Channel Control Register (DM9CNT)	DMA9 Transfer Count Register (DM9TCT)	9-10 9-15
H'0080 045A	DMA9 Source Address Register (DM9SA)		9-13
H'0080 045C	DMA9 Destination Address Register (DM9DA)		9-14
H'0080 045E	(Use inhibited area)		
H'0080 0460	DMA0 Software Request Generation Register (DM0SRI)		9-12
H'0080 0462	DMA1 Software Request Generation Register (DM1SRI)		9-12

SFR Area Register Map (10/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0464	DMA2 Software Request Generation Register (DM2SRI)		9-12
H'0080 0466	DMA3 Software Request Generation Register (DM3SRI)		9-12
H'0080 0468	DMA4 Software Request Generation Register (DM4SRI)		9-12
	(Use inhibited area)		
H'0080 0470	DMA5 Software Request Generation Register (DM5SRI)		9-12
H'0080 0472	DMA6 Software Request Generation Register (DM6SRI)		9-12
H'0080 0474	DMA7 Software Request Generation Register (DM7SRI)		9-12
H'0080 0476	DMA8 Software Request Generation Register (DM8SRI)		9-12
H'0080 0478	DMA9 Software Request Generation Register (DM9SRI)		9-12
	(Use inhibited area)		
H'0080 0600	Dummy access area (Note 1)	Dummy access area (Note 1)	3-31
H'0080 0602	Dummy access area (Note 1)	Dummy access area (Note 1)	3-31
	(Use inhibited area)		
H'0080 0700	P0 Data Register (P0DATA)	P1 Data Register (P1DATA)	8-7
H'0080 0702	P2 Data Register (P2DATA)	P3 Data Register (P3DATA)	8-7
H'0080 0704	P4 Data Register (P4DATA)	(Use inhibited area)	8-7
H'0080 0706	P6 Data Register (P6DATA)	P7 Data Register (P7DATA)	8-7
H'0080 0708	P8 Data Register (P8DATA)	P9 Data Register (P9DATA)	8-7
H'0080 070A	P10 Data Register (P10DATA)	P11 Data Register (P11DATA)	8-7
H'0080 070C	P12 Data Register (P12DATA)	P13 Data Register (P13DATA)	8-7
H'0080 070E	(Use inhibited area)	P15 Data Register (P15DATA)	8-7
H'0080 0710	(Use inhibited area)	P17 Data Register (P17DATA)	8-7
H'0080 0712	(Use inhibited area)	(Use inhibited area)	
H'0080 0714	(Use inhibited area)	(Use inhibited area)	
H'0080 0716	P22 Data Register (P22DATA)	(Use inhibited area)	8-7
	(Use inhibited area)		
H'0080 0720	P0 Direction Register (P0DIR)	P1 Direction Register (P1DIR)	8-8
H'0080 0722	P2 Direction Register (P2DIR)	P3 Direction Register (P3DIR)	8-8
H'0080 0724	P4 Direction Register (P4DIR)	(Use inhibited area)	8-8
H'0080 0726	P6 Direction Register (P6DIR)	P7 Direction Register (P7DIR)	8-8
H'0080 0728	P8 Direction Register (P8DIR)	P9 Direction Register (P9DIR)	8-8
H'0080 072A	P10 Direction Register (P10DIR)	P11 Direction Register (P11DIR)	8-8
H'0080 072C	P12 Direction Register (P12DIR)	P13 Direction Register (P13DIR)	8-8
H'0080 072E	(Use inhibited area)	P15 Direction Register (P15DIR)	8-8

SFR Area Register Map (11/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0730	(Use inhibited area)	P17 Direction Register (P17DIR)	8-8
H'0080 0732	(Use inhibited area)	(Use inhibited area)	
H'0080 0734	(Use inhibited area)	(Use inhibited area)	
H'0080 0736	P22 Direction Register (P22DIR)	(Use inhibited area)	8-8
|	(Use inhibited area)		
H'0080 0744	(Use inhibited area)	Port Input Special Function Control Register (PICNT)	8-15 18-3
H'0080 0746	(Use inhibited area)	P7 Operation Mode Register (P7MOD)	8-9, 15-4 18-7
H'0080 0748	P8 Operation Mode Register (P8MOD)	P9 Operation Mode Register (P9MOD)	8-9 8-10
H'0080 074A	P10 Operation Mode Register (P10MOD)	P11 Operation Mode Register (P11MOD)	8-10 8-11
H'0080 074C	P12 Operation Mode Register (P12MOD)	P13 Operation Mode Register (P13MOD)	8-11 8-12
H'0080 074E	(Use inhibited area)	P15 Operation Mode Register (P15MOD)	8-12
H'0080 0750	(Use inhibited area)	P17 Operation Mode Register (P17MOD)	8-13
H'0080 0752	(Use inhibited area)	(Use inhibited area)	
H'0080 0754	(Use inhibited area)	(Use inhibited area)	
H'0080 0756	P22 Operation Mode Register (P22MOD)	(Use inhibited area)	8-13
|	(Use inhibited area)		
H'0080 0760	Port Group 0, 1 Input Level Setting Register (PG01LEV)	Port Group 3 Input Level Setting Register (PG3LEV)	8-19
H'0080 0762	Port Group 4, 5 Input Level Setting Register (PG45LEV)	Port Group 6, 7 Input Level Setting Register (PG67LEV)	8-19
H'0080 0764	Port Group 8 Input Level Setting Register (PG8LEV)	(Use inhibited area)	8-19
H'0080 0766	(Use inhibited area)	P7 Peripheral Function Select Register (P7SMOD)	8-14
|	(Use inhibited area)		
H'0080 077A	(Use inhibited area)	RTD Write Function Disable Register (WRRDIS)	14-3
|	(Use inhibited area)		
H'0080 077E	(Use inhibited area)	Bus Mode Control Register (BUSMODC)	15-5
|	(Use inhibited area)		
H'0080 0786	Clock Control Register (CLKCR)	(Use inhibited area)	18-5
|	(Use inhibited area)		
H'0080 07E0	Flash Mode Register (FMOD)	Flash Status Register (FSTAT)	6-7 6-8
H'0080 07E2	Flash Control Register 1 (FCNT1)	Flash Control Register 2 (FCNT2)	6-9 6-10
H'0080 07E4	Flash Control Register 3 (FCNT3)	Flash Control Register 4 (FCNT4)	6-11 6-13
H'0080 07E6	(Use inhibited area)		
H'0080 07E8	Virtual Flash L Bank Register 0 (FELBANK0)		6-15
H'0080 07EA	Virtual Flash L Bank Register 1 (FELBANK1)		6-15
|	(Use inhibited area)		
H'0080 07F0	Virtual Flash S Bank Register 0 (FESBANK0)		6-16

SFR Area Register Map (12/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 07F2	Virtual Flash S Bank Register 1 (FESBANK1)		6-16
	(Use inhibited area)		
H'0080 0FE0	TML1 Counter (TML1CTH)	(Upper)	10-114
H'0080 0FE2	TML1 Counter (TML1CTL)	(Lower)	10-114
	(Use inhibited area)		
H'0080 0FEA	(Use inhibited area)	TML1 Control Register (TML1CR)	10-113
	(Use inhibited area)		
H'0080 0FF0	TML1 Measure 3 Register (TML1MR3H)	(Upper)	10-116
H'0080 0FF2	TML1 Measure 3 Register (TML1MR3L)	(Lower)	10-116
H'0080 0FF4	TML1 Measure 2 Register (TML1MR2H)	(Upper)	10-116
H'0080 0FF6	TML1 Measure 2 Register (TML1MR2L)	(Lower)	10-116
H'0080 0FF8	TML1 Measure 1 Register (TML1MR1H)	(Upper)	10-116
H'0080 0FFA	TML1 Measure 1 Register (TML1MR1L)	(Lower)	10-116
H'0080 0FFC	TML1 Measure 0 Register (TML1MR0H)	(Upper)	10-116
H'0080 0FFE	TML1 Measure 0 Register (TML1MR0L)	(Lower)	10-116
	(Use inhibited area)		
H'0080 1000	CAN0 Control Register (CAN0CNT)		13-15
H'0080 1002	CAN0 Status Register (CAN0STAT)		13-18
H'0080 1004	CAN0 Extended ID Register (CAN0EXTID)		13-21
H'0080 1006	CAN0 Configuration Register (CAN0CONF)		13-22
H'0080 1008	CAN0 Timestamp Count Register (CAN0TSTMP)		13-24
H'0080 100A	CAN0 Receive Error Count Register (CAN0REC)	CAN0 Transmit Error Count Register (CAN0TEC)	13-25
H'0080 100C	CAN0 Slot Interrupt Request Status Register (CAN0SLIST)		13-29
H'0080 100E	(Use inhibited area)		
H'0080 1010	CAN0 Slot Interrupt Request Mask Register (CAN0SLIMK)		13-30
H'0080 1012	(Use inhibited area)		
H'0080 1014	CAN0 Error Interrupt Request Status Register (CAN0ERIST)	CAN0 Error Interrupt Request Mask Register (CAN0ERIMK)	13-31 13-32
H'0080 1016	CAN0 Baud Rate Prescaler (CAN0BRP)	CAN0 Cause of Error Register (CAN0EF)	13-26 13-45
H'0080 1018	CAN0 Mode Register (CAN0MOD)	CAN0 DMA Transfer Request Select Register (CAN0DMARQ)	13-47 13-48
	(Use inhibited area)		
H'0080 1028	CAN0 Global Mask Register Standard ID 0 (C0GMSKS0)	CAN0 Global Mask Register Standard ID 1 (C0GMSKS1)	13-49
H'0080 102A	CAN0 Global Mask Register Extended ID 0 (C0GMSKE0)	CAN0 Global Mask Register Extended ID 1 (C0GMSKE1)	13-50
H'0080 102C	CAN0 Global Mask Register Extended ID 2 (C0GMSKE2)	(Use inhibited area)	13-51
H'0080 102E	(Use inhibited area)		
H'0080 1030	CAN0 Local Mask Register A Standard ID 0 (C0LMSKAS0)	CAN0 Local Mask Register A Standard ID 1 (C0LMSKAS1)	13-49

SFR Area Register Map (13/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1032	CAN0 Local Mask Register A Extended ID 0 (C0LMSKAE0)	CAN0 Local Mask Register A Extended ID 1 (C0LMSKAE1)	13-50
H'0080 1034	CAN0 Local Mask Register A Extended ID 2 (C0LMSKAE2)	(Use inhibited area)	13-51
H'0080 1036	(Use inhibited area)		
H'0080 1038	CAN0 Local Mask Register B Standard ID 0 (C0LMSKBS0)	CAN0 Local Mask Register B Standard ID 1 (C0LMSKBS1)	13-49
H'0080 103A	CAN0 Local Mask Register B Extended ID 0 (C0LMSKBE0)	CAN0 Local Mask Register B Extended ID 1 (C0LMSKBE1)	13-50
H'0080 103C	CAN0 Local Mask Register B Extended ID 2 (C0LMSKBE2)	(Use inhibited area)	13-51
H'0080 103E	(Use inhibited area)		
H'0080 1040	CAN0 Single Shot Mode Control Register (CAN0SSMODE)		13-53
H'0080 1042	(Use inhibited area)		
H'0080 1044	CAN0 Single-Shot Interrupt Request Status Register (CAN0SSIST)		13-33
H'0080 1046	(Use inhibited area)		
H'0080 1048	CAN0 Single-Shot Interrupt Request Mask Register (CAN0SSIMK)		13-34
	(Use inhibited area)		
H'0080 1050	CAN0 Message Slot 0 Control Register (C0MSL0CNT)	CAN0 Message Slot 1 Control Register (C0MSL1CNT)	13-54
H'0080 1052	CAN0 Message Slot 2 Control Register (C0MSL2CNT)	CAN0 Message Slot 3 Control Register (C0MSL3CNT)	13-54
H'0080 1054	CAN0 Message Slot 4 Control Register (C0MSL4CNT)	CAN0 Message Slot 5 Control Register (C0MSL5CNT)	13-54
H'0080 1056	CAN0 Message Slot 6 Control Register (C0MSL6CNT)	CAN0 Message Slot 7 Control Register (C0MSL7CNT)	13-54
H'0080 1058	CAN0 Message Slot 8 Control Register (C0MSL8CNT)	CAN0 Message Slot 9 Control Register (C0MSL9CNT)	13-54
H'0080 105A	CAN0 Message Slot 10 Control Register (C0MSL10CNT)	CAN0 Message Slot 11 Control Register (C0MSL11CNT)	13-54
H'0080 105C	CAN0 Message Slot 12 Control Register (C0MSL12CNT)	CAN0 Message Slot 13 Control Register (C0MSL13CNT)	13-54
H'0080 105E	CAN0 Message Slot 14 Control Register (C0MSL14CNT)	CAN0 Message Slot 15 Control Register (C0MSL15CNT)	13-54
	(Use inhibited area)		
H'0080 1100	CAN0 Message Slot 0 Standard ID 0 (C0MSL0SID0)	CAN0 Message Slot 0 Standard ID 1 (C0MSL0SID1)	13-58 13-59
H'0080 1102	CAN0 Message Slot 0 Extended ID 0 (C0MSL0EID0)	CAN0 Message Slot 0 Extended ID 1 (C0MSL0EID1)	13-60 13-61
H'0080 1104	CAN0 Message Slot 0 Extended ID 2 (C0MSL0EID2)	CAN0 Message Slot 0 Data Length Register (C0MSL0DLC)	13-62 13-63
H'0080 1106	CAN0 Message Slot 0 Data 0 (C0MSL0DT0)	CAN0 Message Slot 0 Data 1 (C0MSL0DT1)	13-64 13-65
H'0080 1108	CAN0 Message Slot 0 Data 2 (C0MSL0DT2)	CAN0 Message Slot 0 Data 3 (C0MSL0DT3)	13-66 13-67
H'0080 110A	CAN0 Message Slot 0 Data 4 (C0MSL0DT4)	CAN0 Message Slot 0 Data 5 (C0MSL0DT5)	13-68 13-69
H'0080 110C	CAN0 Message Slot 0 Data 6 (C0MSL0DT6)	CAN0 Message Slot 0 Data 7 (C0MSL0DT7)	13-70 13-71
H'0080 110E	CAN0 Message Slot 0 Timestamp (C0MSL0TSP)		13-72
H'0080 1110	CAN0 Message Slot 1 Standard ID 0 (C0MSL1SID0)	CAN0 Message Slot 1 Standard ID 1 (C0MSL1SID1)	13-58 13-59
H'0080 1112	CAN0 Message Slot 1 Extended ID 0 (C0MSL1EID0)	CAN0 Message Slot 1 Extended ID 1 (C0MSL1EID1)	13-60 13-61
H'0080 1114	CAN0 Message Slot 1 Extended ID 2 (C0MSL1EID2)	CAN0 Message Slot 1 Data Length Register (C0MSL1DLC)	13-62 13-63
H'0080 1116	CAN0 Message Slot 1 Data 0 (C0MSL1DT0)	CAN0 Message Slot 1 Data 1 (C0MSL1DT1)	13-64 13-65
H'0080 1118	CAN0 Message Slot 1 Data 2 (C0MSL1DT2)	CAN0 Message Slot 1 Data 3 (C0MSL1DT3)	13-66 13-67

SFR Area Register Map (14/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 111A	CAN0 Message Slot 1 Data 4 (C0MSL1DT4)	CAN0 Message Slot 1 Data 5 (C0MSL1DT5)	13-68 13-69
H'0080 111C	CAN0 Message Slot 1 Data 6 (C0MSL1DT6)	CAN0 Message Slot 1 Data 7 (C0MSL1DT7)	13-70 13-71
H'0080 111E	CAN0 Message Slot 1 Timestamp (C0MSL1TSP)		13-72
H'0080 1120	CAN0 Message Slot 2 Standard ID 0 (C0MSL2SID0)	CAN0 Message Slot 2 Standard ID 1 (C0MSL2SID1)	13-58 13-59
H'0080 1122	CAN0 Message Slot 2 Extended ID 0 (C0MSL2EID0)	CAN0 Message Slot 2 Extended ID 1 (C0MSL2EID1)	13-60 13-61
H'0080 1124	CAN0 Message Slot 2 Extended ID 2 (C0MSL2EID2)	CAN0 Message Slot 2 Data Length Register (C0MSL2DLC)	13-62 13-63
H'0080 1126	CAN0 Message Slot 2 Data 0 (C0MSL2DT0)	CAN0 Message Slot 2 Data 1 (C0MSL2DT1)	13-64 13-65
H'0080 1128	CAN0 Message Slot 2 Data 2 (C0MSL2DT2)	CAN0 Message Slot 2 Data 3 (C0MSL2DT3)	13-66 13-67
H'0080 112A	CAN0 Message Slot 2 Data 4 (C0MSL2DT4)	CAN0 Message Slot 2 Data 5 (C0MSL2DT5)	13-68 13-69
H'0080 112C	CAN0 Message Slot 2 Data 6 (C0MSL2DT6)	CAN0 Message Slot 2 Data 7 (C0MSL2DT7)	13-70 13-71
H'0080 112E	CAN0 Message Slot 2 Timestamp (C0MSL2TSP)		13-72
H'0080 1130	CAN0 Message Slot 3 Standard ID 0 (C0MSL3SID0)	CAN0 Message Slot 3 Standard ID 1 (C0MSL3SID1)	13-58 13-59
H'0080 1132	CAN0 Message Slot 3 Extended ID 0 (C0MSL3EID0)	CAN0 Message Slot 3 Extended ID 1 (C0MSL3EID1)	13-60 13-61
H'0080 1134	CAN0 Message Slot 3 Extended ID 2 (C0MSL3EID2)	CAN0 Message Slot 3 Data Length Register (C0MSL3DLC)	13-62 13-63
H'0080 1136	CAN0 Message Slot 3 Data 0 (C0MSL3DT0)	CAN0 Message Slot 3 Data 1 (C0MSL3DT1)	13-64 13-65
H'0080 1138	CAN0 Message Slot 3 Data 2 (C0MSL3DT2)	CAN0 Message Slot 3 Data 3 (C0MSL3DT3)	13-66 13-67
H'0080 113A	CAN0 Message Slot 3 Data 4 (C0MSL3DT4)	CAN0 Message Slot 3 Data 5 (C0MSL3DT5)	13-68 13-69
H'0080 113C	CAN0 Message Slot 3 Data 6 (C0MSL3DT6)	CAN0 Message Slot 3 Data 7 (C0MSL3DT7)	13-70 13-71
H'0080 113E	CAN0 Message Slot 3 Timestamp (C0MSL3TSP)		13-72
H'0080 1140	CAN0 Message Slot 4 Standard ID 0 (C0MSL4SID0)	CAN0 Message Slot 4 Standard ID 1 (C0MSL4SID1)	13-58 13-59
H'0080 1142	CAN0 Message Slot 4 Extended ID 0 (C0MSL4EID0)	CAN0 Message Slot 4 Extended ID 1 (C0MSL4EID1)	13-60 13-61
H'0080 1144	CAN0 Message Slot 4 Extended ID 2 (C0MSL4EID2)	CAN0 Message Slot 4 Data Length Register (C0MSL4DLC)	13-62 13-63
H'0080 1146	CAN0 Message Slot 4 Data 0 (C0MSL4DT0)	CAN0 Message Slot 4 Data 1 (C0MSL4DT1)	13-64 13-65
H'0080 1148	CAN0 Message Slot 4 Data 2 (C0MSL4DT2)	CAN0 Message Slot 4 Data 3 (C0MSL4DT3)	13-66 13-67
H'0080 114A	CAN0 Message Slot 4 Data 4 (C0MSL4DT4)	CAN0 Message Slot 4 Data 5 (C0MSL4DT5)	13-68 13-69
H'0080 114C	CAN0 Message Slot 4 Data 6 (C0MSL4DT6)	CAN0 Message Slot 4 Data 7 (C0MSL4DT7)	13-70 13-71
H'0080 114E	CAN0 Message Slot 4 Timestamp (C0MSL4TSP)		13-72
H'0080 1150	CAN0 Message Slot 5 Standard ID 0 (C0MSL5SID0)	CAN0 Message Slot 5 Standard ID 1 (C0MSL5SID1)	13-58 13-59
H'0080 1152	CAN0 Message Slot 5 Extended ID 0 (C0MSL5EID0)	CAN0 Message Slot 5 Extended ID 1 (C0MSL5EID1)	13-60 13-61
H'0080 1154	CAN0 Message Slot 5 Extended ID 2 (C0MSL5EID2)	CAN0 Message Slot 5 Data Length Register (C0MSL5DLC)	13-62 13-63
H'0080 1156	CAN0 Message Slot 5 Data 0 (C0MSL5DT0)	CAN0 Message Slot 5 Data 1 (C0MSL5DT1)	13-64 13-65
H'0080 1158	CAN0 Message Slot 5 Data 2 (C0MSL5DT2)	CAN0 Message Slot 5 Data 3 (C0MSL5DT3)	13-66 13-67
H'0080 115A	CAN0 Message Slot 5 Data 4 (C0MSL5DT4)	CAN0 Message Slot 5 Data 5 (C0MSL5DT5)	13-68 13-69
H'0080 115C	CAN0 Message Slot 5 Data 6 (C0MSL5DT6)	CAN0 Message Slot 5 Data 7 (C0MSL5DT7)	13-70 13-71
H'0080 115E	CAN0 Message Slot 5 Timestamp (C0MSL5TSP)		13-72

SFR Area Register Map (15/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1160	CAN0 Message Slot 6 Standard ID 0 (C0MSL6SID0)	CAN0 Message Slot 6 Standard ID 1 (C0MSL6SID1)	13-58 13-59
H'0080 1162	CAN0 Message Slot 6 Extended ID 0 (C0MSL6EID0)	CAN0 Message Slot 6 Extended ID 1 (C0MSL6EID1)	13-60 13-61
H'0080 1164	CAN0 Message Slot 6 Extended ID 2 (C0MSL6EID2)	CAN0 Message Slot 6 Data Length Register (C0MSL6DLC)	13-62 13-63
H'0080 1166	CAN0 Message Slot 6 Data 0 (C0MSL6DT0)	CAN0 Message Slot 6 Data 1 (C0MSL6DT1)	13-64 13-65
H'0080 1168	CAN0 Message Slot 6 Data 2 (C0MSL6DT2)	CAN0 Message Slot 6 Data 3 (C0MSL6DT3)	13-66 13-67
H'0080 116A	CAN0 Message Slot 6 Data 4 (C0MSL6DT4)	CAN0 Message Slot 6 Data 5 (C0MSL6DT5)	13-68 13-69
H'0080 116C	CAN0 Message Slot 6 Data 6 (C0MSL6DT6)	CAN0 Message Slot 6 Data 7 (C0MSL6DT7)	13-70 13-71
H'0080 116E	CAN0 Message Slot 6 Timestamp (C0MSL6TSP)		13-72
H'0080 1170	CAN0 Message Slot 7 Standard ID 0 (C0MSL7SID0)	CAN0 Message Slot 7 Standard ID 1 (C0MSL7SID1)	13-58 13-59
H'0080 1172	CAN0 Message Slot 7 Extended ID 0 (C0MSL7EID0)	CAN0 Message Slot 7 Extended ID 1 (C0MSL7EID1)	13-60 13-61
H'0080 1174	CAN0 Message Slot 7 Extended ID 2 (C0MSL7EID2)	CAN0 Message Slot 7 Data Length Register (C0MSL7DLC)	13-62 13-63
H'0080 1176	CAN0 Message Slot 7 Data 0 (C0MSL7DT0)	CAN0 Message Slot 7 Data 1 (C0MSL7DT1)	13-64 13-65
H'0080 1178	CAN0 Message Slot 7 Data 2 (C0MSL7DT2)	CAN0 Message Slot 7 Data 3 (C0MSL7DT3)	13-66 13-67
H'0080 117A	CAN0 Message Slot 7 Data 4 (C0MSL7DT4)	CAN0 Message Slot 7 Data 5 (C0MSL7DT5)	13-68 13-69
H'0080 117C	CAN0 Message Slot 7 Data 6 (C0MSL7DT6)	CAN0 Message Slot 7 Data 7 (C0MSL7DT7)	13-70 13-71
H'0080 117E	CAN0 Message Slot 7 Timestamp (C0MSL7TSP)		13-72
H'0080 1180	CAN0 Message Slot 8 Standard ID 0 (C0MSL8SID0)	CAN0 Message Slot 8 Standard ID 1 (C0MSL8SID1)	13-58 13-59
H'0080 1182	CAN0 Message Slot 8 Extended ID 0 (C0MSL8EID0)	CAN0 Message Slot 8 Extended ID 1 (C0MSL8EID1)	13-60 13-61
H'0080 1184	CAN0 Message Slot 8 Extended ID 2 (C0MSL8EID2)	CAN0 Message Slot 8 Data Length Register (C0MSL8DLC)	13-62 13-63
H'0080 1186	CAN0 Message Slot 8 Data 0 (C0MSL8DT0)	CAN0 Message Slot 8 Data 1 (C0MSL8DT1)	13-64 13-65
H'0080 1188	CAN0 Message Slot 8 Data 2 (C0MSL8DT2)	CAN0 Message Slot 8 Data 3 (C0MSL8DT3)	13-66 13-67
H'0080 118A	CAN0 Message Slot 8 Data 4 (C0MSL8DT4)	CAN0 Message Slot 8 Data 5 (C0MSL8DT5)	13-68 13-69
H'0080 118C	CAN0 Message Slot 8 Data 6 (C0MSL8DT6)	CAN0 Message Slot 8 Data 7 (C0MSL8DT7)	13-70 13-71
H'0080 118E	CAN0 Message Slot 8 Timestamp (C0MSL8TSP)		13-72
H'0080 1190	CAN0 Message Slot 9 Standard ID 0 (C0MSL9SID0)	CAN0 Message Slot 9 Standard ID 1 (C0MSL9SID1)	13-58 13-59
H'0080 1192	CAN0 Message Slot 9 Extended ID 0 (C0MSL9EID0)	CAN0 Message Slot 9 Extended ID 1 (C0MSL9EID1)	13-60 13-61
H'0080 1194	CAN0 Message Slot 9 Extended ID 2 (C0MSL9EID2)	CAN0 Message Slot 9 Data Length Register (C0MSL9DLC)	13-62 13-63
H'0080 1196	CAN0 Message Slot 9 Data 0 (C0MSL9DT0)	CAN0 Message Slot 9 Data 1 (C0MSL9DT1)	13-64 13-65
H'0080 1198	CAN0 Message Slot 9 Data 2 (C0MSL9DT2)	CAN0 Message Slot 9 Data 3 (C0MSL9DT3)	13-66 13-67
H'0080 119A	CAN0 Message Slot 9 Data 4 (C0MSL9DT4)	CAN0 Message Slot 9 Data 5 (C0MSL9DT5)	13-68 13-69
H'0080 119C	CAN0 Message Slot 9 Data 6 (C0MSL9DT6)	CAN0 Message Slot 9 Data 7 (C0MSL9DT7)	13-70 13-71
H'0080 119E	CAN0 Message Slot 9 Timestamp (C0MSL9TSP)		13-72

SFR Area Register Map (16/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 11A0	CAN0 Message Slot 10 Standard ID 0 (C0MSL10SID0)	CAN0 Message Slot 10 Standard ID 1 (C0MSL10SID1)	13-58 13-59
H'0080 11A2	CAN0 Message Slot 10 Extended ID 0 (C0MSL10EID0)	CAN0 Message Slot 10 Extended ID 1 (C0MSL10EID1)	13-60 13-61
H'0080 11A4	CAN0 Message Slot 10 Extended ID 2 (C0MSL10EID2)	CAN0 Message Slot 10 Data Length Register (C0MSL10DLC)	13-62 13-63
H'0080 11A6	CAN0 Message Slot 10 Data 0 (C0MSL10DT0)	CAN0 Message Slot 10 Data 1 (C0MSL10DT1)	13-64 13-65
H'0080 11A8	CAN0 Message Slot 10 Data 2 (C0MSL10DT2)	CAN0 Message Slot 10 Data 3 (C0MSL10DT3)	13-66 13-67
H'0080 11AA	CAN0 Message Slot 10 Data 4 (C0MSL10DT4)	CAN0 Message Slot 10 Data 5 (C0MSL10DT5)	13-68 13-69
H'0080 11AC	CAN0 Message Slot 10 Data 6 (C0MSL10DT6)	CAN0 Message Slot 10 Data 7 (C0MSL10DT7)	13-70 13-71
H'0080 11AE	CAN0 Message Slot 10 Timestamp (C0MSL10TSP)		13-72
H'0080 11B0	CAN0 Message Slot 11 Standard ID 0 (C0MSL11SID0)	CAN0 Message Slot 11 Standard ID 1 (C0MSL11SID1)	13-58 13-59
H'0080 11B2	CAN0 Message Slot 11 Extended ID 0 (C0MSL11EID0)	CAN0 Message Slot 11 Extended ID 1 (C0MSL11EID1)	13-60 13-61
H'0080 11B4	CAN0 Message Slot 11 Extended ID 2 (C0MSL11EID2)	CAN0 Message Slot 11 Data Length Register (C0MSL11DLC)	13-62 13-63
H'0080 11B6	CAN0 Message Slot 11 Data 0 (C0MSL11DT0)	CAN0 Message Slot 11 Data 1 (C0MSL11DT1)	13-64 13-65
H'0080 11B8	CAN0 Message Slot 11 Data 2 (C0MSL11DT2)	CAN0 Message Slot 11 Data 3 (C0MSL11DT3)	13-66 13-67
H'0080 11BA	CAN0 Message Slot 11 Data 4 (C0MSL11DT4)	CAN0 Message Slot 11 Data 5 (C0MSL11DT5)	13-68 13-69
H'0080 11BC	CAN0 Message Slot 11 Data 6 (C0MSL11DT6)	CAN0 Message Slot 11 Data 7 (C0MSL11DT7)	13-70 13-71
H'0080 11BE	CAN0 Message Slot 11 Timestamp (C0MSL11TSP)		13-72
H'0080 11C0	CAN0 Message Slot 12 Standard ID 0 (C0MSL12SID0)	CAN0 Message Slot 12 Standard ID 1 (C0MSL12SID1)	13-58 13-59
H'0080 11C2	CAN0 Message Slot 12 Extended ID 0 (C0MSL12EID0)	CAN0 Message Slot 12 Extended ID 1 (C0MSL12EID1)	13-60 13-61
H'0080 11C4	CAN0 Message Slot 12 Extended ID 2 (C0MSL12EID2)	CAN0 Message Slot 12 Data Length Register (C0MSL12DLC)	13-62 13-63
H'0080 11C6	CAN0 Message Slot 12 Data 0 (C0MSL12DT0)	CAN0 Message Slot 12 Data 1 (C0MSL12DT1)	13-64 13-65
H'0080 11C8	CAN0 Message Slot 12 Data 2 (C0MSL12DT2)	CAN0 Message Slot 12 Data 3 (C0MSL12DT3)	13-66 13-67
H'0080 11CA	CAN0 Message Slot 12 Data 4 (C0MSL12DT4)	CAN0 Message Slot 12 Data 5 (C0MSL12DT5)	13-68 13-69
H'0080 11CC	CAN0 Message Slot 12 Data 6 (C0MSL12DT6)	CAN0 Message Slot 12 Data 7 (C0MSL12DT7)	13-70 13-71
H'0080 11CE	CAN0 Message Slot 12 Timestamp (C0MSL12TSP)		13-72
H'0080 11D0	CAN0 Message Slot 13 Standard ID 0 (C0MSL13SID0)	CAN0 Message Slot 13 Standard ID 1 (C0MSL13SID1)	13-58 13-59
H'0080 11D2	CAN0 Message Slot 13 Extended ID 0 (C0MSL13EID0)	CAN0 Message Slot 13 Extended ID 1 (C0MSL13EID1)	13-60 13-61
H'0080 11D4	CAN0 Message Slot 13 Extended ID 2 (C0MSL13EID2)	CAN0 Message Slot 13 Data Length Register (C0MSL13DLC)	13-62 13-63
H'0080 11D6	CAN0 Message Slot 13 Data 0 (C0MSL13DT0)	CAN0 Message Slot 13 Data 1 (C0MSL13DT1)	13-64 13-65
H'0080 11D8	CAN0 Message Slot 13 Data 2 (C0MSL13DT2)	CAN0 Message Slot 13 Data 3 (C0MSL13DT3)	13-66 13-67
H'0080 11DA	CAN0 Message Slot 13 Data 4 (C0MSL13DT4)	CAN0 Message Slot 13 Data 5 (C0MSL13DT5)	13-68 13-69
H'0080 11DC	CAN0 Message Slot 13 Data 6 (C0MSL13DT6)	CAN0 Message Slot 13 Data 7 (C0MSL13DT7)	13-70 13-71
H'0080 11DE	CAN0 Message Slot 13 Timestamp (C0MSL13TSP)		13-72

SFR Area Register Map (17/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 11E0	CAN0 Message Slot 14 Standard ID 0 (C0MSL14SID0)	CAN0 Message Slot 14 Standard ID 1 (C0MSL14SID1)	13-58 13-59
H'0080 11E2	CAN0 Message Slot 14 Extended ID 0 (C0MSL14EID0)	CAN0 Message Slot 14 Extended ID 1 (C0MSL14EID1)	13-60 13-61
H'0080 11E4	CAN0 Message Slot 14 Extended ID 2 (C0MSL14EID2)	CAN0 Message Slot 14 Data Length Register (C0MSL14DLC)	13-62 13-63
H'0080 11E6	CAN0 Message Slot 14 Data 0 (C0MSL14DT0)	CAN0 Message Slot 14 Data 1 (C0MSL14DT1)	13-64 13-65
H'0080 11E8	CAN0 Message Slot 14 Data 2 (C0MSL14DT2)	CAN0 Message Slot 14 Data 3 (C0MSL14DT3)	13-66 13-67
H'0080 11EA	CAN0 Message Slot 14 Data 4 (C0MSL14DT4)	CAN0 Message Slot 14 Data 5 (C0MSL14DT5)	13-68 13-69
H'0080 11EC	CAN0 Message Slot 14 Data 6 (C0MSL14DT6)	CAN0 Message Slot 14 Data 7 (C0MSL14DT7)	13-70 13-71
H'0080 11EE	CAN0 Message Slot 14 Timestamp (C0MSL14TSP)		13-72
H'0080 11F0	CAN0 Message Slot 15 Standard ID 0 (C0MSL15SID0)	CAN0 Message Slot 15 Standard ID 1 (C0MSL15SID1)	13-58 13-59
H'0080 11F2	CAN0 Message Slot 15 Extended ID 0 (C0MSL15EID0)	CAN0 Message Slot 15 Extended ID 1 (C0MSL15EID1)	13-60 13-61
H'0080 11F4	CAN0 Message Slot 15 Extended ID 2 (C0MSL15EID2)	CAN0 Message Slot 15 Data Length Register (C0MSL15DLC)	13-62 13-63
H'0080 11F6	CAN0 Message Slot 15 Data 0 (C0MSL15DT0)	CAN0 Message Slot 15 Data 1 (C0MSL15DT1)	13-64 13-65
H'0080 11F8	CAN0 Message Slot 15 Data 2 (C0MSL15DT2)	CAN0 Message Slot 15 Data 3 (C0MSL15DT3)	13-66 13-67
H'0080 11FA	CAN0 Message Slot 15 Data 4 (C0MSL15DT4)	CAN0 Message Slot 15 Data 5 (C0MSL15DT5)	13-68 13-69
H'0080 11FC	CAN0 Message Slot 15 Data 6 (C0MSL15DT6)	CAN0 Message Slot 15 Data 7 (C0MSL15DT7)	13-70 13-71
H'0080 11FE	CAN0 Message Slot 15 Timestamp (C0MSL15TSP)		13-72
	(Use inhibited area)		
H'0080 1400	CAN1 Control Register (CAN1CNT)		13-15
H'0080 1402	CAN1 Status Register (CAN1STAT)		13-18
H'0080 1404	CAN1 Extended ID Register (CAN1EXTID)		13-21
H'0080 1406	CAN1 Configuration Register (CAN1CONF)		13-22
H'0080 1408	CAN1 Timestamp Count Register (CAN1TSTMP)		13-24
H'0080 140A	CAN1 Receive Error Count Register (CAN1REC)	CAN1 Transmit Error Count Register (CAN1TEC)	13-25
H'0080 140C	CAN1 Slot Interrupt Request Status Register (CAN1SLIST)		13-29
H'0080 140E	(Use inhibited area)		
H'0080 1410	CAN1 Slot Interrupt Request Mask Register (CAN1SLIMK)		13-30
H'0080 1412	(Use inhibited area)		
H'0080 1414	CAN1 Error Interrupt Request Status Register (CAN1ERIST)	CAN1 Error Interrupt Request Mask Register (CAN1ERIMK)	13-31 13-32
H'0080 1416	CAN1 Baud Rate Prescaler (CAN1BRP)	CAN1 Cause of Error Register (CAN1EF)	13-26 13-45
H'0080 1418	CAN1 Mode Register (CAN1MOD)	CAN1 DMA Transfer Request Select Register (CAN1DMARQ)	13-47 13-48
	(Use inhibited area)		
H'0080 1428	CAN1 Global Mask Register Standard ID 0 (C1GMSKS0)	CAN1 Global Mask Register Standard ID 1 (C1GMSKS1)	13-49
H'0080 142A	CAN1 Global Mask Register Extended ID 0 (C1GMSKE0)	CAN1 Global Mask Register Extended ID 1 (C1GMSKE1)	13-50
H'0080 142C	CAN1 Global Mask Register Extended ID 2 (C1GMSKE2)	(Use inhibited area)	13-51
H'0080 142E	(Use inhibited area)		

SFR Area Register Map (18/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1430	CAN1 Local Mask Register A Standard ID 0 (C1LMSKAS0)	CAN1 Local Mask Register A Standard ID 1 (C1LMSKAS1)	13-49
H'0080 1432	CAN1 Local Mask Register A Extended ID 0 (C1LMSKAE0)	CAN1 Local Mask Register A Extended ID 1 (C1LMSKAE1)	13-50
H'0080 1434	CAN1 Local Mask Register A Extended ID 2 (C1LMSKAE2)	(Use inhibited area)	13-51
H'0080 1436	(Use inhibited area)		
H'0080 1438	CAN1 Local Mask Register B Standard ID 0 (C1LMSKBS0)	CAN1 Local Mask Register B Standard ID 1 (C1LMSKBS1)	13-49
H'0080 143A	CAN1 Local Mask Register B Extended ID 0 (C1LMSKBE0)	CAN1 Local Mask Register B Extended ID 1 (C1LMSKBE1)	13-50
H'0080 143C	CAN1 Local Mask Register B Extended ID 2 (C1LMSKBE2)	(Use inhibited area)	13-51
H'0080 143E	(Use inhibited area)		
H'0080 1440	CAN1 Single-Shot Mode Control Register (CAN1SSMODE)		13-53
H'0080 1442	(Use inhibited area)		
H'0080 1444	CAN1 Single-Shot Interrupt Request Status Register (CAN1SSIST)		13-33
H'0080 1446	(Use inhibited area)		
H'0080 1448	CAN1 Single-Shot Interrupt Request Mask Register (CAN1SSIMK)		13-34
	(Use inhibited area)		
H'0080 1450	CAN1 Message Slot 0 Control Register (C1MSL0CNT)	CAN1 Message Slot 1 Control Register (C1MSL1CNT)	13-54
H'0080 1452	CAN1 Message Slot 2 Control Register (C1MSL2CNT)	CAN1 Message Slot 3 Control Register (C1MSL3CNT)	13-54
H'0080 1454	CAN1 Message Slot 4 Control Register (C1MSL4CNT)	CAN1 Message Slot 5 Control Register (C1MSL5CNT)	13-54
H'0080 1456	CAN1 Message Slot 6 Control Register (C1MSL6CNT)	CAN1 Message Slot 7 Control Register (C1MSL7CNT)	13-54
H'0080 1458	CAN1 Message Slot 8 Control Register (C1MSL8CNT)	CAN1 Message Slot 9 Control Register (C1MSL9CNT)	13-54
H'0080 145A	CAN1 Message Slot 10 Control Register (C1MSL10CNT)	CAN1 Message Slot 11 Control Register (C1MSL11CNT)	13-54
H'0080 145C	CAN1 Message Slot 12 Control Register (C1MSL12CNT)	CAN1 Message Slot 13 Control Register (C1MSL13CNT)	13-54
H'0080 145E	CAN1 Message Slot 14 Control Register (C1MSL14CNT)	CAN1 Message Slot 15 Control Register (C1MSL15CNT)	13-54
	(Use inhibited area)		
H'0080 1500	CAN1 Message Slot 0 Standard ID 0 (C1MSL0SID0)	CAN1 Message Slot 0 Standard ID 1 (C1MSL0SID1)	13-58 13-59
H'0080 1502	CAN1 Message Slot 0 Extended ID 0 (C1MSL0EID0)	CAN1 Message Slot 0 Extended ID 1 (C1MSL0EID1)	13-60 13-61
H'0080 1504	CAN1 Message Slot 0 Extended ID 2 (C1MSL0EID2)	CAN1 Message Slot 0 Data Length Register (C1MSL0DLC)	13-62 13-63
H'0080 1506	CAN1 Message Slot 0 Data 0 (C1MSL0DT0)	CAN1 Message Slot 0 Data 1 (C1MSL0DT1)	13-64 13-65
H'0080 1508	CAN1 Message Slot 0 Data 2 (C1MSL0DT2)	CAN1 Message Slot 0 Data 3 (C1MSL0DT3)	13-66 13-67
H'0080 150A	CAN1 Message Slot 0 Data 4 (C1MSL0DT4)	CAN1 Message Slot 0 Data 5 (C1MSL0DT5)	13-68 13-69
H'0080 150C	CAN1 Message Slot 0 Data 6 (C1MSL0DT6)	CAN1 Message Slot 0 Data 7 (C1MSL0DT7)	13-70 13-71
H'0080 150E	CAN1 Message Slot 0 Timestamp (C1MSL0TSP)		13-72

SFR Area Register Map (19/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1510	CAN1 Message Slot 1 Standard ID 0 (C1MSL1SID0)	CAN1 Message Slot 1 Standard ID 1 (C1MSL1SID1)	13-58 13-59
H'0080 1512	CAN1 Message Slot 1 Extended ID 0 (C1MSL1EID0)	CAN1 Message Slot 1 Extended ID 1 (C1MSL1EID1)	13-60 13-61
H'0080 1514	CAN1 Message Slot 1 Extended ID 2 (C1MSL1EID2)	CAN1 Message Slot 1 Data Length Register (C1MSL1DLC)	13-62 13-63
H'0080 1516	CAN1 Message Slot 1 Data 0 (C1MSL1DT0)	CAN1 Message Slot 1 Data 1 (C1MSL1DT1)	13-64 13-65
H'0080 1518	CAN1 Message Slot 1 Data 2 (C1MSL1DT2)	CAN1 Message Slot 1 Data 3 (C1MSL1DT3)	13-66 13-67
H'0080 151A	CAN1 Message Slot 1 Data 4 (C1MSL1DT4)	CAN1 Message Slot 1 Data 5 (C1MSL1DT5)	13-68 13-69
H'0080 151C	CAN1 Message Slot 1 Data 6 (C1MSL1DT6)	CAN1 Message Slot 1 Data 7 (C1MSL1DT7)	13-70 13-71
H'0080 151E	CAN1 Message Slot 1 Timestamp (C1MSL1TSP)		13-72
H'0080 1520	CAN1 Message Slot 2 Standard ID 0 (C1MSL2SID0)	CAN1 Message Slot 2 Standard ID 1 (C1MSL2SID1)	13-58 13-59
H'0080 1522	CAN1 Message Slot 2 Extended ID 0 (C1MSL2EID0)	CAN1 Message Slot 2 Extended ID 1 (C1MSL2EID1)	13-60 13-61
H'0080 1524	CAN1 Message Slot 2 Extended ID 2 (C1MSL2EID2)	CAN1 Message Slot 2 Data Length Register (C1MSL2DLC)	13-62 13-63
H'0080 1526	CAN1 Message Slot 2 Data 0 (C1MSL2DT0)	CAN1 Message Slot 2 Data 1 (C1MSL2DT1)	13-64 13-65
H'0080 1528	CAN1 Message Slot 2 Data 2 (C1MSL2DT2)	CAN1 Message Slot 2 Data 3 (C1MSL2DT3)	13-66 13-67
H'0080 152A	CAN1 Message Slot 2 Data 4 (C1MSL2DT4)	CAN1 Message Slot 2 Data 5 (C1MSL2DT5)	13-68 13-69
H'0080 152C	CAN1 Message Slot 2 Data 6 (C1MSL2DT6)	CAN1 Message Slot 2 Data 7 (C1MSL2DT7)	13-70 13-71
H'0080 152E	CAN1 Message Slot 2 Timestamp (C1MSL2TSP)		13-72
H'0080 1530	CAN1 Message Slot 3 Standard ID 0 (C1MSL3SID0)	CAN1 Message Slot 3 Standard ID 1 (C1MSL3SID1)	13-58 13-59
H'0080 1532	CAN1 Message Slot 3 Extended ID 0 (C1MSL3EID0)	CAN1 Message Slot 3 Extended ID 1 (C1MSL3EID1)	13-60 13-61
H'0080 1534	CAN1 Message Slot 3 Extended ID 2 (C1MSL3EID2)	CAN1 Message Slot 3 Data Length Register (C1MSL3DLC)	13-62 13-63
H'0080 1536	CAN1 Message Slot 3 Standard ID 0 (C1MSL3DT0)	CAN1 Message Slot 3 Standard ID 1 (C1MSL3DT1)	13-64 13-65
H'0080 1538	CAN1 Message Slot 3 Data 2 (C1MSL3DT2)	CAN1 Message Slot 3 Data 3 (C1MSL3DT3)	13-66 13-67
H'0080 153A	CAN1 Message Slot 3 Data 4 (C1MSL3DT4)	CAN1 Message Slot 3 Data 5 (C1MSL3DT5)	13-68 13-69
H'0080 153C	CAN1 Message Slot 3 Data 6 (C1MSL3DT6)	CAN1 Message Slot 3 Data 7 (C1MSL3DT7)	13-70 13-71
H'0080 153E	CAN1 Message Slot 3 Timestamp (C1MSL3TSP)		13-72
H'0080 1540	CAN1 Message Slot 4 Standard ID 0 (C1MSL4SID0)	CAN1 Message Slot 4 Standard ID 1 (C1MSL4SID1)	13-58 13-59
H'0080 1542	CAN1 Message Slot 4 Extended ID 0 (C1MSL4EID0)	CAN1 Message Slot 4 Extended ID 1 (C1MSL4EID1)	13-60 13-61
H'0080 1544	CAN1 Message Slot 4 Extended ID 2 (C1MSL4EID2)	CAN1 Message Slot 4 Data Length Register (C1MSL4DLC)	13-62 13-63
H'0080 1546	CAN1 Message Slot 4 Data 0 (C1MSL4DT0)	CAN1 Message Slot 4 Data 1 (C1MSL4DT1)	13-64 13-65
H'0080 1548	CAN1 Message Slot 4 Data 2 (C1MSL4DT2)	CAN1 Message Slot 4 Data 3 (C1MSL4DT3)	13-66 13-67
H'0080 154A	CAN1 Message Slot 4 Data 4 (C1MSL4DT4)	CAN1 Message Slot 4 Data 5 (C1MSL4DT5)	13-68 13-69
H'0080 154C	CAN1 Message Slot 4 Data 6 (C1MSL4DT6)	CAN1 Message Slot 4 Data 7 (C1MSL4DT7)	13-70 13-71
H'0080 154E	CAN1 Message Slot 4 Timestamp (C1MSL4TSP)		13-72

SFR Area Register Map (20/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1550	CAN1 Message Slot 5 Standard ID 0 (C1MSL5SID0)	CAN1 Message Slot 5 Standard ID 1 (C1MSL5SID1)	13-58 13-59
H'0080 1552	CAN1 Message Slot 5 Extended ID 0 (C1MSL5EID0)	CAN1 Message Slot 5 Extended ID 1 (C1MSL5EID1)	13-60 13-61
H'0080 1554	CAN1 Message Slot 5 Extended ID 2 (C1MSL5EID2)	CAN1 Message Slot 5 Data Length Register (C1MSL5DLC)	13-62 13-63
H'0080 1556	CAN1 Message Slot 5 Data 0 (C1MSL5DT0)	CAN1 Message Slot 5 Data 1 (C1MSL5DT1)	13-64 13-65
H'0080 1558	CAN1 Message Slot 5 Data 2 (C1MSL5DT2)	CAN1 Message Slot 5 Data 3 (C1MSL5DT3)	13-66 13-67
H'0080 155A	CAN1 Message Slot 5 Data 4 (C1MSL5DT4)	CAN1 Message Slot 5 Data 5 (C1MSL5DT5)	13-68 13-69
H'0080 155C	CAN1 Message Slot 5 Data 6 (C1MSL5DT6)	CAN1 Message Slot 5 Data 7 (C1MSL5DT7)	13-70 13-71
H'0080 155E	CAN1 Message Slot 5 Timestamp (C1MSL5TSP)		13-72
H'0080 1560	CAN1 Message Slot 6 Standard ID 0 (C1MSL6SID0)	CAN1 Message Slot 6 Standard ID 1 (C1MSL6SID1)	13-58 13-59
H'0080 1562	CAN1 Message Slot 6 Extended ID 0 (C1MSL6EID0)	CAN1 Message Slot 6 Extended ID 1 (C1MSL6EID1)	13-60 13-61
H'0080 1564	CAN1 Message Slot 6 Extended ID 2 (C1MSL6EID2)	CAN1 Message Slot 6 Data Length Register (C1MSL6DLC)	13-62 13-63
H'0080 1566	CAN1 Message Slot 6 Data 0 (C1MSL6DT0)	CAN1 Message Slot 6 Data 1 (C1MSL6DT1)	13-64 13-65
H'0080 1568	CAN1 Message Slot 6 Data 2 (C1MSL6DT2)	CAN1 Message Slot 6 Data 3 (C1MSL6DT3)	13-66 13-67
H'0080 156A	CAN1 Message Slot 6 Data 4 (C1MSL6DT4)	CAN1 Message Slot 6 Data 5 (C1MSL6DT5)	13-68 13-69
H'0080 156C	CAN1 Message Slot 6 Data 6 (C1MSL6DT6)	CAN1 Message Slot 6 Data 7 (C1MSL6DT7)	13-70 13-71
H'0080 156E	CAN1 Message Slot 6 Timestamp (C1MSL6TSP)		13-72
H'0080 1570	CAN1 Message Slot 7 Standard ID 0 (C1MSL7SID0)	CAN1 Message Slot 7 Standard ID 1 (C1MSL7SID1)	13-58 13-59
H'0080 1572	CAN1 Message Slot 7 Extended ID 0 (C1MSL7EID0)	CAN1 Message Slot 7 Extended ID 1 (C1MSL7EID1)	13-60 13-61
H'0080 1574	CAN1 Message Slot 7 Extended ID 2 (C1MSL7EID2)	CAN1 Message Slot 7 Data Length Register (C1MSL7DLC)	13-62 13-63
H'0080 1576	CAN1 Message Slot 7 Data 0 (C1MSL7DT0)	CAN1 Message Slot 7 Data 1 (C1MSL7DT1)	13-64 13-65
H'0080 1578	CAN1 Message Slot 7 Data 2 (C1MSL7DT2)	CAN1 Message Slot 7 Data 3 (C1MSL7DT3)	13-66 13-67
H'0080 157A	CAN1 Message Slot 7 Data 4 (C1MSL7DT4)	CAN1 Message Slot 7 Data 5 (C1MSL7DT5)	13-68 13-69
H'0080 157C	CAN1 Message Slot 7 Data 6 (C1MSL7DT6)	CAN1 Message Slot 7 Data 7 (C1MSL7DT7)	13-70 13-71
H'0080 157E	CAN1 Message Slot 7 Timestamp (C1MSL7TSP)		13-72
H'0080 1580	CAN1 Message Slot 8 Standard ID 0 (C1MSL8SID0)	CAN1 Message Slot 8 Standard ID 1 (C1MSL8SID1)	13-58 13-59
H'0080 1582	CAN1 Message Slot 8 Extended ID 0 (C1MSL8EID0)	CAN1 Message Slot 8 Extended ID 1 (C1MSL8EID1)	13-60 13-61
H'0080 1584	CAN1 Message Slot 8 Extended ID 2 (C1MSL8EID2)	CAN1 Message Slot 8 Data Length Register (C1MSL8DLC)	13-62 13-63
H'0080 1586	CAN1 Message Slot 8 Data 0 (C1MSL8DT0)	CAN1 Message Slot 8 Data 1 (C1MSL8DT1)	13-64 13-65
H'0080 1588	CAN1 Message Slot 8 Data 2 (C1MSL8DT2)	CAN1 Message Slot 8 Data 3 (C1MSL8DT3)	13-66 13-67
H'0080 158A	CAN1 Message Slot 8 Data 4 (C1MSL8DT4)	CAN1 Message Slot 8 Data 5 (C1MSL8DT5)	13-68 13-69
H'0080 158C	CAN1 Message Slot 8 Data 6 (C1MSL8DT6)	CAN1 Message Slot 8 Data 7 (C1MSL8DT7)	13-70 13-71
H'0080 158E	CAN1 Message Slot 8 Timestamp (C1MSL8TSP)		13-72

SFR Area Register Map (21/22)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1590	CAN1 Message Slot 9 Standard ID 0 (C1MSL9SID0)	CAN1 Message Slot 9 Standard ID 1 (C1MSL9SID1)	13-58 13-59
H'0080 1592	CAN1 Message Slot 9 Extended ID 0 (C1MSL9EID0)	CAN1 Message Slot 9 Extended ID 1 (C1MSL9EID1)	13-60 13-61
H'0080 1594	CAN1 Message Slot 9 Extended ID 2 (C1MSL9EID2)	CAN1 Message Slot 9 Data Length Register (C1MSL9DLC)	13-62 13-63
H'0080 1596	CAN1 Message Slot 9 Data 0 (C1MSL9DT0)	CAN1 Message Slot 9 Data 1 (C1MSL9DT1)	13-64 13-65
H'0080 1598	CAN1 Message Slot 9 Data 2 (C1MSL9DT2)	CAN1 Message Slot 9 Data 3 (C1MSL9DT3)	13-66 13-67
H'0080 159A	CAN1 Message Slot 9 Data 4 (C1MSL9DT4)	CAN1 Message Slot 9 Data 5 (C1MSL9DT5)	13-68 13-69
H'0080 159C	CAN1 Message Slot 9 Data 6 (C1MSL9DT6)	CAN1 Message Slot 9 Data 7 (C1MSL9DT7)	13-70 13-71
H'0080 159E	CAN1 Message Slot 9 Timestamp (C1MSL9TSP)		13-72
H'0080 15A0	CAN1 Message Slot 10 Standard ID 0 (C1MSL10SID0)	CAN1 Message Slot 10 Standard ID 1 (C1MSL10SID1)	13-58 13-59
H'0080 15A2	CAN1 Message Slot 10 Extended ID 0 (C1MSL10EID0)	CAN1 Message Slot 10 Extended ID 1 (C1MSL10EID1)	13-60 13-61
H'0080 15A4	CAN1 Message Slot 10 Extended ID 2 (C1MSL10EID2)	CAN1 Message Slot 10 Data Length Register (C1MSL10DLC)	13-62 13-63
H'0080 15A6	CAN1 Message Slot 10 Data 0 (C1MSL10DT0)	CAN1 Message Slot 10 Data 1 (C1MSL10DT1)	13-64 13-65
H'0080 15A8	CAN1 Message Slot 10 Data 2 (C1MSL10DT2)	CAN1 Message Slot 10 Data 3 (C1MSL10DT3)	13-66 13-67
H'0080 15AA	CAN1 Message Slot 10 Data 4 (C1MSL10DT4)	CAN1 Message Slot 10 Data 5 (C1MSL10DT5)	13-68 13-69
H'0080 15AC	CAN1 Message Slot 10 Data 6 (C1MSL10DT6)	CAN1 Message Slot 10 Data 7 (C1MSL10DT7)	13-70 13-71
H'0080 15AE	CAN1 Message Slot 10 Timestamp (C1MSL10TSP)		13-72
H'0080 15B0	CAN1 Message Slot 11 Standard ID 0 (C1MSL11SID0)	CAN1 Message Slot 11 Standard ID 1 (C1MSL11SID1)	13-58 13-59
H'0080 15B2	CAN1 Message Slot 11 Extended ID 0 (C1MSL11EID0)	CAN1 Message Slot 11 Extended ID 1 (C1MSL11EID1)	13-60 13-61
H'0080 15B4	CAN1 Message Slot 11 Extended ID 2 (C1MSL11EID2)	CAN1 Message Slot 11 Data Length Register (C1MSL11DLC)	13-62 13-63
H'0080 15B6	CAN1 Message Slot 11 Data 0 (C1MSL11DT0)	CAN1 Message Slot 11 Data 1 (C1MSL11DT1)	13-64 13-65
H'0080 15B8	CAN1 Message Slot 11 Data 2 (C1MSL11DT2)	CAN1 Message Slot 11 Data 3 (C1MSL11DT3)	13-66 13-67
H'0080 15BA	CAN1 Message Slot 11 Data 4 (C1MSL11DT4)	CAN1 Message Slot 11 Data 5 (C1MSL11DT5)	13-68 13-69
H'0080 15BC	CAN1 Message Slot 11 Data 6 (C1MSL11DT6)	CAN1 Message Slot 11 Data 7 (C1MSL11DT7)	13-70 13-71
H'0080 15BE	CAN1 Message Slot 11 Timestamp (C1MSL11TSP)		13-72
H'0080 15C0	CAN1 Message Slot 12 Standard ID 0 (C1MSL12SID0)	CAN1 Message Slot 12 Standard ID 1 (C1MSL12SID1)	13-58 13-59
H'0080 15C2	CAN1 Message Slot 12 Extended ID 0 (C1MSL12EID0)	CAN1 Message Slot 12 Extended ID 1 (C1MSL12EID1)	13-60 13-61
H'0080 15C4	CAN1 Message Slot 12 Extended ID 2 (C1MSL12EID2)	CAN1 Message Slot 12 Data Length Register (C1MSL12DLC)	13-62 13-63
H'0080 15C6	CAN1 Message Slot 12 Data 0 (C1MSL12DT0)	CAN1 Message Slot 12 Data 1 (C1MSL12DT1)	13-64 13-65
H'0080 15C8	CAN1 Message Slot 12 Data 2 (C1MSL12DT2)	CAN1 Message Slot 12 Data 3 (C1MSL12DT3)	13-66 13-67
H'0080 15CA	CAN1 Message Slot 12 Data 4 (C1MSL12DT4)	CAN1 Message Slot 12 Data 5 (C1MSL12DT5)	13-68 13-69
H'0080 15CC	CAN1 Message Slot 12 Data 6 (C1MSL12DT6)	CAN1 Message Slot 12 Data 7 (C1MSL12DT7)	13-70 13-71
H'0080 15CE	CAN1 Message Slot 12 Timestamp (C1MSL12TSP)		13-72

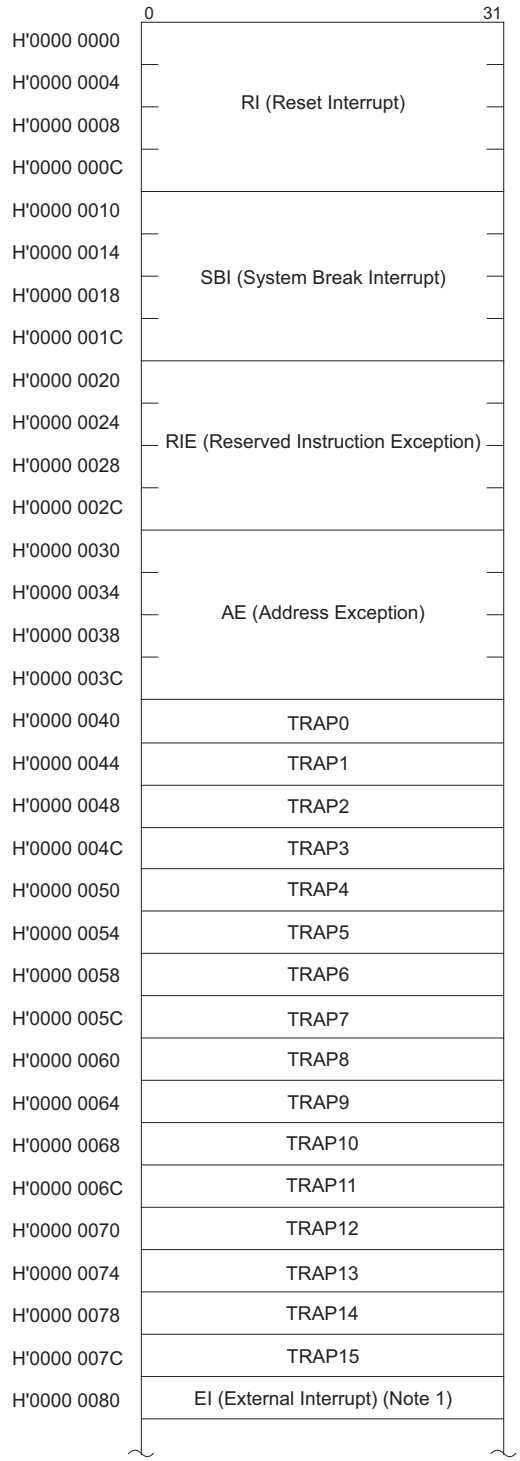
SFR Area Register Map (22/22)

Address	+0 address b0 – b7	+1 address b8 – b15	See pages
H'0080 15D0	CAN1 Message Slot 13 Standard ID 0 (C1MSL13SID0)	CAN1 Message Slot 13 Standard ID 1 (C1MSL13SID1)	13-58 13-59
H'0080 15D2	CAN1 Message Slot 13 Extended ID 0 (C1MSL13EID0)	CAN1 Message Slot 13 Extended ID 1 (C1MSL13EID1)	13-60 13-61
H'0080 15D4	CAN1 Message Slot 13 Extended ID 2 (C1MSL13EID2)	CAN1 Message Slot 13 Data Length Register (C1MSL13DLC)	13-62 13-63
H'0080 15D6	CAN1 Message Slot 13 Data 0 (C1MSL13DT0)	CAN1 Message Slot 13 Data 1 (C1MSL13DT1)	13-64 13-65
H'0080 15D8	CAN1 Message Slot 13 Data 2 (C1MSL13DT2)	CAN1 Message Slot 13 Data 3 (C1MSL13DT3)	13-66 13-67
H'0080 15DA	CAN1 Message Slot 13 Data 4 (C1MSL13DT4)	CAN1 Message Slot 13 Data 5 (C1MSL13DT5)	13-68 13-69
H'0080 15DC	CAN1 Message Slot 13 Data 6 (C1MSL13DT6)	CAN1 Message Slot 13 Data 7 (C1MSL13DT7)	13-70 13-71
H'0080 15DE	CAN1 Message Slot 13 Timestamp (C1MSL13TSP)		13-72
H'0080 15E0	CAN1 Message Slot 14 Standard ID 0 (C1MSL14SID0)	CAN1 Message Slot 14 Standard ID 1 (C1MSL14SID1)	13-58 13-59
H'0080 15E2	CAN1 Message Slot 14 Extended ID 0 (C1MSL14EID0)	CAN1 Message Slot 14 Extended ID 1 (C1MSL14EID1)	13-60 13-61
H'0080 15E4	CAN1 Message Slot 14 Extended ID 2 (C1MSL14EID2)	CAN1 Message Slot 14 Data Length Register (C1MSL14DLC)	13-62 13-63
H'0080 15E6	CAN1 Message Slot 14 Data 0 (C1MSL14DT0)	CAN1 Message Slot 14 Data 1 (C1MSL14DT1)	13-64 13-65
H'0080 15E8	CAN1 Message Slot 14 Data 2 (C1MSL14DT2)	CAN1 Message Slot 14 Data 3 (C1MSL14DT3)	13-66 13-67
H'0080 15EA	CAN1 Message Slot 14 Data 4 (C1MSL14DT4)	CAN1 Message Slot 14 Data 5 (C1MSL14DT5)	13-68 13-69
H'0080 15EC	CAN1 Message Slot 14 Data 6 (C1MSL14DT6)	CAN1 Message Slot 14 Data 7 (C1MSL14DT7)	13-70 13-71
H'0080 15EE	CAN1 Message Slot 14 Timestamp (C1MSL14TSP)		13-72
H'0080 15F0	CAN1 Message Slot 15 Standard ID 0 (C1MSL15SID0)	CAN1 Message Slot 15 Standard ID 1 (C1MSL15SID1)	13-58 13-59
H'0080 15F2	CAN1 Message Slot 15 Extended ID 0 (C1MSL15EID0)	CAN1 Message Slot 15 Extended ID 1 (C1MSL15EID1)	13-60 13-61
H'0080 15F4	CAN1 Message Slot 15 Extended ID 2 (C1MSL15EID2)	CAN1 Message Slot 15 Data Length Register (C1MSL15DLC)	13-62 13-63
H'0080 15F6	CAN1 Message Slot 15 Data 0 (C1MSL15DT0)	CAN1 Message Slot 15 Data 1 (C1MSL15DT1)	13-64 13-65
H'0080 15F8	CAN1 Message Slot 15 Data 2 (C1MSL15DT2)	CAN1 Message Slot 15 Data 3 (C1MSL15DT3)	13-66 13-67
H'0080 15FA	CAN1 Message Slot 15 Data 4 (C1MSL15DT4)	CAN1 Message Slot 15 Data 5 (C1MSL15DT5)	13-68 13-69
H'0080 15FC	CAN1 Message Slot 15 Data 6 (C1MSL15DT6)	CAN1 Message Slot 15 Data 7 (C1MSL15DT7)	13-70 13-71
H'0080 15FE	CAN1 Message Slot 15 Timestamp (C1MSL15TSP)		13-72
|	(Use inhibited area)		
H'0080 3FFE	(Use inhibited area)		

Note1. Address H'0080 0600 - H'0080 0603 are dummy areas.
When there is access to these areas, writing value is disabled and reading value is undefined.
In addition, it does not effect on the other SFR area by writing and reading out operation to dummy access area.

3.5 EIT Vector Entry

The EIT vector entry is located at the beginning of the internal ROM/external extension areas. The branch instruction for jumping to the start address of each EIT event processing handler is written here. Note that it is the branch instruction and not the jump address itself that is written here. For details, see Chapter 4, “EIT.”



Note 1: When flash entry bit = 1 (flash E/W enable mode), the EI vector entry is located at H'0080 4000.

Figure 3.5.1 EIT Vector Entry

3.6 ICU Vector Table

The ICU vector table is used by the internal interrupt controller of the microcomputer. This table has the addresses shown below, at which the start addresses of interrupt handlers for the interrupt requests from respective internal peripheral I/Os are set. For details, see Chapter 5, “Interrupt Controller.”

ICU Vector Table Memory Map (1/2)

Address	+0 address b0 b7	+1 address b8 b15
H'0000 0094	MJT Input Interrupt 4 Handler Start Address (A0–A15)	
H'0000 0096	MJT Input Interrupt 4 Handler Start Address (A16–A31)	
H'0000 0098	MJT Input Interrupt 3 Handler Start Address (A0–A15)	
H'0000 009A	MJT Input Interrupt 3 Handler Start Address (A16–A31)	
H'0000 009C	MJT Input Interrupt 2 Handler Start Address (A0–A15)	
H'0000 009E	MJT Input Interrupt 2 Handler Start Address (A16–A31)	
H'0000 00A0	MJT Input Interrupt 1 Handler Start Address (A0–A15)	
H'0000 00A2	MJT Input Interrupt 1 Handler Start Address (A16–A31)	
H'0000 00A4		
H'0000 00A6		
H'0000 00A8	MJT Output Interrupt 7 Handler Start Address (A0–A15)	
H'0000 00AA	MJT Output Interrupt 7 Handler Start Address (A16–A31)	
H'0000 00AC	MJT Output Interrupt 6 Handler Start Address (A0–A15)	
H'0000 00AE	MJT Output Interrupt 6 Handler Start Address (A16–A31)	
H'0000 00B0	MJT Output Interrupt 5 Handler Start Address (A0–A15)	
H'0000 00B2	MJT Output Interrupt 5 Handler Start Address (A16–A31)	
H'0000 00B4	MJT Output Interrupt 4 Handler Start Address (A0–A15)	
H'0000 00B6	MJT Output Interrupt 4 Handler Start Address (A16–A31)	
H'0000 00B8	MJT Output Interrupt 3 Handler Start Address (A0–A15)	
H'0000 00BA	MJT Output Interrupt 3 Handler Start Address (A16–A31)	
H'0000 00BC	MJT Output Interrupt 2 Handler Start Address (A0–A15)	
H'0000 00BE	MJT Output Interrupt 2 Handler Start Address (A16–A31)	
H'0000 00C0	MJT Output Interrupt 1 Handler Start Address (A0–A15)	
H'0000 00C2	MJT Output Interrupt 1 Handler Start Address (A16–A31)	
H'0000 00C4	MJT Output Interrupt 0 Handler Start Address (A0–A15)	
H'0000 00C6	MJT Output Interrupt 0 Handler Start Address (A16–A31)	
H'0000 00C8	DMA0–4 Interrupt Handler Start Address (A0–A15)	
H'0000 00CA	DMA0–4 Interrupt Handler Start Address (A16–A31)	
H'0000 00CC	SIO1 Receive Interrupt Handler Start Address (A0–A15)	
H'0000 00CE	SIO1 Receive Interrupt Handler Start Address (A16–A31)	
H'0000 00D0	SIO1 Transmit Interrupt Handler Start Address (A0–A15)	
H'0000 00D2	SIO1 Transmit Interrupt Handler Start Address (A16–A31)	
H'0000 00D4	SIO0 Receive Interrupt Handler Start Address (A0–A15)	
H'0000 00D6	SIO0 Receive Interrupt Handler Start Address (A16–A31)	

ICU Vector Table Memory Map (2/2)

Address	+0 address b0 – b7	+1 address b8 – b15
H'0000 00D8	SIO0 Transmit Interrupt Handler Start Address (A0–A15)	
H'0000 00DA	SIO0 Transmit Interrupt Handler Start Address (A16–A31)	
H'0000 00DC	A/D0 Conversion Interrupt Handler Start Address (A0–A15)	
H'0000 00DE	A/D0 Conversion Interrupt Handler Start Address (A16–A31)	
H'0000 00E0		
H'0000 00E2		
H'0000 00E4		
H'0000 00E6		
H'0000 00E8	DMA5–9 Interrupt Handler Start Address (A0–A15)	
H'0000 00EA	DMA5–9 Interrupt Handler Start Address (A16–A31)	
H'0000 00EC	SIO2, 3 Transmit/receive Interrupt Handler Start Address (A0–A15)	
H'0000 00EE	SIO2, 3 Transmit/receive Interrupt Handler Start Address (A16–A31)	
H'0000 00F0	RTD Interrupt Handler Start Address (A0–A15)	
H'0000 00F2	RTD Interrupt Handler Start Address (A16–A31)	
H'0000 00F4		
H'0000 00F6		
H'0000 00F8		
H'0000 00FA		
H'0000 00FC		
H'0000 00FE		
H'0000 0100		
H'0000 0102		
H'0000 0104		
H'0000 0106		
H'0000 0108		
H'0000 010A		
H'0000 010C	CAN0 Transmit/receive & Error Interrupt Handler Start Address (A0–A15)	
H'0000 010E	CAN0 Transmit/receive & Error Interrupt Handler Start Address (A16–A31)	
H'0000 0110	CAN1 Transmit/receive & Error Interrupt Handler Start Address (A0–A15)	
H'0000 0112	CAN1 Transmit/receive & Error Interrupt Handler Start Address (A16–A31)	

3.7 Notes on Address Space

• Virtual flash emulation function

The microcomputer has the function to map up to two 8-kbyte memory blocks of the internal RAM into areas of the internal flash memory (L banks) that are divided in 8-kbyte units, as well as to map up to two 4-kbyte memory blocks of the internal RAM into areas of the internal flash memory (S banks) that are divided in 4-kbyte units. This function is referred to as the virtual flash emulation function. For details about this function, refer to Section 6.6, “Virtual Flash Emulation Function.”

• Dummy access areas

Address H'0080 0600 - H'0080 0603 are dummy areas.
When there is access to these areas, writing value is disabled and reading value is undefined.
In addition, it does not effect on the other SFR area by writing and reading out operation to dummy access area.

This page is blank for reasons of layout.

CHAPTER 4

EIT

4.1 Outline of EIT

If some event occurs when the CPU is executing an ordinary program, it may become necessary to suspend the program being executed and execute another program. Events like this one are referred to by a generic name as EIT (Exception, Interrupt and Trap).

(1) Exception

This is an event related to the context being executed. It is generated by an error or violation during instruction execution. This type of event includes Address Exception (AE) and Reserved Instruction Exception (RIE) .

(2) Interrupt

This is an event generated irrespective of the context being executed. It is generated by a hardware-derived signal from an external source. This type of event includes Reset Interrupt (RI), System Break Interrupt (SBI) and External Interrupt (EI).

(3) Trap

This refers to a software interrupt generated by executing a TRAP instruction. This type of event is intentionally generated in a program as in the OS's system call by the programmer.

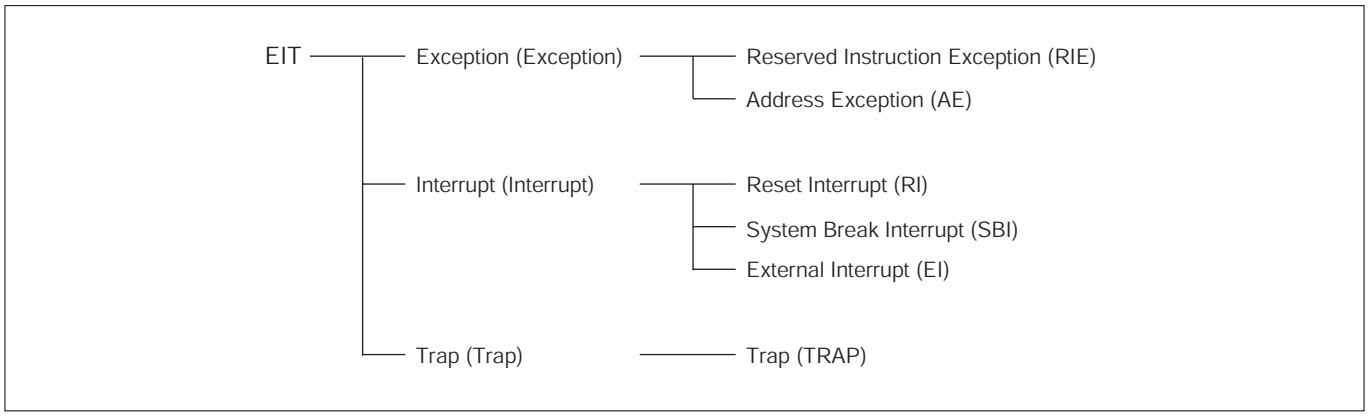


Figure 4.1.1 Classification of EITs

4.2 EIT Events

4.2.1 Exceptions

(1) Reserved Instruction Exception (RIE)

Reserved Instruction Exception (RIE) occurs when execution of a reserved instruction (unimplemented instruction) is detected.

(2) Address Exception (AE)

Address Exception (AE) occurs when an attempt is made to access a misaligned address in Load or Store instructions.

4.2.2 Interrupts

(1) Reset Interrupt (RI)

Reset Interrupt (RI) is always accepted by entering the RESET# signal. The reset interrupt is assigned the highest priority.

(2) System Break Interrupt (SBI)

System Break Interrupt (SBI) is an emergency interrupt which is used when power outage is detected or a fault condition is notified by an external watchdog timer. This interrupt can only be used in cases when after interrupt processing, control will not return to the program that was being executed when the interrupt occurred.

(3) External Interrupt (EI)

External Interrupt (EI) is requested from internal peripheral I/Os managed by the interrupt controller. The internal interrupt controller manages these interrupts by assigning each one of eight priority levels including an interrupt-disabled state.

4.2.3 Trap

Traps are software interrupts which are generated by executing the TRAP instruction. Sixteen distinct vector addresses are provided corresponding to TRAP instruction operands 0–15.

4.3 EIT Processing Procedure

EIT processing consists of two parts, one in which they are handled automatically by hardware, and one in which they are handled by user-created programs (EIT handlers). The procedure for processing EITs when accepted, except for a reset interrupt, is shown below.

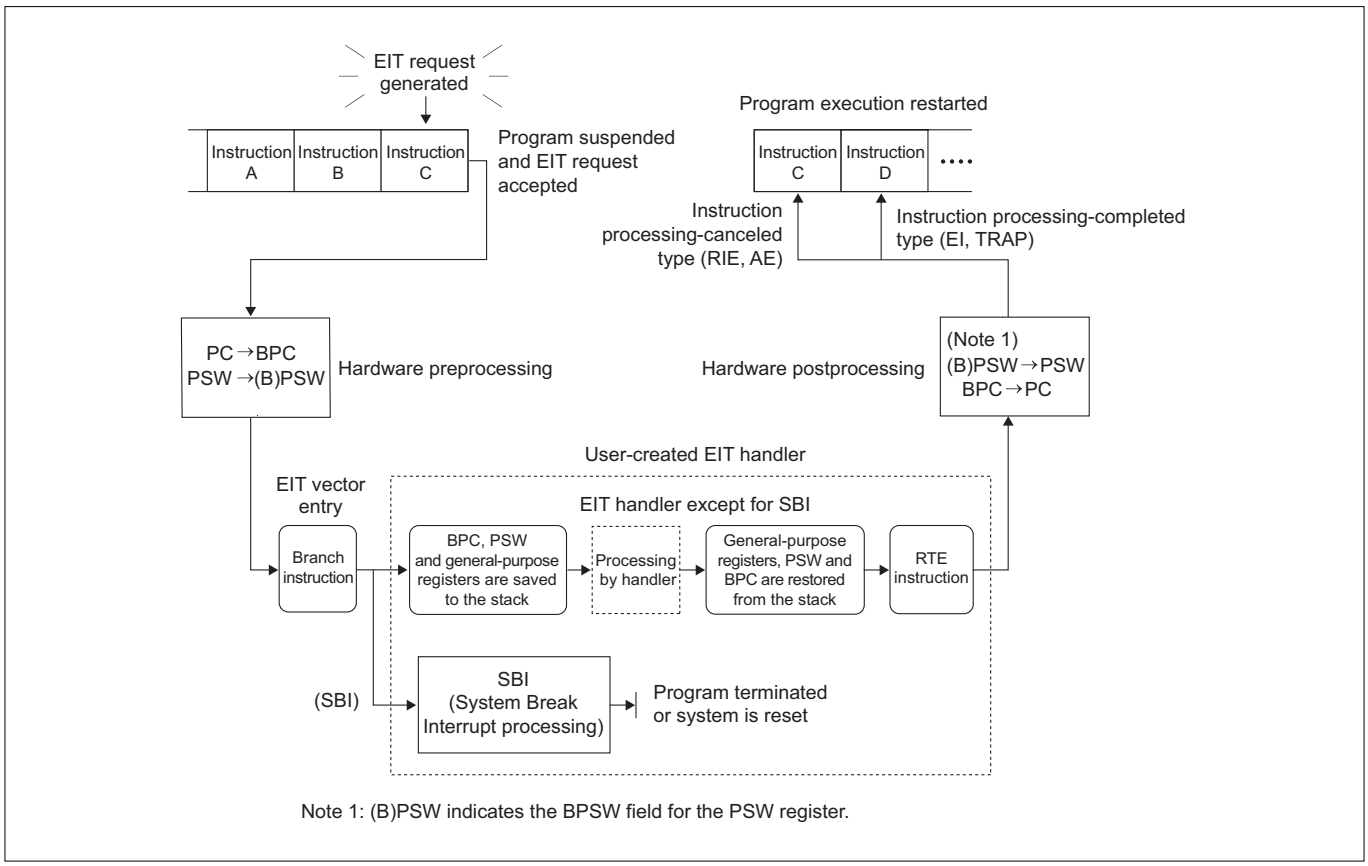


Figure 4.3.1 Outline of the EIT Processing Procedure

When an EIT is accepted, the CPU branches to the EIT vector after hardware preprocessing (as will be described later). The EIT vector has an entry address assigned for each EIT. This is where the BRA (branch) instruction for the EIT handler (not the jump address itself) is written.

In the hardware preprocessing, the content of the PC and PSW registers is transferred to the backup register (BPC register and BPSW field in the PSW register).
Other necessary operations must be performed in the user-created EIT handler. These include saving the BPC register and PSW register (including the BPSW field) and the general-purpose registers to be used in the EIT handler to the stack. In addition, the accumulator must be saved to the stack as necessary. Remember that all these registers must be saved to the stack in a program by the user.

When processing by the EIT handler is completed, restore the saved registers from the stack and finally execute the RTE instruction. Control is thereby returned from the EIT processing to the program that was being executed when the EIT occurred. (This does not apply to the System Break Interrupt, however.)
In the hardware postprocessing, the content of the backup register (BPC register and BPSW field in the PSW register) is returned to the PC and PSW registers. Note that the values stored in the BPC and the PSW register's BPSW field after executing the RTE instruction are undefined.

4.4 EIT Processing Mechanism

The EIT processing mechanism consists of the M32R CPU core and the interrupt controller for internal peripheral I/Os. It also has the backup registers for the PC and PSW (the BPC register and the BPSW field in the PSW register). The EIT processing mechanism is shown below.

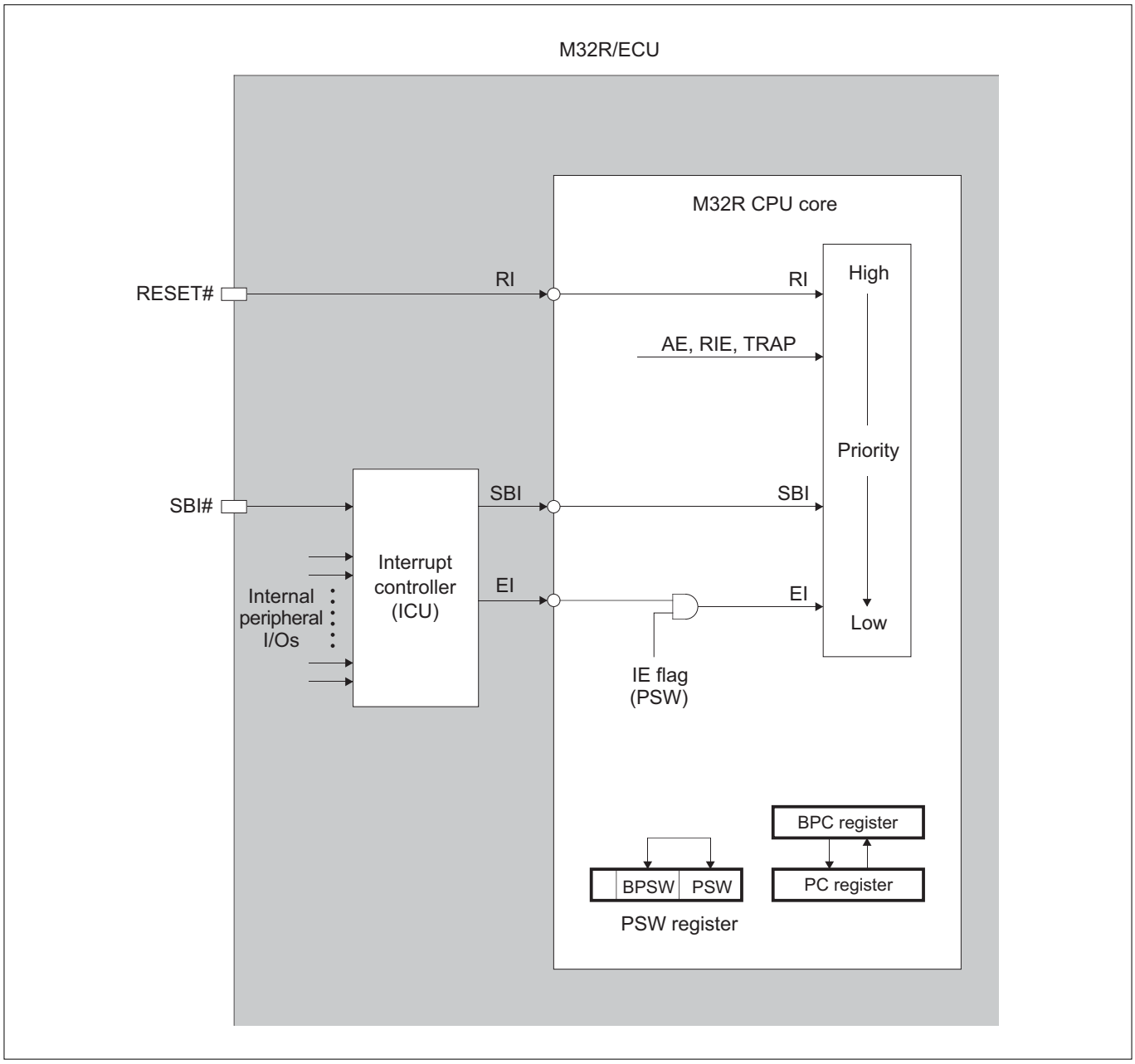


Figure 4.4.1 EIT Processing Mechanism

4.5 Acceptance of EIT Events

When an EIT event occurs, the CPU suspends the program it has hitherto been executing and branches to EIT processing by the relevant handler. Conditions under which each EIT event occurs and the timing at which they are accepted are shown below.

Table 4.5.1 Acceptance of EIT Events

EIT Event	Type of Processing	Acceptance Timing	Values Set in BPC Register
Reserved Instruction Exception (RIE)	Instruction processing-canceled type	During instruction execution	PC value of the instruction that generated RIE
Address Exception (AE)	Instruction processing-canceled type	During instruction execution	PC value of the instruction that generated AE
Reset Interrupt (RI)	Instruction processing-aborted type	Each machine cycle	Undefined value
System Break Interrupt (SBI)	Instruction processing-completed type	Break in instructions (word boundary only)	PC value of the next instruction
External Interrupt (EI)	Instruction processing-completed type	Break in instructions (word boundary only)	PC value of the next instruction
Trap (TRAP)	Instruction processing-completed type	Break in instructions	PC value of TRAP instruction + 4

4.6 Saving and Restoring the PC and PSW

The following describe operation of the microcomputer at the time when it accepts an EIT and when it executes the RTE instruction.

(1) Hardware preprocessing when an EIT is accepted

[1] Save the PSW register's SM, IE and C bits in its backup field.

BSM ← SM
BIE ← IE
BC ← C

[2] Update the PSW register's SM, IE and C bits

SM ← Remains unchanged (RIE, AE, TRAP) or cleared to "0" (SBI, EI, RI)
IE ← Cleared to "0"
C ← Cleared to "0"

[3] Save the PC register

BPC ← PC

[4] Set the vector address in the PC register

Branches to the EIT vector and executes the branch (BRA) instruction written in it, thereby transferring control to the user-created EIT handler.

(2) Hardware postprocessing when the RTE instruction is executed

[A] Restore the PSW register's SM, IE and C bits from its backup field.

SM ← BSM
IE ← BIE
C ← BC

[B] Restore the PC register from the BPC register.

PC ← BPC

Note: • The values stored in the BPC and the PSW register's BSM, BIE and BC bits after executing the RTE instruction are undefined.

[1] Saving the SM, IE and C bits

BSM ← SM
BIE ← IE
BC ← C

[2] Updating the SM, IE and C bits

SM ← Unchanged or 0
IE ← 0
C ← 0

[3] Saving the PC

BPC ← PC

[4] Setting the vector address in the PC

PC ← Vector address

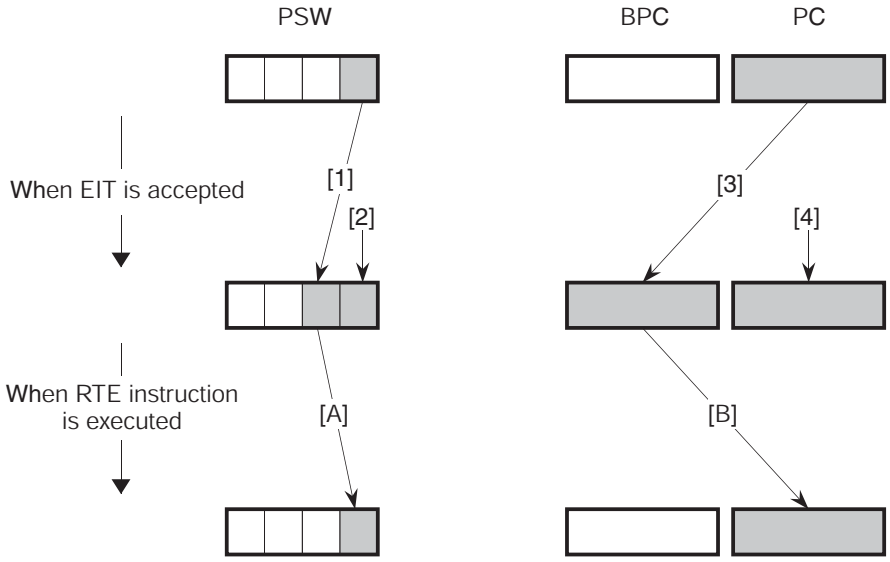
[A] Restoring the SM, IE and C bits from the backup field

SM ← BSM
IE ← BIE
C ← BC

The values stored in the BSM, BIE and BC bits after executing the RTE instruction are undefined.

[B] Restoring the PC from the BPC register

The value stored in the BPC register after executing the RTE instruction is undefined.



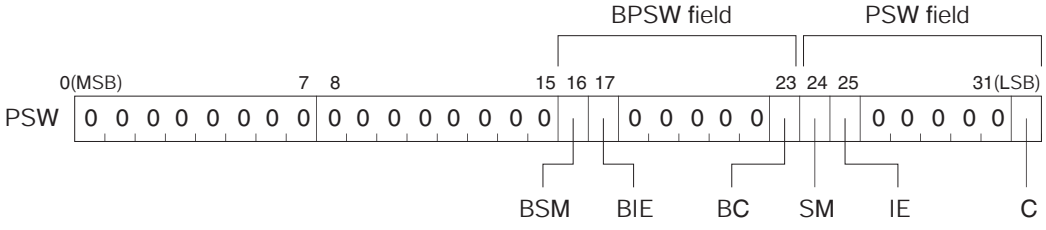


Figure 4.6.1 Saving and Restoring the PC and PSW

4.7 EIT Vector Entry

The EIT vector entry is located in the user space beginning with the address H'0000 0000. The table below lists the EIT vector entry.

Table 4.7.1 EIT Vector Entry

Name	Abbreviation	Vector Address	SM	IE	BPC
Reset Interrupt	RI	H'0000 0000 (Note 1)	0	0	Undefined
System Break Interrupt	SBI	H'0000 0010	0	0	PC of the next instruction
Reserved Instruction Exception	RIE	H'0000 0020	Unchanged	0	PC of the instruction that generated RIE
Address Exception	AE	H'0000 0030	Unchanged	0	PC of the instruction that generated AE
Trap	TRAP0	H'0000 0040	Unchanged	0	PC of TRAP instruction + 4
	TRAP1	H'0000 0044	Unchanged	0	PC of TRAP instruction + 4
	TRAP2	H'0000 0048	Unchanged	0	PC of TRAP instruction + 4
	TRAP3	H'0000 004C	Unchanged	0	PC of TRAP instruction + 4
	TRAP4	H'0000 0050	Unchanged	0	PC of TRAP instruction + 4
	TRAP5	H'0000 0054	Unchanged	0	PC of TRAP instruction + 4
	TRAP6	H'0000 0058	Unchanged	0	PC of TRAP instruction + 4
	TRAP7	H'0000 005C	Unchanged	0	PC of TRAP instruction + 4
	TRAP8	H'0000 0060	Unchanged	0	PC of TRAP instruction + 4
	TRAP9	H'0000 0064	Unchanged	0	PC of TRAP instruction + 4
	TRAP10	H'0000 0068	Unchanged	0	PC of TRAP instruction + 4
	TRAP11	H'0000 006C	Unchanged	0	PC of TRAP instruction + 4
	TRAP12	H'0000 0070	Unchanged	0	PC of TRAP instruction + 4
	TRAP13	H'0000 0074	Unchanged	0	PC of TRAP instruction + 4
	TRAP14	H'0000 0078	Unchanged	0	PC of TRAP instruction + 4
	TRAP15	H'0000 007C	Unchanged	0	PC of TRAP instruction + 4
External Interrupt	EI	H'0000 0080 (Note 2)	0	0	PC of the next instruction

Note 1: During boot mode, the CPU starts executing the boot program after exiting the reset state. For details, see Section 6.5, "Programming the Internal Flash Memory."

Note 2: During flash E/W enable mode, this vector address is moved to the beginning of the internal RAM (address H'0080 4000). For details, see Section 6.5, "Programming the Internal Flash Memory."

4.8 Exception Processing

4.8.1 Reserved Instruction Exception (RIE)

[Occurrence Conditions]

Reserved Instruction Exception (RIE) occurs when a reserved instruction (unimplemented instruction) is detected. Instruction check is performed on the op-code part of the instruction.

When a reserved instruction exception occurs, the instruction that generated it is not executed. If an external interrupt is requested at the same time a reserved instruction exception is detected, it is the reserved instruction exception that is accepted.

[EIT Processing]

(1) Saving SM, IE and C bits

The PSW register's SM, IE and C bits are saved to the respective backup bits: BSM, BIE and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating SM, IE and C bits

The PSW register's SM, IE and C bits are updated as shown below.

SM ← Unchanged
IE ← 0
C ← 0

(3) Saving the PC

The PC value of the instruction that generated the reserved instruction exception is set in the BPC register. For example, if the instruction that generated the reserved instruction exception is at address 4, the value 4 is set in the BPC register. Similarly, if the instruction that generated the reserved instruction exception is at address 6, the value 6 is set in the BPC register. In this case, the value of the BPC register bit 30 indicates whether the instruction that generated the reserved instruction exception resides on a word boundary (BPC[30] = 0) or not on a word boundary (BPC[30] = 1).

However, in either case of the above, the address to which the RTE instruction returns after the EIT handler has terminated is address 4. (This is because the 2 low-order address bits are cleared to '00' when returned to the PC.)

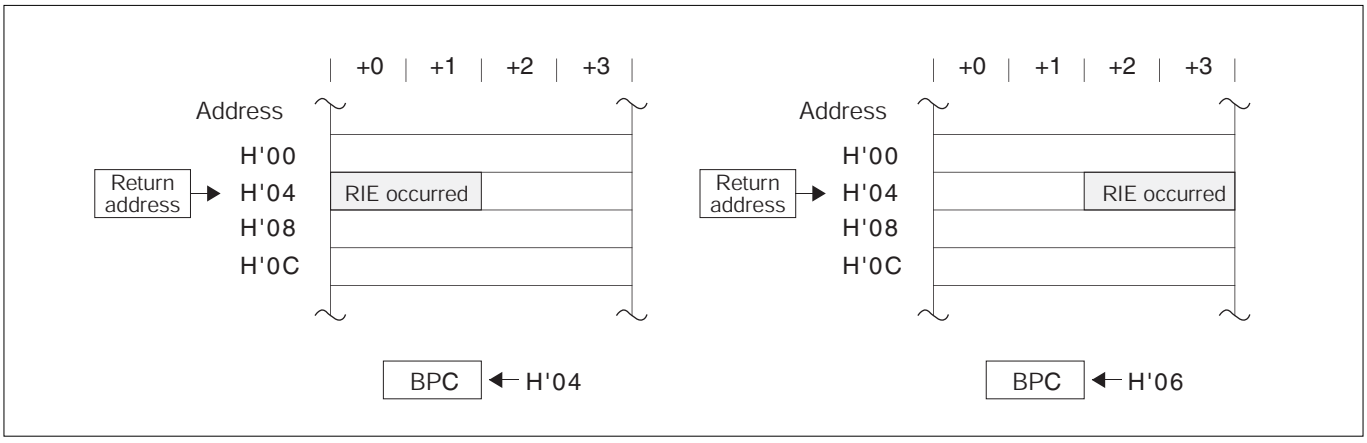


Figure 4.8.1 Example of a Return Address for Reserved Instruction Exception (RIE)

(4) Branching to the EIT vector entry

The CPU branches to the address H'0000 0020 in the user space. This is the last operation performed in hardware preprocessing.

(5) Jumping from the EIT vector entry to the user-created handler

The CPU executes the BRA instruction written by the user at the address H'0000 0020 of the EIT vector entry to jump to the start address of the user-created handler. At the beginning of the user-created EIT handler, first save the BPC and PSW registers and the necessary general-purpose registers to the stack. Also, save the accumulator as necessary.

(6) Returning from the EIT handler

At the end of the EIT handler, restore the saved registers from the stack and execute the RTE instruction. When the RTE instruction is executed, hardware postprocessing is automatically performed. At this time, the CPU restarts from a word-boundary instruction including the instruction that generated a RIE (see Figure 4.8.1). Except when using reserved instruction exceptions intentionally, occurrence of a reserved instruction exception suggests that the system has some fatal fault already existing in it. In such a case, therefore, do not return from the reserved instruction exception handler to the program that was being executed when the exception occurred.

4.8.2 Address Exception (AE)

[Occurrence Conditions]

Address Exception (AE) occurs when an attempt is made to access a misaligned address in Load or Store instructions. The following lists the combination of instructions and accessed addresses that may cause address exceptions to occur.

- Two low-order address bits accessed in the LDH, LDUH or STH instruction are '01' or '11'
- Two low-order address bits accessed in the LD, ST, LOCK or UNLOCK instruction are '01,' '10' or '11'

When an address exception occurs, memory access by the instruction that generated the exception is not performed. If an external interrupt is requested at the same time an address exception is detected, it is the address exception that is accepted.

[EIT Processing]

(1) Saving SM, IE and C bits

The PSW register's SM, IE and C bits are saved to the respective backup bits: BSM, BIE and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating SM, IE and C bits

The PSW register's SM, IE and C bits are updated as shown below.

SM ← Unchanged
IE ← 0
C ← 0

(3) Saving the PC

The PC value of the instruction that generated the address exception is set in the BPC register. For example, if the instruction that generated the address exception is at address 4, the value 4 is set in the BPC register. Similarly, if the instruction that generated the address exception is at address 6, the value 6 is set in the BPC register. In this case, the value of the BPC register bit 30 indicates whether the instruction that generated the reserved instruction exception resides on a word boundary (BPC[30] = 0) or not on a word boundary (BPC[30] = 1).

However, in either case of the above, the address to which the RTE instruction returns after the EIT handler has terminated is address 4. (This is because the 2 low-order address bits are cleared to '00' when returned to the PC.)

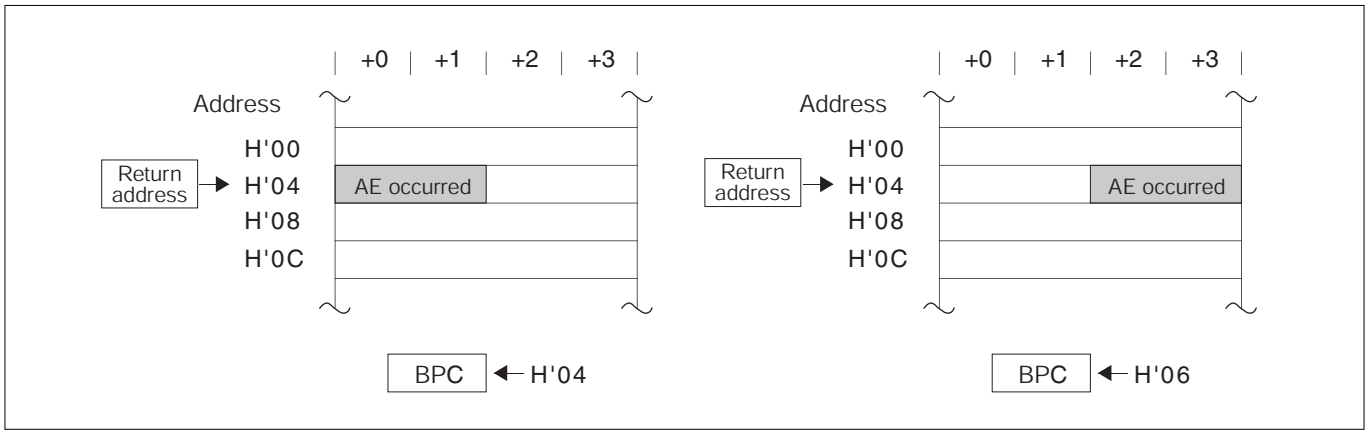


Figure 4.8.2 Example of a Return Address for Address Exception (AE)

(4) Branching to the EIT vector entry

The CPU branches to the address H'0000 0030 in the user space. This is the last operation performed in hardware preprocessing.

(5) Jumping from the EIT vector entry to the user-created handler

The CPU executes the BRA instruction written by the user at the address H'0000 0030 of the EIT vector entry to jump to the start address of the user-created handler. At the beginning of the user-created EIT handler, first save the BPC and PSW registers and the necessary general-purpose registers to the stack. Also, save the accumulator as necessary.

(6) Returning from the EIT handler

At the end of the EIT handler, restore the saved registers from the stack and execute the RTE instruction. When the RTE instruction is executed, hardware postprocessing is automatically performed. At this time, the CPU restarts from a word-boundary instruction including the instruction that generated an AE (see Figure 4.8.2). Except when using address exceptions intentionally, occurrence of an address exception suggests that the system has some fatal fault already existing in it. In such a case, therefore, do not return from the address exception handler to the program that was being executed when the exception occurred.

4.9 Interrupt Processing

4.9.1 Reset Interrupt (RI)

[Occurrence Conditions]

A reset interrupt is accepted in machine cycle by pulling the RESET# input signal "L." The reset interrupt is assigned the highest priority among all EITs.

[EIT Processing]

(1) Initializing SM, IE and C bits

The PSW register's SM, IE and C bits are initialized as shown below.

SM ← 0
IE ← 0
C ← 0

For the reset interrupt, the values of BSM, BIE and BC bits are undefined.

(2) Branching to the EIT vector entry

The CPU branches to the address H'0000 0000 in the user space. However, when operating in boot mode, the CPU jumps to the boot program. For details, see Section 6.5, "Programming the Internal Flash Memory."

(3) Jumping from the EIT vector entry to the user program

The CPU executes the instruction written by the user at the address H'0000 0000 of the EIT vector entry. In the reset vector entry, be sure to initialize the PSW and SPI registers before jumping to the start address of the user program.

4.9.2 System Break Interrupt (SBI)

System Break Interrupt (SBI) is an emergency interrupt which is used when power outage is detected or a fault condition is notified by an external watchdog timer. The system break interrupt cannot be masked by the PSW register IE bit.

Therefore, the system break interrupt can only be used when the system has some fatal event already existing in it when the interrupt is detected. Also, this interrupt must be used on condition that after processing by the SBI handler, control will not return to the program that was being executed when the system break interrupt occurred.

[Occurrence Conditions]

A system break interrupt is accepted by a falling edge on SBI# input pin. (The system break interrupt cannot be masked by the PSW register IE bit.)

In no case will a system break interrupt be activated immediately after executing a 16-bit instruction that starts from a word boundary. (For 16-bit branch instructions, however, the interrupt is accepted immediately after branching.) Note also that because of the instruction processing-completed type, a system break interrupt is accepted after the instruction is completed.

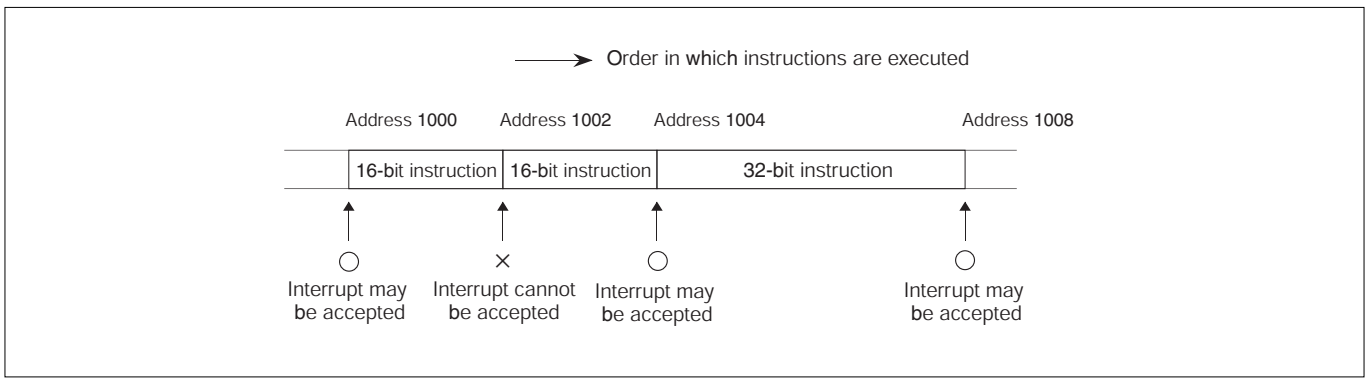


Figure 4.9.1 Timing at Which System Break Interrupt (SBI) is Accepted

[EIT Processing]

(1) Saving SM, IE and C bits

The PSW register's SM, IE and C bits are saved to the respective backup bits: BSM, BIE and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating SM, IE and C bits

The PSW register's SM, IE and C bits are updated as shown below.

SM ← 0
IE ← 0
C ← 0

(3) Saving the PC

The content of the PC register (always on word boundary) is saved to the BPC register. If the interrupt was detected in a branch instruction, then the next instruction is one that exists at the jump address.

(4) Branching to the EIT vector entry

The CPU branches to the address H'0000 0010 in the user space. This is the last operation performed in hardware preprocessing.

(5) Jumping from the EIT vector entry to the user-created handler

The CPU executes the BRA instruction written by the user at the address H'0000 0010 of the EIT vector entry to jump to the start address of the user-created handler.

The system break interrupt can only be used when the system has some fatal event already existing in it when the interrupt is detected. Also, this interrupt must be used on condition that after processing by the SBI handler, control will not return to the program that was being executed when the system break interrupt occurred.

4.9.3 External Interrupt (EI)

An external interrupt is generated upon an interrupt request which is output by the microcomputer's internal interrupt controller. The interrupt controller manages interrupt requests by assigning each one of seven priority levels. For details, see Chapter 5, "Interrupt Controller." For details about the interrupt request sources, see each section in which the relevant internal peripheral I/O is described.

[Occurrence Conditions]

External interrupts are managed by the microcomputer's internal interrupt controller based on interrupt requests from each internal peripheral I/O, and are sent to the CPU via the interrupt controller. The CPU checks these interrupt requests at a break in instructions residing on word boundaries, and when an interrupt request is detected and the PSW register IE flag = "1", accepts it as an external interrupt.

In no case will an external interrupt be activated immediately after executing a 16-bit instruction that starts from a word boundary. (For 16-bit branch instructions, however, the interrupt is accepted immediately after branching.)

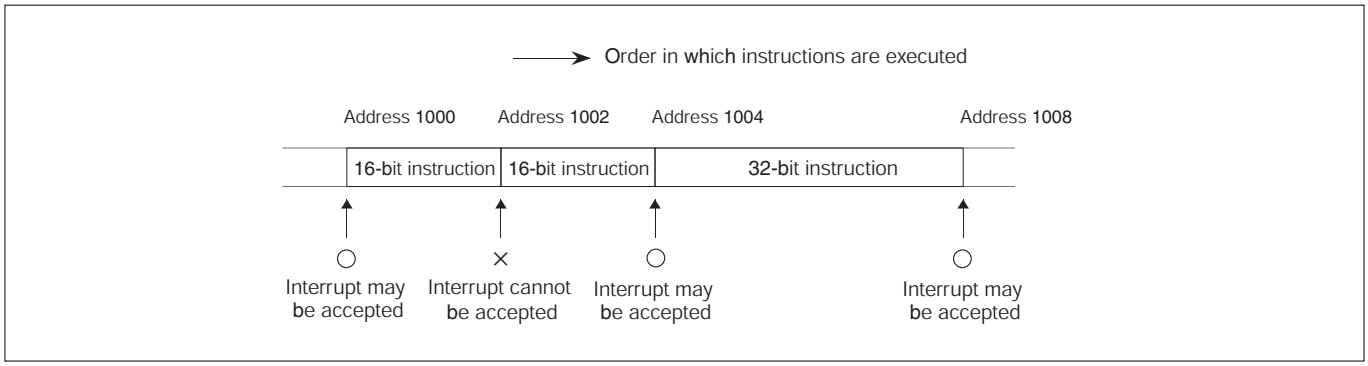


Figure 4.9.2 Timing at Which External Interrupt (EI) is Accepted

[EIT Processing]

(1) Saving SM, IE and C bits

The PSW register's SM, IE and C bits are saved to the respective backup bits: BSM, BIE and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating SM, IE and C bits

The PSW register's SM, IE and C bits are updated as shown below.

SM ← 0
IE ← 0
C ← 0

(3) Saving the PC

The content of the PC register (always on word boundary) is saved to the BPC register.

(4) Branching to the EIT vector entry

The CPU branches to the address H'0000 0080 in the user space. However, when operating in flash E/W enable mode, the CPU goes to the beginning of the internal RAM (address H'0080 4000). (For details, see Section 6.5, "Programming the Internal Flash Memory.") This is the last operation performed in hardware preprocessing.

(5) Jumping from the EIT vector entry to the user-created handler

The CPU executes the BRA instruction written by the user at the address H'0000 0080 of the EIT vector entry to jump to the start address of the user-created handler. At the beginning of the user-created EIT handler, first save the BPC and PSW registers and the necessary general-purpose registers to the stack. Also, save the accumulator as necessary.

(6) Returning from the EIT handler

At the end of the EIT handler, restore the saved registers from the stack and execute the RTE instruction. When the RTE instruction is executed, hardware postprocessing is automatically performed.

4.10 Trap Processing

4.10.1 Trap

[Occurrence Conditions]

Traps are software interrupts which are generated by executing the TRAP instruction. Sixteen traps are generated, each corresponding to one of TRAP instruction operands 0–15. Accordingly, sixteen vector entries are provided.

[EIT Processing]

(1) Saving SM, IE and C bits

The PSW register's SM, IE and C bits are saved to the respective backup bits: BSM, BIE and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating SM, IE and C bits

The PSW register's SM, IE and C bits are updated as shown below.

SM ← Unchanged
IE ← 0
C ← 0

(3) Saving the PC

When the trap instruction is executed, the PC value of TRAP instruction + 4 is set in the BPC register. For example, if the TRAP instruction is located at address 4, the value H'08 is set in the BPC register. Similarly, if the TRAP instruction is located at address 6, the value H'0A is set in the BPC register. The value of the BPC register bit 30 indicates whether the trap instruction resides on a word boundary (BPC register 30 = "0") or not on a word boundary (BPC register 30 = "1").

However, in either case of the above, the address to which the RTE instruction returns after the EIT handler has terminated is address 8. (This is because the 2 low-order address bits are cleared to '00' when returned to the PC.)

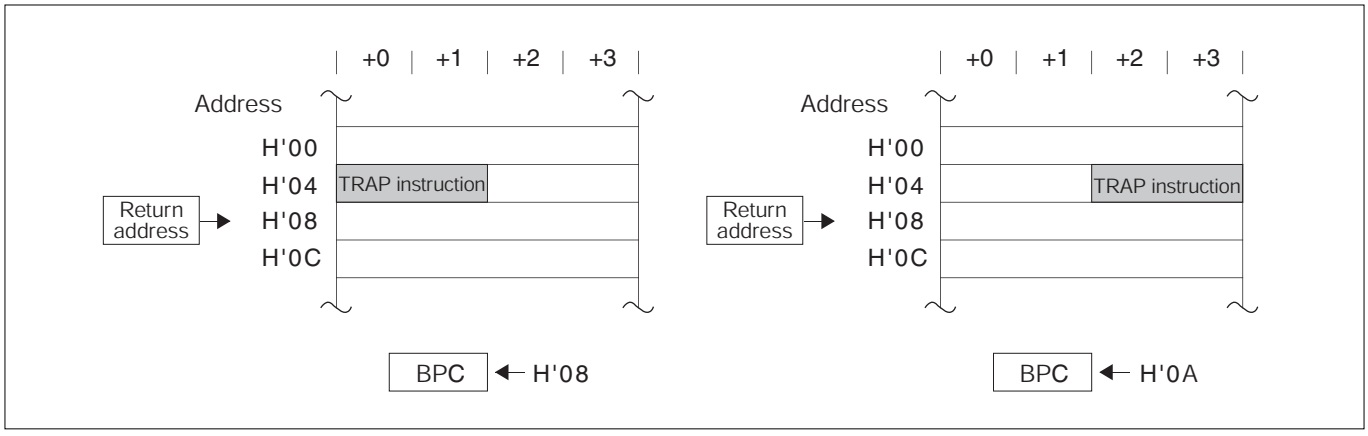


Figure 4.10.1 Example of a Return Address for Trap (TRAP)

(4) Branching to the EIT vector entry

The CPU branches to the addresses H'0000 0040–H'0000 007C in the user space. This is the last operation performed in hardware preprocessing.

(5) Jumping from the EIT vector entry to the user-created handler

The CPU executes the BRA instruction written by the user at the addresses H'0000 0040–H'0000 007C of the EIT vector entry to jump to the start address of the user-created handler. At the beginning of the user-created EIT handler, first save the BPC and PSW registers and the necessary general-purpose registers to the stack. Also, save the accumulator as necessary.

(6) Returning from the EIT handler

At the end of the EIT handler, restore the saved registers from the stack and execute the RTE instruction. When the RTE instruction is executed, hardware postprocessing is automatically performed.

4.11 EIT Priority Levels

The table below lists the priority levels of EIT events. When two or more EITs occur simultaneously, the event with the highest priority is accepted first.

Table 4.11.1 Priority of EIT Events and How Returned from EIT

Priority	EIT Event	Type of Processing	Values Set in BPC Register
1 (Highest)	Reset Interrupt (RI)	Instruction processing-aborted type	Undefined
2	Address Exception (AE)	Instruction processing-canceled type	PC of the instruction that generated AE
	Reserved Instruction Exception (RIE)	Instruction processing-canceled type	PC of the instruction that generated RIE
	Trap (TRAP)	Instruction processing-completed type	TRAP instruction + 4
3	System Break Interrupt (SBI)	Instruction processing-completed type	PC of the next instruction
4	External Interrupt (EI)	Instruction processing-completed type	PC of the next instruction

Note that for External Interrupt (EI), the priority levels of interrupt requests from each peripheral I/O are set by the microcomputer's internal interrupt controller. For details, see Chapter 5, "Interrupt Controller."

4.12 Example of EIT Processing

(1) When RIE, AE, SBI, EI or TRAP occurs singly

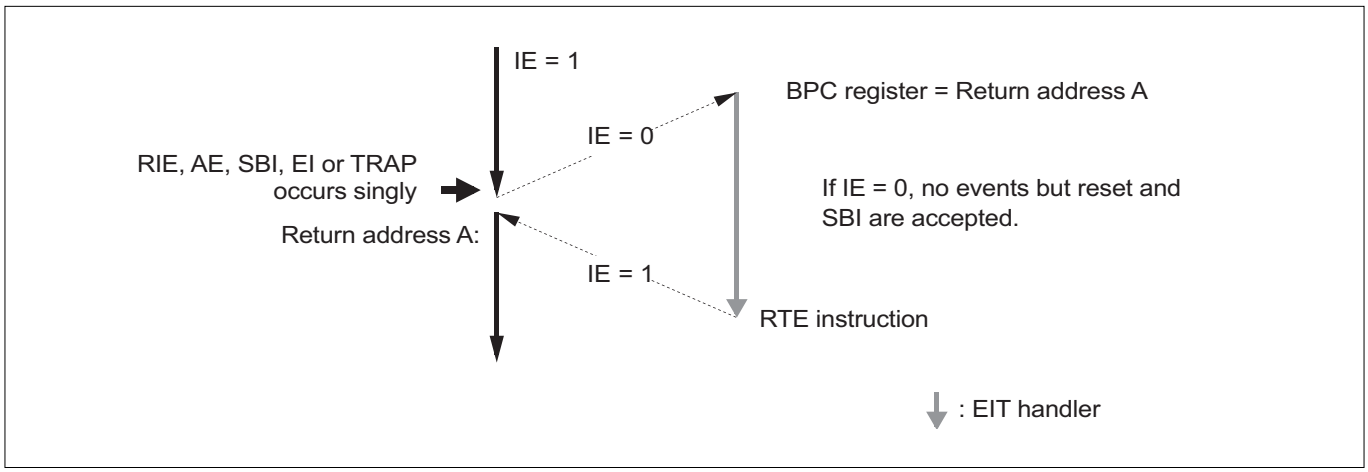


Figure 4.12.1 Processing of Events When RIE, AE, SBI, EI or TRAP Occurs Singly

(2) When RIE, AE or TRAP and EI occur simultaneously

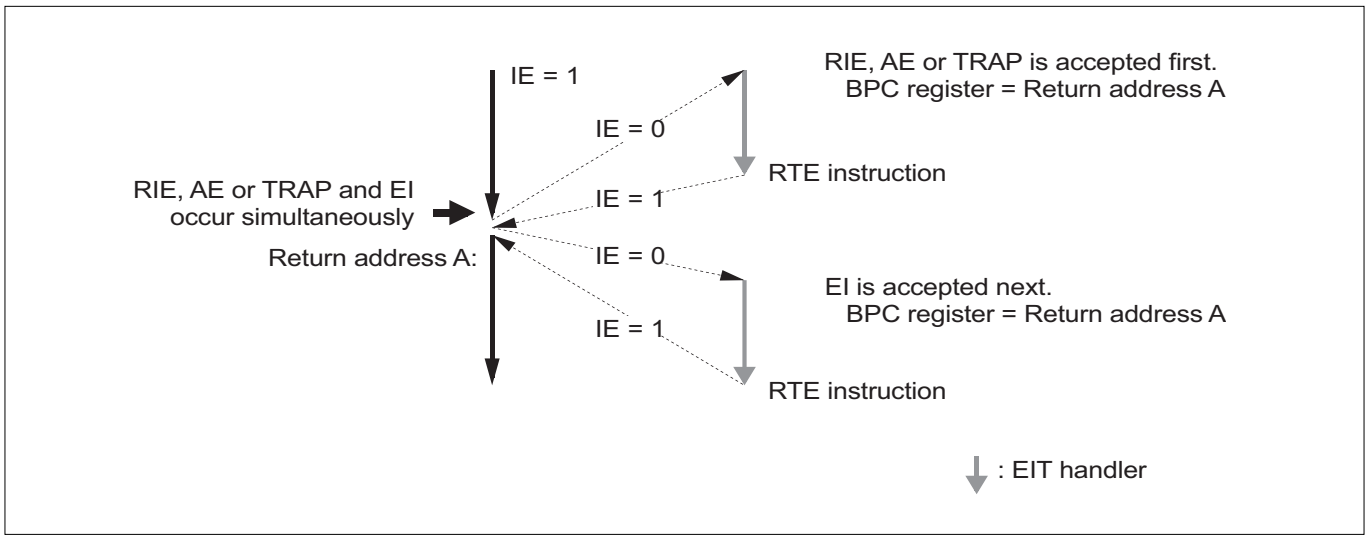
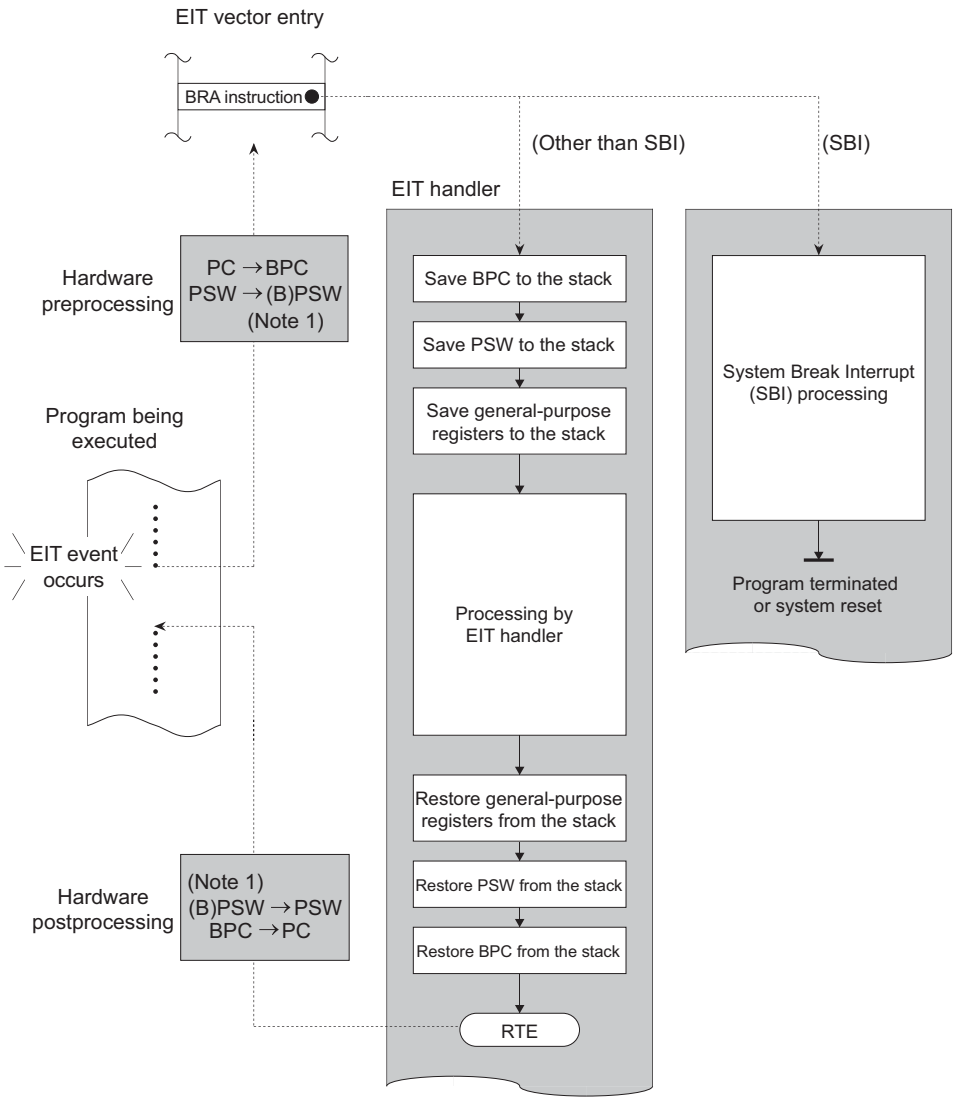


Figure 4.12.2 Processing of Events When RIE, AE or TRAP and EI Occur Simultaneously



Note 1: Indicates saving and restoring the PSW register bits between its PSW and BPSW fields.

Figure 4.12.3 Example of EIT Processing

4.13 Notes on EIT

The Address Exception (AE) requires caution because if one of the instructions that use “register indirect + register update” addressing mode (following three) generates an address exception when it is executed, the values of the registers to be automatically updated (Rsrc and Rsrc2) become undefined.
Except that the values of Rsrc and Rsrc2 become undefined, these instructions behave the same way as when used in other addressing modes.

- **Applicable instructions**

```
LD      Rdest, @Rsrc+
ST      Rsrc1, @-Rsrc2
ST      Rsrc1, @+Rsrc2
```

If the above case applies, consider the fact that the register values become undefined when you design the processing to be performed after executing said instructions. (If an address exception occurs, it means that the system has some fatal fault already existing in it. Therefore, address exceptions must be used on condition that control will not be returned from the address exception handler to the program that was being executed when the exception occurred.)

CHAPTER 5

INTERRUPT CONTROLLER (ICU)

5.1 Outline of the Interrupt Controller

The Interrupt Controller (ICU) manages maskable interrupts from internal peripheral I/Os and a system break interrupt (SBI). The maskable interrupts from internal peripheral I/Os are sent to the M32R CPU as external interrupts (EI).

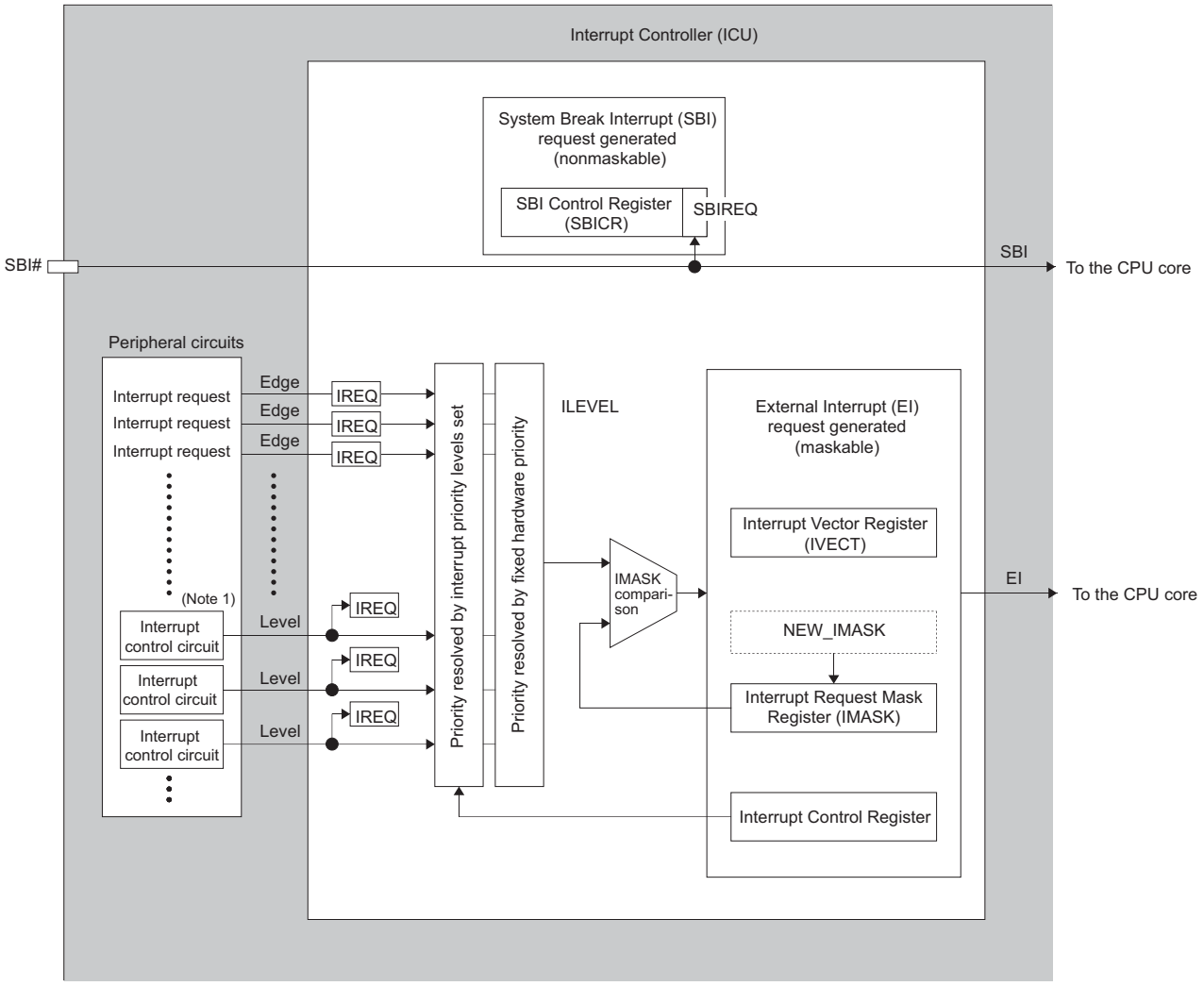
The maskable interrupts from internal peripheral I/Os are managed by assigning them one of eight priority levels including an interrupt-disabled state. If two or more interrupt requests with the same priority level occur at the same time, their priorities are resolved by predetermined hardware priority. The source of an interrupt request generated in internal peripheral I/Os is identified by reading the relevant interrupt status register provided for internal peripheral I/Os.

On the other hand, the system break interrupt (SBI) is recognized when a falling edge occurs on the SBI# signal input pin. This interrupt is used for emergency purposes such as when power outage is detected or a fault condition is notified by an external watchdog timer, so that it is always accepted irrespective of the PSW register IE bit status. After processing of an SBI, shut down or reset the system without returning to the program that was being executed when the interrupt occurred.
Specifications of the Interrupt Controller are outlined below.

Table 5.1.1 Outline of the Interrupt Controller (ICU)

Item	Specification
Interrupt request source	Maskable interrupt requests from internal peripheral I/Os: 23 sources (Note 1) System break interrupt request: 1 source (input from SBI# pin)
Priority management	8 priority levels including an interrupt-disabled state (However, interrupts with the same priority level have their priorities resolved by fixed hardware priority.)

Note 1: This is the number of interrupt requests divided into groups. There are actually a total of 123 interrupt request sources when counted individually.



Note 1. Interrupt control circuit indicates Interrupt request status register and Interrupt request mask register in each peripheral function.

Figure 5.1.1 Block Diagram of the Interrupt Controller

5.2 ICU Related Registers

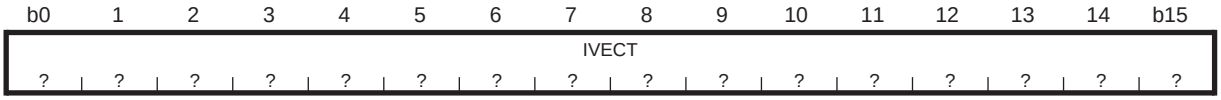
The diagram below shows a register map associated with the Interrupt Controller (ICU).

ICU Related Register Map

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0000	Interrupt Vector Register (IVECT)		5-5
H'0080 0002	(Use inhibited area)		
H'0080 0004	Interrupt Request Mask Register (IMASK)	(Use inhibited area)	5-6
H'0080 0006	SBI Control Register (SBICR)	(Use inhibited area)	5-7
	(Use inhibited area)		
H'0080 0060	CAN0 Transmit/Receive & Error Interrupt Control Register (ICAN0CR)	(Use inhibited area)	5-8
H'0080 0062	(Use inhibited area)		
H'0080 0064	(Use inhibited area)		
H'0080 0066	(Use inhibited area)	RTD Interrupt Control Register (IRTDCR)	5-8
H'0080 0068	SIO2,3 Transmit/Receive Interrupt Control Register (ISIO23CR)	DMA5–9 Interrupt Control Register (IDMA59CR)	5-8
H'0080 006A	(Use inhibited area)		
H'0080 006C	A/D0 Conversion Interrupt Control Register (IAD0CCR)	SIO0 Transmit Interrupt Control Register (ISIO0TXCR)	5-8
H'0080 006E	SIO0 Receive Interrupt Control Register (ISIO0RXCR)	SIO1 Transmit Interrupt Control Register (ISIO1TXCR)	5-8
H'0080 0070	SIO1 Receive Interrupt Control Register (ISIO1RXCR)	DMA0–4 Interrupt Control Register (IDMA04CR)	5-8
H'0080 0072	MJT Output Interrupt Control Register 0 (IMJTOCR0)	MJT Output Interrupt Control Register 1 (IMJTOCR1)	5-8
H'0080 0074	MJT Output Interrupt Control Register 2 (IMJTOCR2)	MJT Output Interrupt Control Register 3 (IMJTOCR3)	5-8
H'0080 0076	MJT Output Interrupt Control Register 4 (IMJTOCR4)	MJT Output Interrupt Control Register 5 (IMJTOCR5)	5-8
H'0080 0078	MJT Output Interrupt Control Register 6 (IMJTOCR6)	MJT Output Interrupt Control Register 7 (IMJTOCR7)	5-8
H'0080 007A	(Use inhibited area)	MJT Input Interrupt Control Register 1 (IMJTICR1)	5-8
H'0080 007C	MJT Input Interrupt Control Register 2 (IMJTICR2)	MJT Input Interrupt Control Register 3 (IMJTICR3)	5-8
H'0080 007E	MJT Input Interrupt Control Register 4 (IMJTICR4)	CAN1 Transmit/Receive & Error Interrupt Control Register (ICAN1CR)	5-8

5.2.1 Interrupt Vector Register

Interrupt Vector Register (IVECT) <Address: H'0080 0000>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0-15	IVECT 16 low-order bits of ICU vector table address	When an interrupt request is accepted, the 16-low-order bits of the ICU vector table address for the accepted interrupt request source are stored in this register.	R	N

Note: • This register must always be accessed in halfwords (2 bytes). (This is a read-only register.)

The Interrupt Vector Register (IVECT) is used when an interrupt request is accepted to store the 16-low-order bits of the ICU vector table address for the accepted interrupt request source.

Before this function can work, the ICU vector table (addresses H'0000 0094 through H'0000 0113) must have set in it the start addresses of interrupt handlers for each internal peripheral I/O. When an interrupt request is accepted, the 16-low-order bits of the ICU vector table address for the accepted interrupt request source are stored in the IVECT register. In the EIT handler, read the content of this IVECT register using the LDH instruction to get the ICU vector table address.

When the IVECT register is read, operations (1) to (4) below are automatically performed in hardware.

(1) The interrupt priority level (ILEVEL) of the accepted interrupt request source is set in the IMASK register as a new IMASK value. (Interrupts with lower priority levels than that of the accepted interrupt request source are masked.)
(2) The interrupt request bit for the accepted interrupt request source is cleared (not cleared for level-recognized interrupt request sources).
(3) The interrupt request (EI) to the CPU core is deasserted.
(4) The ICU's internal sequencer is activated to start internal processing (interrupt priority resolution).

Notes: • Do not read the Interrupt Vector Register (IVECT) in the EIT handler unless interrupts are disabled (PSW register IE bit = "0"). In the EIT handler, furthermore, read the Interrupt Request Mask Register (IMASK) first before reading the IVECT register.
• To reenable interrupts (by setting the IE bit to "1") after reading the Interrupt Vector Register (IVECT), perform a dummy access to the internal memory, etc. before reenabling interrupts. (The ICU vector table readout in the EI handler processing example in Figure 5.5.2 Typical Handler Operation for Interrupts from Internal Peripheral I/O is an access to the internal ROM and, therefore, does not require adding a dummy access.)

5.2.2 Interrupt Request Mask Register

Interrupt Request Mask Register (IMASK) <Address: H'0080 0004>

b0	1	2	3	4	5	6	b7
					IMASK		
0	0	0	0	0	1	1	1

<Upon exiting reset: H'07>

b	Bit Name	Function	R	W
0–4	No function assigned. Fix to "0"		0	0
5–7	IMASK Interrupt request mask bit	000: Disable maskable interrupts 001: Accept interrupts with priority level 0 010: Accept interrupts with priority levels 0–1 011: Accept interrupts with priority levels 0–2 100: Accept interrupts with priority levels 0–3 101: Accept interrupts with priority levels 0–4 110: Accept interrupts with priority levels 0–5 111: Accept interrupts with priority levels 0–6	R	W

The Interrupt Request Mask Register (IMASK) is used to finally determine whether or not to accept an interrupt request after comparing its priority with the priority levels (Interrupt Control Register ILEVEL bits) that have been set for each interrupt request source.
When the Interrupt Vector Register (IVECT) described above is read, the interrupt priority level of the accepted interrupt request source is set in this IMASK register as a new mask value.

When any value is written to the IMASK register, operations (1) to (2) below are automatically performed in hardware.

(1) The interrupt request (EI) to the CPU core is deasserted.
(2) The ICU's internal sequencer is activated to start internal processing (interrupt priority resolution).

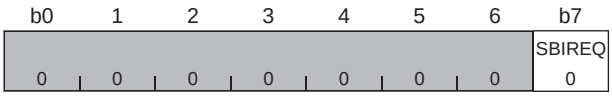
Notes: • Do not write to the Interrupt Request Mask Register (IMASK) unless interrupts are disabled (PSW register IE bit = "0").
• To reenable interrupts (by setting the IE bit to "1") after writing to the Interrupt Request Mask Register (IMASK), perform a dummy access to the internal memory, etc. before reenabling interrupts.

(1) Write to the Interrupt Request Mask Register (IMASK)
(2) Perform a dummy access to the internal memory and SFR once or more
(3) Issue one or more instructions (Note 1)
(4) Enable interrupts (by setting the IE bit to "1")

Note 1: Any instructions other than NOP that does not require clock cycles (one that is automatically inserted by the assembler for alignment adjustment: instruction code H'F000).

5.2.3 SBI (System Break Interrupt) Control Register

SBI (System Break Interrupt) Control Register (SBICR) <Address: H'0080 0006>



<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–6	No function assigned. Fix to "0"		0	0
7	SBIREQ SBI request bit	0: SBI not requested 1: SBI requested	R (Note 1)	

Note 1: This bit can only be cleared (see below)

The System Break Interrupt (SBI) is an interrupt request generated by a falling edge on the SBI# signal input pin.
When a falling edge on the SBI# signal input pin is detected and this bit is set to "1", a system break interrupt (SBI) request is generated to the CPU.
This bit cannot be set to "1" in software, it can only be cleared.
To clear this bit to "0", follow the procedure described below.

1. Write "1" to the SBI request bit.
2. Write "0" to the SBI request bit.

Notes: • Unless this bit is set to "1", do not perform the above clearing operation.
• If falling edge is inputted to SBI# pin again, system break is not occurred while SBI request bit is set to "1."

5.2.4 Interrupt Control Registers

CAN0 Transmit/Receive & Error Interrupt Control Register (ICAN0CR)	<Address: H'0080 0060>
RTD Interrupt Control Register (IRTDCR)	<Address: H'0080 0067>
SIO2,3 Transmit/Receive Interrupt Control Register (ISIO23CR)	<Address: H'0080 0068>
DMA5–9 Interrupt Control Register (IDMA59CR)	<Address: H'0080 0069>
A/D0 Conversion Interrupt Control Register (IAD0CCR)	<Address: H'0080 006C>
SIO0 Transmit Interrupt Control Register (ISIO0TXCR)	<Address: H'0080 006D>
SIO0 Receive Interrupt Control Register (ISIO0RXCR)	<Address: H'0080 006E>
SIO1 Transmit Interrupt Control Register (ISIO1TXCR)	<Address: H'0080 006F>
SIO1 Receive Interrupt Control Register (ISIO1RXCR)	<Address: H'0080 0070>
DMA0–4 Interrupt Control Register (IDMA04CR)	<Address: H'0080 0071>
MJT Output Interrupt Control Register 0 (IMJTOCR0)	<Address: H'0080 0072>
MJT Output Interrupt Control Register 1 (IMJTOCR1)	<Address: H'0080 0073>
MJT Output Interrupt Control Register 2 (IMJTOCR2)	<Address: H'0080 0074>
MJT Output Interrupt Control Register 3 (IMJTOCR3)	<Address: H'0080 0075>
MJT Output Interrupt Control Register 4 (IMJTOCR4)	<Address: H'0080 0076>
MJT Output Interrupt Control Register 5 (IMJTOCR5)	<Address: H'0080 0077>
MJT Output Interrupt Control Register 6 (IMJTOCR6)	<Address: H'0080 0078>
MJT Output Interrupt Control Register 7 (IMJTOCR7)	<Address: H'0080 0079>
MJT Input Interrupt Control Register 1 (IMJTICR1)	<Address: H'0080 007B>
MJT Input Interrupt Control Register 2 (IMJTICR2)	<Address: H'0080 007C>
MJT Input Interrupt Control Register 3 (IMJTICR3)	<Address: H'0080 007D>
MJT Input Interrupt Control Register 4 (IMJTICR4)	<Address: H'0080 007E>
CAN1 Transmit/Receive & Error Interrupt Control Register (ICAN1CR)	<Address: H'0080 007F>

b0 (b8)	1 (9)	2 (10)	3 (11)	4 (12)	5 (13)	6 (14)	b7 (b15)
			IREQ		ILEVEL		
0	0	0	0	0	1	1	1

<Upon exiting reset: H'07>

b	Bit Name	Function	R	W
0–2 (8–10)	No function assigned. Fix to "0"		0	0
3 (11)	IREQ Interrupt request bit	<Edge recognized type> At read 0: Interrupt not requested 1: Interrupt requested At write 0: Clear interrupt request 1: Generate interrupt request	R	W
		<Level-recognized type> At read 0: Interrupt not requested 1: Interrupt requested	R	0
4 (12)	No function assigned. Fix to "0"		0	0
5–7 (13–15)	ILEVEL Interrupt priority level bits	000: Interrupt priority level 0 001: Interrupt priority level 1 010: Interrupt priority level 2 011: Interrupt priority level 3 100: Interrupt priority level 4 101: Interrupt priority level 5 110: Interrupt priority level 6 111: Interrupt priority level 7 (interrupt disabled)	R	W

(1) IREQ (Interrupt Request) bit (Bit 3 or 11)

When an interrupt request from some internal peripheral I/O occurs, the corresponding IREQ (Interrupt Request) bit is set to "1."
This bit can be set and cleared in software for only edge-recognized interrupt request sources (and not for level-recognized interrupt request sources). Also, when this bit is set by an edge-recognized interrupt request generated, it is automatically cleared to "0" by reading the Interrupt Vector Register (IVECT) (not cleared in the case of level-recognized interrupt request).

If the IREQ bit is cleared in software at the same time it is set by an interrupt request generated, clearing in software has priority. Also, if the IREQ bit is cleared by reading the Interrupt Vector Register (IVECT) at the same time it is set by an interrupt request generated, clearing by a read of the IVECT register has priority.

Note: • External Interrupt (EI) to the CPU core is not deasserted by clearing the IREQ bit. External Interrupt (EI) to the CPU core can only be deasserted by the following operation:
(1) Reset
(2) IVECT register read
(3) Write to the IMASK register

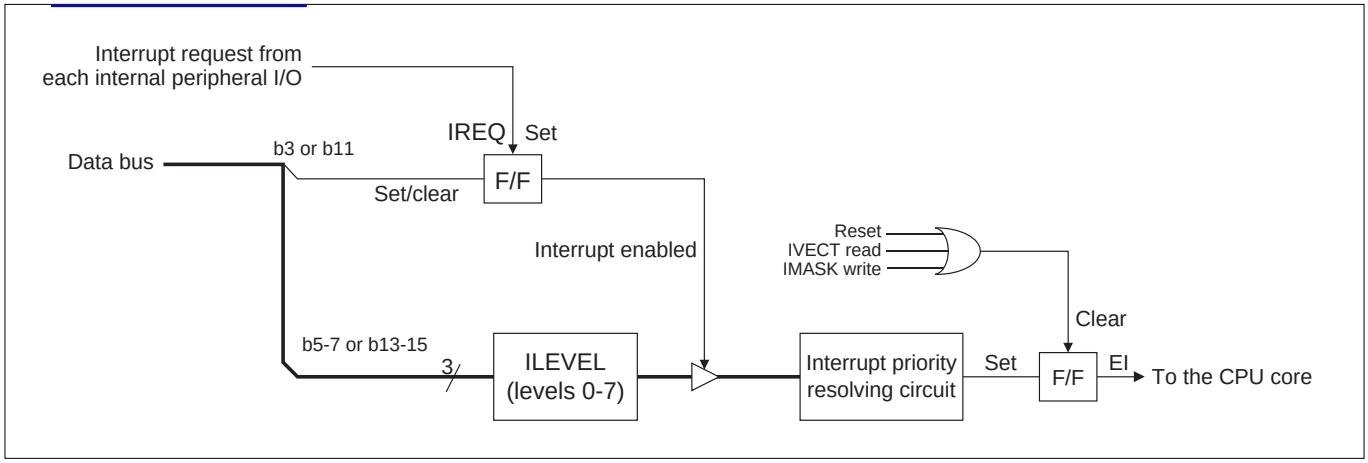


Figure 5.2.1 Configuration of the Interrupt Control Register (Edge-recognized Type)

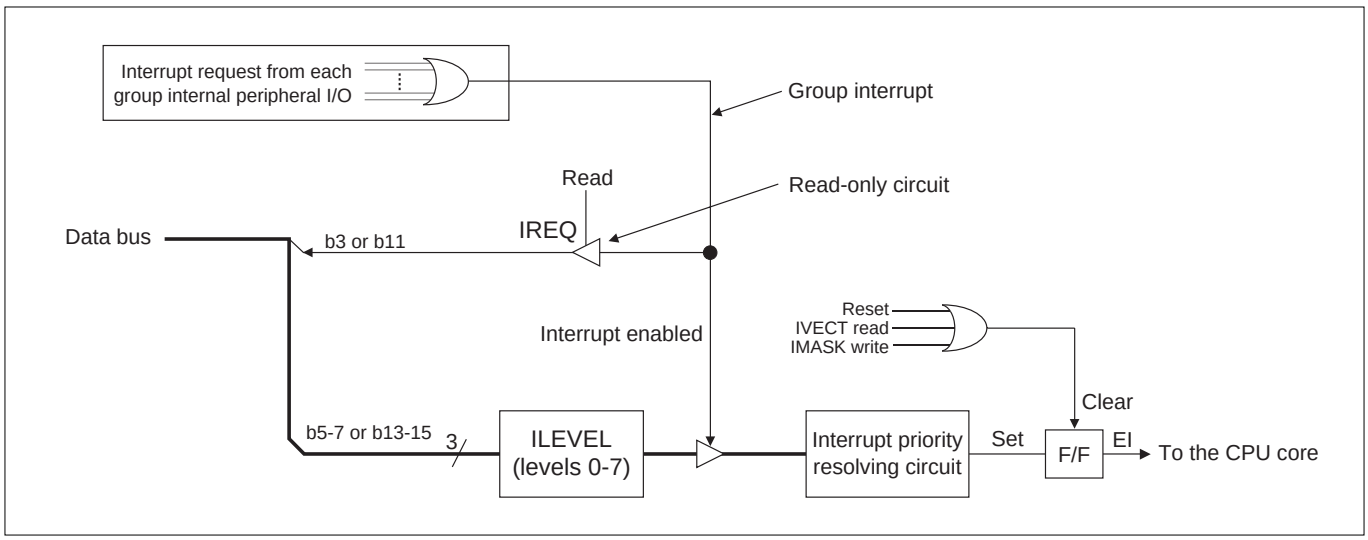


Figure 5.2.2 Configuration of the Interrupt Control Register (Level-recognized Type)

(2) ILEVEL (Interrupt Priority Level) (Bits 5–7 or bits 13–15)

These bits set the priority levels of interrupt requests from each internal peripheral I/O. Set these bits to '111' to disable or any value '000' through '110' to enable the interrupt from some internal peripheral I/O.

When an interrupt occurs, the Interrupt Controller resolves priority between this interrupt and other interrupt sources based on ILEVEL settings and finally compares priority with the IMASK value to determine whether to forward an EI request to the CPU or keep the interrupt request pending.

The table below shows the relationship between ILEVEL settings and the IMASK values at which interrupts are accepted.

Table 5.2.1 ILEVEL Settings and Accepted IMASK Values

ILEVEL values set	IMASK values at which interrupts are accepted
0 (ILEVEL = "000")	Accepted when IMASK is 1–7
1 (ILEVEL = "001")	Accepted when IMASK is 2–7
2 (ILEVEL = "010")	Accepted when IMASK is 3–7
3 (ILEVEL = "011")	Accepted when IMASK is 4–7
4 (ILEVEL = "100")	Accepted when IMASK is 5–7
5 (ILEVEL = "101")	Accepted when IMASK is 6–7
6 (ILEVEL = "110")	Accepted when IMASK is 7
7 (ILEVEL = "111")	Not accepted (interrupts disabled)

5.3 Interrupt Request Sources in Internal Peripheral I/O

The Interrupt Controller receives as inputs the interrupt requests from MJT (multijunction timer), DMAC, serial interface, A/D converter, RTD and CAN. For details about these interrupts, see each section in which the relevant internal peripheral I/O is described.

Table 5.3.1 Interrupt Request Sources in Internal Peripheral I/O

Interrupt Request Sources	Contents	Number of Input Sources	ICU Type of Input Source (Note 1)
A/D0 conversion interrupt request	A/D0 converter's scan mode one-shot operation, single mode or comparate mode completed	1	Edge-recognized
SIO0 transmit interrupt request	SIO0 transmission-completed or transmit buffer empty interrupt	1	Edge-recognized
SIO0 receive interrupt request	SIO0 reception-completed or receive error interrupt	1	Edge-recognized
SIO1 transmit interrupt request	SIO1 transmission-completed or transmit buffer empty interrupt	1	Edge-recognized
SIO1 receive interrupt request	SIO1 reception-completed or receive error interrupt	1	Edge-recognized
SIO2,3 transmit/receive interrupt request	SIO2,3 reception-completed or receive error interrupt, transmission-completed or transmit buffer empty interrupt	4	Level-recognized
RTD interrupt request	RTD interrupt generation command	1	Edge-recognized
DMA transfer interrupt request 0	DMA0–4 transfer completed	5	Level-recognized
DMA transfer interrupt request 1	DMA5–9 transfer completed	5	Level-recognized
CAN0 transmit/receive & error interrupt request	CAN0 transmission or reception completed, CAN0 errorpassive, CAN0 error bus-off, CAN0 bus error, CAN0 single shot	35	Level-recognized
CAN1 transmit/receive & error interrupt request	CAN1 transmission or reception completed, CAN1 error passive, CAN1 error bus-off, CAN1 bus error, CAN1 single shot	35	Level-recognized
MJT output interrupt request 7	MJT output interrupt rgroup 7 (TMS0, TMS1 output)	2	Level-recognized
MJT output interrupt request 6	MJT output interrupt rgroup 6 (TOP8, TOP9 output)	2	Level-recognized
MJT output interrupt request 5	MJT output interrupt rgroup 5 (TOP10 output)	1	Edge-recognized
MJT output interrupt request 4	MJT output interrupt rgroup 4 (TIO4–TIO7 outputs)	4	Level-recognized
MJT output interrupt request 3	MJT output interrupt rgroup 3 (TIO8, TIO9 outputs)	2	Level-recognized
MJT output interrupt request 2	MJT output interrupt rgroup 2 (TOP0–TOP5 outputs)	6	Level-recognized
MJT output interrupt request 1	MJT output interrupt rgroup 1 (TOP6,TOP7 outputs)	2	Level-recognized
MJT output interrupt request 0	MJT output interrupt rgroup 0 (TIO0–TIO3 outputs)	4	Level-recognized
MJT input interrupt request 4	MJT input interrupt group 4 (TIN3 input)	1	Level-recognized
MJT input interrupt request 3	MJT input interrupt group 3 (TIN20–TIN23 inputs)	4	Level-recognized
MJT input interrupt request 2	MJT input interrupt group 2 (TIN16–TIN19 inputs)	4	Level-recognized
MJT input interrupt request 1	MJT input interrupt group 1 (TIN0 input)	1	Level-recognized

Note 1: ICU type of input source

- Edge-recognized: Interrupt requests are generated on a falling edge of the interrupt signal supplied to the ICU.
- Level-recognized: Interrupt requests are generated when the interrupt signal supplied to the ICU is held low. For this type of interrupt, the ICU's Interrupt Control Register IRQ bit cannot be set or cleared in software.

5.4 ICU Vector Table

The ICU vector table is used to set the start addresses of interrupt handlers for each internal peripheral I/O. The 23-source interrupt requests are assigned the following vector table addresses.

Table 5.4.1 ICU Vector Table Addresses

Priority	Interrupt Request Source	ICU Vector Table Addresses	Number of Input Source	Input Source Type (Note 1)
High	MJT input interrupt 4 (TIN3 input)	H'0000 0094 – H'0000 0097	1	Level-recognized
↑	MJT input interrupt 3 (TIN20–23 input)	H'0000 0098 – H'0000 009B	4	Level-recognized
	MJT input interrupt 2 (TIN16–19 input)	H'0000 009C – H'0000 009F	4	Level-recognized
	MJT input interrupt 1 (TIN0 input)	H'0000 00A0 – H'0000 00A3	1	Level-recognized
	MJT output interrupt 7 (TMS0,1 output)	H'0000 00A8 – H'0000 00AB	2	Level-recognized
	MJT output interrupt 6 (TOP8,9 output)	H'0000 00AC–H'0000 00AF	2	Level-recognized
	MJT output interrupt 5 (TOP10 output)	H'0000 00B0 – H'0000 00B3	1	Edge-recognized
	MJT output interrupt 4 (TIO4–7 output)	H'0000 00B4 – H'0000 00B7	4	Level-recognized
	MJT output interrupt 3 (TIO8,9 output)	H'0000 00B8 – H'0000 00BB	2	Level-recognized
	MJT output interrupt 2 (TOP0–5 output)	H'0000 00BC – H'0000 00BF	6	Level-recognized
	MJT output interrupt 1 (TOP6,7 output)	H'0000 00C0 – H'0000 00C3	2	Level-recognized
	MJT output interrupt 0 (TIO0–3 output)	H'0000 00C4 – H'0000 00C7	4	Level-recognized
	DMA0–4 interrupt	H'0000 00C8 – H'0000 00CB	5	Level-recognized
	SIO1 receive interrupt	H'0000 00CC – H'0000 00CF	1	Edge-recognized
	SIO1 transmit interrupt	H'0000 00D0 – H'0000 00D3	1	Edge-recognized
	SIO0 receive interrupt	H'0000 00D4 – H'0000 00D7	1	Edge-recognized
	SIO0 transmit interrupt	H'0000 00D8 – H'0000 00DB	1	Edge-recognized
	A/D0 conversion interrupt	H'0000 00DC – H'0000 00DF	1	Edge-recognized
	DMA5–9 interrupt	H'0000 00E8 – H'0000 00EB	5	Level-recognized
	SIO2,3 transmit/receive interrupt	H'0000 00EC – H'0000 00EF	4	Level-recognized
	RTD interrupt	H'0000 00F0 – H'0000 00F3	1	Edge-recognized
↓	CAN0 transmit/receive & error interrupt	H'0000 010C – H'0000 010F	35	Level-recognized
Low	CAN1 transmit/receive & error interrupt	H'0000 0110 – H'0000 0113	35	Level-recognized

Note 1: ICU type of input source

- Edge-recognized: Interrupt requests are generated on a falling edge of the interrupt signal supplied to the ICU.
- Level-recognized: Interrupt requests are generated when the interrupt signal supplied to the ICU is held low. For this type of interrupt, the ICU's Interrupt Control Register IRQ bit cannot be set or cleared in software.

5.5 Description of Interrupt Operation

5.5.1 Acceptance of Internal Peripheral I/O Interrupts

An interrupt request from any internal peripheral I/O is checked to see whether or not to accept by comparing its ILEVEL value set in the Interrupt Control Register and the IMASK value of the Interrupt Request Mask Register. If its priority is higher than the IMASK value, the interrupt request is accepted. However, if two or more interrupt requests occur simultaneously, the Interrupt Controller resolves priority between these interrupt requests following the procedure described below.

1) The ILEVEL values set in the Interrupt Control Registers for the respective internal peripheral I/Os are compared with each other.
2) If the ILEVEL values are the same, priorities are resolved according to the predetermined hardware priority.
3) The ILEVEL and IMASK values are compared.

If two or more interrupt requests occur simultaneously, the Interrupt Controller first compares their priority levels set in each Interrupt Control Register's ILEVEL bit to select an interrupt request that has the highest priority. If the interrupt requests have the same ILEVEL value, their priorities are resolved according to the hardware fixed priority. The interrupt request thus selected has its ILEVEL value compared with the IMASK value and if its priority is higher than the IMASK value, the Interrupt Controller sends an EI request to the CPU.

Interrupt requests may be masked by setting the Interrupt Request Mask Register and the Interrupt Control Register's ILEVEL bit (disabled at level 7) provided for each internal peripheral I/O and the PSW register IE bit.

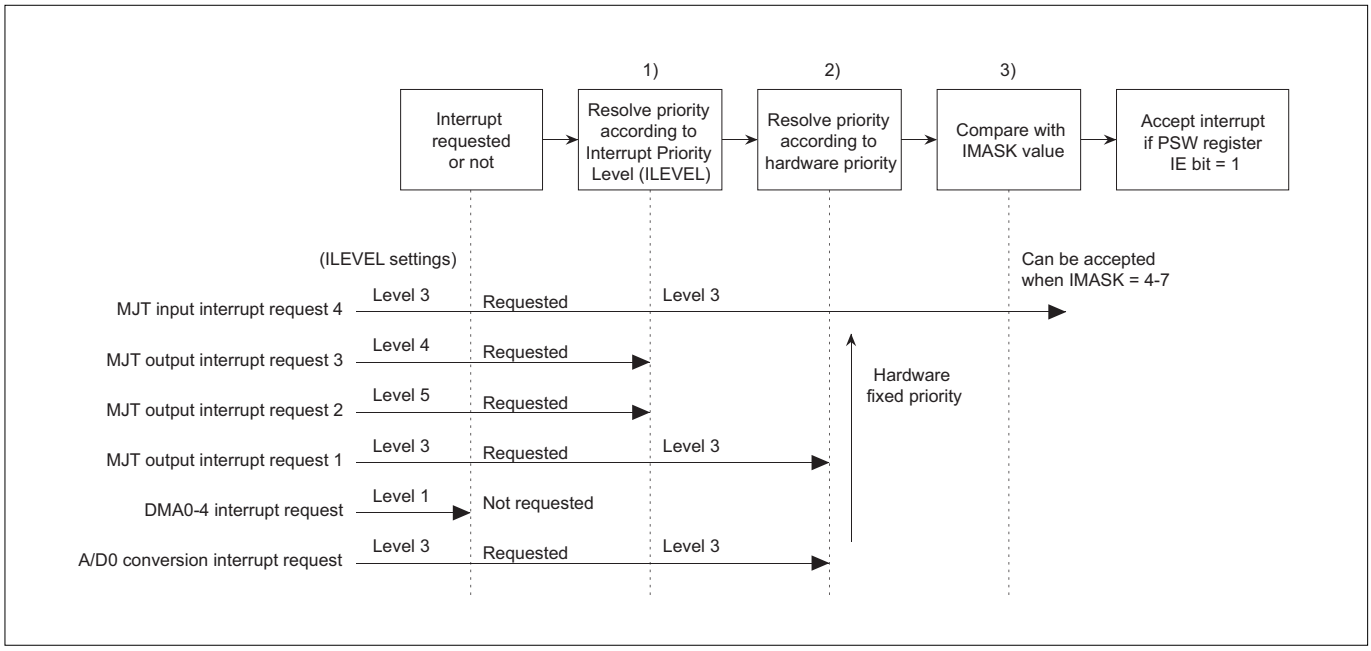


Figure 5.5.1 Example of Priority Resolution when Accepting Interrupt Requests

Table 5.5.1 ILEVEL Settings and Accepted IMASK Values

ILEVEL values set	IMASK values at which interrupts are accepted
0 (ILEVEL = "000")	Accepted when IMASK is 1–7
1 (ILEVEL = "001")	Accepted when IMASK is 2–7
2 (ILEVEL = "010")	Accepted when IMASK is 3–7
3 (ILEVEL = "011")	Accepted when IMASK is 4–7
4 (ILEVEL = "100")	Accepted when IMASK is 5–7
5 (ILEVEL = "101")	Accepted when IMASK is 6–7
6 (ILEVEL = "110")	Accepted when IMASK is 7
7 (ILEVEL = "111")	Not accepted (interrupts disabled)

5.5.2 Processing by Internal Peripheral I/O Interrupt Handlers

(1) Branching to the interrupt handler

Upon accepting an interrupt request, the CPU branches to the EIT vector entry after performing the hardware preprocessing as described in Section 4.3, “EIT Processing Procedure.” The EIT vector entry for External Interrupt (EI) is located at the address H’0000 0080. This address is where the instruction (not the jump address itself) for branching to the beginning of the interrupt handler routine for external interrupt requests is written.

(2) Processing in the External Interrupt (EI) handler

A typical operation of the External Interrupt (EI) handler (for interrupts from internal peripheral I/O) is shown in Figure 5.5.2.

[1] Saving each register to the stack

Save the BPC, PSW and general-purpose registers to the stack. Also, save the accumulator as necessary.

[2] Reading the Interrupt Request Mask Register (IMASK) and saving to the stack

Read the Interrupt Request Mask Register and save its content to the stack.

[3] Reading the Interrupt Vector Register (IVECT)

Read the Interrupt Vector Register. This register holds the 16 low-order address bits of the ICU vector table for the accepted interrupt request source that was stored in it when accepting an interrupt request. When the Interrupt Vector Register is read, the following processing is automatically performed in hardware:

- The interrupt priority level of the accepted interrupt request (ILEVEL) is set in the IMASK register as a new IMASK value. (Interrupts with lower priority levels than that of the accepted interrupt request source are masked.)
- The accepted interrupt request source is cleared (not cleared for level-recognized interrupt request sources).
- The interrupt request (EI) to the CPU core is dropped.
- The ICU’s internal sequencer is activated to start internal processing (interrupt priority resolution).

[4] Reading and overwriting the Interrupt Request Mask Register (IMASK)

Read the Interrupt Request Mask Register and overwrite it with the read value. This write to the IMASK register causes the following processing to be automatically performed in hardware:

- The interrupt request (EI) to the CPU core is dropped.
- The ICU’s internal sequencer is activated to start internal processing (interrupt priority resolution).

Note: • Processing in [4] here is unnecessary when multiple interrupts are to be enabled in [6] below.

[5] Reading the ICU vector table

Read the ICU vector table for the accepted interrupt request source. The relevant ICU vector table address can be obtained by zero-extending the content of the Interrupt Vector Register that was read in [3] (i.e., the 16 low-order address bits of the ICU vector table for the accepted interrupt request source). The ICU vector table must have set in it the start address of the interrupt handler for the interrupt request source concerned.)

[6] Enabling multiple interrupts

To enable another higher priority interrupt while processing the accepted interrupt (i.e., enabling multiple interrupts), set the PSW register IE bit to "1".

Note: • There are precautions to be take when reenabling interrupts (by setting the IE bit to "1") after writing the Interrupt Mask Register (IMASK). For details, see the Section 5.2.2, "Interrupt MASK Register."

[7] Branching to the internal peripheral I/O interrupt handler

Branch to the start address of the interrupt handler that was read out in [5].

[8] Processing in the internal peripheral I/O interrupt handler

[9] Disabling interrupts

Clear the PSW register IE bit to "0" to disable interrupts.

[10] Restoring the Interrupt Request Mask Register (IMASK)

Restore the Interrupt Request Mask Register that was saved to the stack in [2].

[11] Restoring registers from the stack

Restore the registers that were saved to the stack in [1].

[12] Completion of external interrupt processing

Execute the RTE instruction to complete the external interrupt processing. The program returns to the state in which it was before the currently processed interrupt request was accepted.

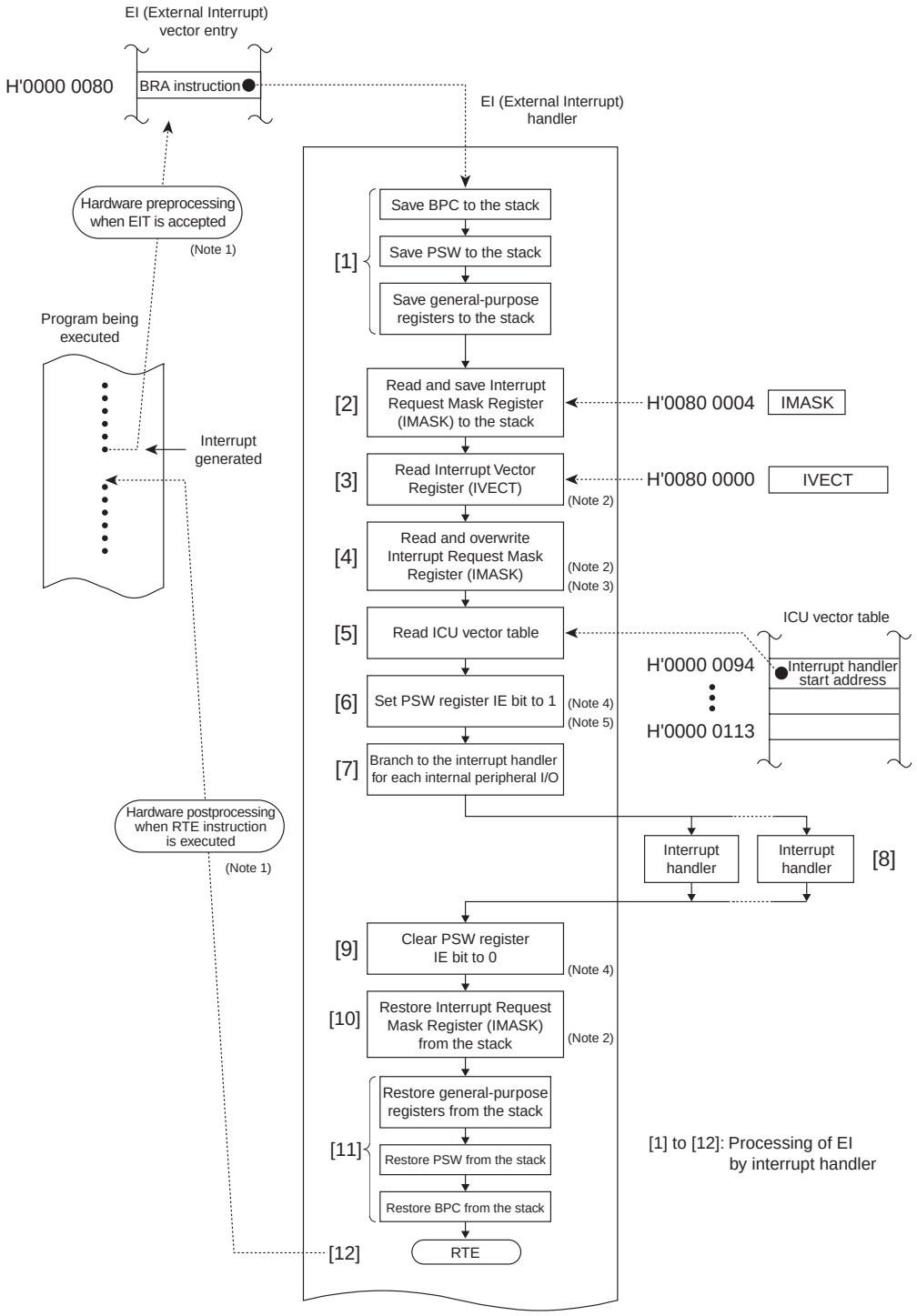
(3) Identifying the source of the interrupt request generated

If any internal peripheral I/O has two or more interrupt request sources, check the Interrupt Request Status Register provided for each internal peripheral I/O to identify the source of the interrupt request generated.

(4) Enabling multiple interrupts

To enable multiple interrupts in the interrupt handler, set the PSW register IE (Interrupt Enable) bit to enable interrupt requests to be accepted. However, before writing "1" to the IE bit, be sure to save each register (BPC, PSW, general-purpose registers and IMASK) to the stack.

Note: • Before enabling multiple interrupts, read the Interrupt Vector Register (IVECT) and then the ICU vector table, as shown in Figure 5.5.2, “Typical Handler Operation for Interrupts from Internal Peripheral I/O.”



Note 1: For operations at EIT acceptance and return from EIT, also see Section 4.3, "EIT Processing Procedure."
Note 2: Do not read the Interrupt Vector Register (IVECT) or write to the Interrupt Request Mask Register (IMASK) in the EIT handler unless interrupts are disabled (PSW register IE bit = 0).
Note 3: When multiple interrupts are disabled, execute processing in [4]. Processing in [4] is unnecessary if multiple interrupts are enabled by executing processing in [6] and [9].
Note 4: To enable multiple interrupts, execute processing in [6] and [9].
Note 5: To reenable interrupts (by setting the IE bit to 1) after reading the Interrupt Vector Register (IVECT), perform a dummy access to the internal memory, etc. before reenabling interrupts. In the example here, there is no need to add a dummy access because the ICU vector table is read after reading the IVECT register. Similarly, to reenable interrupts (by setting the IE bit to 1) after writing to the Interrupt Request Mask Register (IMASK), there are precautions to be taken when reenabling interrupts (by setting the IE bit to "1") after writing the Interrupt Mask Register (IMASK). For details, see the section 5.5.2 "Interrupt Mask Register."

Figure 5.5.2 Typical Handler Operation for Interrupts from Internal Peripheral I/O

5.6 Description of System Break Interrupt (SBI) Operation

5.6.1 Acceptance of SBI

System Break Interrupt (SBI) is an emergency interrupt which is used when power outage is detected or a fault condition is notified by an external watchdog timer. The system break interrupt is accepted anytime upon detection of a falling edge on the SBI# signal input pin no matter how the PSW register IE bit is set, and cannot be masked. If falling edge is inputted to SBI# pin again, system break is not occured while SBI request bit is set to "1."

5.6.2 SBI Processing by Handler

When the system break interrupt generated has been serviced, shut down or reset the system without returning to the program that was being executed when the interrupt occurred.

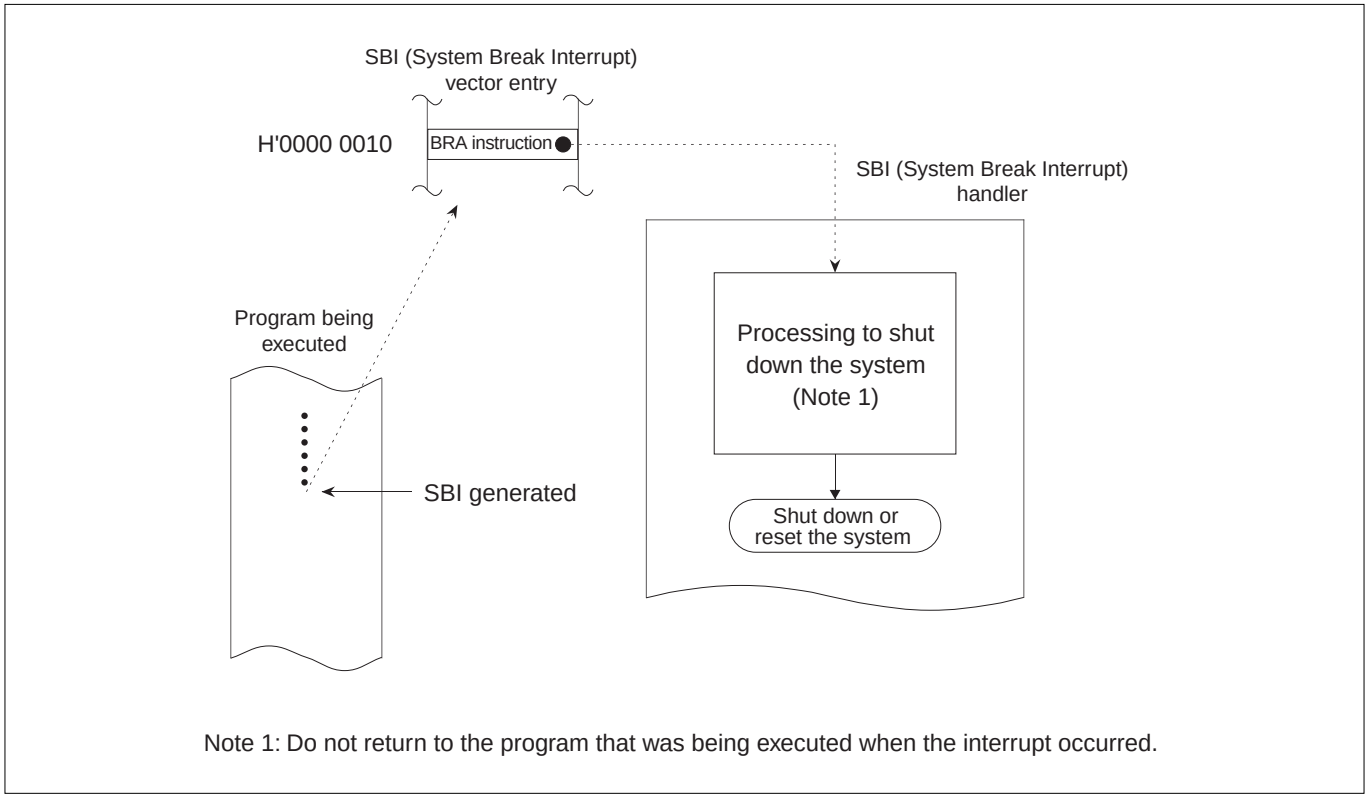


Figure 5.6.1 Typical SBI Operation

CHAPTER 6

INTERNAL MEMORY

6.1 Outline of the Internal Memory

The 32176 internally contains the following types of memory:

- 24-Kbyte RAM
- 512-Kbyte, 384-Kbyte or 256-Kbyte flash memory

6.2 Internal RAM

Specifications of the internal RAM are shown below.

Table 6.2.1 Specifications of the Internal RAM

Item	Specification
Size	24 Kbytes
Location address	H'0080 4000 to H'0080 9FFF
Wait insertion	Operates with zero wait states
Internal bus connection	Connected by 32-bit bus
Dual port	By using the Real-Time Debugger (RTD), data can be read (monitored) or written to any area of the internal RAM via serial communication from external devices independently of the CPU. (See Chapter 14, "Real-Time Debugger.")

Notes: • Immediately after power-on reset (for the power-on case in which VDDE also goes up from GND), the value of the RAM is undefined.
• If the RAM is reset during RAM backup (power for only VDDE is on), the RAM retains the value it had immediately before being reset.

6.3 Internal Flash Memory

Specifications of the internal flash memory are shown below.

Table 6.3.1 Specifications of the Internal Flash Memory

Item	Specification		
Size	M32176F4: 512 Kbytes M32176F3: 384 Kbytes M32176F2: 256 Kbytes		
Location address	M32176F4: H'0000 0000 to H'0007 FFFF M32176F3: H'0000 0000 to H'0005 FFFF M32176F2: H'0000 0000 to H'0003 FFFF		
Wait insertion	Operates with zero wait state		
Durability	Standard product	: 100 times	
	10000 (10k) times rewritable -product (Note 1)	: 4-Kbyte block (Note 2)	: 10000 (10k) times
		: Other blocks	: 1000 (1k) times
Internal bus connection	Connected by 32-bit bus		
Other	Virtual flash emulation function is incorporated. (See Section 6.6, "Virtual Flash Emulation Function.")		

Note 1: The 10000 (10k) times rewritable product is offered as an optional item. For details about it, please contact your nearest office of Renesas or its distributor.
Note 2: Block 1: H'0000 2000 to H'0000 2FFF
Block 2: H'0000 3000 to H'0000 3FFF

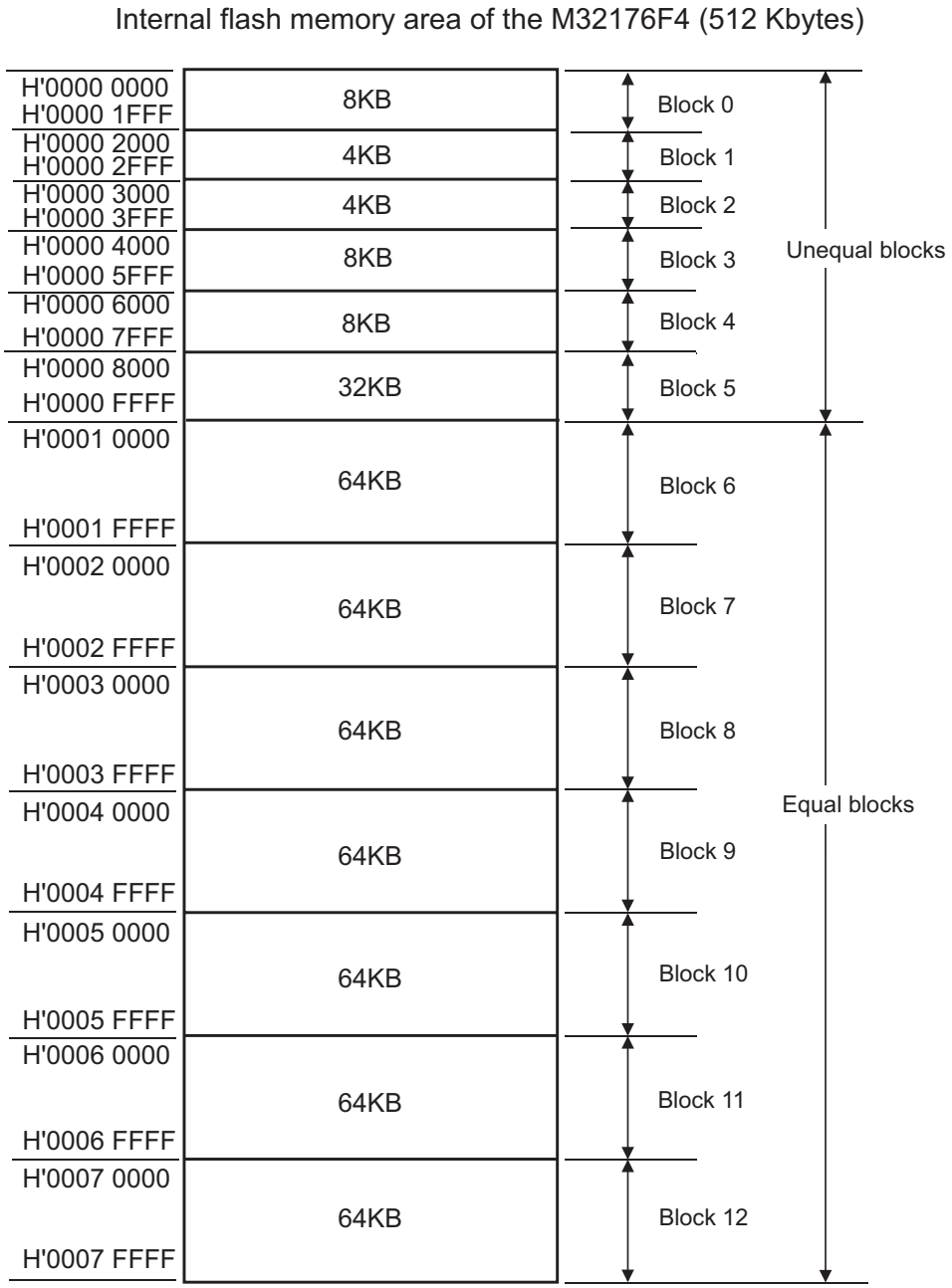


Figure 6.3.1 Block Configuration of the M32176F4's Internal Flash Memory

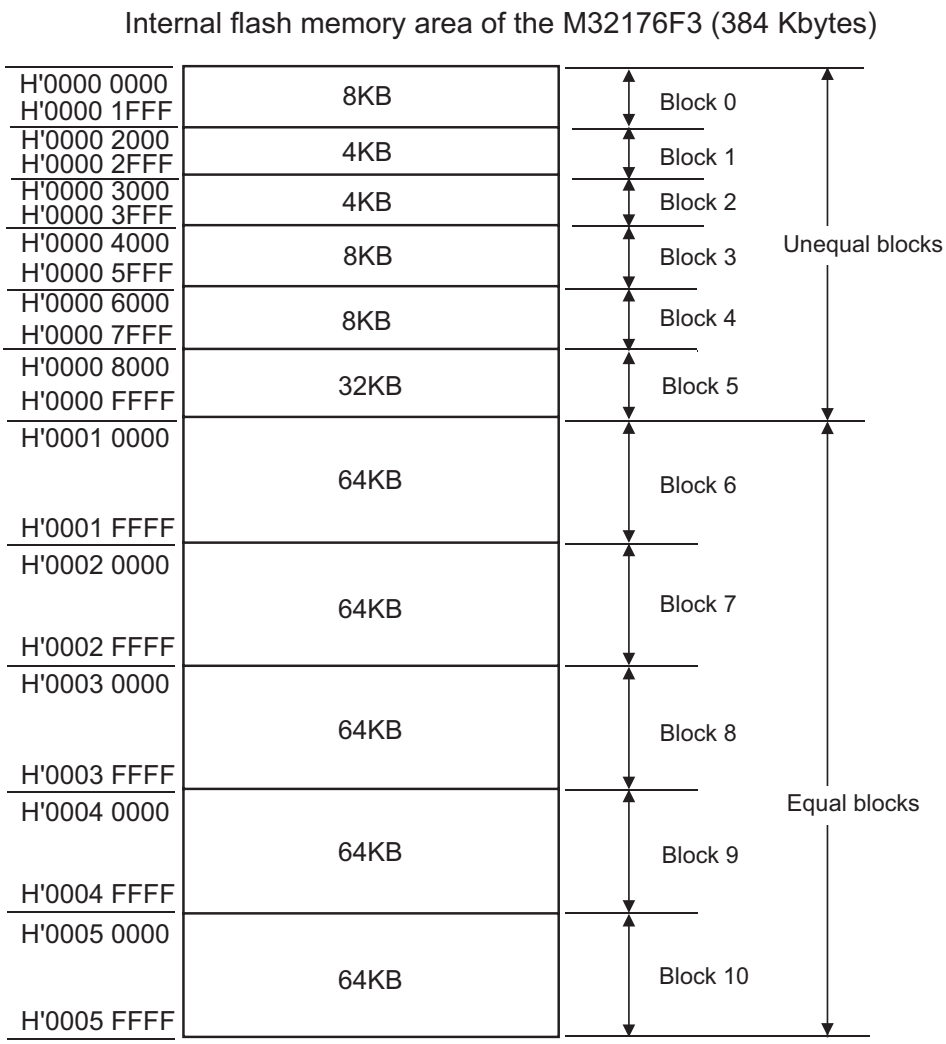


Figure 6.3.2 Block Configuration of the M32176F3's Internal Flash Memory

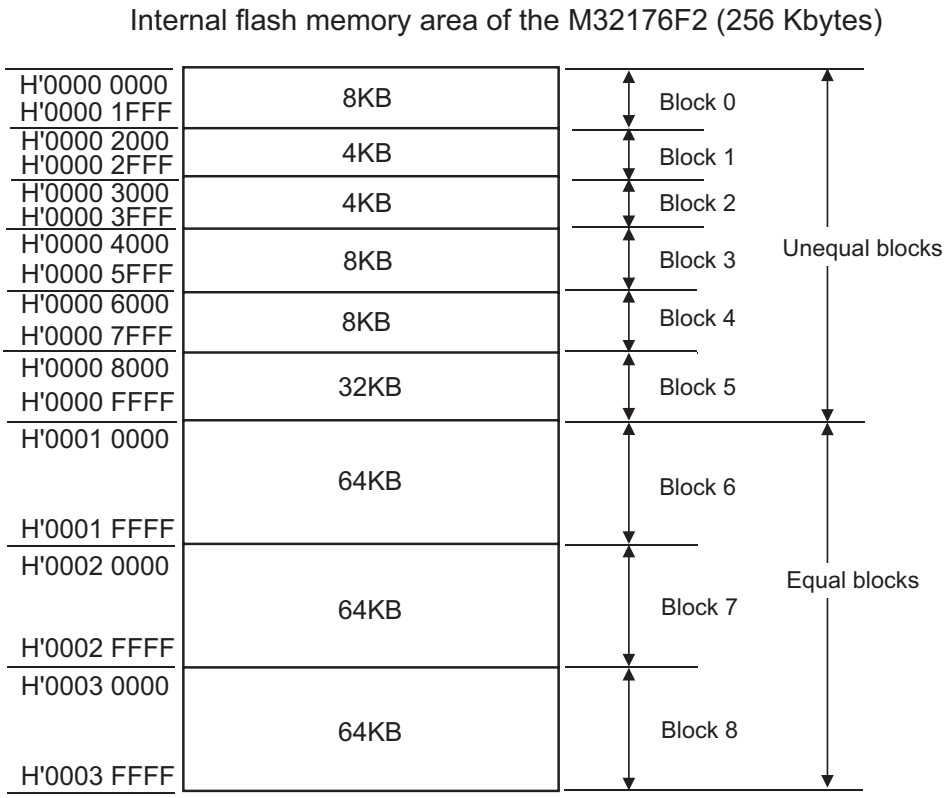


Figure 6.3.3 Block Configuration of the M32176F2's Internal Flash Memory

6.4 Registers Associated with the Internal Flash Memory

A register map associated with the internal flash memory is shown below.

Internal Flash Memory Related Register Map

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 07E0	Flash Mode Register (FMOD)	Flash Status Register (FSTAT)	6-7 6-8
H'0080 07E2	Flash Control Register 1 (FCNT1)	Flash Control Register 2 (FCNT2)	6-9 6-10
H'0080 07E4	Flash Control Register 3 (FCNT3)	Flash Control Register 4 (FCNT4)	6-11 6-13
H'0080 07E6	(Use inhibited area)		
H'0080 07E8	Virtual Flash L Bank Register 0 (FELBANK0)		6-15
H'0080 07EA	Virtual Flash L Bank Register 1 (FELBANK1)		6-15
⏐	(Use inhibited area)		
H'0080 07F0	Virtual Flash S Bank Register 0 (FESBANK0)		6-16
H'0080 07F2	Virtual Flash S Bank Register 1 (FESBANK1)		6-16

6.4.1 Flash Mode Register

Flash Mode Register (FMOD) <Address: H'0080 07E0>

b0	1	2	3	4	5	6	b7
0	0	0	FAENS 1	0	0	0	FPMOD ?

<Upon exiting reset: H'1?>

b	Bit Name	Function	R	W
0–2	No function assigned. Fix to "0"		0	0
3	FAENS Flash access enable status bit	0: Flash access disabled 1: Flash access enabled	R	–
4–6	No function assigned. Fix to "0"		0	0
7	FPMOD External FP pin status bit	0: FP pin = "L" 1: FP pin = "H"	R	–

(1) FAENS (Flash Access Enable Status) bit (Bit 3)

The FAENS bit shows whether access to the flash memory is enabled or disabled. When the flash memory is reset by the FRESET bit in Flash Control Register 4 (FCNT4) or accessed for programming/erasing, this bit is cleared to "0", resulting in the flash memory being disabled against access. When the flash memory becomes ready for access, this bit is set to "1." However, it requires up to 20µs for FAENS bit to be "1" from "0" after exiting Flash reset by FRESET bit or executing programming and erasing operation for Flash memory.

(2) FPMOD (External FP Pin Status) bit (Bit 7)

The FPMOD is a status bit which indicates the FP (Flash Protect) pin status.
The internal flash memory is enabled for programming and erasing operation only when FPMOD = "1", and is protected against programming and erasing operation when FPMOD = "0".

6.4.2 Flash Status Register

Flash Status Register (FSTAT) <Address: H'0080 07E1>

b8	9	10	11	12	13	14	b15
FBUSY		ERASE	WRERR		FESQ1	FESQ2	
1	0	0	0	0	0	0	0

<Upon exiting reset: H'80>

b	Bit Name	Function	R	W
8	FBUSY Flash busy bit	0: Being programmed or erased 1: Ready state	R	–
9	No function assigned. Fix to "0".		0	0
10	ERASE Erase status confirmation bit	0: Erase normally operating or teminated 1: Erase error occurred	R	–
11	WRERR Write status confirmation bit	0: Programming normally operating or terminated 1: Programming error occurred	R	–
12	No function assigned. Fix to "0".		0	0
13	FESQ1 Reserved bit		?	–
14	FESQ2 Reserved bit		?	–
15	No function assigned. Fix to "0".		0	0

Flash Status Register (FSTAT) consists of the following status bits that indicate the operation condition of the flash memory.

(1) FBUSY (Flash Busy) bit (Bit 8)

The FBUSY bit is used to determine whether the operation on the flash memory is finished when it is being programmed or erased. When FBUSY = "0", it means that the programming or erasing operation is being executed; when FBUSY = "1", the operation is finished.

(2) ERASE (Erase Status Confirmation) bit (Bit 10)

The ERASE bit is used to determine after execution of processing whether the erasing operation performed on the flash memory resulted in an error. When ERASE = "0", it means that the erasing operation terminated normally; when ERASE = "1", the erasing operation terminated in an error. Also, this bit is set to "1" when invalid command is issued.

(3) WRERR (Write Status Confirmation) bit (Bit 11)

The WRERR bit is used to determine after completion of processing whether the programming operation performed on the flash memory resulted in an error. When WRERR = "0", it means that the programming operation terminated normally; when WRERR = "1", the programming operation terminated in an error. Also, this bit is set to "1" when invalid command is issued.

Note: • Except when programming/erasing processing on the flash memory is forcibly terminated, do not manipulate the FRESET bit in Flash Control Register 4 (FCNT4) while the FBUSY bit = "0" (programming/erasure in progress).

6.4.3 Flash Control Registers

Flash Control Register 1 (FCNT1) <Address: H'0080 07E2>

b0	1	2	3	4	5	6	b7
0	0	0	FENTRY 0	0	0	0	FEMMOD 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–2	No function assigned. Fix to "0".		0	0
3	FENTRY Flash E/W enable mode entry bit	0: Normal read 1: Program/erase enable	R	W
4–6	No function assigned. Fix to "0".		0	0
7	FEMMOD Virtual flash emulation mode bit	0: Normal mode 1: Virtual flash emulation mode	R	W

Flash Control Register 1 (FCNT1) consists of the following two bits to control the internal flash memory.

(1) FENTRY (Flash E/W Enable Mode Entry) bit (Bit 3)

The FENTRY bit controls entry to flash E/W enable mode. Flash E/W enable mode can only be entered when FENTRY = "1".

To set the FENTRY bit to "1", write "0" and then "1" to the FENTRY bit in succession while the FP pin = "H". To clear the FENTRY bit, check to see that the Flash Status Register (FSTAT) FBUSY bit = "1" (ready), issue Read Array commands (or Flash memory reset by FRESET bit), make sure that FAENS bit = "1", and then write "0" to the FENTRY bit.

Note that the following operations cannot be performed while programming or erasing the internal flash memory (FSTAT register FBUSY bit = "0"). If one of these operations is attempted, the FENTRY bit is cleared to "0" in hardware.

1) Writing "0" to the FENTRY bit
2) Entering a "L" level signal to the FP pin
3) Entering a "L" level signal to the RESET# pin

When running a program resident in the internal flash memory while the FENTRY bit = "0", the EI vector entry is located at the address H'0000 0080 of the internal flash memory. When running the flash write/erase program in the RAM while the FENTRY bit = "1", the EI vector entry is located at the address H'0080 4000 of the RAM, allowing the flash programming/erasing operation to be controlled using interrupts.

Table 6.4.1 Changes of the EI Vector Entry by FENTRY

FENTRY	EI Vector Entry	Address
0	Internal flash memory area	H'0000 0080
1	Internal RAM area	H'0080 4000

(2) FEMMOD (Virtual Flash Emulation Mode) bit (Bit 7)

The FEMMOD bit controls entry to virtual flash emulation mode. Virtual flash emulation mode is entered by setting the FEMMOD bit to "1" while the FENTRY bit = "0". (For details, see Section 6.6, "Virtual Flash Emulation Function.")

Flash Control Register 2 (FCNT2) <Address: H'0080 07E3>

b8	9	10	11	12	13	14	b15
0	0	0	FLOCKS 0	0	0	0	FPROT 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	FLOCKS Lock bit read mode select bit	0: Memory area read mode 1: Register read mode	R (Note 1)	
12–14	No function assigned. Fix to "0".		0	0
15	FPROT Lock bit protect control bit	0: Protection by lock bit effective 1: Protection by lock bit invalidated	R (Note 1)	

Note 1: It can be accessed for write only during the Flash E/W entry mode (FENTRY bit = "1").

(1) FLOCKS (Lock Bit Read Mode Select) bit (Bit 11)

The FLOCKS bit is used to select a method for reading out the lock bit status. When the FLOCKS bit = "0", the internal flash memory is placed in memory area read mode, so that it is possible to inspect the lock bit status by issuing command data H'7171 to any address of the flash memory and then reading the last even address of the target block. When the FLOCKS bit = "1", the internal flash memory is placed in register read mode, so that it is possible to inspect the lock bit status by first issuing command data H'7171 and H'D0D0 to any address of the target block in succession and then, when the FBUSY bit is set to "1", by reading the FLOCKST bit in Flash Control Register 4.
The FLOCKS bit can only be accessed for write when the FENTRY bit = "1".

If one of the following operations is attempted, the FLOCKS bit is cleared to "0".

1) Writing "0" to the FLOCKS bit
2) Entering a "L" level signal to the FP pin
3) Clearing the FENTRY bit to "0"
4) Entering a "L" level signal to the RESET# pin

(2) FPROT (Lock Bit Protect Control) bit (Bit 15)

The FPROT bit controls invalidation of the internal flash memory protection by a lock bit (protection against programming/erasing operation). Protection of the internal flash memory is invalidated by setting the FPROT bit to "1", so that any blocks protected by a lock bit can now be programmed or erased.

To set the FPROT bit to "1", write "0" and then "1" to the FPROT bit in succession while the FENTRY bit = "1". To clear the FPROT bit to "0", write "0" to the FPROT bit.

If one of the following operations is attempted, the FPROT bit is cleared to "0".

1) Writing "0" to the FPROT bit
2) Entering a "L" level signal to the FP pin
3) Clearing the FENTRY bit to "0"
4) Entering a "L" level signal to the RESET# pin

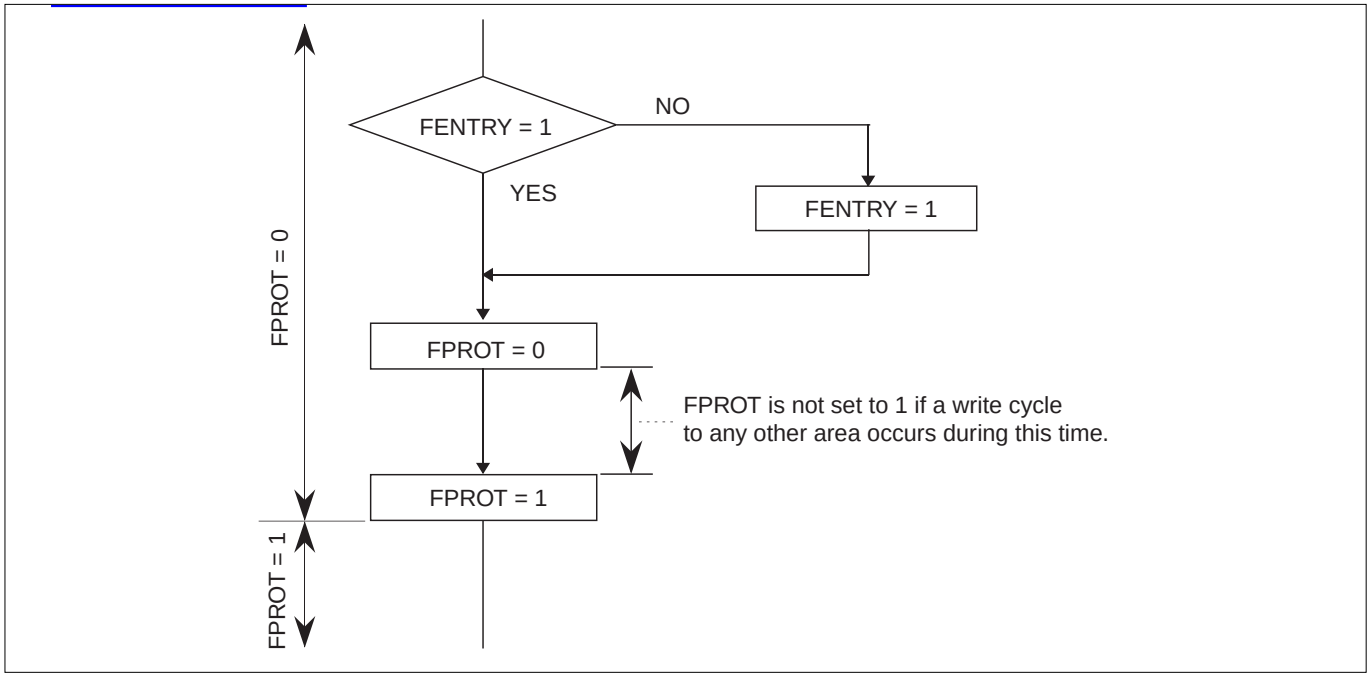


Figure 6.4.1 Protection Unlocking Flow

Flash Control Register 3 (FCNT3) <Address: H'0080 07E4>

b0	1	2	3	4	5	6	b7
0	0	0	FBSYCK 1	0	0	0	FPBSYCK 1

<Upon exiting reset: H'11>

b	Bit Name	Function	R	W
0–2	No function assigned. Fix to "0".		0	0
3	FBSYCK Busy check bit	0: Command accepted normally 1: Command not accepted normally	R	–
4–6	No function assigned. Fix to "0".		0	0
7	FPBSYCK Prebusy check bit	0: Command accepted normally 1: Command not accepted normally	R	–

Flash Control Register 3 (FCNT3) is used when developing an internal flash memory write/erase program to check whether commands have been accepted normally. This register does not need to be used for a program that has been verified to be able to operate properly.

(1) FBSYCK (Busy Check) bit (Bit 3)

The FBSYCK bit is used to check whether a 2-cycle command (confirmation command H'D0D0 or a command that requires write data) issued to the flash memory during flash E/W enable mode has been accepted normally. If the FBSYCK bit is found to be "0" after issuing a command in the second cycle (confirmation command H'D0D0 or write data), it means that the command in the second cycle has been accepted normally. Conversely, if the FBSYCK bit is found to be "1", it means that the command in the second cycle has not been accepted normally.

In addition to the above, the FBSYCK bit is set to "1" in the following cases:

1) When a command in the first cycle of 2-cycle commands has been accepted
2) When the FRESET bit = "1"
3) When input on RESET# pin is pulled "L"

(2) FPBSYCK (Prebusy Check) bit (Bit 7)

The FPBSYCK bit is used to check whether a 2-cycle command (confirmation command H'D0D0 or a command that requires write data) issued to the flash memory during flash E/W enable mode has been accepted normally. If the FPBSYCK bit is found to be "0" after issuing a command in the first cycle, it means that the command in the first cycle has been accepted normally. Conversely, if the FPBSYCK bit is found to be "1", it means that the command in the first cycle has not been accepted normally.

In addition to the above, the FPBSYCK bit is set to "1" in the following cases:

1) When in a ready state (FBUSY = "H" after a command in the second cycle has been accepted)
2) When the Clear Status Register command is issued
3) When the FRESET bit = "1"
4) When input on RESET# pin is pulled "L"

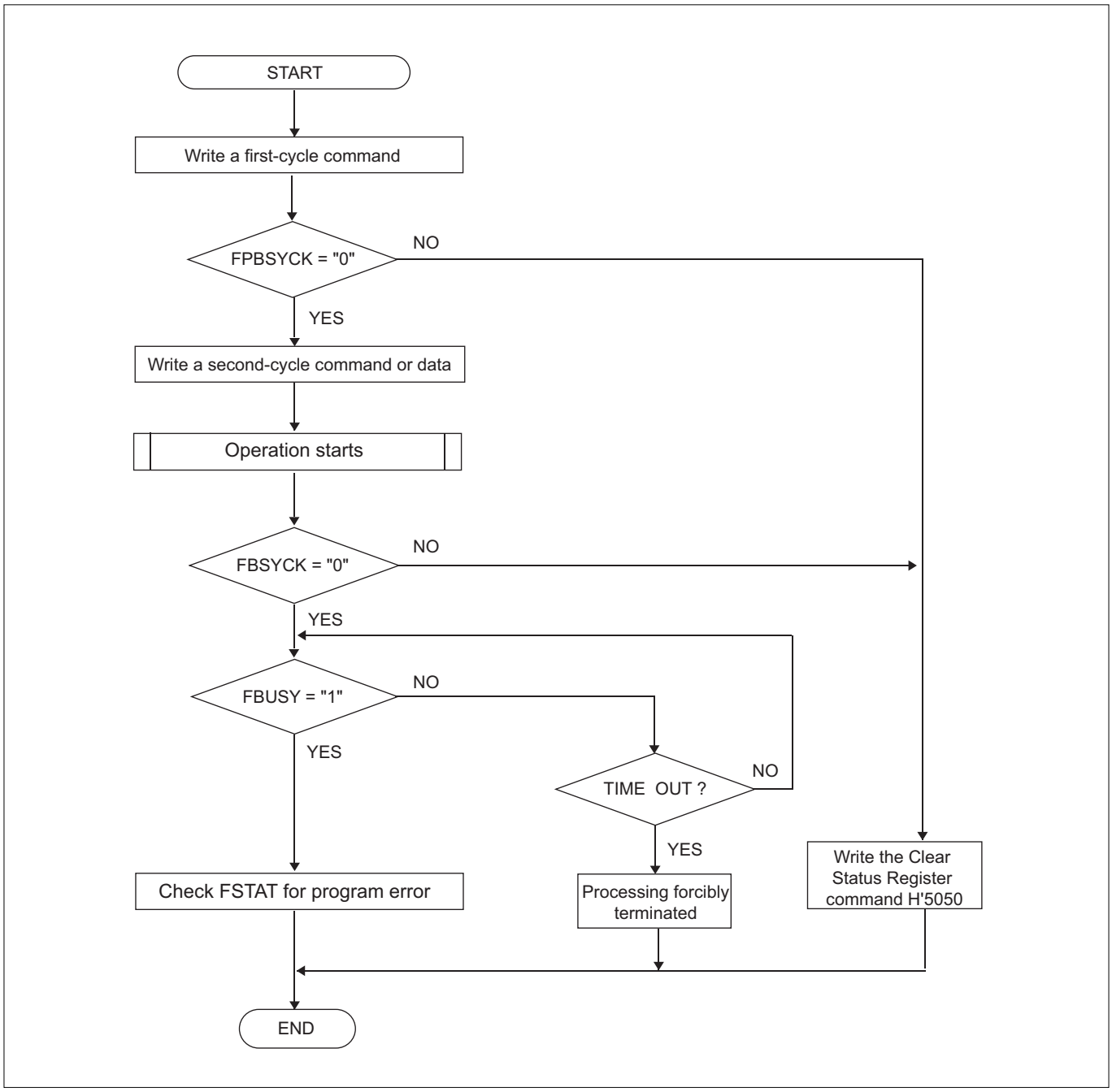


Figure 6.4.2 Method to Confirm the Command Acceptance by Checking FCNT3

Flash Control Register 4 (FCNT4) <Address: H'0080 07E5>

b8	9	10	11	12	13	14	b15
0	0	0	FLOCKST 0	0	0	0	FRESET 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	FLOCKST Lock bit status bit	0: Protected 1: Unprotected	(Note 1)	–
12–14	No function assigned. Fix to "0".		0	0
15	FRESET Flash reset bit	0: No operation 1: Reset	R	W

Note 1. Under setting FLOCKS bit of the flash control register 2 as "1" (register read mode), only the reading out value becomes effective after issuing read rock bit status command. The reading out value is undefined after issuing that read rock bit status command under setting FLOCKS bit as "0" (memory area read mode) and that other internal flash control command.

(1) FLOCKST (Lock Bit Status) bit (Bit 11)

The FLOCKST bit is used to read the lock bit status. If the FLOCKST bit = "0", it means that the relevant memory block is protected. If the FLOCKST bit = "1", it means that the relevant memory block is not protected.

Confirmation of the lock bit status by the FLOCKST bit is possible when the FLOCKS bit = "1". In this case, the lock bit status can be checked by first issuing command data H'7171 and H'D0D0 to any address of the target block in succession and then, when the FBUSY bit is set to "1", by reading the FLOCKST bit.

(2) FRESET (Flash Reset) bit (Bit 15)

The FRESET bit controls forcible termination of the internal flash memory programming/erasing operation, initialization (to H'80) of each status bit in the Flash Status Register (FSTAT), and initialization of the FPBSYCK bit in Flash Control Register 3 (FCNT3).

Setting the FRESET bit to "1" forcibly terminates programming/erasing operation and initializes each status bit in the FSTAT (to H'80) and the FPBSYCK bit in FCNT3. Make sure FRESET is held high (= "1") for at least 10 µs during a flash reset.

After a flash reset, the internal flash memory is disabled against access until the FAENS bit is set to "1".

The FRESET bit is effective only when the FENTRY bit = "1". Unless the FENTRY bit = "1", settings made to the FRESET bit are ignored. Make sure the FRESET bit = "0" while programming or erasing the flash memory.

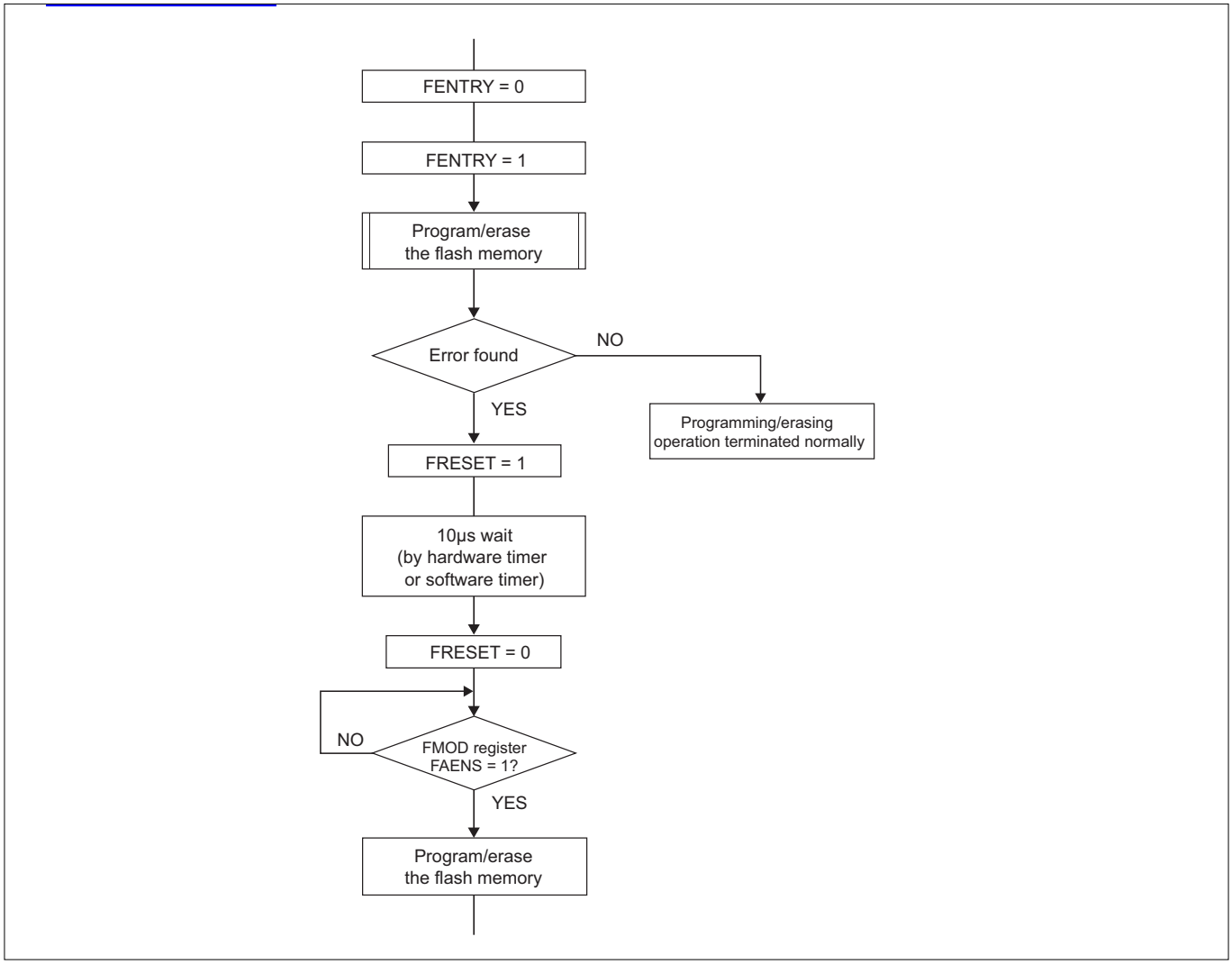


Figure 6.4.3 Example of FRESET Bit 1 (Initializing Flash Status Register 2)

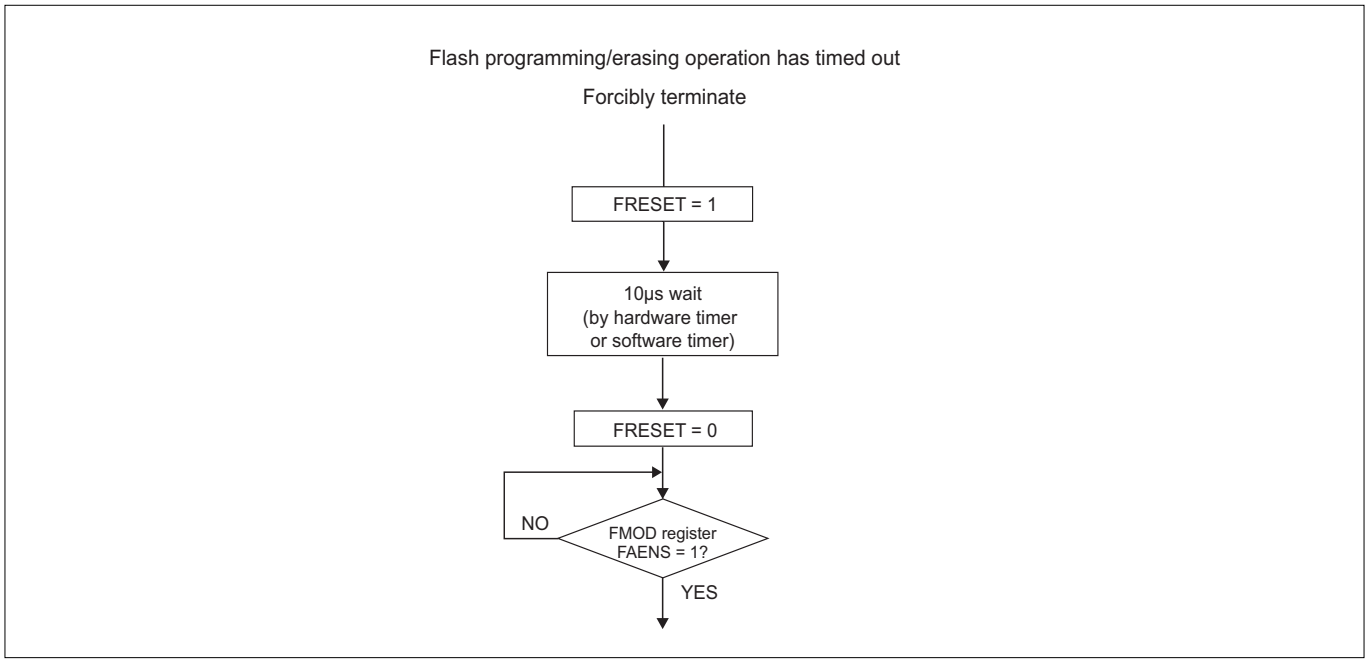
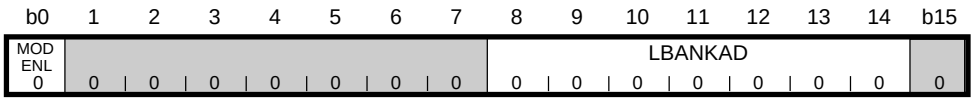


Figure 6.4.4 Example of FRESET Bit 2 (Forcibly Terminating Programming/Erasing Operation)

6.4.4 Virtual Flash L Bank Registers

Virtual Flash L Bank Register 0 (FELBANK0) <Address: H'0080 07E8>
Virtual Flash L Bank Register 1 (FELBANK1) <Address: H'0080 07EA>



<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	MODENL Virtual flash emulation L enable bit	0: Disable virtual flash emulation function 1: Enable virtual flash emulation function	R	W
1–7	No function assigned. Fix to "0".		0	0
8–14	LBANKAD L bank address bit	Start address A12–A18 of the relevant L bank	R	W
15	No function assigned. Fix to "0".		0	0

Note: • These registers must always be accessed in halfwords.

(1) MODENL (Virtual Flash Emulation L Enable) bit (Bit 0)

The MODENL bit can be set to "1" after entering virtual flash emulation mode (by setting the FEMMOD bit to "1" while the FENTRY bit = "0"). This causes the virtual flash emulation function to be enabled for the L bank area selected by the LBANKAD bits.

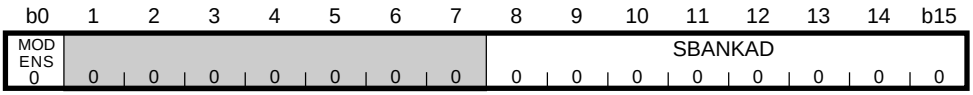
(2) LBANKAD (L Bank Address) bits (Bits 8–14)

The LBANKAD bits are provided for selecting one of the L banks that are separated every 8 Kbytes. Use these LBANKAD bits to set the seven bits A12–A18 (b8 corresponds to the address A12, and b14 corresponds to the address A18) of the 32-bit start address of the desired L bank.

Note: • For details, see Section 6.6, "Virtual Flash Emulation Function."

6.4.5 Virtual Flash S Bank Registers

Virtual Flash S Bank Register 0 (FESBANK0) <Address: H'0080 07F0>
Virtual Flash S Bank Register 1 (FESBANK1) <Address: H'0080 07F2>



<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	MODENS Virtual flash emulation S enable bit	0: Disable virtual flash emulation function 1: Enable virtual flash emulation function	R	W
1–7	No function assigned. Fix to "0".		0	0
8–15	SBANKAD S bank address bit	Start address A12–A19 of the relevant S bank	R	W

Note: • These registers must always be accessed in halfwords.

(1) MODENS (Virtual Flash Emulation S Enable) bit (Bit 0)

The MODENS bit can be set to "1" after entering virtual flash emulation mode (by setting the FEMMOD bit to "1" while the FENTRY bit = "0"). This causes the virtual flash emulation function to be enabled for the S bank area selected by the SBANKAD bits.

(2) SBANKAD (S Bank Address) bits (Bits 8–15)

The SBANKAD bits are provided for selecting one of the S banks that are separated every 4 Kbytes. Use these SBANKAD bits to set the eight bits A12–A19 (b8 corresponds to the address A12, and b15 corresponds to the address A19) of the 32-bit start address of the desired S bank.

Note: • For details, see Section 6.6, "Virtual Flash Emulation Function."

6.5 Programming the Internal Flash Memory

6.5.1 Outline of Internal Flash Memory Programming

To program or erase the internal flash memory, there are following two methods to choose depending on the situation:

(1) When the flash write/erase program does not exist in the internal flash memory
(2) When the flash write/erase program already exists in the internal flash memory

For (1), set the FP pin = "H", MOD0 = "H" and MOD1 = "L" to enter boot mode. In this case, the CPU starts running the boot program immediately after reset.
The boot program transfers the flash write/erase program into the internal RAM. After the transfer, jump to a location in the RAM and use the RAM-resident program to set the Flash Control Register 1 (FCNT1) FENTRY bit to "1" to make the internal flash memory ready for programming/erasing operation (i.e., placed in boot mode + flash E/W enable mode).
When the above is done, use the flash write/erase program that has been transferred into the internal RAM to program or erase the internal flash memory.

For (2), set the FP pin = "H", MOD0 = "L" and MOD1 = "L" to enter single-chip mode. Transfer the flash write/erase program from the internal flash memory in which it has been prepared into the internal RAM. After the transfer, jump to the RAM and use the program transferred into the RAM to set the Flash Control Register 1 (FCNT1) FENTRY bit to "1" to make the internal flash memory ready for programming/erasing operation (i.e., placed in single-chip mode + flash E/W enable mode).
When the above is done, use the flash write/erase program that has been transferred into the internal RAM to program or erase the internal flash memory. Or flash E/W enable mode can be entered from external extension mode by setting the FP pin = "H", MOD0 = "L" and MOD1 = "H."

During flash E/W enable mode (FP pin = 1, FENTRY = 1), the EIT vector entry for External Interrupt (EI) is relocated to the start address (H'0080 4000) of the internal RAM. During normal mode, it is located in the flash area (H'0000 0080).

To use an external interrupt (EI) in flash E/W enable mode, write at the beginning of the internal RAM an instruction for branching to the external interrupt (EI) handler that has been transferred into the internal RAM. Furthermore, because the IVECT register which is read out in the external interrupt (EI) handler has stored in it the flash memory address of the ICU vector table, make sure the ICU vector table to be used during flash E/W enable mode is prepared in the internal RAM so that the value of the IVECT register will be converted into the internal RAM address of the ICU vector table (for example, by adding an offset) before performing branch processing.

When started by boot mode, internal RAM value is indefinite after started by boot mode in order to "Flash writing/erasing program" is transferred to internal RAM.

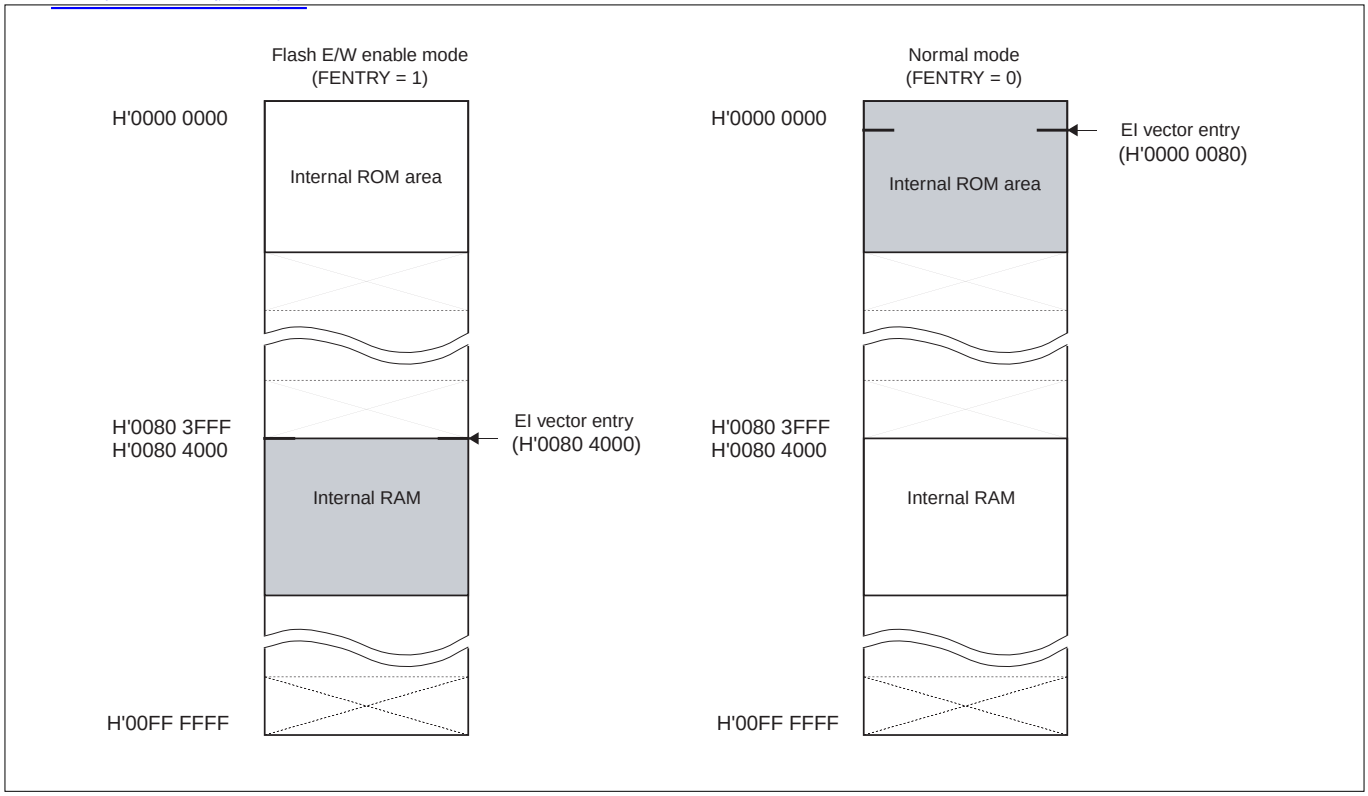
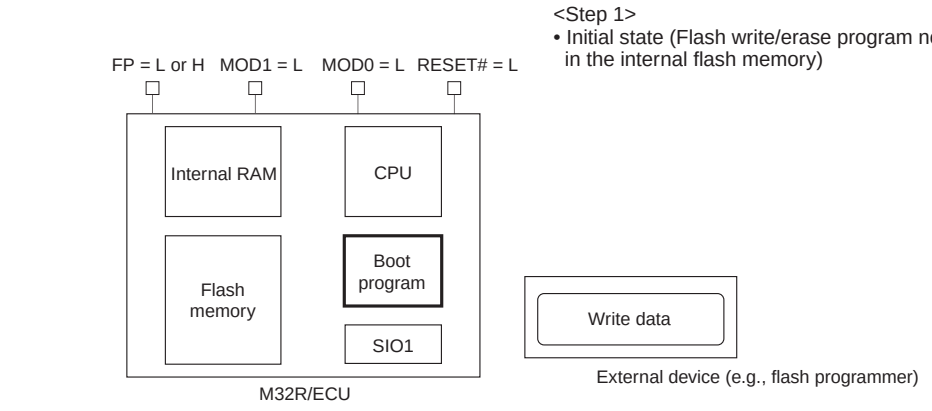
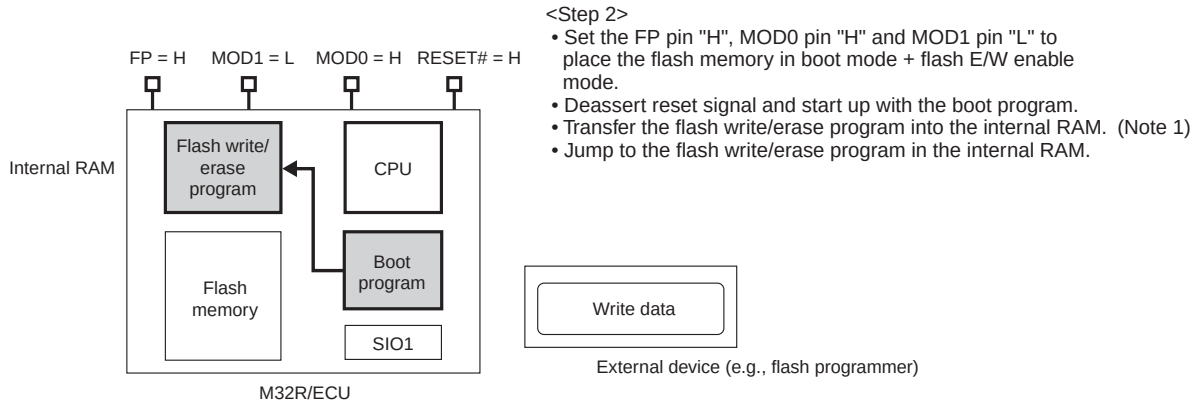


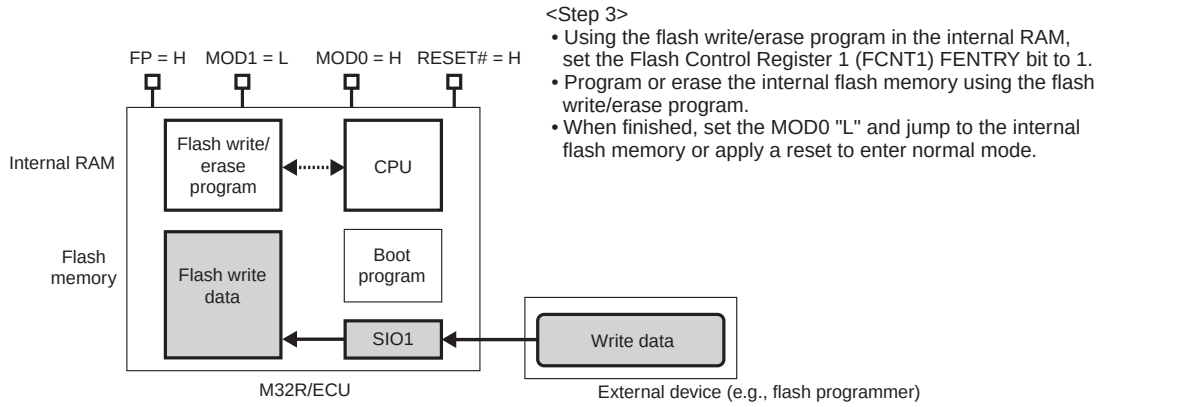
Figure 6.5.1 EI Vector Entry during Flash E/W Enable Mode

(1) When the flash write/erase program does not exist in the internal flash memory

In this case, the boot program is used to program or erase the internal flash memory. To transfer the write data, use SIO1 in clock-synchronized serial interface or clock-asynchronized serial interface mode.
To program or erase the internal flash memory using a flash programmer, follow the procedure described below.







Note 1. When started by boot mode, internal RAM value is indefinite after started by boot mode in order to "Flash writing/ erasing program" is transferred to internal RAM.

Figure 6.5.2 Procedure for Programming/Erasing the Internal Flash Memory (when the flash write/erase program does not exist in it)

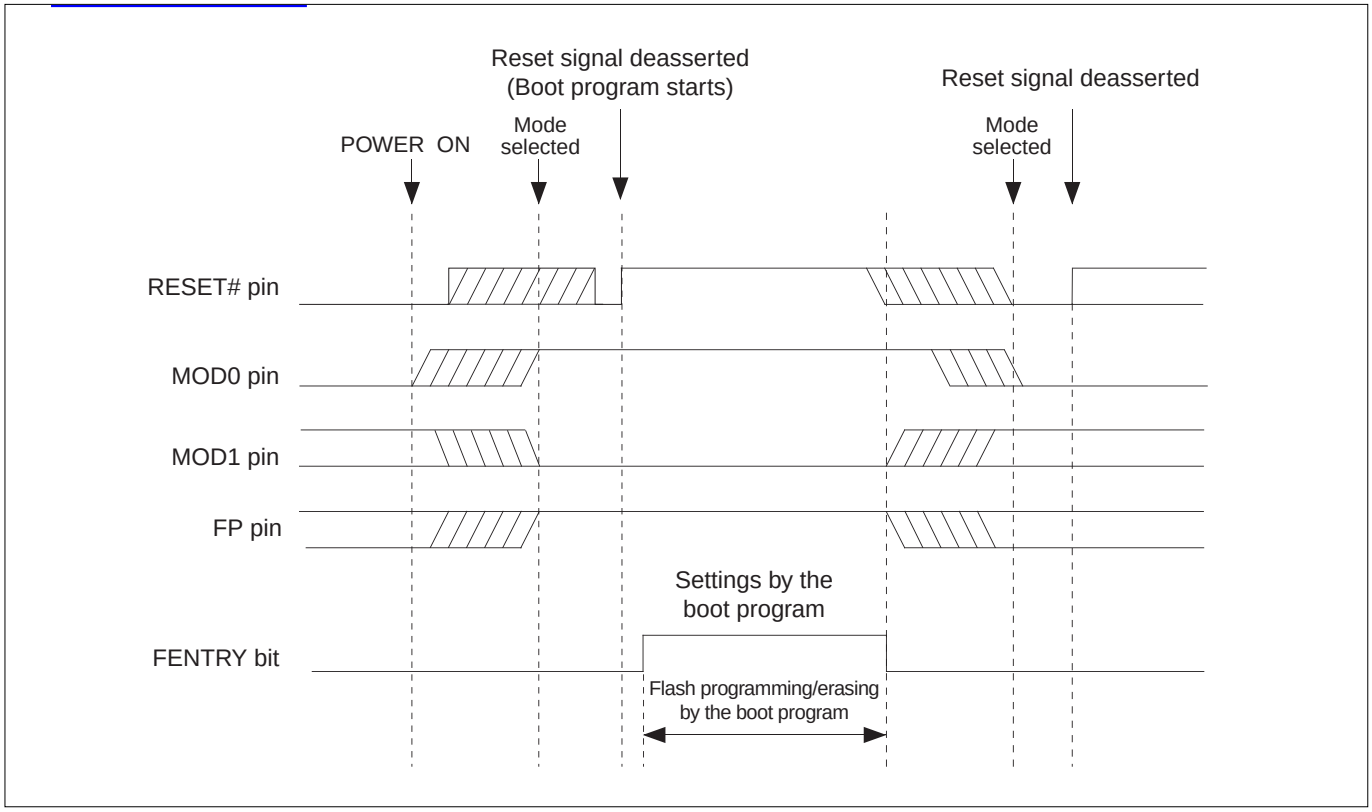


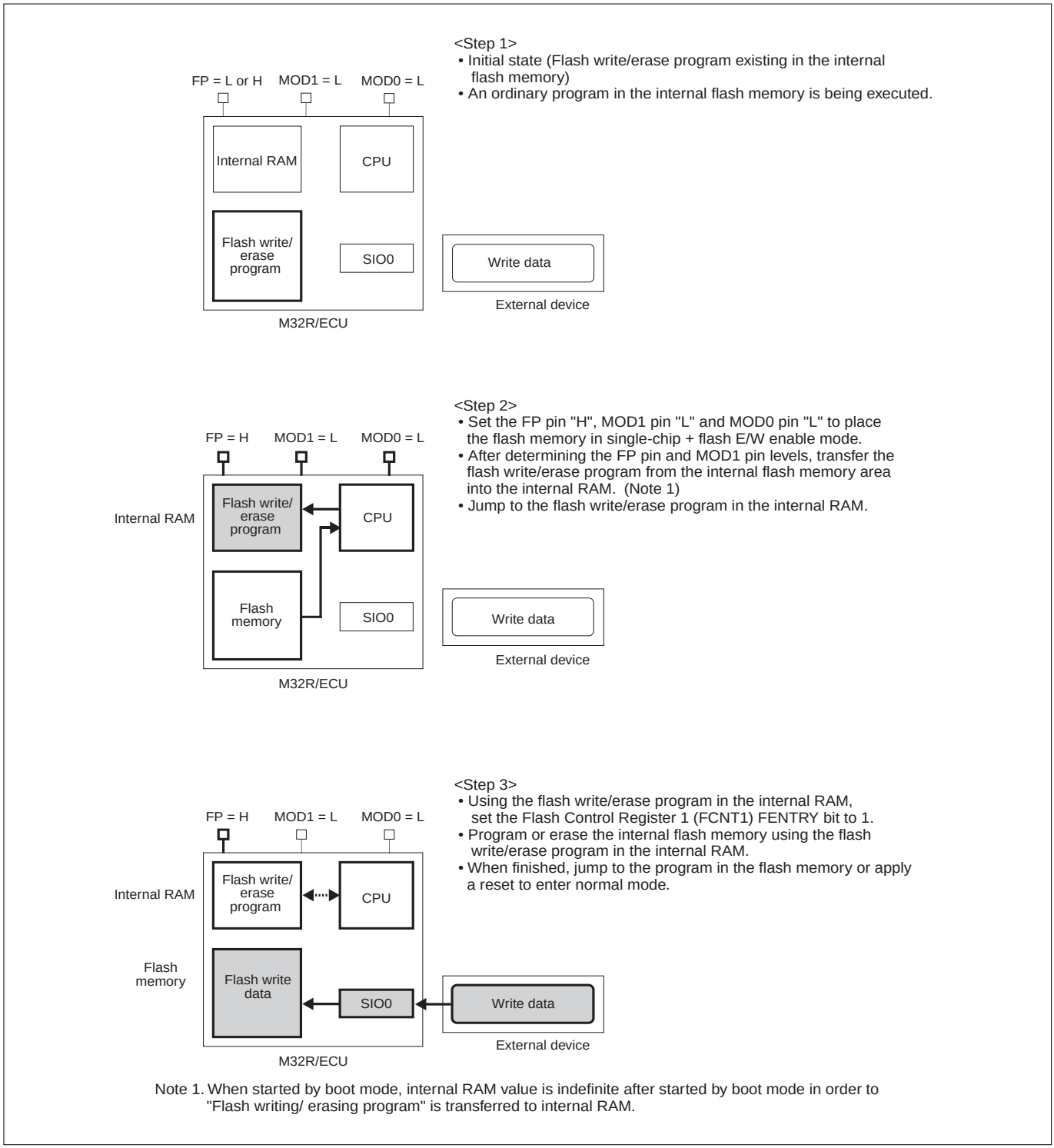
Figure 6.5.3 Internal Flash Memory Write/Erase Timing (when the flash write/erase program does not exist in it)

(2) When the flash write/erase program already exists in the internal flash memory

In this case, the flash write/erase program prepared in the internal flash memory is used to program or erase the internal flash memory.

For programming/erasing operation here, use the internal peripheral circuits in the manner suitable for the programming system. (All resources of the internal peripheral circuits such as the data bus, serial interface and ports can be used.)

The following shows an example for programming or erasing the internal flash memory by using SIO0 in single-chip mode.



Note 1. When started by boot mode, internal RAM value is indefinite after started by boot mode in order to "Flash writing/ erasing program" is transferred to internal RAM.

Figure 6.5.4 Procedure for Programming/Erasing the Internal Flash Memory (when the flash write/erase program already exists in it)

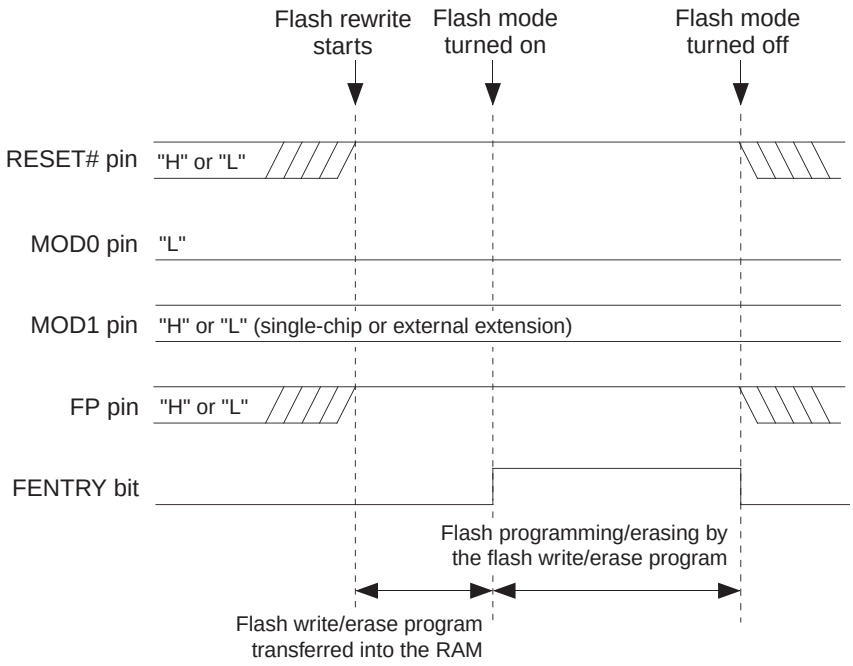


Figure 6 .5.5 Internal Flash Memory Write/Erase Timing (when the flash write/erase program already exists in it)

6.5.2 Controlling Operation Modes during Flash Programming

The microcomputer's operation mode is set by MOD0, MOD1 and Flash Control Register 1 (FCNT1) FENTRY bit. The table below lists operation modes that may be used when programming or erasing the internal flash memory.

Table 6.5.1 Operation Modes Set during Flash Programming/erasing

FP	MOD0	MOD1	FENTRY (Note 1)	Operation Mode	Reset Vector Entry	EI Vector Entry
0	0	0	0	Single-chip mode	Start address of internal flash memory (H'0000 0000)	Flash area (H'0000 0080)
1	0	0	0			
0	1	0	0	Processor mode	Start address of external area (H'0000 0000)	External area (H'0000 0080)
0	0	1	0	External extension mode	Start address of internal flash memory (H'0000 0000)	Flash area (H'0000 0080)
1	0	1	0			
1	0	0	1	Single-chip mode + flash E/W enable	Start address of internal flash memory (H'0000 0000)	Beginning of internal RAM (H'0080 4000)
1	1	0	0	Boot mode	Boot program startup address	Flash area (H'0000 0080)
1	1	0	1	Boot mode + flash E/W enable	Boot program startup address	Beginning of internal RAM (H'0080 4000)
1	0	1	1	External extension mode + flash E/W enable	Start address of internal flash memory (H'0000 0000)	Beginning of internal RAM (H'0080 4000)
–	1	1	–	Use inhibited	–	–

Note 1: Indicates the Flash Control Register 1 (FCNT1) FENTRY bit status (– denotes "Don't care"). However, if FP = "0", writing "1" to FENTRY only results in it cleared to "0".

Note 2: Always make sure the MOD2 pin is connected low (= 0) to ground (GND).

(1) Flash E/W enable mode

Flash E/W enable mode is a mode in which the internal flash memory can be programmed or erased. In flash E/W enable mode, no programs can be executed in the internal flash memory. Therefore, the necessary program must be transferred into the internal RAM before entering flash E/W enable mode, so that it can be executed in the internal RAM.

(2) Entering flash E/W enable mode

Flash E/W enable mode can only be entered when operating in single-chip, external extension or boot mode. Furthermore, it is only when the FP pin = "H" and the Flash Control Register 1 (FCNT1) FENTRY bit = "1" that flash E/W enable mode can be entered. Flash E/W enable mode cannot be entered when operating in processor mode or the FP pin = "L."

(3) Detecting the MOD0 and MOD1 pin levels

The MOD0 and MOD1 pin levels ("H" or "L") can be known by checking the P8 Data Register (Port Data Register, H'0080 0708) MOD0DT and MOD1DT bits.

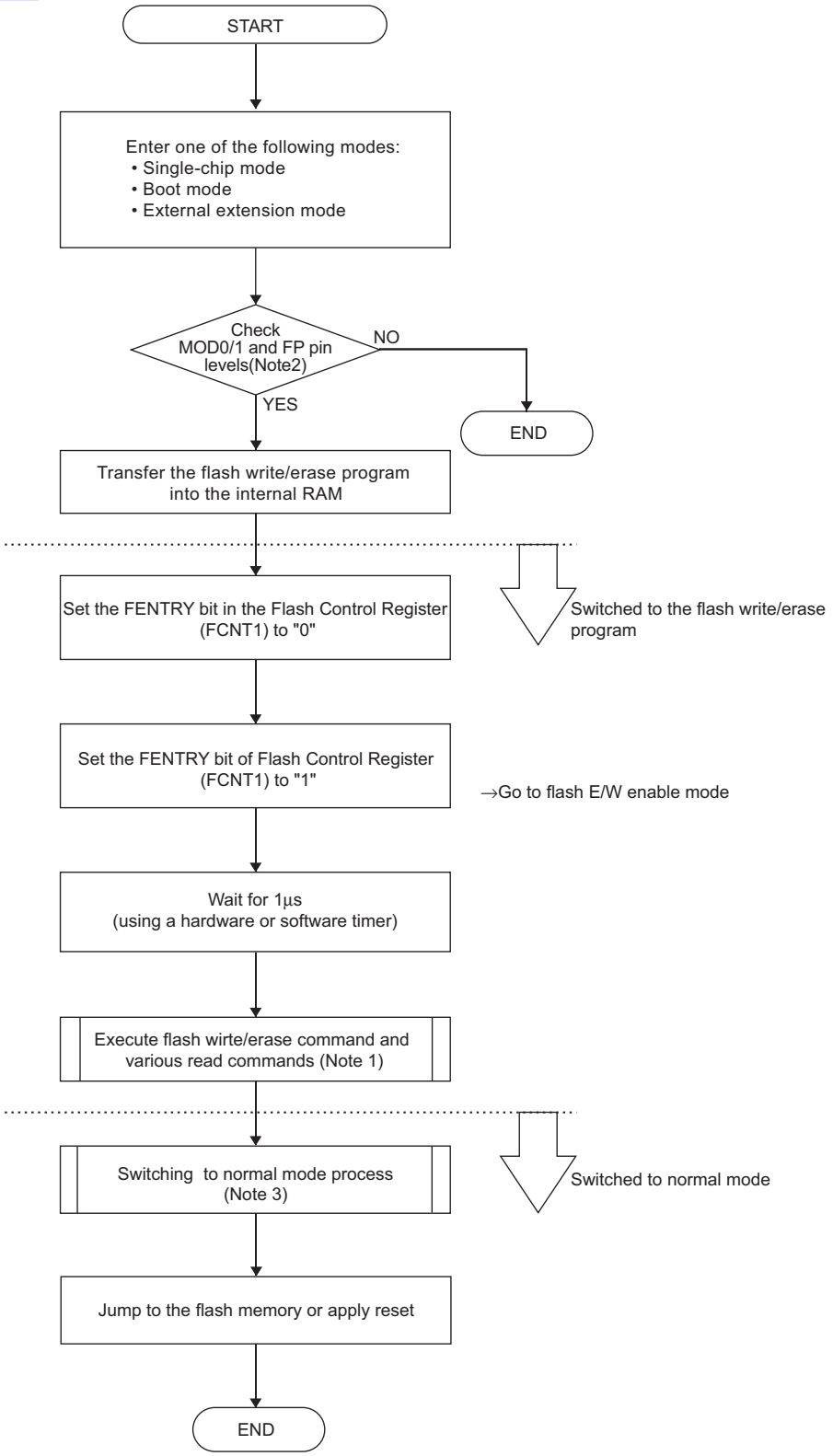
P8 Data Register (P8DATA) <Address: H'0080 0708>

b0	1	2	3	4	5	6	b7
MOD0DT	MOD1DT	P82DT	P83DT	P84DT	P85DT	P86DT	P87DT
?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

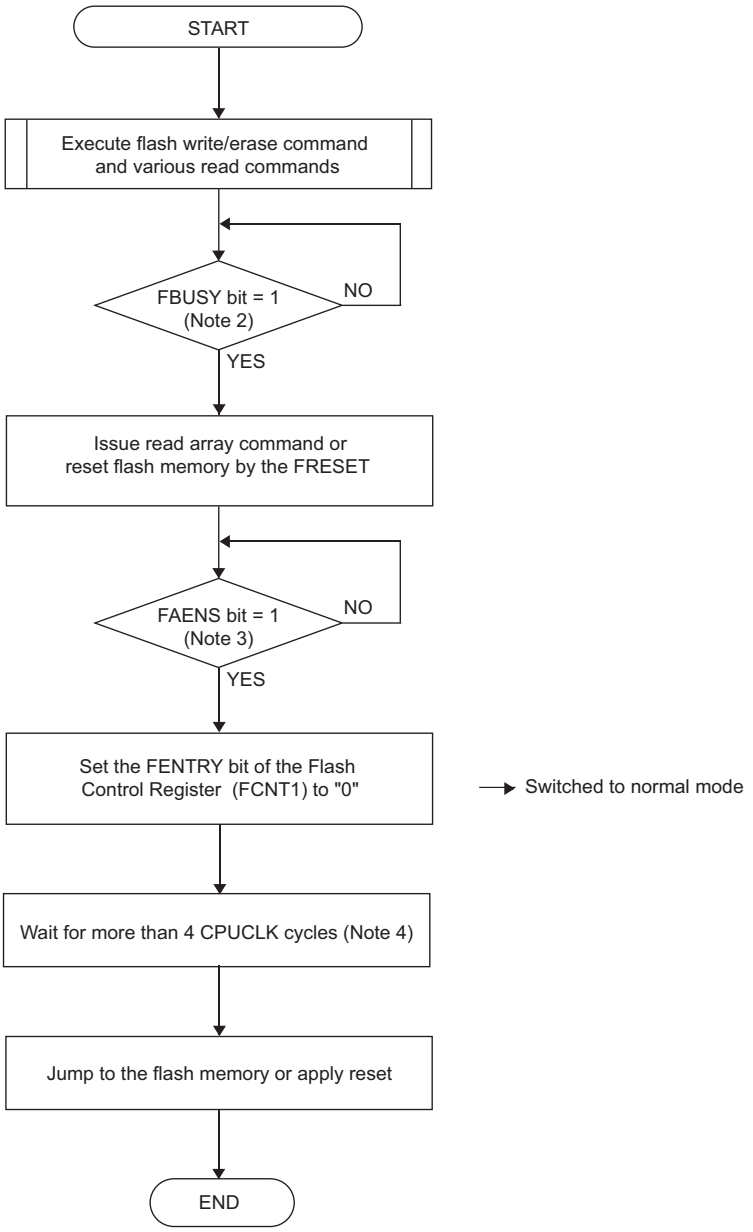
b	Bit Name	Function	R	W
0	MOD0DT MOD0 data bit	0: MOD0 pin = "L" 1: MOD0 pin = "H"	R	–
1	MOD1DT MOD1 data bit	0: MOD1 pin = "L" 1: MOD1 pin = "H"	R	–
2	P82DT Port P82 data bit	At read Depends on how the Port Direction Register is set • If direction bit = "0" (input mode) 0: Port input pin = "L" 1: Port input pin = "H" • If direction bit = "1" (output mode) (Note 1) 0: Port output latch = "0" / Port pin level = "L" 1: Port output latch = "1" / Port pin level = "H" At write Write to the port output latch	R	W
3	P83DT Port P83 data bit			
4	P84DT Port P84 data bit			
5	P85DT Port P85 data bit			
6	P86DT Port P86 data bit			
7	P87DT Port P87 data bit			

Note 1: To select the port data to read, use the Port Input Special Function Control Register's port input data select bit (PISEL).



Note 1: For details about each command, see Section 6.6.3, "Procedure for Programming/Erasing the Internal Flash Memory."
Note 2: Checked by FPMOD bit in FMOD register and MOD0DT and MOD1DT bits in P8DATA register
Note 3: For details about the procedure for switching process to normal mode, see Figure 6.5.7 " Procedure for switching to nomal mode."

Figure 6.5.6 Procedure for Entering Flash E/W Enable Mode



Note1: For details about each command, see Section 6.6.3, "Procedure for Programming/Erasing the Internal Flash Memory."
Note2: If it is checked that the value of FBUSY bit in Flash Status Register (FSTAT) is "1" after executing the command in flash E/W enable mode, it is not necessary to check that the value of FBUSY bit is "1".
Note3: If flash memory reset by FRESET bit in Flash Control Register 4 (FCNT4) is not executed, it is not necessary to check that the value of FAENS bit in Flash Mode Register (FMOD) is "1".
Note4: Insert any instructions for more than 4 CPUCLK waits other than NOP that do not require clock cycles (one that is automatically inserted by the assembler for alignment adjustment: instruction code H'F000).
Note: · When switching to normal mode by entering a "L" level signal to the RESET# pin in flash E/W enable mode, enter the signal to the RESET# pin after checking that the value of FBUSY bit is "1"(ready).

Figure 6.5.7 Procedure for switching to normal mode

6.5.3 Procedure for Programming/Erasing the Internal Flash Memory

To program or erase the internal flash memory, set up chip mode to enter flash E/W enable mode and execute the flash write/erase program in the internal RAM into which it has been transferred from the internal flash memory.
In flash E/W enable mode, because the internal flash memory cannot be accessed for read as in normal mode, no programs present in it can be executed. Therefore, the flash write/erase program must be made available in the internal RAM before entering flash E/W enable mode. Once flash E/W enable mode is entered into, only flash command and no other commands can be used to access the internal flash memory.
To access the internal flash memory in flash E/W enable mode, issue commands for the internal flash memory address to be operated on. The table below lists the commands that can be issued in flash E/W enable mode.

Note: • During flash E/W enable mode, the internal flash memory cannot be accessed for read or write wordwise.

Table 6.5.2 Commands in Flash E/W Enable Mode

Command Name	Issued Command Data
Read Array command	H'FFFF
Halfword Program command	H'4040
Lock Bit Program command	H'7777
Block Erase command	H'2020
Clear Status Register command	H'5050
Read Lock Bit Status command	H'7171
Verify command (Note 1)	H'D0D0

Note 1: • This command must be issued immediately after the Lock Bit Program, Block Erase or Read Lock Bit Status command. If the Lock Bit Program, Block Erase or Read Lock Bit Status command is followed by other than the Verify (H'D0D0) command, the Lock Bit Program, Block Erase or Read Lock Bit Status command is not executed normally and terminated in error.

(1) Read Array command

Writing the Read Array command (H'FFFF) to any address of the internal flash memory places it in read mode. Then read the desired flash memory address, and the content of that address will be read out. Before exiting flash E/W enable mode, always be sure to execute the Read Array command.

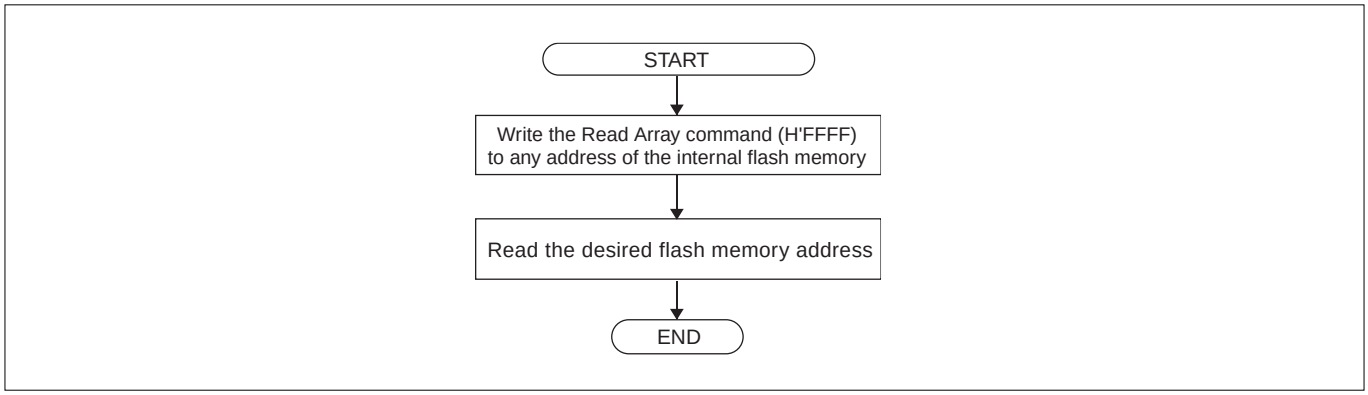


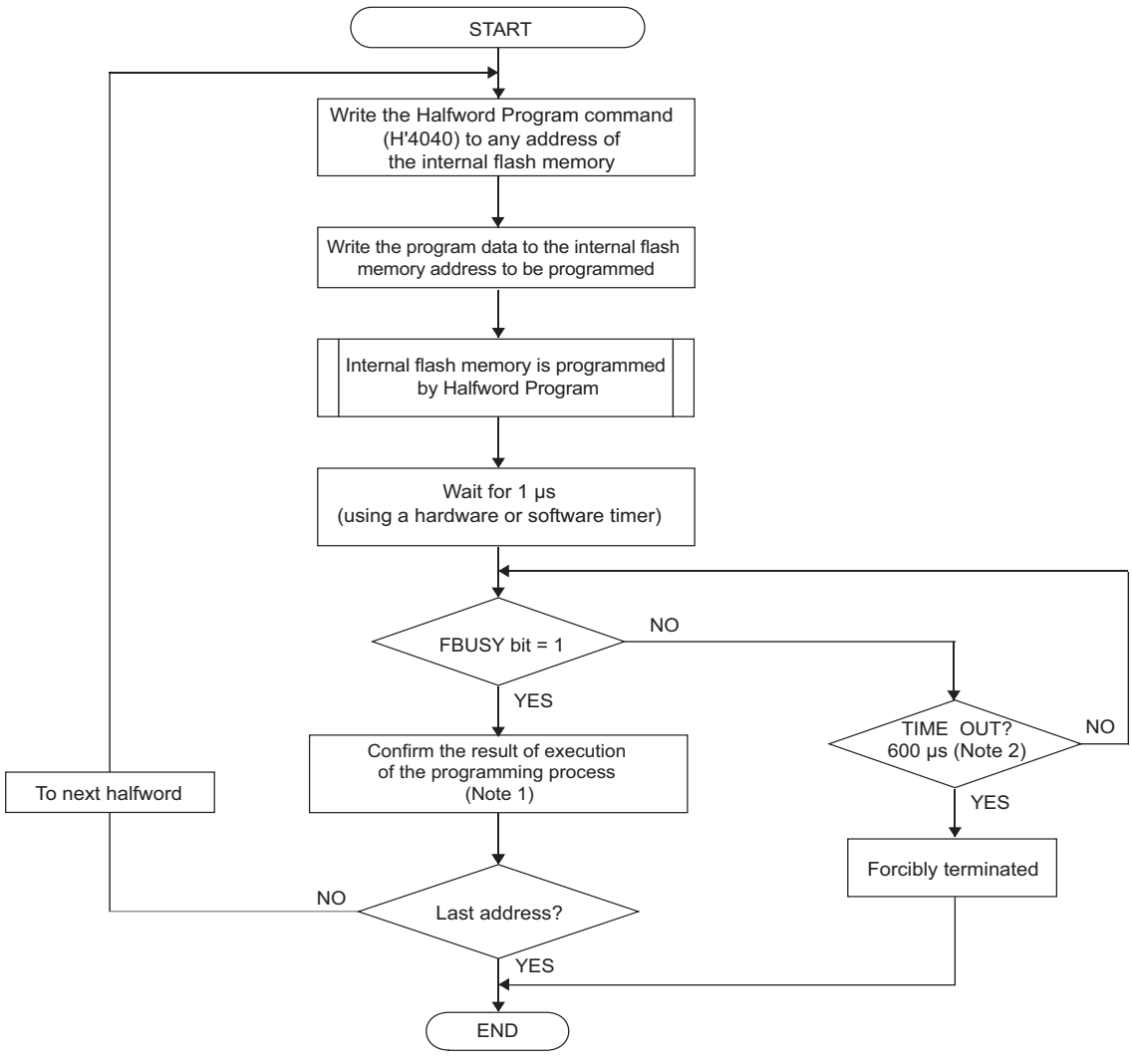
Figure 6.5.8 Read Array

(2) Halfword Program command

The internal flash memory is programmed a halfword at a time, each halfword consisting of 2 bytes. To program the flash memory, write the Program command (H'4040) to any address of the internal flash memory and then the program data to the address to be programmed.

The protected flash memory blocks cannot be accessed for write by the Halfword Program command.

Halfword programming is automatically performed by the internal control circuit, and whether the Halfword Program command has finished can be known by checking the Flash Status Register FBUSY bit. While the FBUSY bit = "0", the next programming cannot be performed.



Note 1: Check Flash Status Register (FSTAT) ERASE bit (for the erase status) and WRERR bit (for the write status).
Note 2: It is the TIME OUT period of 4K-byte block, which is 10k times rewritable product. The TIME OUT period of Other blocks and standard products (Flash rewrite durability: 100 times) is 200µs.

Figure 6.5.9 Halfword Program

(3) Lock Bit Program command

The internal flash memory can be protected against programming/erasing operation one block at a time. The Lock Bit Program command is provided for protecting the flash memory blocks.

Write the Lock Bit Program command (H'7777) to any address of the internal flash memory. Next, write the Verify command (H'D0D0) to the last even address of the flash memory block to be protected, and this memory block is thereby protected against programming/erasing operation. To remove protection, use the Flash Control Register 2 (FCNT2) FPROT bit to invalidate protection by a lock bit and erase the flash memory block whose protection is to be removed. (The content of that memory block is also erased.)

Lock bit programming is automatically performed by the internal control circuit, and whether the Lock Bit Program command has finished can be known by checking the Flash Status Register (FSTAT) FBUSY bit. While the FBUSY bit = "0", the next programming cannot be performed.
The table below lists the target flash memory blocks and their addresses to be specified when writing the Verify command data.

Table 6.5.3 M32176F4 Target Blocks and Specified Addresses

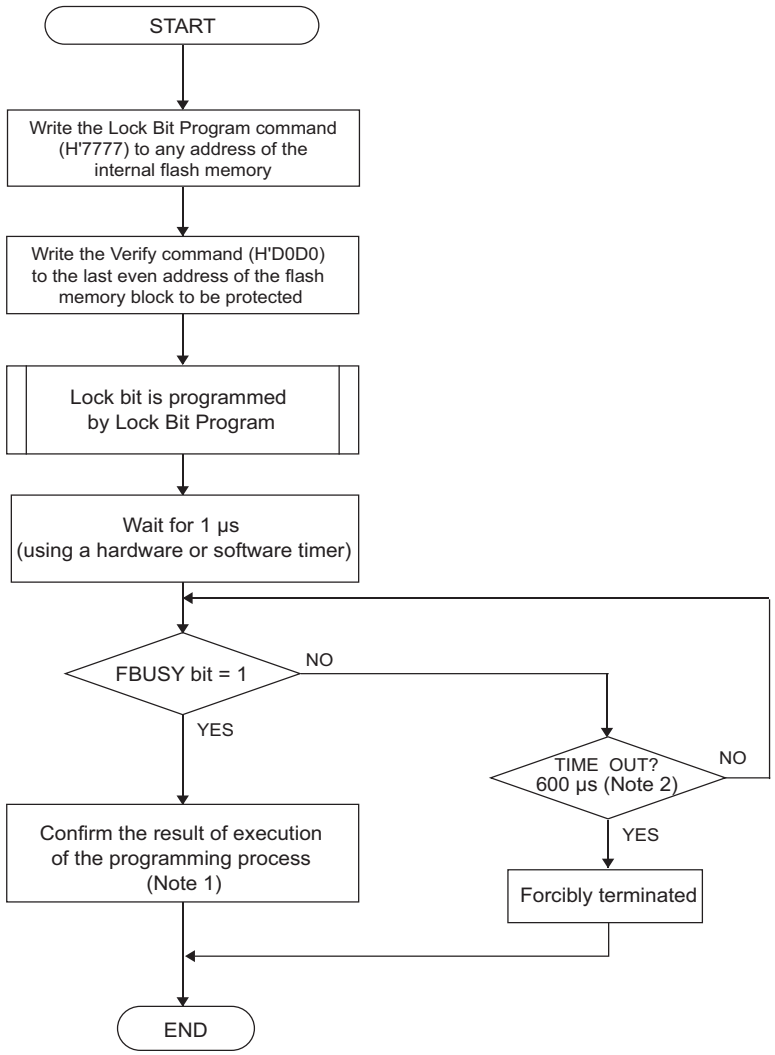
Target Block	Specified Address
0	H'0000 1FFE
1	H'0000 2FFE
2	H'0000 3FFE
3	H'0000 5FFE
4	H'0000 7FFE
5	H'0000 FFFE
6	H'0001 FFFE
7	H'0002 FFFE
8	H'0003 FFFE
9	H'0004 FFFE
10	H'0005 FFFE
11	H'0006 FFFE
12	H'0007 FFFE

Table 6.5.4 M32176F3 Target Blocks and Specified Addresses

Target Block	Specified Address
0	H'0000 1FFE
1	H'0000 2FFE
2	H'0000 3FFE
3	H'0000 5FFE
4	H'0000 7FFE
5	H'0000 FFFE
6	H'0001 FFFE
7	H'0002 FFFE
8	H'0003 FFFE
9	H'0004 FFFE
10	H'0005 FFFE

Table 6.5.5 M32176F2 Target Blocks and Specified Addresses

Target Block	Specified Address
0	H'0000 1FFE
1	H'0000 2FFE
2	H'0000 3FFE
3	H'0000 5FFE
4	H'0000 7FFE
5	H'0000 FFFE
6	H'0001 FFFE
7	H'0002 FFFE
8	H'0003 FFFE



Note 1: Check Flash Status Register (FSTAT) ERASE bit (for the erase status) and WRERR bit (for the write status).
Note 2: It is the TIME OUT period of 4K-byte block, which is 10k times rewritable product. The TIME OUT period of Other blocks and standard products (Flash rewrite durability: 100 times) is 200µs.

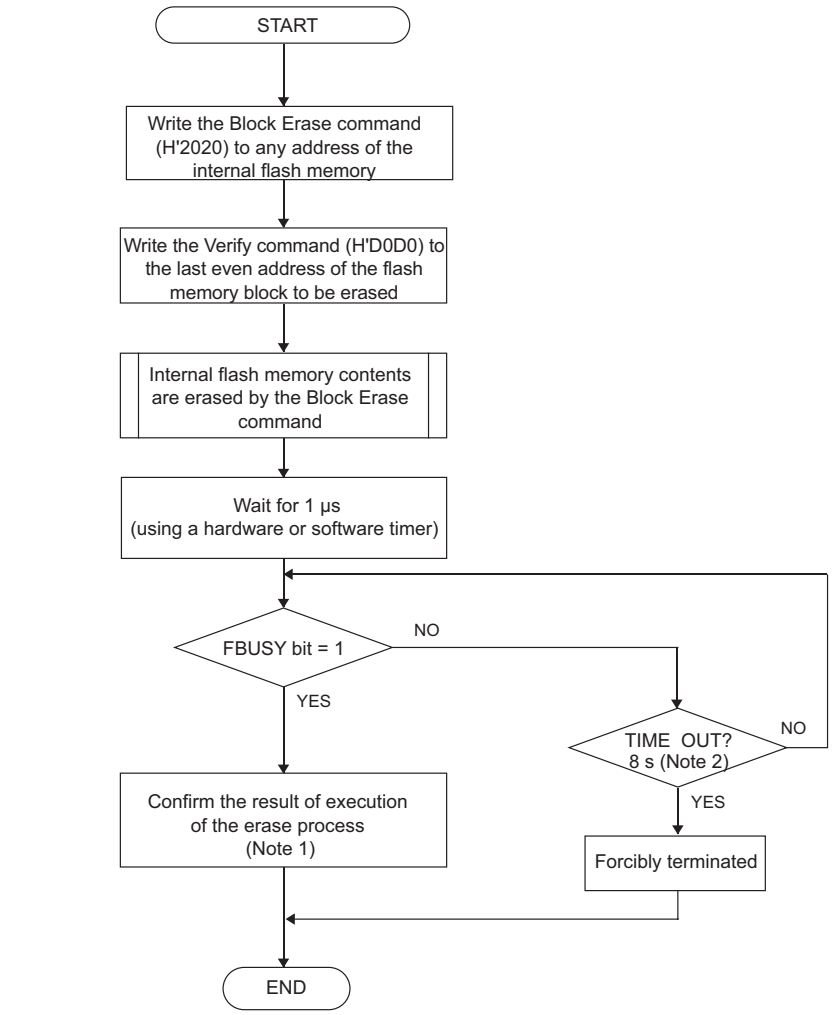
Figure 6.5.10 Lock Bit Program

(4) Block Erase command

The Block Erase command erases the content of the internal flash memory one block at a time. To perform this operation, write the command data (H'2020) to any address of the internal flash memory. Next, write the Verify command (H'D0D0) to the last even address of the flash memory block to be erased (see Tables 6.5.3, 6.5.4 and 6.5.5, "M32176 Target Blocks and Specified Addresses").

The protected flash memory blocks cannot be erased by the Block Erase command.

Block erase operation is automatically performed by the internal control circuit, and whether the Block Erase command has finished can be known by checking the Flash Status Register (FSTAT) FBUSY bit. (See Section 6.4.2, "Flash Status Registers.") While the FBUSY bit = "0", the next block erase operation cannot be performed.



Note 1: Check Flash Status Register (FSTAT) ERASE bit (for the erase status) and WRERR bit (for the write status).
Note 2: It is the TIME OUT period of 4K-byte block, which is 10k times rewritable product. The TIME OUT period of Other blocks and standard products (Flash rewrite durability: 100 times) is 6s.

Figure 6.5.11 Block Erase

(7) Clear Status Register command

The Clear Status Register command clears the Flash Status Register (FSTAT) ERASE (erase status), and WRERR (write status) bits to "0". Write the command data (H'5050) to any address of the internal flash memory, and Flash Status Register is thereby initialized. Also, issue the Clear Status Register command, and Flash Status Register 3 (FCNT3) is initialized.

If an error occurs when programming or erasing the flash memory and the Flash Status Register (FSTAT) ERASE (erase status) or WRERR (write status) bit is set to "1", the next programming or erasing operation cannot be executed unless each status bit is cleared to "0".

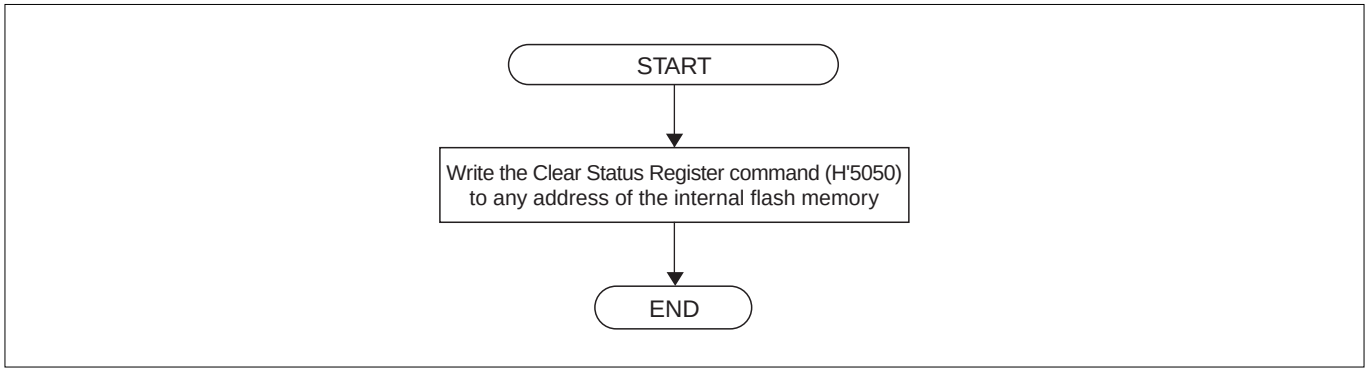


Figure 6.5.12 Clear Status Register

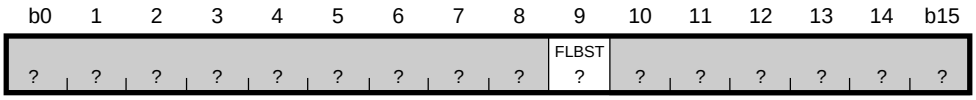
(8) Read Lock Bit Status command

The Read Lock Bit Status command is provided for checking whether a flash memory block is protected against programming/erasing operation. The method for reading lock bit can be chosen from the following depends on the setting for Flash Control Register 2 (FCNT2) FLOCKS (Lock bit read mode select) bit.

1) Memory area read mode (FLOCKS bit = 0)

Write the command data (H'7171) to any address of the internal flash memory. Next, read the last even address of the flash memory block to be checked (see Tables 6.5.3, 6.5.4 and 6.5.5, "M32176 Target Blocks and Specified Addresses"), and the read data shows whether the target block is protected.
If the FLBST (lock bit) in the read data is "0", it means that the target memory block is protected. If the FLBST (lock bit) is "1", it means that the target memory block is not protected.

Lock Bit Status Register (FLBST)



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–8	No function assigned.		?	0
9	FLBST Lock bit	0: Protected 1: Not protected	R	–
10–15	No function assigned.		?	0

The Lock Bit Status Register is a read-only register, which is included for each memory block independently of one another. To read this register, Flash Control Register 2 (FCNT2) FLOCKS bit must be set to "0".

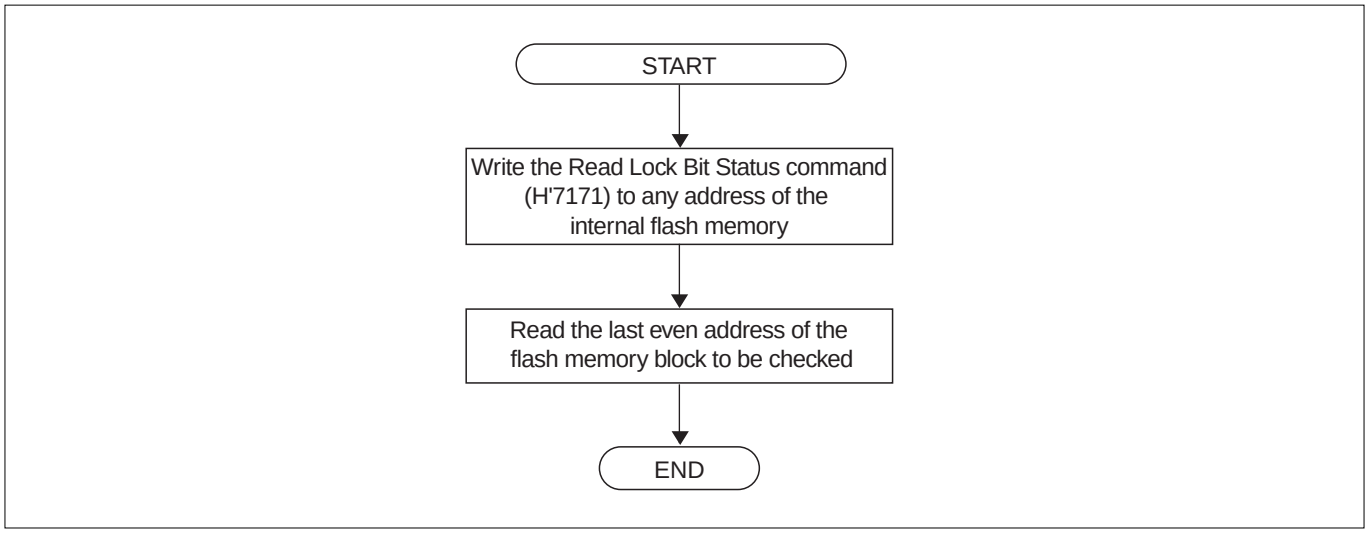
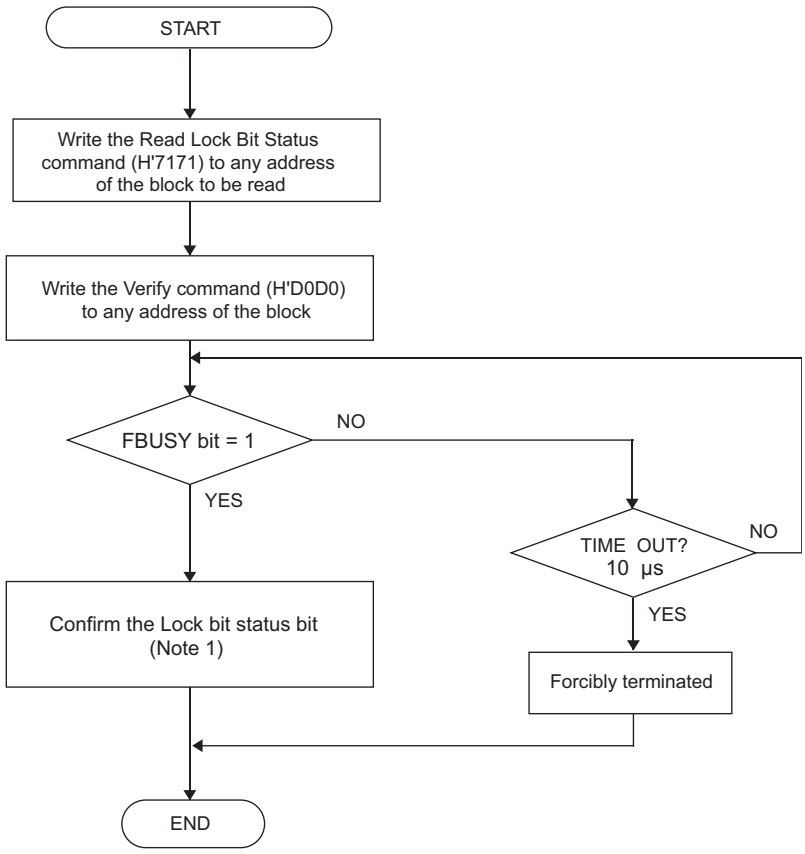


Figure 6.5.13 Read Lock Bit Status (Memory Area Read Mode)

2) Register read mode (FLOCKS bit = 1)

Write the command data (H’7171) to any address of the target block. Next, write the verify command data (H'D0D0), and the Flash Control Register 4 (FCNT4) FLOCKST (Lock Bit Status) bit shows whether the target block is protected.



Note 1: Check Flash Status Register 4 (FSTAT4) FLOCKST bit.

Figure 6.5.14 Read Lock Bit Status (Register Read Mode)

The following describes how to write to the lock bit.

a) To clear the lock bit to "0" (flash protected)
Issue the Lock Bit Program command (H’7777) to the memory block to be protected.

b) To set the lock bit to "1" (flash unprotected)
After setting the FPROT bit in Flash Control Register 2 to 1 (protection by lock bit disabled), use the Block Erase command (H’2020) to erase the memory block to be unprotected.
The lock bit cannot be set to "1" directly by writing to it.

c) Lock bit status when reset
Because the lock bit is a nonvolatile bit, it remains unaffected when the microcomputer is reset or powered off.

6.5.4 Flash Programming Time (Reference)

The following shows the time needed to program internal flash memory for reference.

(1) M32176F4

[1] Time required for transfer by SIO (for a transfer data size of 512 Kbytes)

1/57,600 bps × 1 (frame) × 11 (number of bits transferred) × 512 KB = approx. 100.1 [s]

[2] Time required for programming the flash memory

512 KB / 2-byte × 25 µs = approx. 6.6 [s]

[3] Time required for erasing the entire area

0.3 s × 5 (blocks) + 0.5 s × 1 (block) + 0.8 s × 7 (blocks) = 7.6 [s]

[4] Total flash programming time (entire 512 Kbytes area)

When communicating at 57,600 bps via UART, the flash programming time can be ignored because it is very short compared to the serial communication time. Therefore, the total flash programming time can be calculated using the equation below.

[1] + [3] = approx. 108 [s]

If the transfer time can be ignored by speeding up the serial communication or by other means, the fastest programming time possible can be calculated using the equation below.

[2] + [3] = approx. 15 [s]

(2) M32176F3

1) Time required for transfer by SIO (for a transfer data size of 384 Kbytes)

1/57,600 bps × 1 (frame) × 11 (number of bits transferred) × 384 KB = approx. 75.1 [s]

2) Time required for programming the flash memory

384 KB / 2-byte × 25 µs = approx. 4.9 [s]

3) Time required for erasing the entire area

0.3 s × 5 (blocks) + 0.5 s × 1 (block) + 0.8 s × 5 (blocks) = 6 [s]

4) Total flash programming time (entire 384 Kbytes area)

When communicating at 57,600 bps via UART, the flash programming time can be ignored because it is very short compared to the serial communication time. Therefore, the total flash programming time can be calculated using the equation below.

[1] + [3] = approx. 82 [s]

If the transfer time can be ignored by speeding up the serial communication or by other means, the fastest programming time possible can be calculated using the equation below.

[2] + [3] = approx. 11 [s]

(3) M32176F2

1) Time required for transfer by SIO (for a transfer data size of 256 Kbytes)

1/57,600 bps × 1 (frame) × 11 (number of bits transferred) × 256 KB = approx. 50.1 [s]

2) Time required for programming the flash memory

256 KB / 2-byte × 25 µs = approx. 3.3 [s]

3) Time required for erasing the entire area

0.3 s × 5 (blocks) + 0.5 s × 1 (block) + 0.8 s × 3 (blocks) = 4.4 [s]

4) Total flash programming time (entire 256 Kbytes area)

When communicating at 57,600 bps via UART, the flash programming time can be ignored because it is very short compared to the serial communication time. Therefore, the total flash programming time can be calculated using the equation below.

[1] + [3] = approx. 55 [s]

If the transfer time can be ignored by speeding up the serial communication or by other means, the fastest programming time possible can be calculated using the equation below.

[2] + [3] = approx. 8 [s]

6.6 Virtual Flash Emulation Function

The microcomputer has the function to map 8-Kbyte memory blocks of the internal RAM (max. 2 blocks) into areas (L banks) of the internal flash memory that are divided in 8-Kbyte units and to map 4-Kbyte memory blocks of the internal RAM (max. 2 blocks) into areas (S banks) of the internal flash memory that are divided in 4-Kbyte units. This functions is referred to as the Virtual Flash Emulation Function.

This function allows the data located in 4-Kbyte or 8-Kbyte blocks of the internal RAM to be changed with the contents of internal flash memory at the addresses specified by the Virtual Flash Bank Register. That way, the relevant internal RAM data can read out by reading the content of internal flash memory.

For applications that require modifying the contents of internal flash memory (e.g., data table) during operation, this function enables dynamic data modification by modifying the relevant internal RAM data.

The internal RAM blocks allocated for virtual flash emulation can be accessed for read and write the same way as in usual internal RAM.

This function, when used in combination with the microcomputer's internal Real-Time Debugger (RTD), allows the data table, etc. created in the internal flash memory to be referenced or rewritten from the outside, thereby facilitating data table tuning from an external device.

Note: • Before programming/erasing the internal flash memory, always be sure to exit this virtual flash emulation mode.

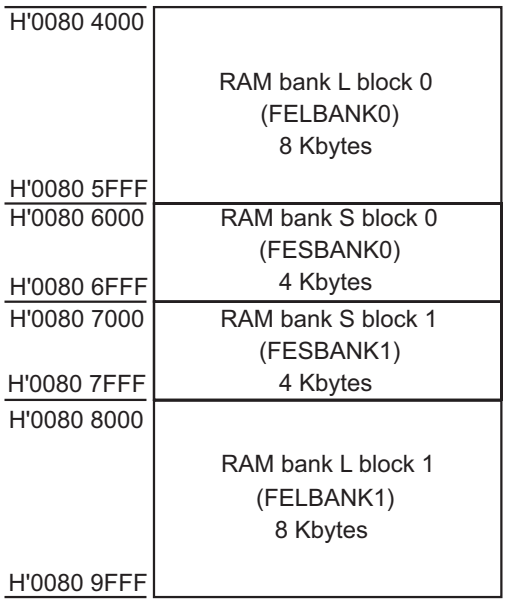


Figure 6.6.1 Internal RAM Bank Configuration of the M32176

6.6.1 Virtual Flash Emulation Area

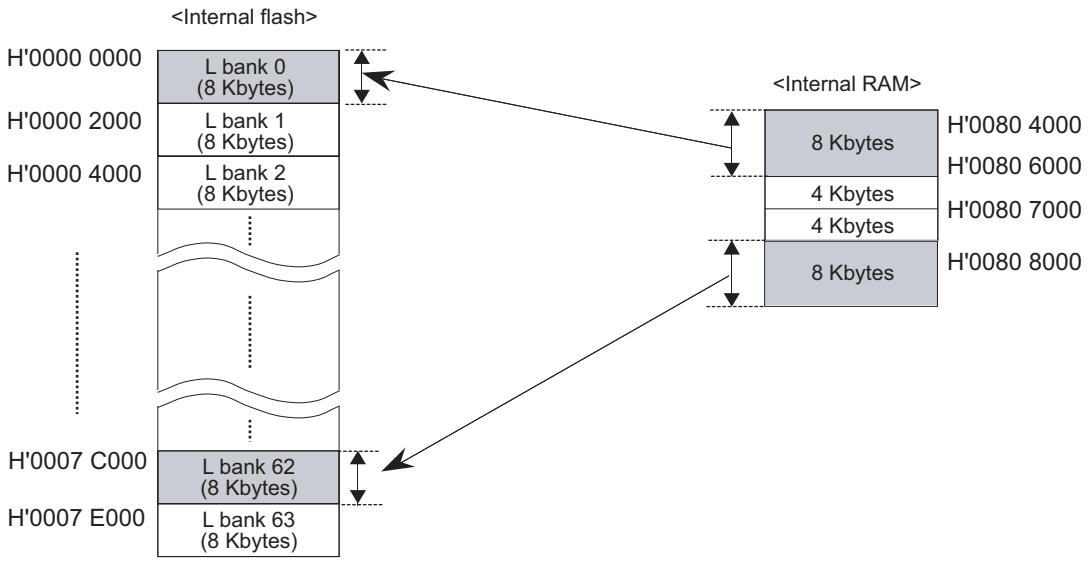
The following shows the internal flash memory areas in which the Virtual Flash Emulation Function is applicable.

Using the Virtual Flash L Bank Register (FELBANK0, FELBANK1), select one among all L banks of internal flash memory that are divided in 8-Kbyte units (by setting the seven start address bits A12–A18 of the desired L bank in the Virtual Flash L Bank Register LBANKAD bits). Then set the Virtual Flash L Bank Register's flash emulation L enable bit (MODENL) to "1", and the selected L bank area will be replaced with 8-Kbyte blocks of the internal RAM, up to two blocks in all.
Using the Virtual Flash S Bank Register (FESBANK0, FESBANK1), select one among all S banks of internal flash memory that are divided in 4-Kbyte units (by setting the eight start address bits A12–A19 of the desired S bank in the Virtual Flash S Bank Register SBANKAD bits). Then set the Virtual Flash S Bank Register's flash emulation S enable bit (MODENS) to "1", and the selected S bank area will be replaced with 4-Kbyte blocks of the internal RAM, up to two blocks in all.
Two 8-Kbyte units L banks and two 4-Kbyte units S banks, total of four banks (maximum), can be selected.

Notes: • If the same bank area is set in two or more Virtual Flash Bank Registers and each register's flash emulation enable bit is enabled, the bank is assigned the corresponding internal RAM area (8-Kbyte or 4-Kbyte) according to the priority given below.

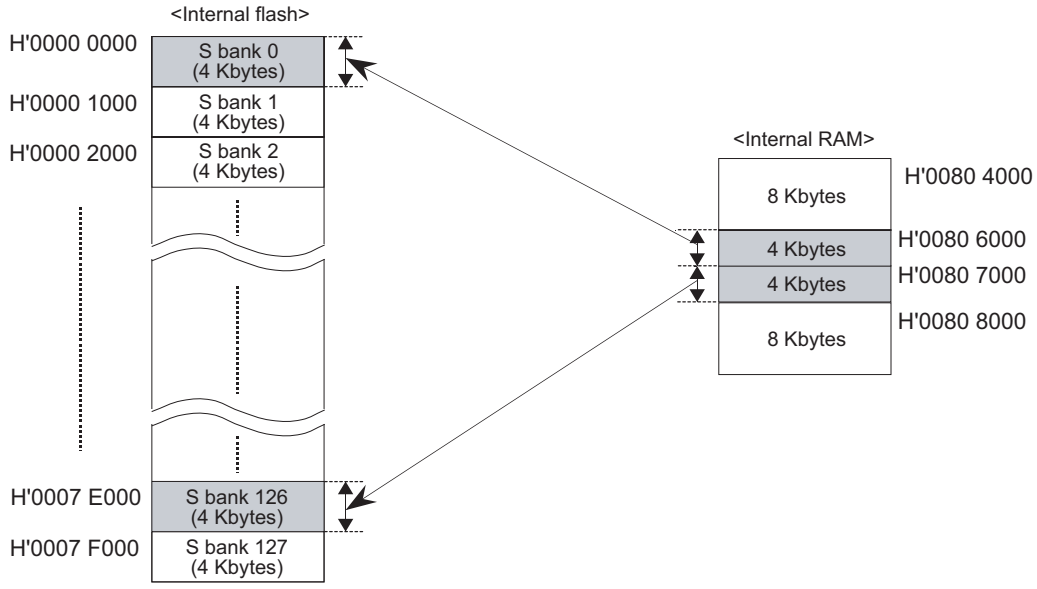
FELBANK0 > FESBANK0 > FESBANK1 > FELBANK1

- During virtual flash emulation mode, RAM can be accessed for read and write from the internal RAM area and the virtual flash set area.
- Before reading any virtual flash area after setting the Flash Control Register 1 virtual flash emulation mode bit to "1", be sure that there must be an interval of at least three clocks (CPU clocks).
- Before reading any virtual flash area after setting the Virtual Flash Bank Register (L bank and S bank registers) virtual flash emulation enable bit and bank address bits, be sure that there must be an interval of at least three clocks (CPU clocks).



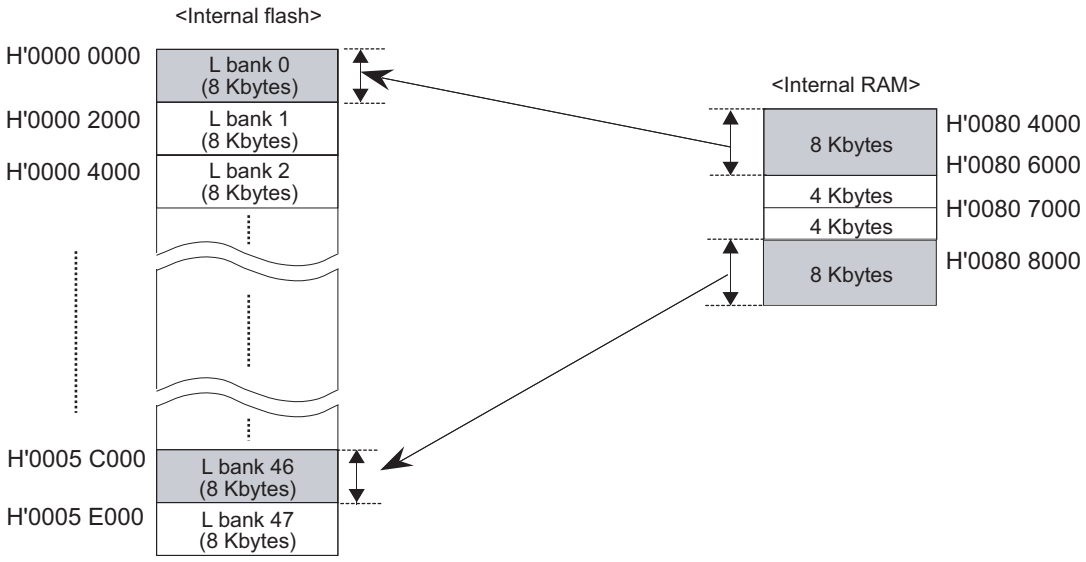
Notes: • If the same bank area is set in two or more Virtual Flash Bank Registers and each register's flash emulation enable bit is enabled, the bank is assigned the corresponding internal RAM area in order of priority: FELBANK0 > FESBANK0 > FESBANK1 > FELBANK1.
• If any 8-Kbyte area (L bank) specified by the Virtual Flash L Bank Registers 0 and 1 is accessed, its corresponding internal RAM area is accessed. During virtual flash emulation mode, RAM can be accessed for read and write from both the internal RAM area and the virtual flash set area.

Figure 6.6.2 M32176F4 Virtual Flash Emulation Area divided in 8-Kbyte units



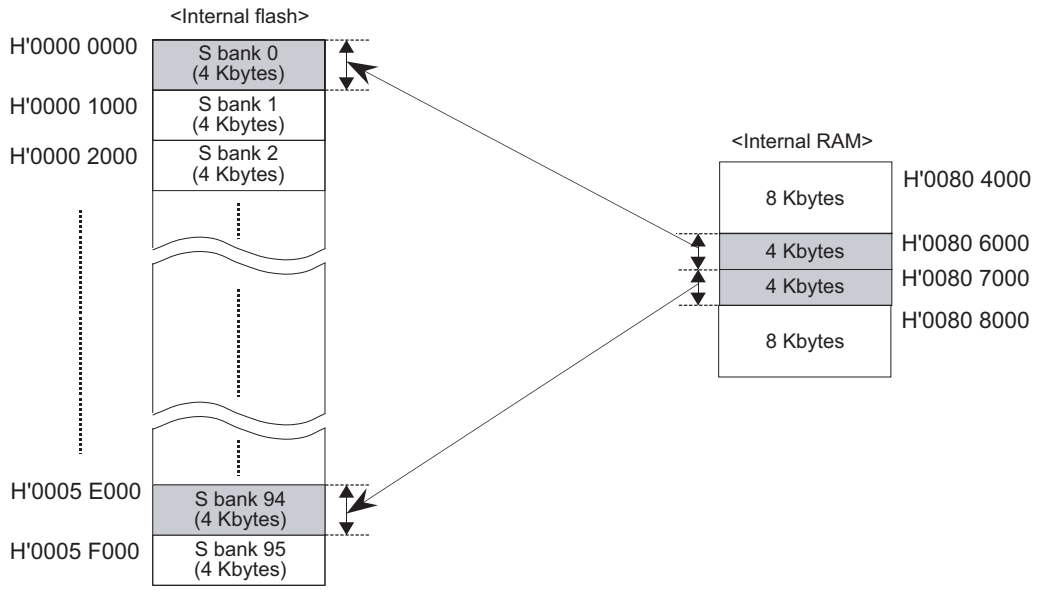
Notes: • If the same bank area is set in two or more Virtual Flash Bank Registers and each register's flash emulation enable bit is enabled, the bank is assigned the corresponding internal RAM area in order of priority: FELBANK0 > FESBANK0 > FESBANK1 > FELBANK1.
• If any 4-Kbyte area (S bank) specified by the Virtual Flash S Bank Registers 0 and 1 is accessed, its corresponding internal RAM area is accessed. During virtual flash emulation mode, RAM can be accessed for read and write from both the internal RAM area and the virtual flash set area.

Figure 6.6.3 M32176F4 Virtual Flash Emulation Area divided in 4-Kbyte units



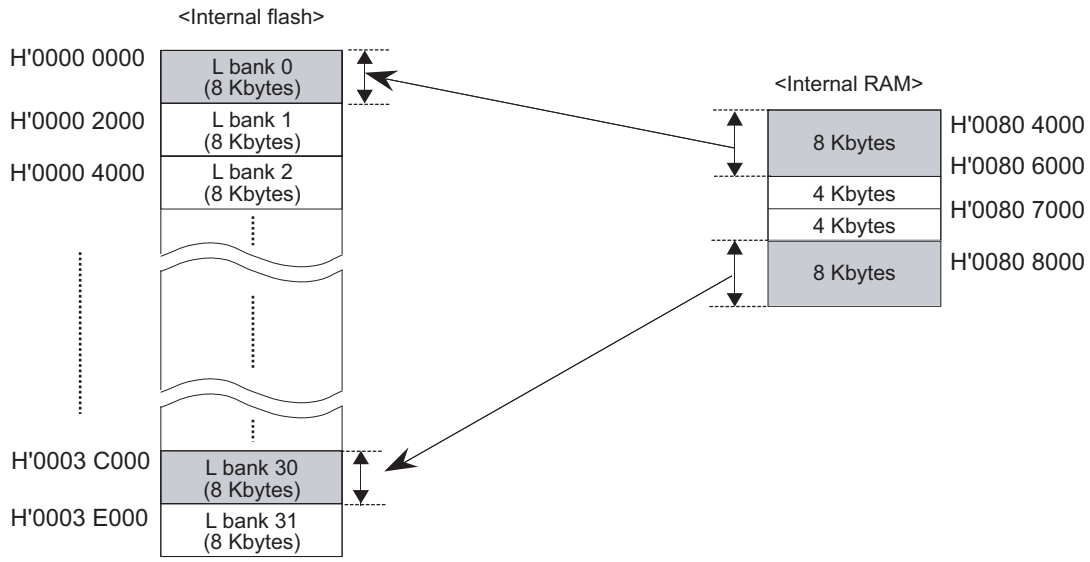
Notes: • If the same bank area is set in two or more Virtual Flash Bank Registers and each register's flash emulation enable bit is enabled, the bank is assigned the corresponding internal RAM area in order of priority: FELBANK0 > FESBANK0 > FESBANK1 > FELBANK1.
• If any 8-Kbyte area (L bank) specified by the Virtual Flash L Bank Registers 0 and 1 is accessed, its corresponding internal RAM area is accessed. During virtual flash emulation mode, RAM can be accessed for read and write from both the internal RAM area and the virtual flash set area.

Figure 6.6.4 M32176F3 Virtual Flash Emulation Area divided in 8-Kbyte units



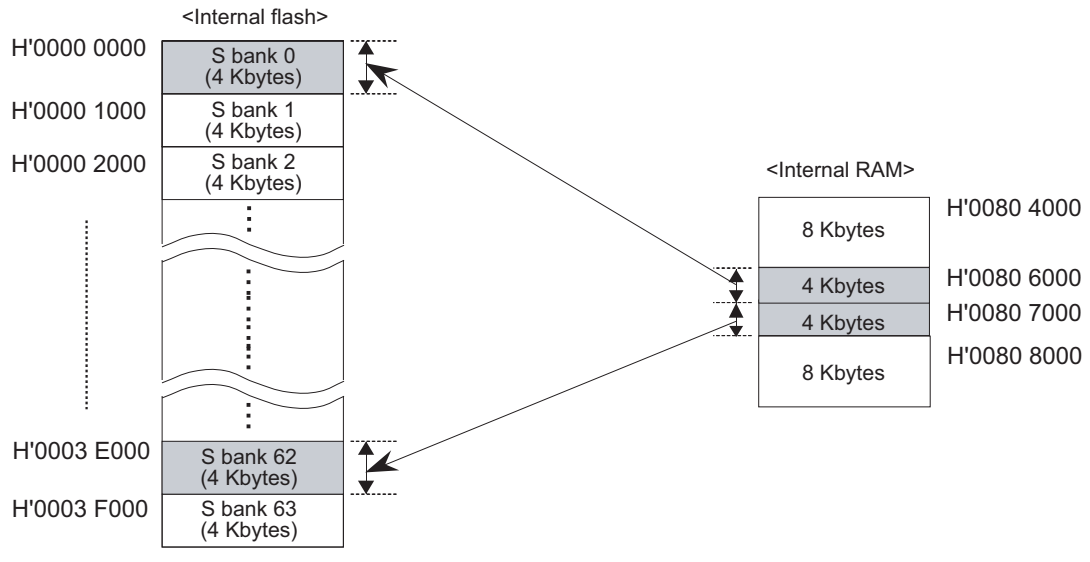
Notes: • If the same bank area is set in two or more Virtual Flash Bank Registers and each register's flash emulation enable bit is enabled, the bank is assigned the corresponding internal RAM area in order of priority: FELBANK0 > FESBANK0 > FESBANK1 > FELBANK1.
• If any 4-Kbyte area (S bank) specified by the Virtual Flash S Bank Registers 0 and 1 is accessed, its corresponding internal RAM area is accessed. During virtual flash emulation mode, RAM can be accessed for read and write from both the internal RAM area and the virtual flash set area.

Figure 6.6.5 M32176F3 Virtual Flash Emulation Area divided in 4-Kbyte units



Notes: • If the same bank area is set in two or more Virtual Flash Bank Registers and each register's flash emulation enable bit is enabled, the bank is assigned the corresponding internal RAM area in order of priority: FELBANK0 > FESBANK0 > FESBANK1 > FELBANK1.
• If any 8-Kbyte area (L bank) specified by the Virtual Flash L Bank Registers 0 and 1 is accessed, its corresponding internal RAM area is accessed. During virtual flash emulation mode, RAM can be accessed for read and write from both the internal RAM area and the virtual flash set area.

Figure 6.6.6 M32176F2 Virtual Flash Emulation Area divided in 8-Kbyte units



Notes: • If the same bank area is set in two or more Virtual Flash Bank Registers and each register's flash emulation enable bit is enabled, the bank is assigned the corresponding internal RAM area in order of priority: FELBANK0 > FESBANK0 > FESBANK1 > FELBANK1.
• If any 4-Kbyte area (S bank) specified by the Virtual Flash S Bank Registers 0 and 1 is accessed, its corresponding internal RAM area is accessed. During virtual flash emulation mode, RAM can be accessed for read and write from both the internal RAM area and the virtual flash set area.

Figure 6.6.7 M32176F2 Virtual Flash Emulation Area divided in 4-Kbyte units

L bank	Start address of bank in flash memory	Values set in L bank address (LBANKAD) bit
L bank 0	H'0000 0000 (Note 1)	H'00
L bank 1	H'0000 2000 (Note 1)	H'02
L bank 2	H'0000 4000 (Note 1)	H'04
≈	⋮	≈
L bank 62	H'0007 C000 (Note 1)	H'7C
L bank 63	H'0007 E000 (Note 1)	H'7E

Note 1: Set the seven start address bits A12-A18 of each L bank of internal flash memory that is divided in 8-Kbyte units in the Virtual Flash L Bank Register's L bank address (LBANKAD) bits.

Figure 6.6.8 Values Set in the M32176F4's Virtual Flash Bank Register when divided in 8-Kbyte units

S bank	Start address of bank in flash memory	Values set in S bank address (SBANKAD) bit
S bank 0	H'0000 0000 (Note 1)	H'00
S bank 1	H'0000 1000 (Note 1)	H'01
S bank 2	H'0000 2000 (Note 1)	H'02
≈	⋮	≈
S bank 126	H'0007 E000 (Note 1)	H'7E
S bank 127	H'0007 F000 (Note 1)	H'7F

Note 1: Set the eight start address bits A12-A19 of each S bank of internal flash memory that is divided in 4-Kbyte units in the Virtual Flash S Bank Register's S bank address (SBANKAD) bits.

Figure 6.6.9 Values Set in the M32176F4's Virtual Flash Bank Register when divided in 4-Kbyte units

L bank	Start address of bank in flash memory	Values set in L bank address (LBANKAD) bit
L bank 0	H'0000 0000 (Note 1)	H'00
L bank 1	H'0000 2000 (Note 1)	H'02
L bank 2	H'0000 4000 (Note 1)	H'04
⋮	⋮	⋮
L bank 46	H'0005 C000 (Note 1)	H'5C
L bank 47	H'0005 E000 (Note 1)	H'5E

Note 1: Set the seven start address bits A12-A18 of each L bank of internal flash memory that is divided in 8-Kbyte units in the Virtual Flash L Bank Register's L bank address (LBANKAD) bits.

Figure 6.6.10 Values Set in the M32176F3's Virtual Flash Bank Register when divided in 8-Kbyte units

S bank	Start address of bank in flash memory	Values set in S bank address (SBANKAD) bit
S bank 0	H'0000 0000 (Note 1)	H'00
S bank 1	H'0000 1000 (Note 1)	H'01
S bank 2	H'0000 2000 (Note 1)	H'02
⋮	⋮	⋮
S bank 94	H'0005 E000 (Note 1)	H'5E
S bank 95	H'0005 F000 (Note 1)	H'5F

Note 1: Set the eight start address bits A12-A19 of each S bank of internal flash memory that is divided in 4-Kbyte units in the Virtual Flash S Bank Register's S bank address (SBANKAD) bits.

Figure 6.6.11 Values Set in the M32176F3's Virtual Flash Bank Register when divided in 4-Kbyte units

L bank	Start address of bank in flash memory	Values set in L bank address (LBANKAD) bit
L bank 0	H'0000 0000 (Note 1)	H'00
L bank 1	H'0000 2000 (Note 1)	H'02
L bank 2	H'0000 4000 (Note 1)	H'04
≈	⋮	≈
L bank 30	H'0003 C000 (Note 1)	H'3C
L bank 31	H'0003 E000 (Note 1)	H'3E

Note 1: Set the seven start address bits A12-A18 of each L bank of internal flash memory that is divided in 8-Kbyte units in the Virtual Flash L Bank Register's L bank address (LBANKAD) bits.

Figure 6.6.12 Values Set in the M32176F2's Virtual Flash Bank Register when divided in 8-Kbyte units

S bank	Start address of bank in flash memory	Values set in S bank address (SBANKAD) bit
S bank 0	H'0000 0000 (Note 1)	H'00
S bank 1	H'0000 1000 (Note 1)	H'01
S bank 2	H'0000 2000 (Note 1)	H'02
≈	⋮	≈
S bank 62	H'0003 E000 (Note 1)	H'3E
S bank 63	H'0003 F000 (Note 1)	H'3F

Note 1: Set the eight start address bits A12-A19 of each S bank of internal flash memory that is divided in 4-Kbyte units in the Virtual Flash S Bank Register's S bank address (SBANKAD) bits.

Figure 6.6.13 Values Set in the M32176F2's Virtual Flash Bank Register when divided in 4-Kbyte units

6.6.2 Entering Virtual Flash Emulation Mode

To enter virtual flash emulation mode, set the Flash Control Register 1 (FCNT1) FEMMOD bit by writing "1". After entering virtual flash emulation mode, set the Virtual Flash Bank Register MODEN bit to "1" to enable the Virtual Flash Emulation Function.

Even during virtual flash emulation mode, the internal RAM area (H'0080 4000 through H'0080 9FFF) can be accessed the same way as in usual internal RAM.

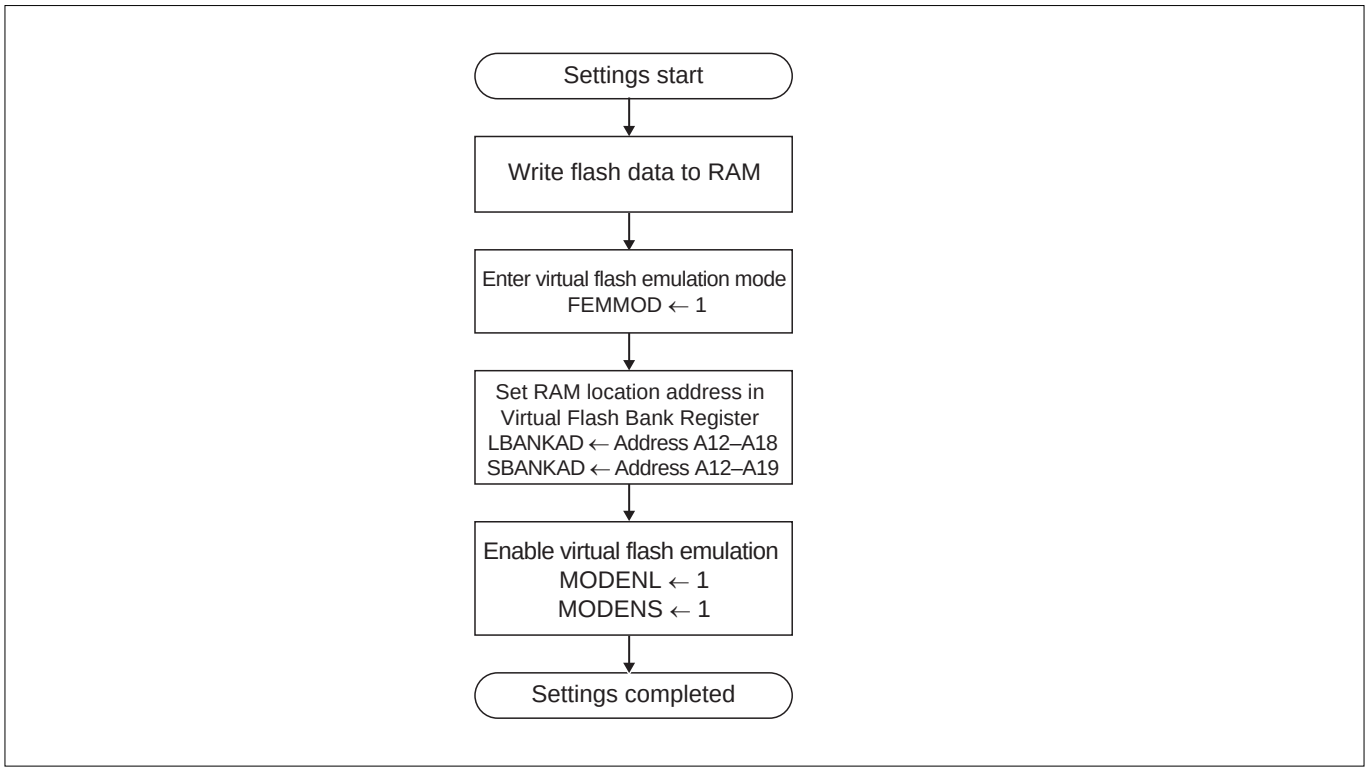


Figure 6.6.14 Virtual Flash Emulation Mode Sequence

6.6.3 Application Example of Virtual Flash Emulation Mode

By using two RAM areas that have been set in the same flash area by the Virtual Flash Emulation Function, the data in the flash memory can be replaced successively.

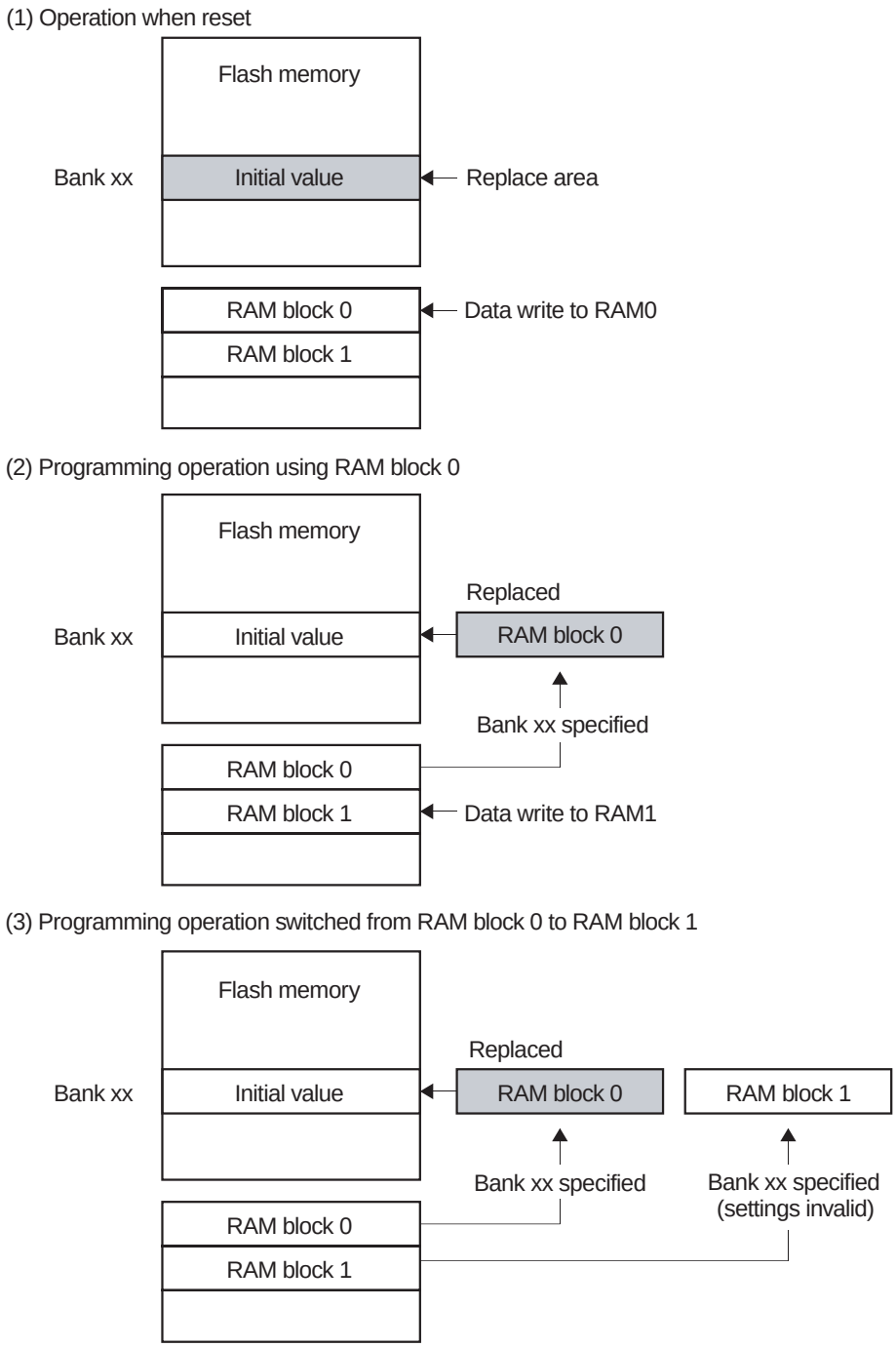


Figure 6.6.15 Application Example of Virtual Flash Emulation Mode (1/2)

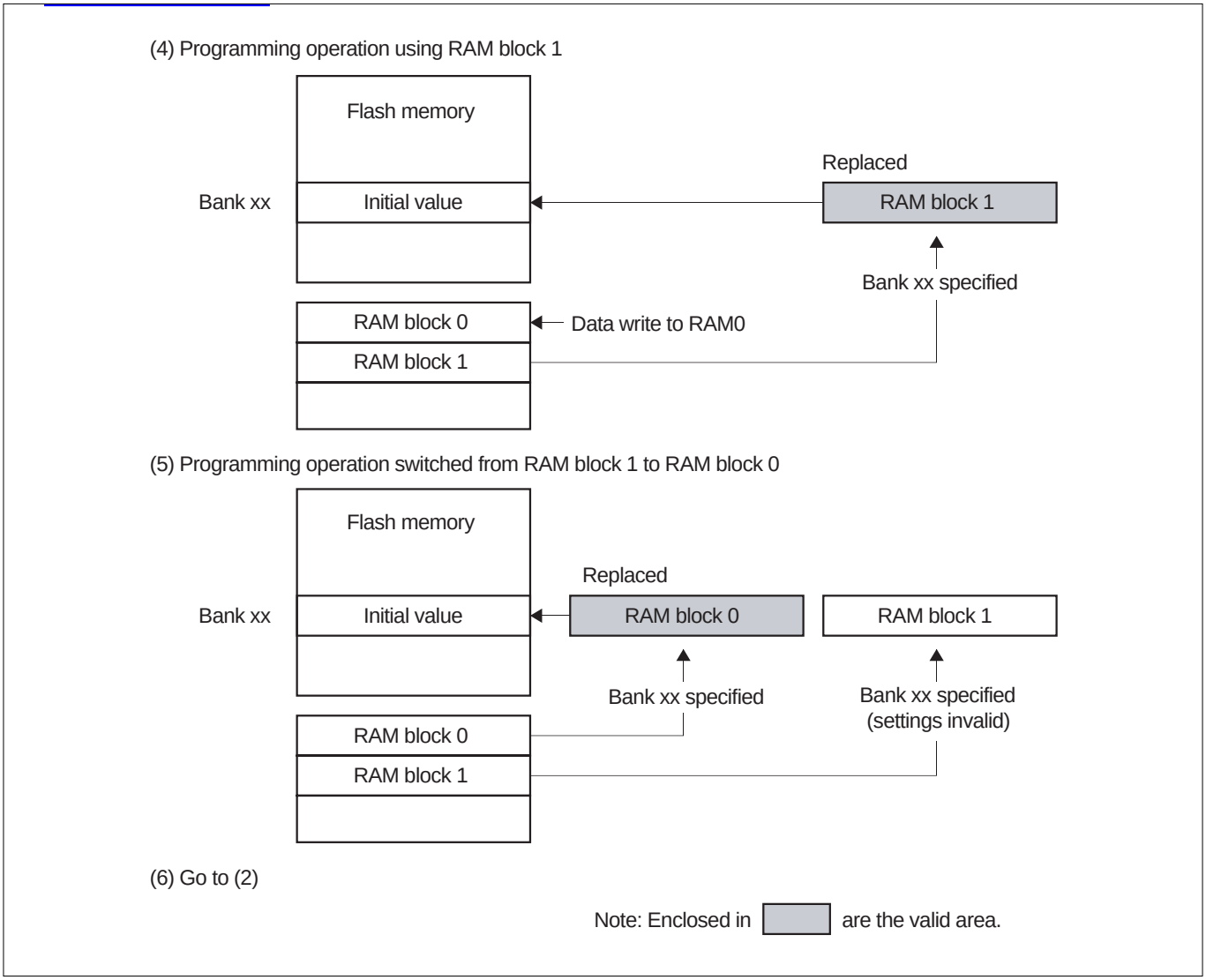


Figure 6.6.16 Application Example of Virtual Flash Emulation Mode (2/2)

6.7 Connecting to a Serial Programmer (CSIO Mode)

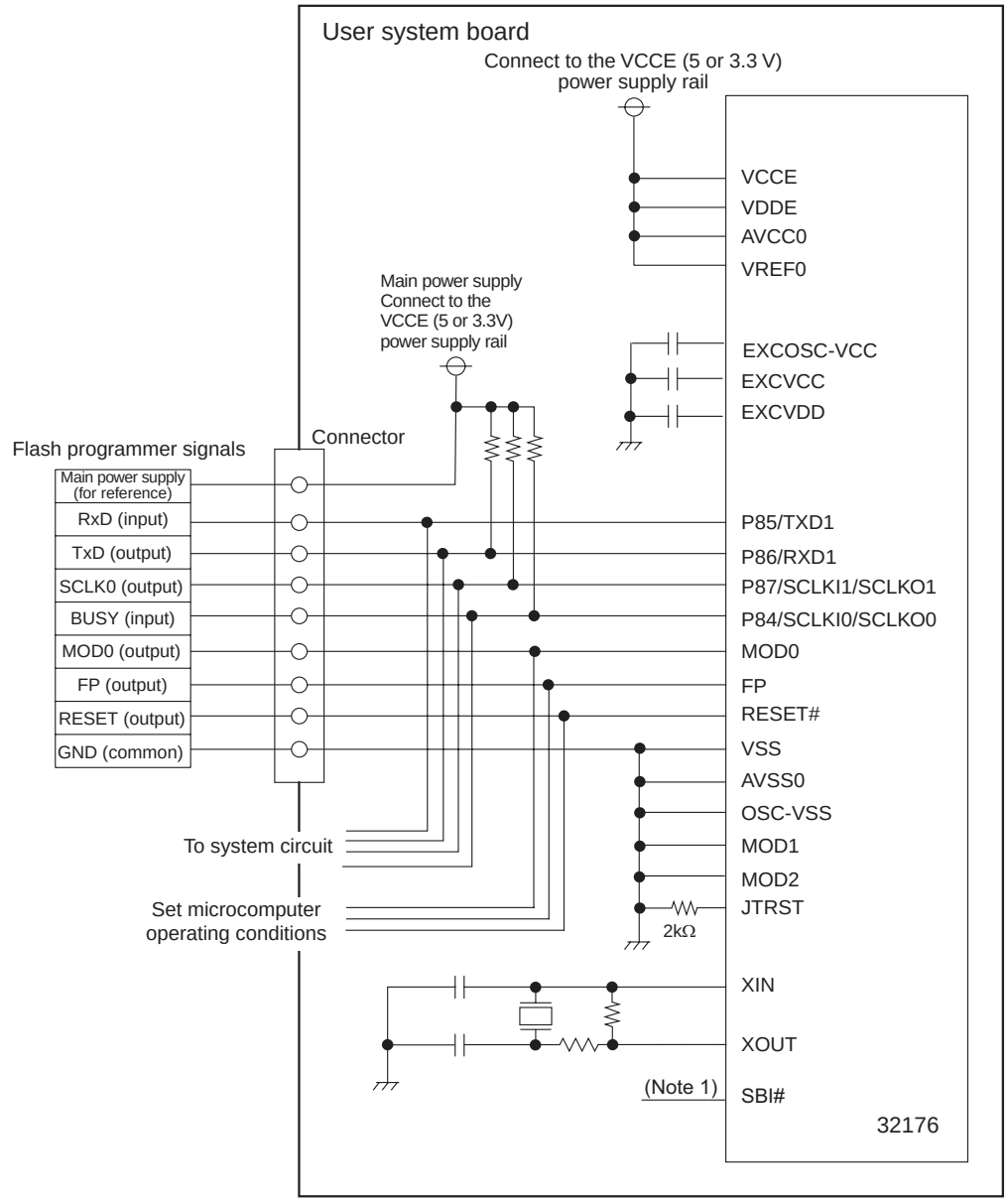
For the internal flash memory to be rewritten in boot mode + flash E/W enable mode by using a general-purpose serial programmer, several pins on the microcomputer must be processed to make them suitable for the serial programmer, as shown below.

Table 6.7.1 Processing Microcomputer Pins before Using a Serial Programmer

Pin Name	Pin No.	Function	Remark
SCLKI1	71	Transfer clock input	Pull high
RXD1	70	Serial data input (received data)	Pull high
TXD1	69	Serial data output (transmit data)	
P84	68	Transmit/receive enable output	Pull high
FP	94	Flash memory protect	Pull high
MOD0	92	Operation mode 0	Connect to the main power supply
MOD1	93	Operation mode 1	Connect to ground
MOD2	123	Operation mode 2	Connect to ground
RESET#	91	Reset	After setting MOD0/MOD1, ground and back to main power supply
XIN	4	Clock input	
XOUT	5	Clock output	
SBI#	77	System Break interrupt (SBI) input	Pull high or low
VREF0	42	Reference voltage input for A/D converter	Connect to the main power supply
AVCC0	43	Analog power supply	Connect to the main power supply
AVSS0	60	Analog ground	Connect to ground
VDDE	108	RAM backup power supply	Connect to the main power supply
VCCE	20, 65, 95, 132	Main power supply	5 V +/- 10% or 3.3 V +/- 10%
EXCVCC	61, 137	Connects external capacitance for the internal power supply	Need to be grounded to earth via capacitor
EXCVDD	73	Connects external capacitance for the RAM power supply	Need to be grounded to earth via capacitor
EXCOSC-VCC	6	Connects external capacitance for the oscillator power supply	Need to be grounded to earth via capacitor
VSS	3, 21, 62, 72, 96, 138	Ground	0V
JTRST	111	JTAG reset input	Pull low (0-100 kΩ)

Notes • Pin processing is not required for those that are not listed above.

The diagram below shows an example of a user system configuration which has had a serial programmer connected. After the user system is powered on, the serial programmer writes to the internal flash memory in clock-synchronized serial mode. No communication problems associated with the oscillator frequency may occur. If the system uses any pins that are to be connected to a serial programmer, care must be taken to prevent adverse effects on the system when a serial programmer is connected. Note that the serial programmer uses the addresses H'0000 0084 through H'0000 0093 as an area in which to check the ID for flash memory protection. If the internal flash memory needs to be protected, set any ID in this area.



Note 1: SBI# must be fixed "H" or "L" to ensure that no interrupts will be generated.

Notes:
- Turn on the power for the user system before writing to the internal flash memory.
- If P84-P87 are used in the system circuit, connection to a serial programmer must be taken into consideration.
- SBI# must be fixed "H" or "L" to ensure that no interrupts will be generated.
- The pullup resistance values of P84, P86 and P87 must be selected to suit the system design condition.
- The typical pullup resistance values of P84, P86 and P87 are 4.7 to 10 KΩ.
- The status of any other ports that are not shown here will not affect flash memory programming.
- Make sure the mode setting pin/power supply voltages do not fluctuate to prevent unintended changes of modes while rewriting the internal flash memory.

Figure 6.7.1 Pin Connection Diagram

6.8 Connecting to a Serial Programmer (UART Mode)

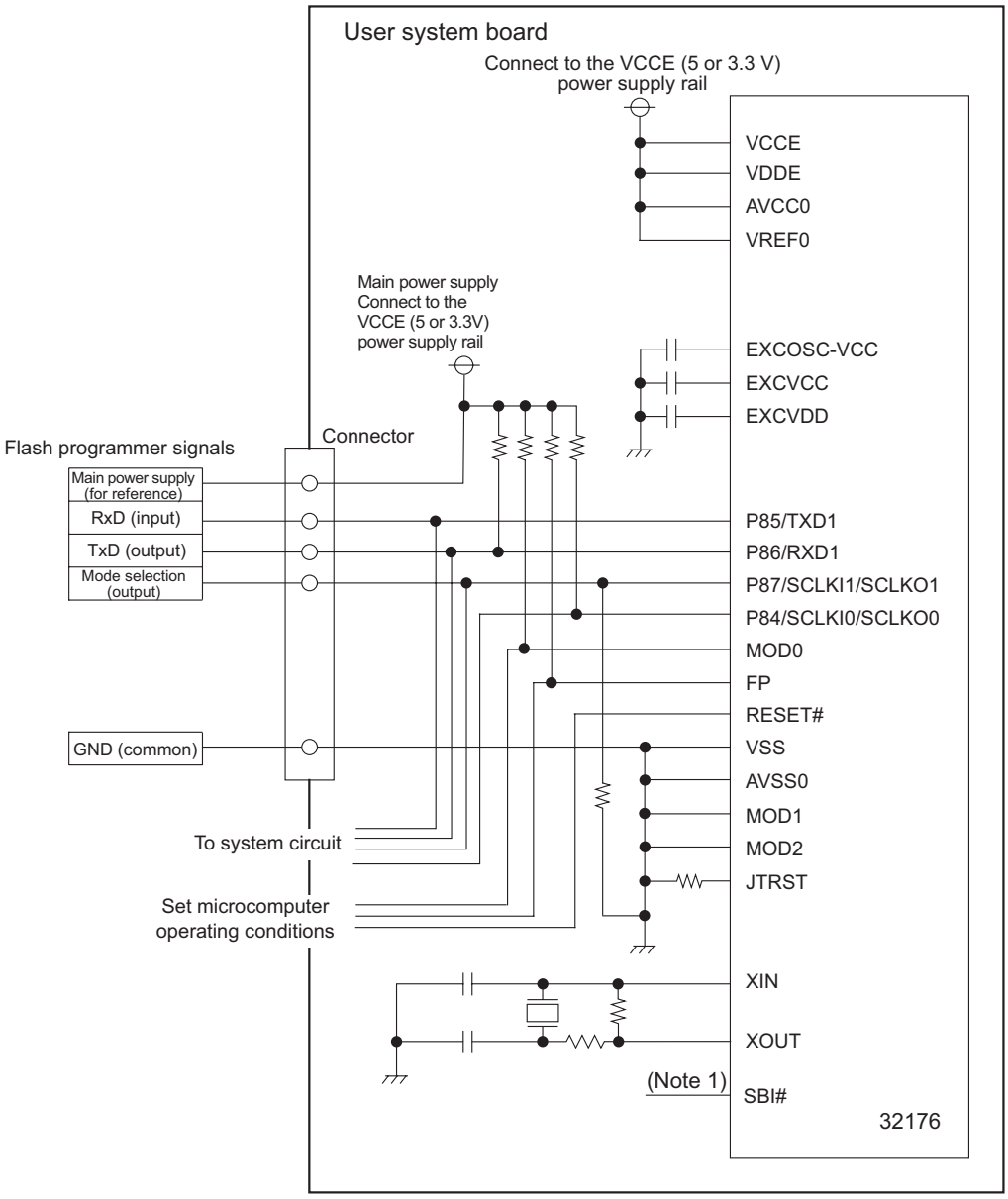
For the internal flash memory to be rewritten in boot mode + flash E/W enable mode by using a general-purpose serial programmer, several pins on the microcomputer must be processed to make them suitable for the serial programmer, as shown below.

Table 6.8.1 Processing Microcomputer Pins before Using a Serial Programmer for 32176 (UART Mode)

Pin Name	Pin No.	Function	Remark
SCLKI1	71	SIO mode selection	Pull low ("L" level input)
RXD1	70	Serial data input (received data)	Pull high
TXD1	69	Serial data output (transmit data)	
P84	68	General-purpose port input	Not used during UART mode Pull high or pull low
FP	94	Flash memory protect	Pull high
MOD0	92	Operation mode 0	Connect to the main power supply
MOD1	93	Operation mode 1	Connect to ground
MOD2	123	Operation mode 2	Connect to ground
RESET#	91	Reset	
XIN	4	Clock input	
XOUT	5	Clock output	
SBI#	77	System Break interrupt (SBI) input	Pull high or low
VREF0	42	Reference voltage input for A/D converter	Connect to the main power supply
AVCC0	43	Analog power supply	Connect to the main power supply
AVSS0	60	Analog ground	Connect to ground
VDDE	108	RAM backup power supply	Connect to the main power supply
VCCE	20, 65, 95, 132	Main power supply	5 V +/- 10% or 3.3 V +/- 10%
EXCVCC	61, 137	Connects external capacitance for the internal power supply	Need to be grounded to earth via capacitor
EXCVDD	73	Connects external capacitance for the RAM power supply	Need to be grounded to earth via capacitor
EXCOSC-VCC	6	Connects external capacitance for the oscillator power supply	Need to be grounded to earth via capacitor
VSS	3, 21, 62, 72, 96, 138	Ground	0V
JTRST	111	JTAG reset input	Pull low (0-100 kΩ)

Note • Pin processing is not required for those that are not listed above.

The diagram below shows an example of a user system configuration which has had a serial programmer connected. After the user system is powered on, the serial programmer writes to the internal flash memory in clock-asynchronous serial mode (UART mode). No communication problems associated with the oscillator frequency may occur. If the system uses any pins that are to be connected to a serial programmer, care must be taken to prevent adverse effects on the system when a serial programmer is connected. Note that the serial programmer uses the addresses H'0000 0084 through H'0000 0093 as an area in which to check the ID for flash memory protection. If the internal flash memory needs to be protected, set any ID in this area.



Note 1: SBI# must be fixed "H" or "L" to ensure that no interrupts will be generated.
Notes: • Turn on the power for the user system before writing to the internal flash memory.
• If P84-P87 are used in the system circuit, connection to a serial programmer must be taken into consideration.
• The pullup/pulldown resistance values of P84, P86, P87, FP and MOD0 must be selected to suit the system design condition.
• The typical pullup/pulldown resistance values of P84, P86, P87, FP and MOD0 are 4.7 to 10 KΩ.
• The status of any other ports that are not shown here will not affect flash memory programming.
• Make sure the mode setting pin/power supply voltages do not fluctuate to prevent unintended changes of modes while rewriting the internal flash memory.

Figure 6.8.1 Pin Connection Diagram (UART Mode)

6.9 Internal Flash Memory Protect Function

The internal flash memory has the following four types of protect functions to prevent it from being inadvertently rewritten or illegally copied, programmed or erased.

(1) Flash memory protect ID

When using a tool to program/erase the internal flash memory such as a general-purpose programmer or emulator, the ID entered by a tool and the ID stored in the internal flash memory are collated. Unless the correct ID is entered, the internal flash memory cannot be read out, programmed nor erased. (For some tools, tool execution is enabled after erasing the entire flash memory area, and the internal flash memory becomes accessible for write.)

(2) Protection by FP pin

The internal flash memory is protected in hardware against programming/erasing operation by pulling the FP (Flash Protect) pin "L" level. For systems that do not require rewriting flash memory or systems in which flash reprogramming is prohibited as in the case of automotive applications, make sure the FP pin is fixed "L" level except when programming or erasing the internal flash memory. Furthermore, because the FP pin level can be known by reading the Flash Mode Register (FMOD)'s FPMOD (external FP pin status) bit in the flash write/erase program, the internal flash memory can also be protected in software. For systems that do not require protection by setting external pins, the FP pin may be fixed high to simplify the operation to program/erase the internal flash memory. However, to prevent the flash memory from being inadvertently rewritten by an erratic operation in software, use the protection by a lock bit described in (4) below.
When programming/erasing via JTAG, the flash memory can be programmed or erased regardless of the pin state because the FP pin is controlled internally within the chip.

(3) Protection by FENTRY bit

Flash E/W enable mode cannot be entered into unless the Flash Control Register 1 (FCNT1)'s FENTRY (flash mode entry) bit is set to "1". To set the FENTRY bit to "1", write "0" and then "1" in succession while the FP pin is high.

(4) Protection by a lock bit

Any block of internal flash memory can be protected by setting the lock bit provided for it to "0". That memory block is disabled against programming/erasing operation.

6.10 Notes on the Internal RAM

The following describes notes on the internal RAM

- When started by boot mode, internal RAM value is indefinite after started by boot mode in order to "Flash writing/erasing program" is transferred to internal RAM.

6.11 Notes on the Internal Flash Memory

The following describes precautions to be taken when programming/erasing the internal flash memory.

- When the internal flash memory is programmed or erased, a high voltage is generated internally. Because mode transitions during programming/erasing operation may cause the chip to break down, make sure the mode setting pin/power supply voltages do not fluctuate to prevent unintended changes of modes.
- If the system uses any pins that are to be used by a general-purpose programming/erasing tool, care must be taken to prevent adverse effects on the system when the tool is connected.
- If the internal flash memory needs to be protected while using a general-purpose programming/erasing tool, set any ID in the flash memory protect ID verification area (H’0000 0084 to H’0000 0093).
- If the internal flash memory does not need to be protected while using a general-purpose programming/erasing tool, fill the entire flash memory protect ID verification area (H’0000 0084 to H’0000 0093) with H’FF.
- If the Flash Status Register (FSTAT)’s each error status is to be cleared (initialized to H’80) by resetting the Flash Control Register 4 (FCNT4) FRESET bit, check to see that the Flash Status Register (FSTAT) FBUSY bit = "1" (ready) before clearing the error status.
- Before resetting the Flash Control Register 1 (FCNT1) FENTRY bit from "1" to "0", check to see that the Flash Status Register (FSTAT) FBUSY bit = "1" (ready).
- Do not clear the FENTRY bit if the Flash Control Register 1 (FCNT1) FENTRY bit = "1" and the Flash Status Register (FSTAT) FBUSY bit = "0" (being programmed or erased).
- When programming/erasing via JTAG, the flash memory can be programmed or erased regardless of the pin state because the FP pin is controlled internally within the chip.

This page is blank for reasons of layout.

CHAPTER 7

RESET

7.1 Outline of Reset

The microcomputer is reset by applying a "L" level signal to the RESET# input pin. The microcomputer is gotten out of a reset state by releasing the RESET# input back high, upon which the reset vector entry address is set in the Program Counter (PC) and the CPU starts executing from the reset vector entry.

7.2 Reset Operation

When a "L" level signal in width of more than 200 ns (a duration needed for noise cancellation) is applied to the RESET# pin, the microcomputer enters a reset state. At this time, the internal circuits (including the CPU) are reset. (For details about the pin state when reset, see Table 1.4.1, “Pin Assignments”)
When the RESET# input is returned "H", the internal circuits get out of a reset state 512-513 BCLK periods after that.

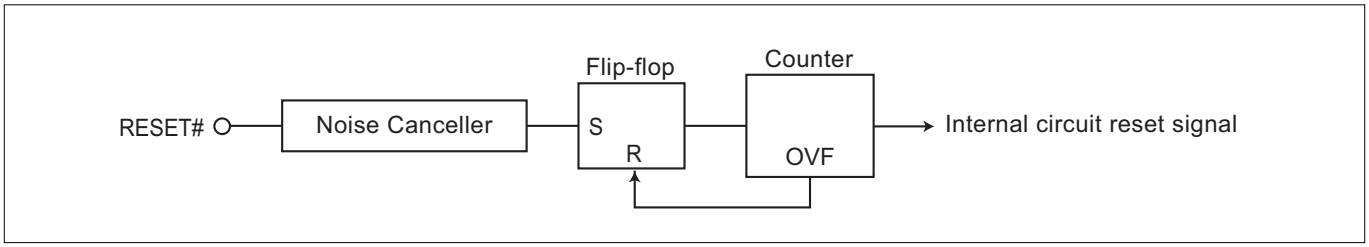


Figure 7.2.1 Reset Circuit

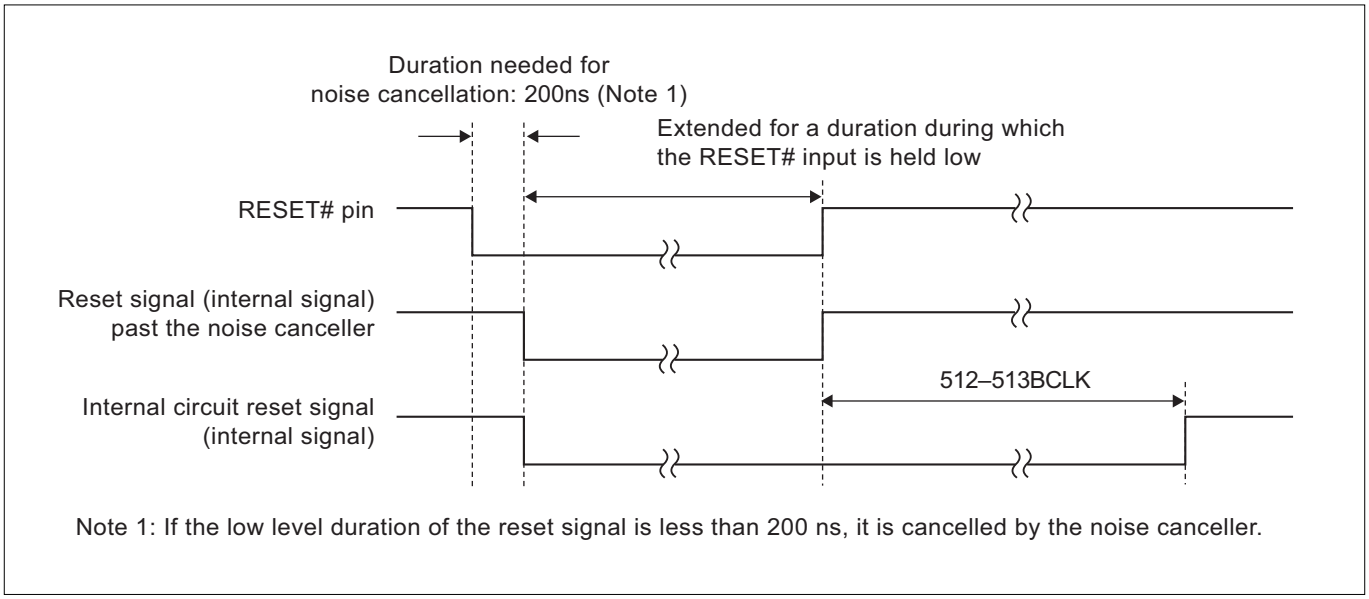


Figure 7.2.2 Reset Sequence

7.2.1 Reset at Power-on

When powering on the microcomputer, hold the RESET# signal input pin "L" level until the rated power supply voltage is reached and the microcomputer's internal x4 clock generator becomes oscillating stably.

7.2.2 Reset during Operation

To reset the microcomputer during operation, hold the RESET# signal input pin "L" level for more than 200 ns.

7.2.3 Reset Vector Relocation during Flash Programming

When the microcomputer is reset after entering boot mode, the reset vector entry address is moved to the boot program startup address. The boot program starts running after the reset state is deasserted. For details, see Section 6.5, "Programming the Internal Flash Memory."

7.3 Internal State Immediately after Exiting Reset

The table below lists the internal state of the microcomputer immediately after it has gotten out of a reset state. For details about the initial register state of each internal peripheral I/O, see each section in this manual in which the relevant internal peripheral I/O is described.

Table 7.3.1 Internal State Immediately after Exiting Reset

Register		State after Reset
PSW	(CR0)	B'0000 0000 0000 0000 ??00 000? 0000 0000 (BSM, BIE, BC bits = undefined)
CBR	(CR1)	H'0000 0000 (C bits = 0)
SPI	(CR2)	Undefined
SPU	(CR3)	Undefined
BPC	(CR6)	Undefined
PC		H'0000 0000 (Executed beginning with the address H'0000 0000) (Note 1)
R0–R15		Undefined
ACC (accumulator)		Undefined
RAM		Undefined when reset at power-on. (However, if the RAM is gotten out of reset after returning from backup mode, it retains the content it had before being reset.)

Note 1: When in boot mode, the CPU executes the boot program.

7.4 Things to Be Considered after Exiting Reset

- **Input/output ports**

  After exiting the reset state, the microcomputer's input/output ports are disabled against input in order to prevent current from flowing through the port. To use any ports in input mode, set the Port Input Special Function Control Register (PICNT) PIEN0 bit to enable them for input. For details, see Section 8.3, "Input/Output Port Related Registers."

CHAPTER 8

INPUT/OUTPUT PORTS AND PIN FUNCTIONS

8.1 Outline of Input/Output Ports

The 32176 has a total of 96 input/output ports from P0-P13, P15, P17 and P22 (except P5, which is reserved for future use). These input/output ports can be used as input or output ports by setting the respective direction registers.

Each input/output port is a dual-function or triple-function pin, sharing the pin with other internal peripheral I/O or external extension bus signal line. Pin functions are selected depending on the current operation mode or by setting the input/output port operation mode registers. (If any internal peripheral I/O has still another function, it is also necessary to set the register provided for that peripheral I/O.)

The microcomputer also has a port input function enable bit that can be used to prevent current from flowing into the input ports. This helps to simplify the software and hardware processing to be performed immediately after reset or during flash programming. Note that before any ports can be used in input mode, this port input function enable bit must be set accordingly.
The input/output ports are outlined below.

Table 8.1.1 Outline of Input/Output Ports

Item	Specification
Number of ports	Total 96 ports P0 : P00–P07 (8 ports) P1 : P10–P17 (8 ports) P2 : P20–P27 (8 ports) P3 : P30–P37 (8 ports) P4 : P41–P47 (7 ports) P6 : P61–P63 (3 ports) P7 : P70–P77 (8 ports) P8 : P82–P87 (6 ports) P9 : P93–P97 (5 ports) P10 : P100–P107 (8 ports) P11 : P110–P117 (8 ports) P12 : P124–P127 (4 ports) P13 : P130–P137 (8 ports) P15 : P150, P153 (2 ports) P17 : P174, P175 (2 ports) P22 : P220, P221, P225 (3 ports)
Port function	The input/output ports can individually be set for input or output mode using the direction control register provided for each input/output port. (However, P221 is a CAN input-only port.)
Pin function	Shared with peripheral I/O or external extension signals to serve dual-functions (or shared with two or more peripheral I/O functions to serve triple-functions)
Pin function selection	P0–P4, P225: Depends on the CPU operation mode (that is set by MOD0 and MOD1 pins). P6–P22: As set by each input/output port's operation mode register. (However, peripheral I/O pin functions are selected by peripheral I/O registers.)

Note: • P5, P14, P16, P18-P21 are nonexist.

8.2 Selecting Pin Functions

Each input/output port serves dual functions sharing the pin with other internal peripheral I/O or external extension bus signal line (or triple functions sharing the pin with two or more peripheral I/O functions). Pin functions are selected depending on the current operation mode or by setting the input/output port operation mode registers.

P0–P4 and P225, when the CPU is set to operate in external extension mode or processor mode, all are switched to serve as signal pins for external access. The CPU operation mode is determined depending on how the MOD0 and MOD1 pins are set (see the table below).

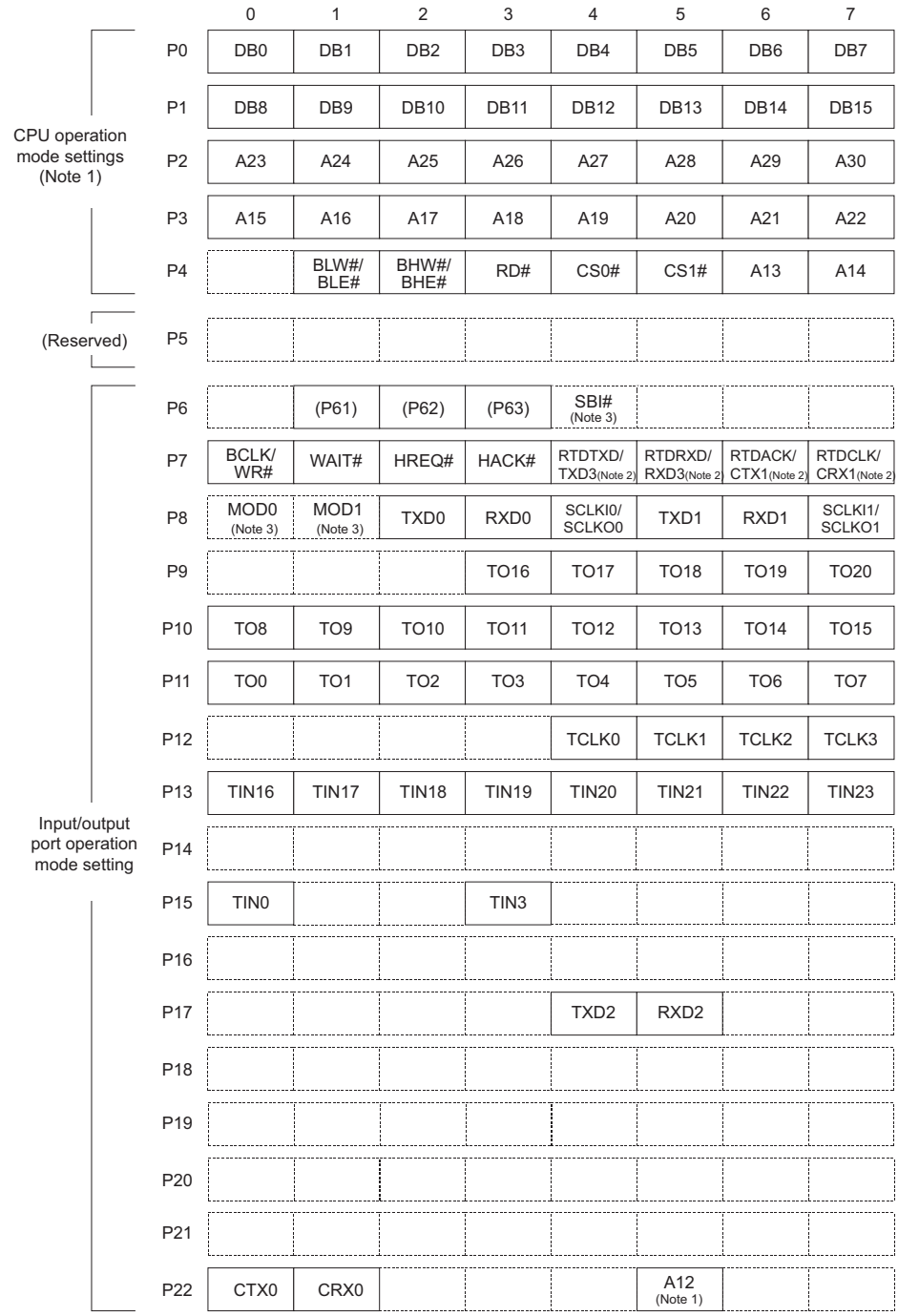
Table 8.2.1 CPU Operation Modes and P0–P4 and P225 Pin Functions

MOD0	MOD1	Operation Mode	P0–P4 and P225 Pin Function
VSS	VSS	Single-chip mode	Input/output port pin
VSS	VCCE	External extension mode	External extension signal pin
VCCE	VSS	Processor mode	
VCCE	VCCE	Reserved (use inhibited)	–

Note: • VCCE and VSS are connected to main power supply and GND, respectively.

Each input/output port has their functions switched between input/output port pins and internal peripheral I/O pins by setting the respective port operation mode registers. If any internal peripheral I/O has two or more pin functions, use the register provided for that peripheral I/O to select the desired pin function.

Note that FP and MOD1 pin settings during internal flash memory programming do not affect the pin functions.



Note 1: The pin function changes depending on the setting for MOD0 and MOD1 pins.
Note 2: These are triple-function pins. Their desired output function must be selected using the port peripheral function select register.
Note 3: These ports cannot be used for input/output port function. The SBI#, MOD0 and MOD1 pin input levels can be read from these ports.
Note: • P5, P14, P16, P18, P19, P20 and P21 are not provided.

Figure 8.2.1 Input/Output Ports and Pin Function Assignments

8.3 Input/Output Port Related Registers

The input/output port related registers included in the microcomputer consists of the port data register, port direction register and port operation mode register.
Note that P5 is reserved for future use. The tables below show an input/output port related register map.

Input/Output Port Related Register Map (1/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0700	P0 Data Register (P0DATA)	P1 Data Register (P1DATA)	8-7
H'0080 0702	P2 Data Register (P2DATA)	P3 Data Register (P3DATA)	8-7
H'0080 0704	P4 Data Register (P4DATA)	(Use inhibited area)	8-7
H'0080 0706	P6 Data Register (P6DATA)	P7 Data Register (P7DATA)	8-7
H'0080 0708	P8 Data Register (P8DATA)	P9 Data Register (P9DATA)	8-7
H'0080 070A	P10 Data Register (P10DATA)	P11 Data Register (P11DATA)	8-7
H'0080 070C	P12 Data Register (P12DATA)	P13 Data Register (P13DATA)	8-7
H'0080 070E	(Use inhibited area)	P15 Data Register (P15DATA)	8-7
H'0080 0710	(Use inhibited area)	P17 Data Register (P17DATA)	8-7
H'0080 0712	(Use inhibited area)	(Use inhibited area)	
H'0080 0714	(Use inhibited area)	(Use inhibited area)	
H'0080 0716	P22 Data Register (P22DATA)	(Use inhibited area)	8-7
⏐	(Use inhibited area)		
H'0080 0720	P0 Direction Register (P0DIR)	P1 Direction Register (P1DIR)	8-8
H'0080 0722	P2 Direction Register (P2DIR)	P3 Direction Register (P3DIR)	8-8
H'0080 0724	P4 Direction Register (P4DIR)	(Use inhibited area)	8-8
H'0080 0726	P6 Direction Register (P6DIR)	P7 Direction Register (P7DIR)	8-8
H'0080 0728	P8 Direction Register (P8DIR)	P9 Direction Register (P9DIR)	8-8
H'0080 072A	P10 Direction Register (P10DIR)	P11 Direction Register (P11DIR)	8-8
H'0080 072C	P12 Direction Register (P12DIR)	P13 Direction Register (P13DIR)	8-8
H'0080 072E	(Use inhibited area)	P15 Direction Register (P15DIR)	8-8
H'0080 0730	(Use inhibited area)	P17 Direction Register (P17DIR)	8-8
H'0080 0732	(Use inhibited area)	(Use inhibited area)	
H'0080 0734	(Use inhibited area)	(Use inhibited area)	
H'0080 0736	P22 Direction Register (P22DIR)	(Use inhibited area)	8-8

Input/Output Port Related Register Map (2/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0744	(Use inhibited area)	Port Input Special Function Control Register (PICNT)	8-15
H'0080 0746	(Use inhibited area)	P7 Operation Mode Register (P7MOD)	8-9
H'0080 0748	P8 Operation Mode Register (P8MOD)	P9 Operation Mode Register (P9MOD)	8-9 8-10
H'0080 074A	P10 Operation Mode Register (P10MOD)	P11 Operation Mode Register (P11MOD)	8-10 8-11
H'0080 074C	P12 Operation Mode Register (P12MOD)	P13 Operation Mode Register (P13MOD)	8-11 8-12
H'0080 074E	(Use inhibited area)	P15 Operation Mode Register (P15MOD)	8-12
H'0080 0750	(Use inhibited area)	P17 Operation Mode Register (P17MOD)	8-13
H'0080 0752	(Use inhibited area)	(Use inhibited area)	
H'0080 0754	(Use inhibited area)	(Use inhibited area)	
H'0080 0756	P22 Operation Mode Register (P22MOD)	(Use inhibited area)	8-13
|	(Use inhibited area)		
H'0080 0760	Port Group 0,1 Input Level Setting Register (PG01LEV)	Port Group 3 Input Level Setting Register (PG3LEV)	8-19
H'0080 0762	Port Group 4,5 Input Level Setting Register (PG45LEV)	Port Group 6,7 Input Level Setting Register (PG67LEV)	8-19
H'0080 0764	Port Group 8 Input Level Setting Register (PG8LEV)	(Use inhibited area)	8-19
H'0080 0766	(Use inhibited area)	P7 Peripheral Function Select Register (P7SMOD)	8-14

8.3.1 Port Data Registers

P0 Data Register (P0DATA)	<Address: H'0080 0700>
P1 Data Register (P1DATA)	<Address: H'0080 0701>
P2 Data Register (P2DATA)	<Address: H'0080 0702>
P3 Data Register (P3DATA)	<Address: H'0080 0703>
P4 Data Register (P4DATA)	<Address: H'0080 0704>
P6 Data Register (P6DATA)	<Address: H'0080 0706>
P7 Data Register (P7DATA)	<Address: H'0080 0707>
P8 Data Register (P8DATA)	<Address: H'0080 0708>
P9 Data Register (P9DATA)	<Address: H'0080 0709>
P10 Data Register (P10DATA)	<Address: H'0080 070A>
P11 Data Register (P11DATA)	<Address: H'0080 070B>
P12 Data Register (P12DATA)	<Address: H'0080 070C>
P13 Data Register (P13DATA)	<Address: H'0080 070D>
P15 Data Register (P15DATA)	<Address: H'0080 070F>
P17 Data Register (P17DATA)	<Address: H'0080 0711>
P22 Data Register (P22DATA)	<Address: H'0080 0716>

b0 (b8	1 9	2 10	3 11	4 12	5 13	6 14	b7 b15)
Pn0DT	Pn1DT	Pn2DT	Pn3DT	Pn4DT	Pn5DT	Pn6DT	Pn7DT
?	?	?	?	?	?	?	?

n = 0–13, 15, 17, 22 (not including P5)

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0(b8)	Pn0DT (Port Pn0 data bit)	<At read>	R	W
1(b9)	Pn1DT (Port Pn1 data bit)	Depends on how the Port Direction Register is set		
2(b10)	Pn2DT (Port Pn2 data bit)	If direction bit = "0" (input mode)		
3(b11)	Pn3DT (Port Pn3 data bit)	0: Port input pin = "L"		
4(b12)	Pn4DT (Port Pn4 data bit)	1: Port input pin = "H"		
5(b13)	Pn5DT (Port Pn5 data bit)	If direction bit = "1" (output mode) (Note 1)		
6(b14)	Pn6DT (Port Pn6 data bit)	0: Port output latch = "0" / Port pin level = "L"		
7(b15)	Pn7DT (Port Pn7 data bit)	1: Port output latch = "1" / Port pin level = "H"		
		<At write>		
		Write to the port output latch		

Note 1: To select the port data to read, use the Port Input Special Function Control Register's port input data select bit (PISEL).

Notes: • Following bits are not provided (read as "0", writing has no effect):
P40, P60, P65–P67, P90–P92, P120–P123, P151, P152, P154–P157, P170–P173, P176, P177, P222–P224, P226, P227

• The SBI# pin level can be read out by reading the P64DT bit. Writing to the P64DT bit has no effect.

• The MOD0 and MOD1 pin levels can be read out by reading the P80DT and P81DT bits, respectively. Writing to the P80DT and P81DT bits has no effect.

• P221 is an input-only port. Writing to the P221DT bit has no effect.

8.3.2 Port Direction Registers

P0 Direction Register (P0DIR)	<Address: H'0080 0720>
P1 Direction Register (P1DIR)	<Address: H'0080 0721>
P2 Direction Register (P2DIR)	<Address: H'0080 0722>
P3 Direction Register (P3DIR)	<Address: H'0080 0723>
P4 Direction Register (P4DIR)	<Address: H'0080 0724>
P6 Direction Register (P6DIR)	<Address: H'0080 0726>
P7 Direction Register (P7DIR)	<Address: H'0080 0727>
P8 Direction Register (P8DIR)	<Address: H'0080 0728>
P9 Direction Register (P9DIR)	<Address: H'0080 0729>
P10 Direction Register (P10DIR)	<Address: H'0080 072A>
P11 Direction Register (P11DIR)	<Address: H'0080 072B>
P12 Direction Register (P12DIR)	<Address: H'0080 072C>
P13 Direction Register (P13DIR)	<Address: H'0080 072D>
P15 Direction Register (P15DIR)	<Address: H'0080 072F>
P17 Direction Register (P17DIR)	<Address: H'0080 0731>
P22 Direction Register (P22DIR)	<Address: H'0080 0736>

b0	1	2	3	4	5	6	b7
(b8	9	10	11	12	13	14	b15)
Pn0DIR	Pn1DIR	Pn2DIR	Pn3DIR	Pn4DIR	Pn5DIR	Pn6DIR	Pn7DIR
0	0	0	0	0	0	0	0

n = 0–13, 15, 17, 22 (not including P5)

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0(b8)	Pn0DR (Port Pn0 direction bit)	0: Input mode	R	W
1(b9)	Pn1DR (Port Pn1 direction bit)	1: Output mode		
2(b10)	Pn2DR (Port Pn2 direction bit)			
3(b11)	Pn3DR (Port Pn3 direction bit)			
4(b12)	Pn4DR (Port Pn4 direction bit)			
5(b13)	Pn5DR (Port Pn5 direction bit)			
6(b14)	Pn6DR (Port Pn6 direction bit)			
7(b15)	Pn7DR (Port Pn7 direction bit)			

Notes: • Following bits are not provided (read as 0, writing has no effect):
P40, P60, P64–P67, P80, P81, P90–P92, P120–P123, P151, P152, P154–P157, P170–P173, P176, P177, P221, P222-–P224, P226, P227
• All ports are set for input mode upon exiting the reset state.

8.3.3 Port Operation Mode Registers

P7 Operation Mode Register (P7MOD) <Address: H'0080 0747>

b8	9	10	11	12	13	14	b15
P70MOD	P71MOD	P72MOD	P73MOD	P74MOD	P75MOD	P76MOD	P77MOD
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	P70MOD Port P70 operation mode bit (Note 2)	0: P70 1: BCLK/WR#	R	W
9	P71MOD Port P71 operation mode bit	0: P71 1: WAIT#	R	W
10	P72MOD Port P72 operation mode bit	0: P72 1: HREQ#	R	W
11	P73MOD Port P73 operation mode bit	0: P73 1: HACK#	R	W
12	P74MOD Port P74 operation mode bit	0: P74 1: RTDTXD/TXD3 (Note 1)	R	W
13	P75MOD Port P75 operation mode bit	0: P75 1: RTDRXD/RXD3 (Note 1)	R	W
14	P76MOD Port P76 operation mode bit	0: P76 1: RTDACK/CTX1 (Note 1)	R	W
15	P77MOD Port P77 operation mode bit	0: P77 1: RTDCLK/CRX1 (Note 1)	R	W

Note 1: These functions are selected using the P7 Peripheral Function Select Register.

Note 2: When BUSMOD bit of the BUSMODC register is set to 1 (byte enable separate mode) in the external extension mode, regardless of setting P7MOD register , P70/BCLK/WR# pin becomes a pin to output WR# signal.

P8 Operation Mode Register (P8MOD) <Address: H'0080 0748>

b0	1	2	3	4	5	6	b7
		P82MOD	P83MOD	P84MOD	P85MOD	P86MOD	P87MOD
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0,1	No function assigned. Fix to "0".		0	0
2	P82MOD Port P82 operation mode bit	0: P82 1: TXD0	R	W
3	P83MOD Port P83 operation mode bit	0: P83 1: RXD0	R	W
4	P84MOD Port P84 operation mode bit	0: P84 1: SCLKI0/SCLKO0	R	W
5	P85MOD Port P85 operation mode bit	0: P85 1: TXD1	R	W
6	P86MOD Port P86 operation mode bit	0: P86 1: RXD1	R	W
7	P87MOD Port P87 operation mode bit	0: P87 1: SCLKI1/SCLKO1	R	W

Note: • Ports P80 and P81 are nonexistent.

P9 Operation Mode Register (P9MOD) <Address: H'0080 0749>

b8	9	10	11	12	13	14	b15
0	0	0	P93MOD 0	P94MOD 0	P95MOD 0	P96MOD 0	P97MOD 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	P93MOD Port P93 operation mode bit	0: P93 1: TO16	R	W
12	P94MOD Port P94 operation mode bit	0: P94 1: TO17	R	W
13	P95MOD Port P95 operation mode bit	0: P95 1: TO18	R	W
14	P96MOD Port P96 operation mode bit	0: P96 1: TO19	R	W
15	P97MOD Port P97 operation mode bit	0: P97 1: TO20	R	W

Note: • Ports P90–P92 are nonexistent.

P10 Operation Mode Register (P10MOD) <Address: H'0080 074A>

b0	1	2	3	4	5	6	b7
P100MOD 0	P101MOD 0	P102MOD 0	P103MOD 0	P104MOD 0	P105MOD 0	P106MOD 0	P107MOD 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	P100MOD Port P100 operation mode bit	0: P100 1: TO8	R	W
1	P101MOD Port P101 operation mode bit	0: P101 1: TO9	R	W
2	P102MOD Port P102 operation mode bit	0: P102 1: TO10	R	W
3	P103MOD Port P103 operation mode bit	0: P103 1: TO11	R	W
4	P104MOD Port P104 operation mode bit	0: P104 1: TO12	R	W
5	P105MOD Port P105 operation mode bit	0: P105 1: TO13	R	W
6	P106MOD Port P106 operation mode bit	0: P106 1: TO14	R	W
7	P107MOD Port P107 operation mode bit	0: P107 1: TO15	R	W

P11 Operation Mode Register (P11MOD) <Address: H'0080 074B>

b8	9	10	11	12	13	14	b15
P110MOD	P111MOD	P112MOD	P113MOD	P114MOD	P115MOD	P116MOD	P117MOD
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	P110MOD Port P110 operation mode bit	0: P110 1: TO0	R	W
9	P111MOD Port P111 operation mode bit	0: P111 1: TO1	R	W
10	P112MOD Port P112 operation mode bit	0: P112 1: TO2	R	W
11	P113MOD Port P113 operation mode bit	0: P113 1: TO3	R	W
12	P114MOD Port P114 operation mode bit	0: P114 1: TO4	R	W
13	P115MOD Port P115 operation mode bit	0: P115 1: TO5	R	W
14	P116MOD Port P116 operation mode bit	0: P116 1: TO6	R	W
15	P117MOD Port P117 operation mode bit	0: P117 1: TO7	R	W

P12 Operation Mode Register (P12MOD) <Address: H'0080 074C>

b0	1	2	3	4	5	6	b7
0	0	0	0	P124MOD 0	P125MOD 0	P126MOD 0	P127MOD 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–3	No function assigned. Fix to "0".		0	0
4	P124MOD Port P124 operation mode bit	0: P124 1: TCLK0	R	W
5	P125MOD Port P125 operation mode bit	0: P125 1: TCLK1	R	W
6	P126MOD Port P126 operation mode bit	0: P126 1: TCLK2	R	W
7	P127MOD Port P127 operation mode bit	0: P127 1: TCLK3	R	W

Note: • Ports P120–P123 are nonexistent.

P13 Operation Mode Register (P13MOD) <Address: H'0080 074D>

b8	9	10	11	12	13	14	b15
P130MOD 0	P131MOD 0	P132MOD 0	P133MOD 0	P134MOD 0	P135MOD 0	P136MOD 0	P137MOD 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	P130MOD Port P130 operation mode bit	0: P130 1: TIN16	R	W
9	P131MOD Port P131 operation mode bit	0: P131 1: TIN17	R	W
10	P132MOD Port P132 operation mode bit	0: P132 1: TIN18	R	W
11	P133MOD Port P133 operation mode bit	0: P133 1: TIN19	R	W
12	P134MOD Port P134 operation mode bit	0: P134 1: TIN20	R	W
13	P135MOD Port P135 operation mode bit	0: P135 1: TIN21	R	W
14	P136MOD Port P136 operation mode bit	0: P136 1: TIN22	R	W
15	P137MOD Port P137 operation mode bit	0: P137 1: TIN23	R	W

P15 Operation Mode Register (P15MOD) <Address: H'0080 074F>

b8	9	10	11	12	13	14	b15
P150MOD 0	0	0	P153MOD 0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	P150MOD Port P150 operation mode bit	0: P150 1: TIN0	R	W
9, 10	No function assigned. Fix to "0".		0	0
11	P153MOD Port P153 operation mode bit	0: P153 1: TIN3	R	W
12-15	No function assigned. Fix to "0".		0	0

Note : • Ports P151, P152 and P154–P157 are nonexistent.

P17 Operation Mode Register (P17MOD) <Address: H'0080 0751>

b8	9	10	11	12	13	14	b15
0	0	0	0	P174MOD 0	P175MOD 0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–11	No function assigned. Fix to "0".		0	0
12	P174MOD Port P174 operation mode bit	0: P174 1: TXD2	R	W
13	P175MOD Port P175 operation mode bit	0: P175 1: RXD2	R	W
14, 15	No function assigned. Fix to "0".		0	0

Notes: • Ports P170–P173, P176 and P177 are nonexistent.

P22 Operation Mode Register (P22MOD) <Address: H'0080 0756>

b0	1	2	3	4	5	6	b7
P220MOD 0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	P220MOD Port P220 operation mode bit	0: P220 1: CTX0	R	W
1– 7	No function assigned. Fix to "0".		0	0

Note 1: Port P221 is a CAN input-only pin.

Note 2: The pin function for P225 changes depending on the MOD0 and MOD1pin settings.

Note 3: Ports P222–P224, P226 and P227 are nonexistent.

8.3.4 Port Peripheral Function Select Register

P7 Peripheral Function Select Register (P7SMOD) <Address: H'0080 0767>

b8	9	10	11	12	13	14	b15
0	0	0	0	P74SMOD 0	P75SMOD 0	P76SMOD 0	P77SMOD 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–11	No function assigned. Fix to "0".		0	0
12	P74SMOD Port P74 peripheral function select bit	0: RTDTXD 1: TXD3	R	W
13	P75SMOD Port P75 peripheral function select bit	0: RTDRXD 1: RXD3	R	W
14	P76SMOD Port P76 peripheral function select bit	0: RTDACK 1: CTX1	R	W
15	P77SMOD Port P77 peripheral function select bit	0: RTDCLK 1: CRX1	R	W

The P7 Peripheral Function Select Register is used to select a peripheral function when the corresponding bit in the P7 Operation Mode Register = "1".
To use this register, first rewrite it when the P7 Operation Mode Register = "0", and then set the P7 Operation Mode Register to "1" to enable peripheral functions.

8.3.5 Port Input Special Function Control Register

Port Input Special Function Control Register (PICNT) <Address: H'0080 0745>

b8	9	10	11	12	13	14	b15
0	0	0	XSTAT 0	0	0	PISEL 0	PIEN0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	XSTAT XIN oscillation status bit	0: XIN oscillating 1: XIN inactive	R (Note 1)	
12, 13	No function assigned. Fix to "0".		0	0
14	PISEL Port input data select bit	0: Content of port output latch 1: Port pin level	R	W
15	PIEN0 Port input enable bit	0: Disable input 1: Enable input	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

(1) XSTAT (XIN oscillation status) bit (Bit 11)

1) Conditions under which XSTAT is set to "1"

XSTAT is set to "1" upon detecting that XIN oscillation has stopped. When XIN remains at the same level for a predetermined time (3 BCLK periods up to 4 BCLK periods), XIN oscillation is assumed to have stopped. When operating normally, XIN changes state ("H" or "L") once every BCLK period.

2) Conditions under which XSTAT is cleared to "0"

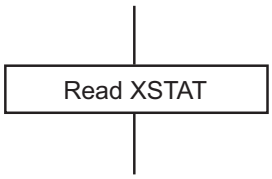
XSTAT is cleared to "0" by a system reset or by writing "0". If XSTAT is cleared at the same time it is set in (1) above, the former has priority. Writing "1" to XSTAT is ignored.

3) Method for using XSTAT to detect XIN oscillation stoppage

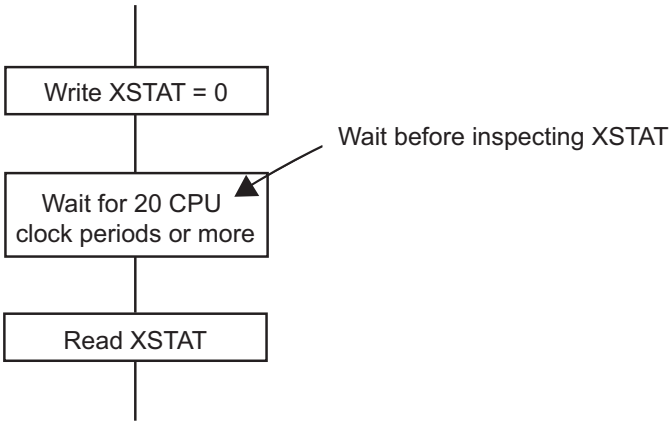
Because the M32R/ECU internally contains a PLL, the internal clock remains active even when XIN oscillation has stopped.
By reading XSTAT without clearing it never once after reset, it is possible to know whether XIN has ever stopped since the reset signal was deasserted. Similarly, by reading XSTAT after clearing it by writing "0", it is possible to know the current oscillating status of XIN. However, there must be an interval of at least 10 BCLK periods (20 CPU clock periods) between read and write.

Pay attention about processing when XSTAT bit is set to "1," make double check after clearing XSTAT bit etc.

(1) To know whether XIN oscillation has ever stopped after being reset



(2) To know the current status of XIN oscillation



Note. Pay attention about processing when XSTAT bit is set to "1," make double check after clearing XSTAT bit etc.

Figure 8.3.1 Procedure for Setting XSTAT

(2) PISEL (Port input data select) bit (Bit 14)

When the Port Direction Register is set for output, this bit selects the target data to be read from the Port Data Register. At this time, this bit is unaffected by the Port Operation Mode Register.

Table 8.3.1 PISEL Bit Settings and the Target Data To Be Read from the Port Data Register

Direction Register	PISEL Settings	Target Data to Be Read
0 (input)	0/1	Port pin level
1 (output)	0	Port output latch
	1	Port pin level

(3) PIEN0 (Port input enable) bit (Bit 15)

This bit is used to prevent current from flowing into the port input pins.
Because the input/output ports are disabled against input after reset, if any ports need to be used in input mode they must be enabled for input by setting this bit to "1".
When disabled against input, the input/output ports are in a state equivalent to a situation where the pin has a "L" level input applied. Consequently, if a peripheral input function is selected for any port while disabled against input by using the Port Operation Mode Register, the port may operate unexpectedly due to the "L" level input on it.

The following shows the procedure for selecting a peripheral input function.

(1) Enable the port for input when its pin level is valid ("H" or "L")
(2) Select a function using the port operation mode bit

During boot mode, the pins shared with serial interface functions are enabled for input and can therefore be protected against current flowing in from the pins other than serial interface functions during flash programming by clearing PIEN0.
The table below lists the pins that can be controlled by the PIEN0 bit in each operation mode.

Table 8.3.2 Pins Controllable by PIEN0 Bit

Mode Name	Controllable Pins	Uncontrolled Pins
Single-chip	P00–P07, P10–P17, P20–P27 P30–P37, P41–P47, P61–P63 P70–P77, P82–P87, P93–P97 P100–P107, P110–P117, P124–P127 P130–P137, 150, P153, P174, P175 P220, P225	P221, FP, SBI#, MOD0, MOD1, MOD2, RESET#
External extension Microprocessor	P61–P63, P70–P77, P82–P87 P93–P97, P100–P107, P110–P117 P124–P127, P130–P137 P150, P153, P174, P175, P220	P00–P07, P10–P17 P20–P27, P30–P37 P41–P47, P221, P225 FP, SBI#, MOD0, MOD1, MOD2, RESET#
Boot (single-chip)	P00–P07, P10–P17, P20–P27 P30–P37, P41–P47, P61–P63 P67, P70–P77, P93–P97 P100–P107, P110–P117, P124–P127 P130–P137, P150, P153, P220, P225	P82–P87, P174, P175 P221, FP, SBI#, MOD0, MOD1, MOD2, RESET#

8.4 Port Input Level Switching Function

The port input level switching function allows the port threshold to be switched to one of three voltage levels (with or without Schmitt as selected) in units of the following port group. This can be set to the following registers in units of group.

Group 0: P00–P07, P10–P17, P20–P27, P30–P37, P41–P47, P70–P73, P225
Group 1: P82–P87, P174–P177
Group 3: P93–P97, P110–P117
Group 4: P124–P127
Group 5: P61–P63, SBI#
Group 6: P74–P77, P100–P107
Group 7: P220, P221
Group 8: P130–P137, P150–P153

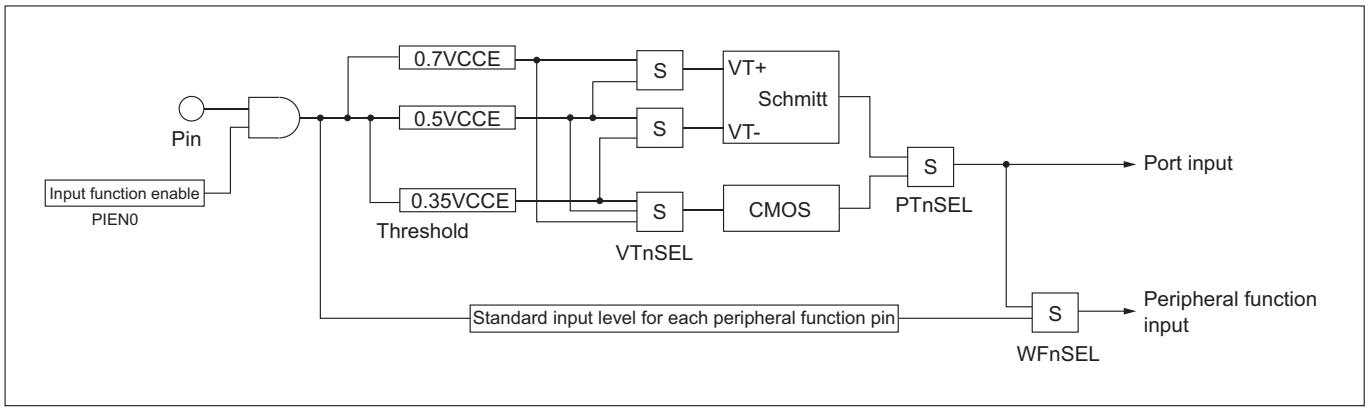


Figure 8.4.1 Port Level Switching Function

Port Group 0,1 Input Level Setting Register (PG01LEV) <Address: H'0080 0760>

b0	1	2	3	4	5	6	b7
WF0SEL	PT0SEL	VT0SEL0	VT0SEL1	WF1SEL	PT1SEL	VT1SEL0	VT1SEL1
0	0	0	1	0	0	0	1

Port Group 3 Input Level Setting Register (PG3LEV) <Address: H'0080 0761>

b8	9	10	11	12	13	14	b15
0	0	0	0	WF3SEL	PT3SEL	VT3SEL0	VT3SEL1
				0	0	0	1

Note: • The PG3LEV register bits 8–11 have no functions assigned.

Port Group 4,5 Input Level Setting Register (PG45LEV) <Address: H'0080 0762>

b0	1	2	3	4	5	6	b7
WF4SEL	PT4SEL	VT4SEL0	VT4SEL1	WF5SEL	PT5SEL	VT5SEL0	VT5SEL1
0	0	0	1	0	0	0	1

Port Group 6,7 Input Level Setting Register (PG67LEV) <Address: H'0080 0763>

b8	9	10	11	12	13	14	b15
WF6SEL	PT6SEL	VT6SEL0	VT6SEL1	WF7SEL	PT7SEL	VT7SEL0	VT7SEL1
0	0	0	1	0	0	0	1

Port Group 8 Input Level Setting Register (PG8LEV) <Address: H'0080 0764>

b0	1	2	3	4	5	6	b7
WF8SEL	PT8SEL	VT8SEL0	VT8SEL1	0	0	0	0
0	0	0	1				

Note: • The PG8LEV register bits 4–7 have no functions assigned.

<Upon exiting reset: :H'11, H'01, H'10> (Note 2)

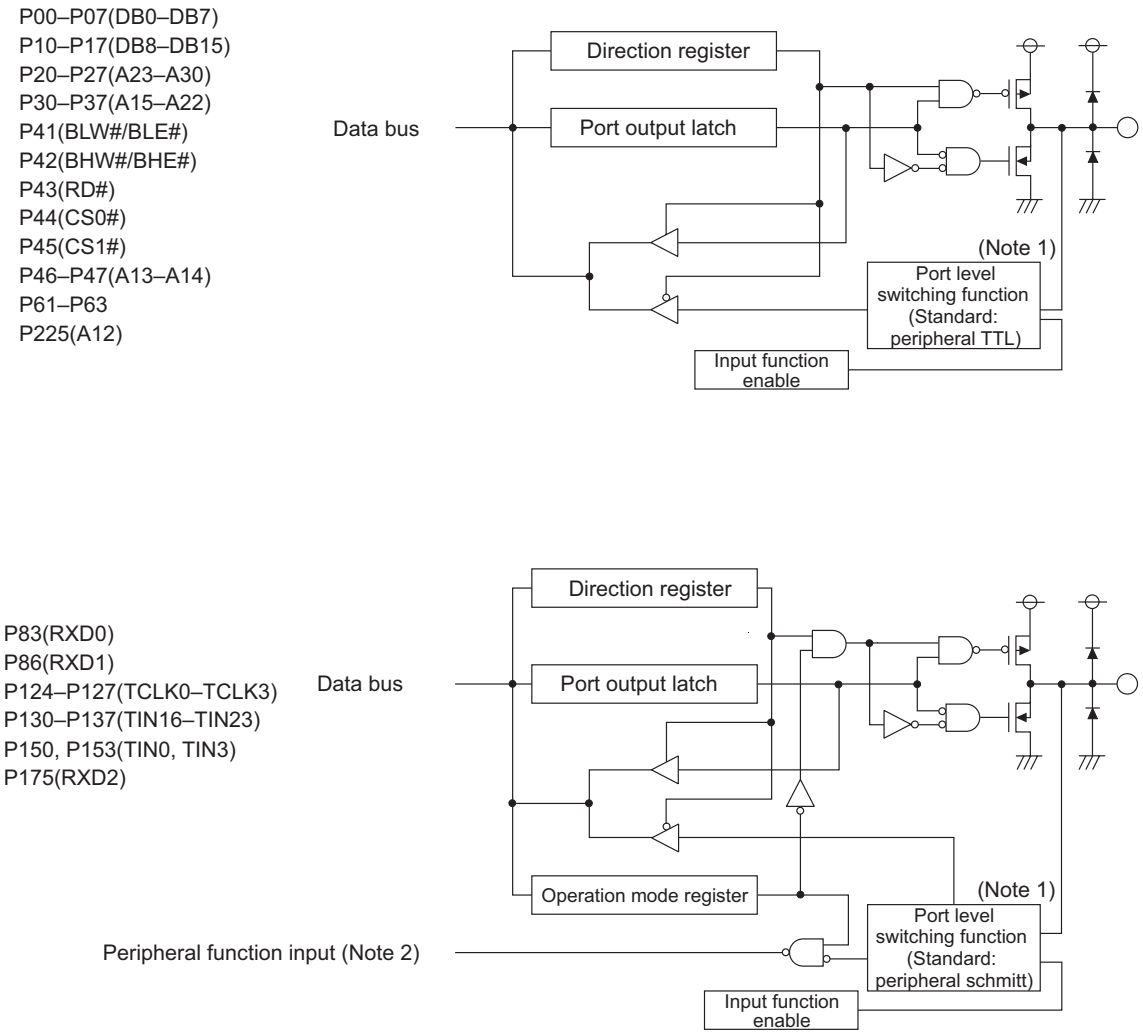
b	Bit Name	Function	R	W
0(8)	WFnSEL (Note 1) Group n dual-function input select bit	0: Select standard input for each pin 1: Select threshold switching function	R	W
1–3 (9–11)	PTnSEL (Group n port input select bit) VTnSEL0, VTnSEL1 (Group n input threshold select bit)	000 : Input CMOS, Select 0.35VCCE 001 : Input CMOS, Select 0.50VCCE 010 : Input CMOS, Select 0.70VCCE 011 : Settings inhibited 100 : Schmitt input , VT += 0.50VCCE, VT -= 0.35VCCE 101 : Settings inhibited 110 : Schmitt input , VT += 0.70VCCE, VT -= 0.35VCCE 111 : Schmitt input , VT += 0.70VCCE, VT -= 0.50VCCE	R	W
4(12)	WFnSEL (Note 1) Group n dual-function input select bit	0: Select standard input for each pin 1: Select threshold switching function	R	W
5–7 (13 –15)	PTnSEL (Group n port input select bit) VTnSEL0, VTnSEL1 (Group n input threshold select bit)	000 : Input CMOS, Select 0.35VCCE 001 : Input CMOS, Select 0.50VCCE 010 : Input CMOS, Select 0.70VCCE 011 : Settings inhibited 100 : Schmitt input , VT += 0.50VCCE, AVT -= 0.35VCCE 101 : Settings inhibited 110 : Schmitt input , VT += 0.70VCCE, AVT -= 0.35VCCE 111 : Schmitt input , VT += 0.70VCCE, AVT -= 0.50VCCE	R	W

Note 1.When the multipurpose port function pin is selected (Set bit corresponding Px operation mode register(PxMOD) to "0".), setting value for WFnSEL is invalid and threshold switch function is effective.

Note 2. Upon exiting reset, VTnSEL1 bit value is "1" and the other bit is set to "0" .

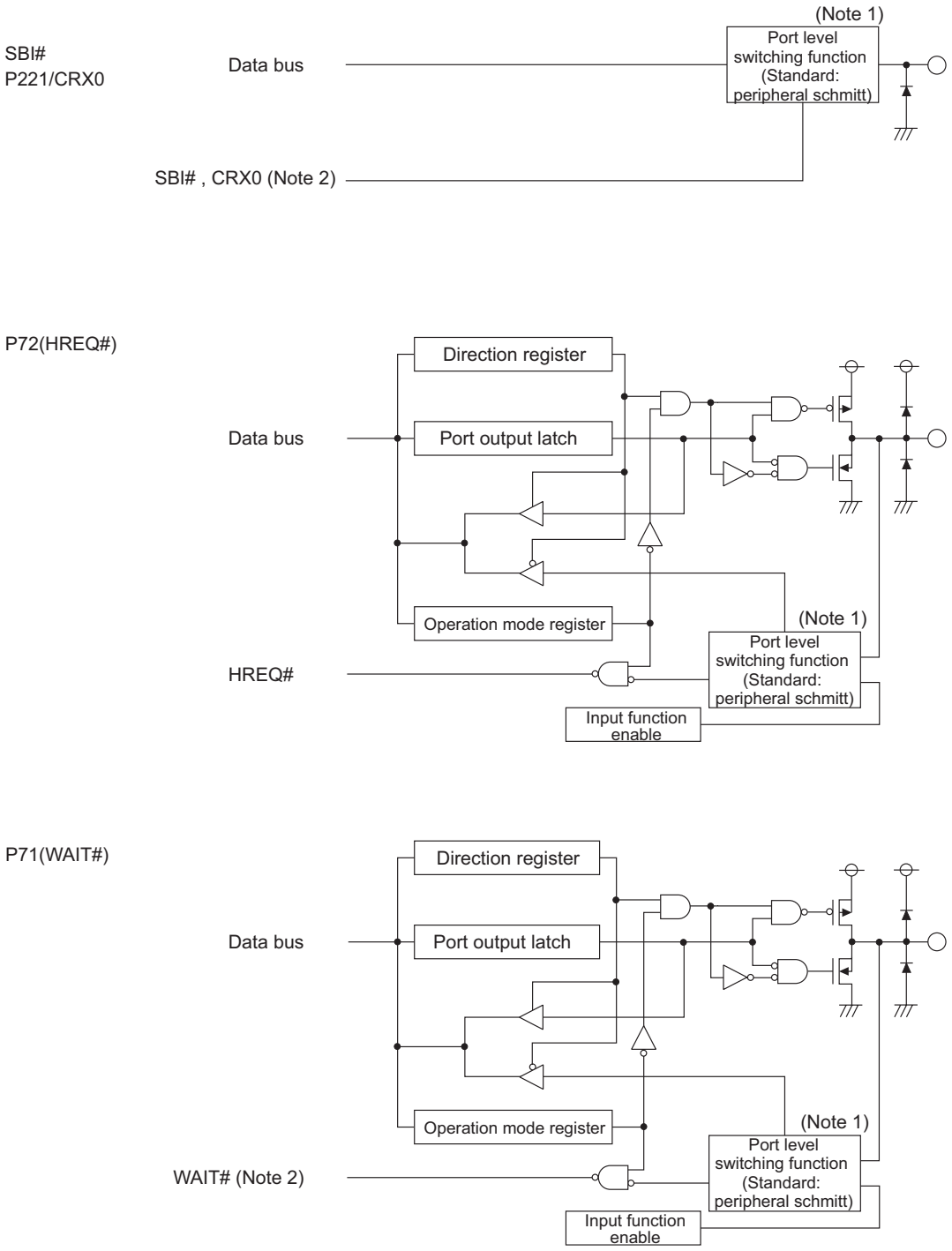
8.5 Port Peripheral Circuits

Figures 8.5.1 through 8.5.5 show the peripheral circuit diagrams of the input/output ports described in the preceding pages.



Note 1: For details about the port level switching function, see Section 8.4, "Port Input Level Switching Function."
Note 2: "H" level is entered to the peripheral function input when it is set to the general-purpose port in the operation mode register.
Notes: • During external extension and processor modes, P00-P07, P10-P17, P20-P27, P30-P37, P41-P47, and P225 are external bus interface control signal pins, but their functional description in this block diagram is omitted.
• The circle denotes a pin.
• The symbol ⊣◂ denotes a parasitic diode. Make sure the voltage applied to each pin does not exceed the VCCE voltage.
• The input capacitance of each pin is approximately 10 pF.

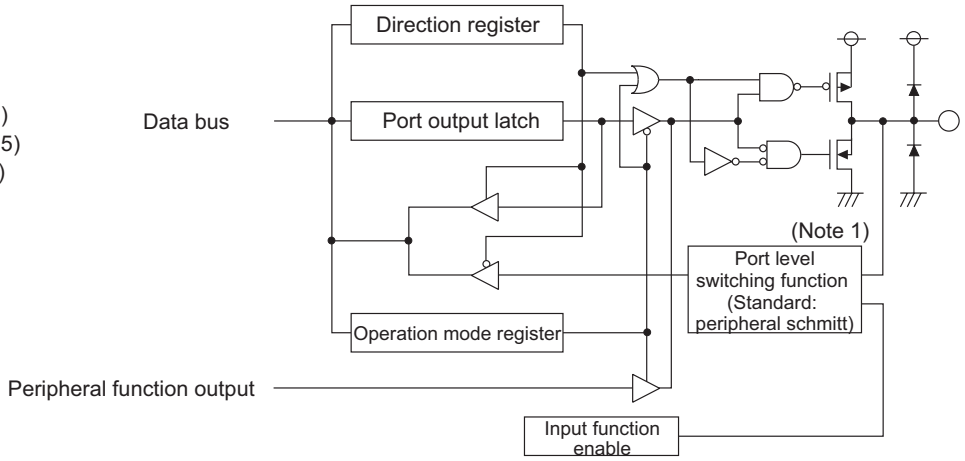
Figure 8.5.1 Port Peripheral Circuit Diagram (1)



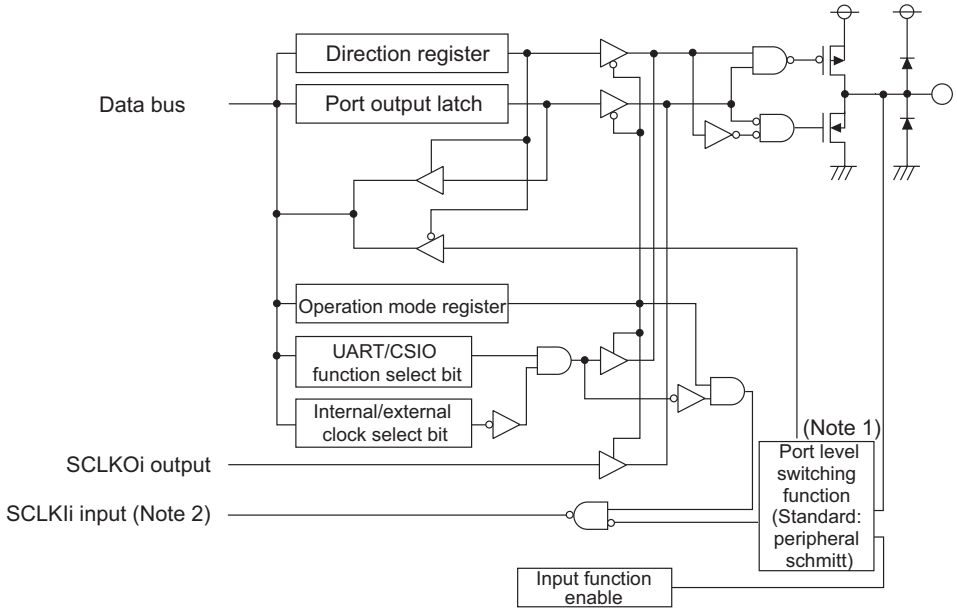
Note 1: For details about the port level switching function, see Section 8.4, "Port Input Level Switching Function."
Note 2: "H" level is entered to the peripheral function input when it is set to the general-purpose port in the operation mode register.
Notes: • The circle denotes a pin.
• The symbol ⊣◂ denotes a parasitic diode. Make sure the voltage applied to each pin does not exceed the VCCE voltage.
• The input capacitance of each pin is approximately 10 pF.

Figure 8.5.2 Port Peripheral Circuit Diagram (2)

P70(BCLK/WR#)
P73(HACK#)
P82(TXD0)
P85(TXD1)
P93–P97(TO16–TO20)
P100–P107(TO8–TO15)
P110–P117(TO0–TO7)
P174(TXD2)
P220(CTX0)

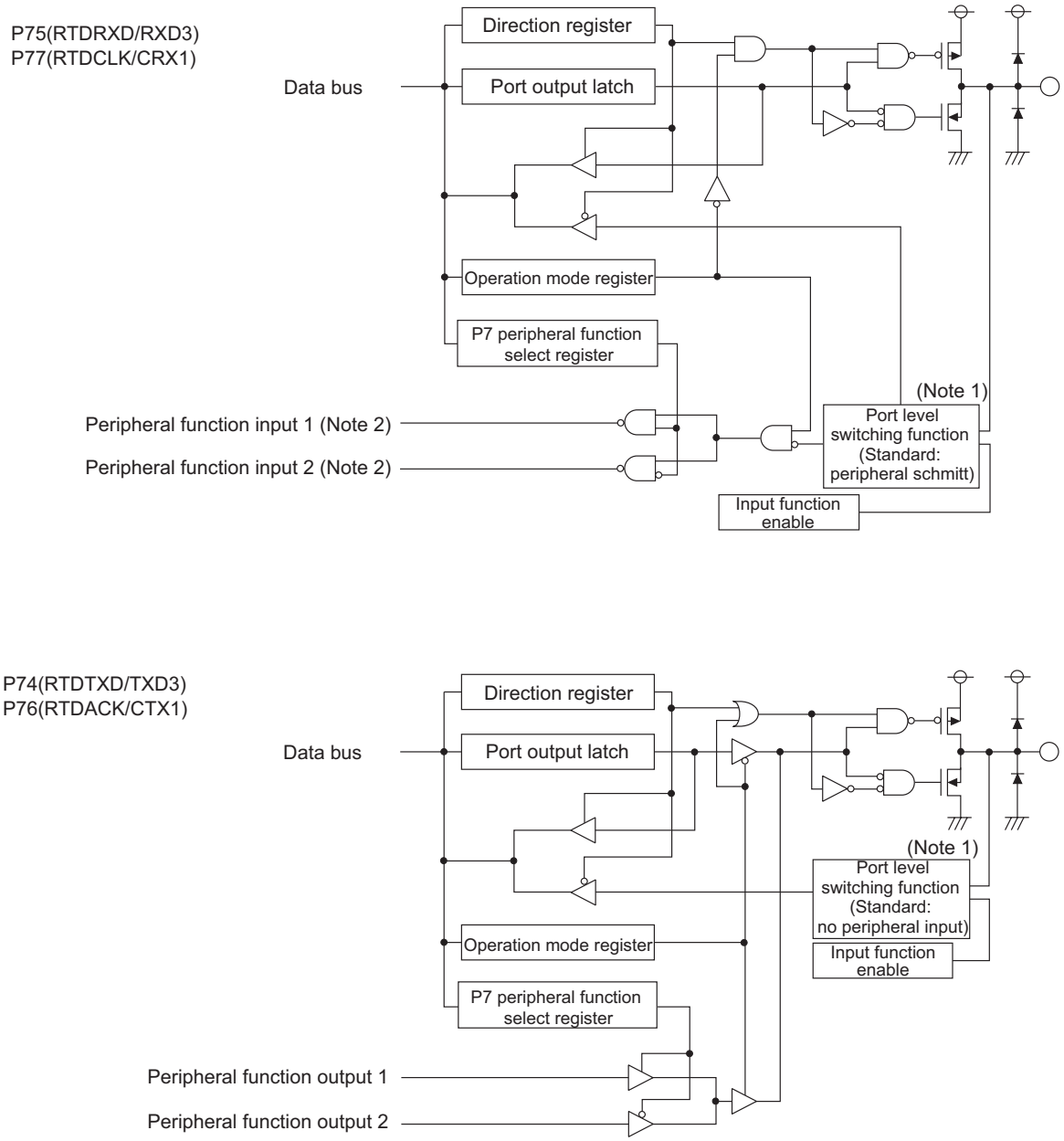


P84(SCLKI0,SCLKO0)
P87(SCLKI1,SCLKO1)



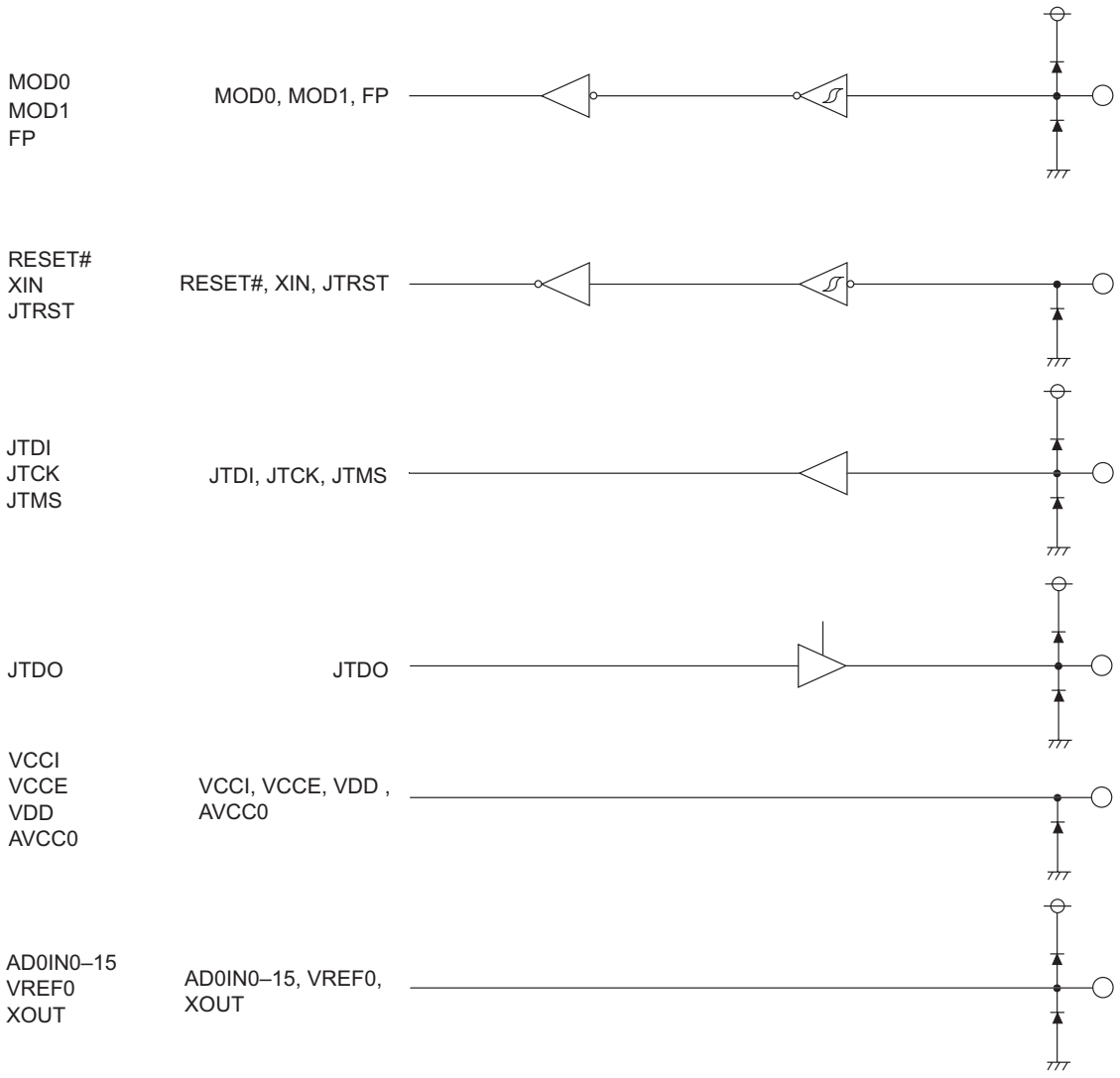
Note 1: For details about the port level switching function, see Section 8.4, "Port Input Level Switching Function."
Note 2: "H" level is entered to the peripheral function input when it is set to the general-purpose port in the operation mode register.
Notes: • The circle denotes a pin.
• The symbol ⊣◂ denotes a parasitic diode. Make sure the voltage applied to each pin does not exceed the VCCE voltage.
• The input capacitance of each pin is approximately 10 pF.

Figure 8.5.3 Port Peripheral Circuit Diagram (3)



Note 1: For details about the port level switching function, see Section 8.4, "Port Input Level Switching Function."
Note 2: "H" level is entered to the peripheral function input when it is set to the general-purpose port in the operation mode register.
Notes: • The circle denotes a pin.
• The symbol ⊣◂ denotes a parasitic diode. Make sure the voltage applied to each pin does not exceed the VCCE voltage.
• The input capacitance of each pin is approximately 10 pF.

Figure 8.5.4 Port Peripheral Circuit Diagram (4)



Notes: • The circle denotes a pin.
• The symbol ⊣◂ denotes a parasitic diode. Make sure the voltage applied to each pin does not exceed the VCCE voltage.
• The input capacitance of each pin is approximately 10 pF.

Figure 8.5.5 Port Peripheral Circuit Diagram (5)

8.6 Notes on Input/Output Ports

• When using input/output ports in output mode

Because the value of the Port Data Register is undefined when exiting the reset state, the Port Data Register must have its initial value set in it before the Port Direction Register can be set for output. Conversely, if the Port Direction Register is set for output before setting data in the Port Data Register, the Port Data Register outputs an undefined value until any data is written into it.

• About the port input disable function

Because the input/output ports are disabled against input after reset, they must be enabled for input by setting the Port Input Enable (PIEN0) bit to "1" before their input functions can be used.
When disabled against input, the input/output ports are in a state equivalent to a situation where the pin has a "L" level input applied. Consequently, if a peripheral input function is selected for any port while disabled against input by using the Port Operation Mode Register, the port may operate unexpectedly due to the "L" level input on it.

• About the peripheral function input when it is set to the general purpose port

In the pin for both peripheral function input and general-purpose port, "H" level is entered to the peripheral function input when it is set to the general-purpose port in the operation mode register. Therefore, when "L" level is entered to the peripheral function input pin, edge signal is entered to the peripheral function input at manipulating operation mode register.

This page is blank for reasons of layout.

CHAPTER 9

DMAC

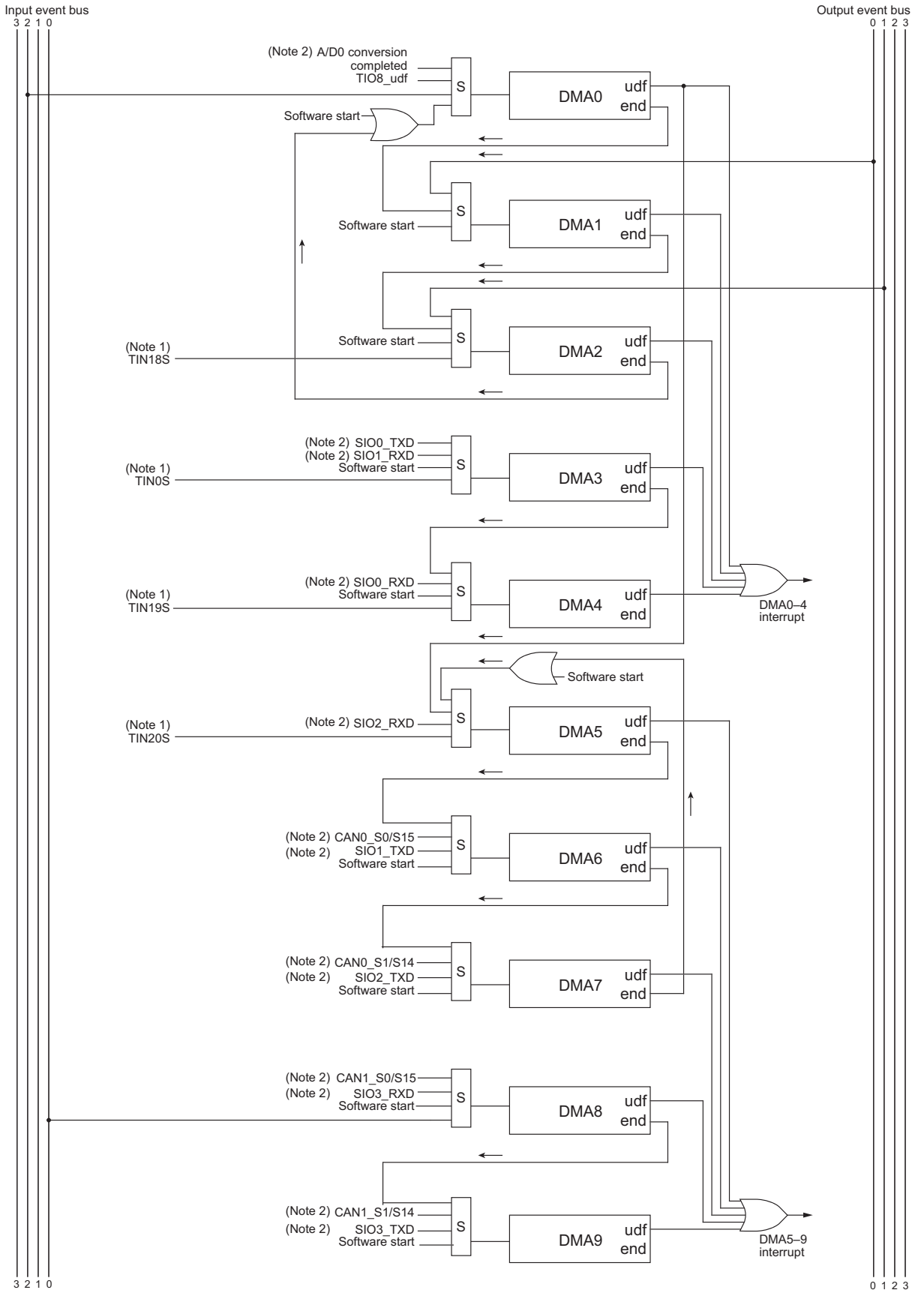
9.1 Outline of the DMAC

The microcomputer internally contains a 10-channel DMAC (Direct Memory Access Controller). It allows data to be transferred at high speed between internal peripheral I/Os, between internal RAM and internal peripheral I/O, or between internal RAMs, as initiated by a software trigger or requested from an internal peripheral I/O.

Table 9.1.1 Outline of the DMAC

Item	Description
Number of channels	10 channels
Transfer request sources	• Software trigger • Request from internal peripheral I/Os: A/D converter, multijunction timer, serial interface (reception completed, transmit buffer empty) or CAN • DMA channels can be cascaded (Note 1)
Maximum number of times transferred	256 times
Transferable address space	• 64 Kbytes (address space from H'0080 0000 to H'0080 FFFF) • Transfers between internal peripheral I/Os, between internal RAM and internal peripheral I/O, and between internal RAMs are supported.
Transfer data size	16 or 8 bits
Transfer method	Single transfer DMA (control of the internal bus is relinquished for each transfer performed), dual-address transfer
Transfer mode	Single transfer mode
Direction of transfer	One of three modes can be selected for the source and destination: • Address fixed • Address incremental • Ring buffered
Channel priority	DMA0 > DMA1 > DMA2 > DMA3 > DMA4 > DMA5 > DMA6 > DMA7 > DMA8 > DMA9 (Priority is fixed)
Maximum transfer rate	13.3 Mbytes per second (when internal peripheral clock BCLK = 20 MHz)
Interrupt request	Group interrupt request can be generated when each transfer count register underflows.
Transfer area	64 Kbytes from H'0080 0000 to H'0080 FFFF

Note 1: The DMA channels can be cascaded in the manner described below.
- Start DMA transfer on DMA1 upon completion of one DMA transfer on DMA0
- Start DMA transfer on DMA2 upon completion of one DMA transfer on DMA1
- Start DMA transfer on DMA0 upon completion of one DMA transfer on DMA2
- Start DMA transfer on DMA4 upon completion of one DMA transfer on DMA3
- Start DMA transfer on DMA6 upon completion of one DMA transfer on DMA5
- Start DMA transfer on DMA7 upon completion of one DMA transfer on DMA6
- Start DMA transfer on DMA5 upon completion of one DMA transfer on DMA7
- Start DMA transfer on DMA9 upon completion of one DMA transfer on DMA8
- Start DMA transfer on DMA5 upon completion of all DMA transfers on DMA0 (upon underflow of the transfer count register)



Note 1: Indicates edge select output at the timer input pin.

Note 2: Indicates an input signal from each peripheral circuit.

Figure 9.1.1 Block Diagram of the DMAC

9.2 DMAC Related Registers

The diagram below shows a memory map of the DMAC related registers.

DMAC Related Register Map (1/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0400	DMA0–4 Interrupt Request Status Register (DM04ITST)	DMA0–4 Interrupt Request Mask Register (DM04ITMK)	9-18 9-19
	(Use inhibited area)		
H'0080 0408	DMA5–9 Interrupt Request Status Register (DM59ITST)	DMA5–9 Interrupt Request Mask Register (DM59ITMK)	9-18 9-19
	(Use inhibited area)		
H'0080 0410	DMA0 Channel Control Register (DM0CNT)	DMA0 Transfer Count Register (DM0TCT)	9-6 9-15
H'0080 0412	DMA0 Source Address Register (DM0SA)		9-13
H'0080 0414	DMA0 Destination Address Register (DM0DA)		9-14
H'0080 0416	(Use inhibited area)		
H'0080 0418	DMA5 Channel Control Register (DM5CNT)	DMA5 Transfer Count Register (DM5TCT)	9-8 9-15
H'0080 041A	DMA5 Source Address Register (DM5SA)		9-13
H'0080 041C	DMA5 Destination Address Register (DM5DA)		9-14
H'0080 041E	(Use inhibited area)		
H'0080 0420	DMA1 Channel Control Register (DM1CNT)	DMA1 Transfer Count Register (DM1TCT)	9-6 9-15
H'0080 0422	DMA1 Source Address Register (DM1SA)		9-13
H'0080 0424	DMA1 Destination Address Register (DM1DA)		9-14
H'0080 0426	(Use inhibited area)		
H'0080 0428	DMA6 Channel Control Register (DM6CNT)	DMA6 Transfer Count Register (DM6TCT)	9-9 9-15
H'0080 042A	DMA6 Source Address Register (DM6SA)		9-13
H'0080 042C	DMA6 Destination Address Register (DM6DA)		9-14
H'0080 042E	(Use inhibited area)		
H'0080 0430	DMA2 Channel Control Register (DM2CNT)	DMA2 Transfer Count Register (DM2TCT)	9-7 9-15
H'0080 0432	DMA2 Source Address Register (DM2SA)		9-13
H'0080 0434	DMA2 Destination Address Register (DM2DA)		9-14
H'0080 0436	(Use inhibited area)		
H'0080 0438	DMA7 Channel Control Register (DM7CNT)	DMA7 Transfer Count Register (DM7TCT)	9-9 9-15
H'0080 043A	DMA7 Source Address Register (DM7SA)		9-13
H'0080 043C	DMA7 Destination Address Register (DM7DA)		9-14
H'0080 043E	(Use inhibited area)		

DMAC Related Register Map (2/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0440	DMA3 Channel Control Register (DM3CNT)	DMA3 Transfer Count Register (DM3TCT)	9-7 9-15
H'0080 0442	DMA3 Source Address Register (DM3SA)		9-13
H'0080 0444	DMA3 Destination Address Register (DM3DA)		9-14
H'0080 0446	(Use inhibited area)		
H'0080 0448	DMA8 Channel Control Register (DM8CNT)	DMA8 Transfer Count Register (DM8TCT)	9-10 9-15
H'0080 044A	DMA8 Source Address Register (DM8SA)		9-13
H'0080 044C	DMA8 Destination Address Register (DM8DA)		9-14
H'0080 044E	(Use inhibited area)		
H'0080 0450	DMA4 Channel Control Register (DM4CNT)	DMA4 Transfer Count Register (DM4TCT)	9-8 9-15
H'0080 0452	DMA4 Source Address Register (DM4SA)		9-13
H'0080 0454	DMA4 Destination Address Register (DM4DA)		9-14
H'0080 0456	(Use inhibited area)		
H'0080 0458	DMA9 Channel Control Register (DM9CNT)	DMA9 Transfer Count Register (DM9TCT)	9-10 9-15
H'0080 045A	DMA9 Source Address Register (DM9SA)		9-13
H'0080 045C	DMA9 Destination Address Register (DM9DA)		9-14
H'0080 045E	(Use inhibited area)		
H'0080 0460	DMA0 Software Request Generation Register (DM0SRI)		9-12
H'0080 0462	DMA1 Software Request Generation Register (DM1SRI)		9-12
H'0080 0464	DMA2 Software Request Generation Register (DM2SRI)		9-12
H'0080 0466	DMA3 Software Request Generation Register (DM3SRI)		9-12
H'0080 0468	DMA4 Software Request Generation Register (DM4SRI)		9-12
⋮	(Use inhibited area)		
H'0080 0470	DMA5 Software Request Generation Register (DM5SRI)		9-12
H'0080 0472	DMA6 Software Request Generation Register (DM6SRI)		9-12
H'0080 0474	DMA7 Software Request Generation Register (DM7SRI)		9-12
H'0080 0476	DMA8 Software Request Generation Register (DM8SRI)		9-12
H'0080 0478	DMA9 Software Request Generation Register (DM9SRI)		9-12

9.2.1 DMA Channel Control Registers

DMA0 Channel Control Register (DM0CNT) <Address: H'0080 0410>

b0	1	2	3	4	5	6	b7
MDSEL0	TREQF0	REQSL0		TENL0	TSZSL0	SADSL0	DADSL0
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL0 DMA0 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF0 DMA0 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL0 DMA0 transfer request source select bit	00: Software start or one DMA2 transfer completed 01: A/D0 conversion completed 10: MJT (TIO8_udf) 11: MJT (input event bus 2)	R	W
4	TENL0 DMA0 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL0 DMA0 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL0 DMA0 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL0 DMA0 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA1 Channel Control Register (DM1CNT) <Address: H'0080 0420>

b0	1	2	3	4	5	6	b7
MDSEL1	TREQF1	REQSL1		TENL1	TSZSL1	SADSL1	DADSL1
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL1 DMA1 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF1 DMA1 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL1 DMA1 transfer request source select bit	00: Software start 01: MJT (output event bus 0) 10: Settings inhibited 11: One DMA0 transfer completed	R	W
4	TENL1 DMA1 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL1 DMA1 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL1 DMA1 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL1 DMA1 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA2 Channel Control Register (DM2CNT) <Address: H'0080 0430>

b0	1	2	3	4	5	6	b7
MDSEL2	TREQF2	REQSL2		TENL2	TSZSL2	SADSL2	DADSL2
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL2 DMA2 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF2 DMA2 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL2 DMA2 transfer request source select bit	00: Software start 01: MJT (output event bus 1) 10: MJT (TIN18S) 11: One DMA1 transfer completed	R	W
4	TENL2 DMA2 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL2 DMA2 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL2 DMA2 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL2 DMA2 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA3 Channel Control Register (DM3CNT) <Address: H'0080 0440>

b0	1	2	3	4	5	6	b7
MDSEL3	TREQF3	REQSL3		TENL3	TSZSL3	SADSL3	DADSL3
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL3 DMA3 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF3 DMA3 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL3 DMA3 transfer request source select bit	00: Software start 01: SIO0_TXD (transmit buffer empty) 10: SIO1_RXD (reception completed) 11: MJT (TIN0S)	R	W
4	TENL3 DMA3 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL3 DMA3 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL3 DMA3 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL3 DMA3 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA4 Channel Control Register (DM4CNT) <Address: H'0080 0450>

b0	1	2	3	4	5	6	b7
MDSEL4	TREQF4	REQSL4		TENL4	TSZSL4	SADSL4	DADSL4
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL4 DMA4 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF4 DMA4 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL4 DMA4 transfer request source select bit	00: Software start 01: One DMA3 transfer completed 10: SIO0_RXD (reception completed) 11: MJT (TIN19S)	R	W
4	TENL4 DMA4 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL4 DMA4 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL4 DMA4 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL4 DMA4 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA5 Channel Control Register (DM5CNT) <Address: H'0080 0418>

b0	1	2	3	4	5	6	b7
MDSEL5	TREQF5	REQSL5		TENL5	TSZSL5	SADSL5	DADSL5
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL5 DMA5 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF5 DMA5 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL5 DMA5 transfer request source select bit	00: Software start or one DMA7 transfer completed 01: All DMA0 transfers completed 10: SIO2_RXD (reception completed) 11: MJT (TIN20S)	R	W
4	TENL5 DMA5 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL5 DMA5 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL5 DMA5 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL5 DMA5 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA6 Channel Control Register (DM6CNT) <Address: H'0080 0428>

b0	1	2	3	4	5	6	b7
MDSEL6	TREQF6	REQSL6		TENL6	TSZSL6	SADSL6	DADSL6
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL6 DMA6 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF6 DMA6 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL6 DMA6 transfer request source select bit	00: Software start 01: SIO1_TXD (transmit buffer empty) 10: CAN (CAN0_S0/S15) 11: One DMA5 transfer completed	R	W
4	TENL6 DMA6 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL6 DMA6 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL6 DMA6 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL6 DMA6 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA7 Channel Control Register (DM7CNT) <Address: H'0080 0438>

b0	1	2	3	4	5	6	b7
MDSEL7	TREQF7	REQSL7		TENL7	TSZSL7	SADSL7	DADSL7
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL7 DMA7 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF7 DMA7 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL7 DMA7 transfer request source select bit	00: Software start 01: SIO2_TXD (transmit buffer empty) 10: CAN (CAN0_S1/S14) 11: One DMA6 transfer completed	R	W
4	TENL7 DMA7 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL7 DMA7 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL7 DMA7 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL7 DMA7 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA8 Channel Control Register (DM8CNT) <Address: H'0080 0448>

b0	1	2	3	4	5	6	b7
MDSEL8	TREQF8	REQSL8		TENL8	TSZSL8	SADSL8	DADSL8
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL8 DMA8 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF8 DMA8 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL8 DMA8 transfer request source select bit	00: Software start 01: MJT (input event bus 0) 10: SIO3_RXD (reception completed) 11: CAN (CAN1_S0/S15)	R	W
4	TENL8 DMA8 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL8 DMA8 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL8 DMA8 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL8 DMA8 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA9 Channel Control Register (DM9CNT) <Address: H'0080 0458>

b0	1	2	3	4	5	6	b7
MDSEL9	TREQF9	REQSL9		TENL9	TSZSL9	SADSL9	DADSL9
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	MDSEL9 DMA9 transfer mode select bit	0: Normal mode 1: Ring buffer mode	R	W
1	TREQF9 DMA9 transfer request flag bit	0: Transfer not requested 1: Transfer requested	R(Note 1)	
2, 3	REQSL9 DMA9 transfer request source select bit	00: Software start 01: SIO3_TXD (transmit buffer empty) 10: CAN (CAN1_S1/S14) 11: One DMA8 transfer completed	R	W
4	TENL9 DMA9 transfer enable bit	0: Disable transfer 1: Enable transfer	R	W
5	TSZSL9 DMA9 transfer size select bit	0: 16 bits 1: 8 bits	R	W
6	SADSL9 DMA9 source address direction select bit	0: Fixed 1: Increment	R	W
7	DADSL9 DMA9 destination address direction select bit	0: Fixed 1: Increment	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

The DMA Channel Control Register consists of the bits to select DMA transfer mode on each channel, set the DMA transfer request flag, select the cause or source of DMA request and enable DMA transfer, as well as those to set the transfer size and the source/destination address directions.

(1) MDSELn (DMAn Transfer Mode Select) bit (Bit 0)

When performing DMA transfer in single transfer mode, this bit selects normal mode or ring buffer mode. Setting this bit to "0" selects normal mode and setting it to "1" selects ring buffer mode.
In ring buffer mode, transfer begins from the transfer start address and after performing transfers 32 times, control is returned back to the transfer start address, from which transfer operation is repeated. In this case, the Transfer Count Register counts in free-run mode, during which time transfer operation is continued until the transfer enable bit is reset to "0" (to disable transfer). In ring buffer mode, no interrupt is generated at completion of DMA transfer.

(2) TREQFn (DMAn Transfer Request Flag) bit (Bit 1)

This flag indicates if there are DMA transfer requests for each channel. This bit is set to "1", when DMA transfer requests are occurred in spite of TENLn bit setting value and then after completing transmission it is cleared to "0."
And when write "0" to this bit, it clear DMA transfer requests occurred. When write "1", it keeps value which before writing.
If a new DMA transfer request occurs on a channel for which the DMA transfer request flag has already been set to "1", the next DMA transfer request is not accepted until the transfer being performed on that channel is completed.

(3) REQSLn (DMAn Transfer Request Source Select) bits (Bits 2–3)

These bits select the cause or source of DMA transfer request on each DMA channel.

(4) TENLn (DMAn Transfer Enable) bit (Bit 4)

When setting this bit to "1" (enable transfer), DMA transfer is enable and when all transmissions are completed (underflow of transfer count register), it is "0" cleared. And when DMA transfer request is already occurred and set to transfer enable, DMA transfer starts immediately so that make sure not to do that. When setting this bit to "0" (disable transfer), DMA transfer is disable. However, if a transfer request has already been accepted, transfers on that channel are not disabled until after the requested transfer is completed.

(5) TSZSLn (DMAn Transfer Size Select) bit (Bit 5)

This bit selects the number of bits to be transferred in one DMA transfer operation (the unit of one transfer). The unit of one transfer is 16 bits when TSZSL = "0" or 8 bits when TSZSL = "1".

(6) SADSLn (DMAn Source Address Direction Select) bit (Bit 6)

This bit selects the direction in which the source address changes. This mode can be selected from two choices: Address fixed or Address incremental.

(7) DADSLn (DMAn Destination Address Direction Select) bit (Bit 7)

This bit selects the direction in which the destination address changes. This mode can be selected from two choices: Address fixed or Address incremental.

9.2.2 DMA Software Request Generation Registers

DMA0 Software Request Generation Register (DM0SRI)	<Address: H'0080 0460>
DMA1 Software Request Generation Register (DM1SRI)	<Address: H'0080 0462>
DMA2 Software Request Generation Register (DM2SRI)	<Address: H'0080 0464>
DMA3 Software Request Generation Register (DM3SRI)	<Address: H'0080 0466>
DMA4 Software Request Generation Register (DM4SRI)	<Address: H'0080 0468>
DMA5 Software Request Generation Register (DM5SRI)	<Address: H'0080 0470>
DMA6 Software Request Generation Register (DM6SRI)	<Address: H'0080 0472>
DMA7 Software Request Generation Register (DM7SRI)	<Address: H'0080 0474>
DMA8 Software Request Generation Register (DM8SRI)	<Address: H'0080 0476>
DMA9 Software Request Generation Register (DM9SRI)	<Address: H'0080 0478>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
DM0SRI-DM9SRI															
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	DM0SRI–DM9SRI DMA software request generation	DMA transfer request is generated by writing any data to these bits.	?	W

Note: • This register may be accessed in either bytes or halfwords.

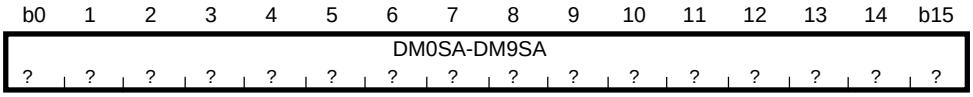
The DMA Software Request Generation Register is used to generate DMA transfer requests in software. A DMA transfer request can be generated by writing any data to this register when “Software start” has been selected for the cause of DMA request.

(1) DM0SRI–DM9SRI (DMA Software Request Generation)

A software DMA transfer request is generated by writing any data to this register in halfword (16 bits) or in byte (8 bits) beginning with an even or odd address when “Software start” is selected as the cause of DMA transfer request (by setting the DMAn Channel Control Register bits 2–3 to ‘00’).

9.2.3 DMA Source Address Registers

DMA0 Source Address Register (DM0SA)	<Address: H'0080 0412>
DMA1 Source Address Register (DM1SA)	<Address: H'0080 0422>
DMA2 Source Address Register (DM2SA)	<Address: H'0080 0432>
DMA3 Source Address Register (DM3SA)	<Address: H'0080 0442>
DMA4 Source Address Register (DM4SA)	<Address: H'0080 0452>
DMA5 Source Address Register (DM5SA)	<Address: H'0080 041A>
DMA6 Source Address Register (DM6SA)	<Address: H'0080 042A>
DMA7 Source Address Register (DM7SA)	<Address: H'0080 043A>
DMA8 Source Address Register (DM8SA)	<Address: H'0080 044A>
DMA9 Source Address Register (DM9SA)	<Address: H'0080 045A>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	DM0SA–DMA9SA DMA source address	Source address bits A16–A31 (A0–A15 are fixed to H'0080)	R	W

Note: • This register must always be accessed in halfwords.

The DMA Source Address Register is used to set the source address of DMA transfer in such a way that bit 0 and bit 15 correspond to A16 and A31, respectively. Because this register is comprised of a current register, the values read from this register are always the current value.
When DMA transfer finishes (i.e., the Transfer Count Register underflows), the value in this register if "Address fixed" is selected, is the same source address that was set in it before the DMA transfer began; if "Address incremental" is selected, the value in this register is the last transfer address + 1 (for 8-bit transfer) or the last transfer address + 2 (for 16-bit transfer).

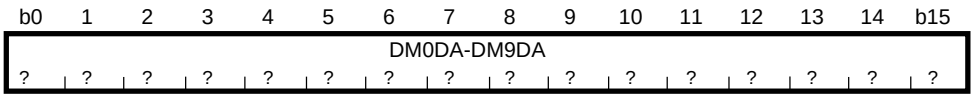
The DMA Source Address Register must always be accessed in halfwords (16 bits) beginning with an even address. If accessed in bytes, the value in this register is undefined.

(1) DM0SA–DM9SA (Source Address A16–A31)

Set this register to specify the source address of DMA transfer in the SFR area or internal RAM space from the address H'0080 0000 to the address H'0080 FFFF.
The 16 high-order source address bits (A0–A15) are always fixed to H'0080. Use this register to set the 16 low-order source address bits (with bit 0 corresponding to the source address A16, and bit 15 corresponding to the source address A31).

9.2.4 DMA Destination Address Registers

DMA0 Destination Address Register (DM0DA)	<Address: H'0080 0414>
DMA1 Destination Address Register (DM1DA)	<Address: H'0080 0424>
DMA2 Destination Address Register (DM2DA)	<Address: H'0080 0434>
DMA3 Destination Address Register (DM3DA)	<Address: H'0080 0444>
DMA4 Destination Address Register (DM4DA)	<Address: H'0080 0454>
DMA5 Destination Address Register (DM5DA)	<Address: H'0080 041C>
DMA6 Destination Address Register (DM6DA)	<Address: H'0080 042C>
DMA7 Destination Address Register (DM7DA)	<Address: H'0080 043C>
DMA8 Destination Address Register (DM8DA)	<Address: H'0080 044C>
DMA9 Destination Address Register (DM9DA)	<Address: H'0080 045C>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	DM0DA–DM9DA DMA destination address	Destination address bits A16–A31 (A0–A15 are fixed to H'0080)	R	W

Note: • This register must always be accessed in halfwords

The DMA Destination Address Register is used to set the destination address of DMA transfer in such a way that bit 0 and bit 15 correspond to A16 and A31, respectively. Because this register is comprised of a current register, the values read from this register are always the current value.
When DMA transfer finishes (i.e., the Transfer Count Register underflows), the value in this register if "Address fixed" is selected, is the same source address that was set in it before the DMA transfer began; if "Address incremental" is selected, the value in this register is the last transfer address + 1 (for 8-bit transfer) or the last transfer address + 2 (for 16-bit transfer).

The DMA Destination Address Register must always be accessed in halfwords (16 bits) beginning with an even address. If accessed in bytes, the value in this register is undefined.

(1) DM0DA–DM9DA (Destination Address bits A16–A31)

Set this register to specify the destination address of DMA transfer in the SFR area or internal RAM space from the address H'0080 0000 to the address H'0080 FFFF.
The 16 high-order destination address bits (A0–A15) are always fixed to H'0080. Use this register to set the 16 low-order destination address bits (with bit 0 corresponding to the destination address A16, and bit 15 corresponding to the destination address A31).

9.2.5 DMA Transfer Count Registers

DMA0 Transfer Count Register (DM0TCT)	<Address: H'0080 0411>
DMA1 Transfer Count Register (DM1TCT)	<Address: H'0080 0421>
DMA2 Transfer Count Register (DM2TCT)	<Address: H'0080 0431>
DMA3 Transfer Count Register (DM3TCT)	<Address: H'0080 0441>
DMA4 Transfer Count Register (DM4TCT)	<Address: H'0080 0451>
DMA5 Transfer Count Register (DM5TCT)	<Address: H'0080 0419>
DMA6 Transfer Count Register (DM6TCT)	<Address: H'0080 0429>
DMA7 Transfer Count Register (DM7TCT)	<Address: H'0080 0439>
DMA8 Transfer Count Register (DM8TCT)	<Address: H'0080 0449>
DMA9 Transfer Count Register (DM9TCT)	<Address: H'0080 0459>

b8	9	10	11	12	13	14	b15
DM0TCT-DM9TCT							
?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–15	DM0TCT–DM9TCT	DMA transfer count	R	W
	DMA transfer count	(Has no effect during 32-channel ring buffer mode)		

The DMA Transfer Count Register is used to set the number of times data is transferred on each channel. However, the value in this register has no effect during ring buffer mode.

The transfer count is the (value set in the transfer count register + 1). Because the DMA Transfer Count Register is comprised of a current register, the values read from this register are always the current value. (However, if the register is read in a cycle immediately after transfer, the value obtained is one that was stored in the count register before the transfer began.) When transfer finishes, this count register underflows and the value read from it is H'FF.

When transfer is enabled, this register is protected in hardware and cannot be accessed for write.

During ring buffer mode, the register counts down in free-run mode and continues counting until transfer is disabled. No interrupt is generated at underflow.

If any cascaded channel exists, each time one DMA transfer (byte or halfword) is completed or when all transfers on a channel are completed (i.e., the transfer count register underflows), transfer on the cascaded channel starts.

9.2.6 DMA Interrupt Related Registers

The DMA interrupt related registers are used to control the interrupt request signals sent from the DMAC to the Interrupt Controller.

(1) Interrupt request status bit

This status bit is used to determine whether there is an interrupt request. When an interrupt request occurs, this bit is set in hardware (cannot be set in software). The status bit is cleared by writing "0". Writing "1" has no effect; the bit retains the status it had before the write. Because this status bit is unaffected by the interrupt request mask bit, it can be used to inspect the operating status of peripheral functions.
In interrupt handling, make sure that within the grouped interrupt request status, only the status bit for the interrupt request that has been serviced is cleared. If the status bit for any interrupt request that has not been serviced is cleared, the pending interrupt request is cleared simultaneously with its status bit.

(2) Interrupt request mask bit

This bit is used to disable unnecessary interrupt requests within the grouped interrupt request. Set this bit to "0" to enable interrupt requests or "1" to disable interrupt requests.

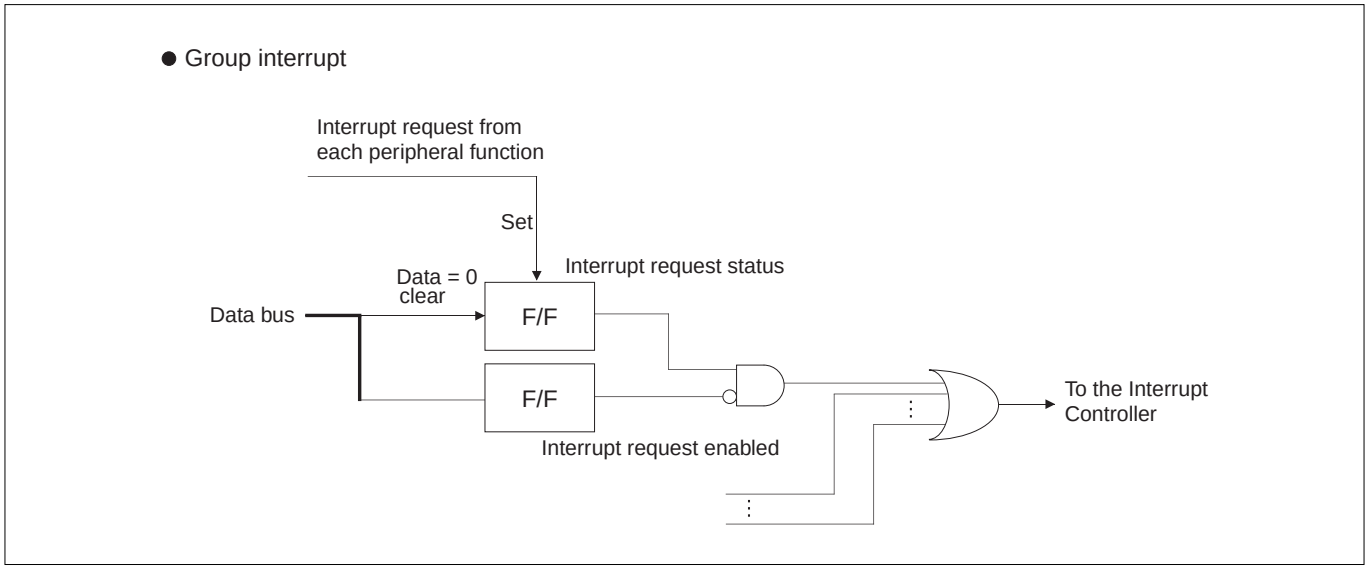
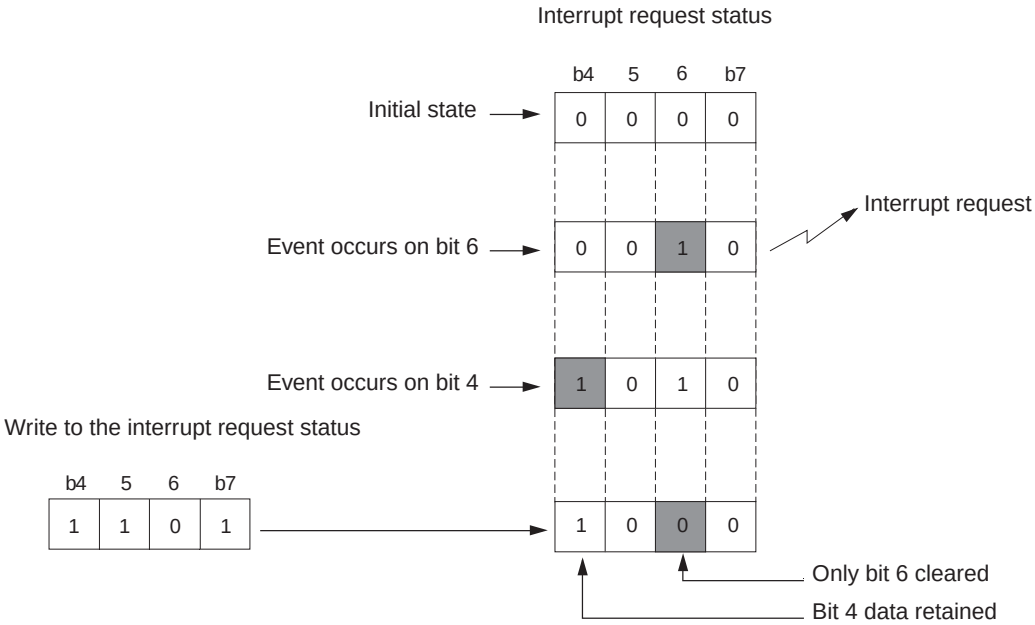


Figure 9.2.1 Interrupt Request Status and Mask Registers

● Example for clearing interrupt request status



● Program example

• To clear the Interrupt Request Status Register 0 (ISTREG) interrupt request status 1, ISTAT1 (0x02 bit)

○ ISTREG = 0xfd; /* Clear ISTAT1 (0x02 bit) only */

To clear an interrupt request status, always be sure to write "1" to all other interrupt request status bits. At this time, avoid using a logic operation like the one shown below. Because it requires three step-ISTREG read, logic operation and write, if another interrupt request occurs between the read and write, status may be inadvertently cleared.

× ISTREG &= 0xfd; /* Clear ISTAT1 (0x02 bit) only */

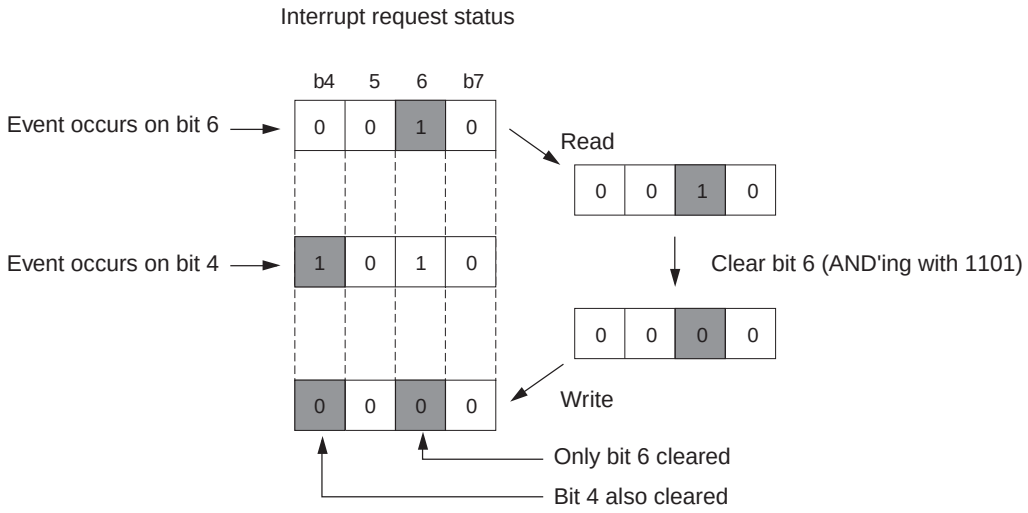


Figure 9.2.2 Example for Clearing Interrupt Request Status

DMA0–4 Interrupt Request Status Register (DM04ITST) <Address: H'0080 0400>

b0	1	2	3	4	5	6	b7
0	0	0	DMITST4 0	DMITST3 0	DMITST2 0	DMITST1 0	DMITST0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–2	No function assigned. Fix to "0".		0	0
3	DMITST4 (DMA4 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
4	DMITST3 (DMA3 interrupt request status bit)	1: Interrupt requested		
5	DMITST2 (DMA2 interrupt request status bit)			
6	DMITST1 (DMA1 interrupt request status bit)			
7	DMITST0 (DMA0 interrupt request status bit)			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

DMA5–9 Interrupt Request Status Register (DM59ITST) <Address: H'0080 0408>

b0	1	2	3	4	5	6	b7
0	0	0	DMITST9 0	DMITST8 0	DMITST7 0	DMITST6 0	DMITST5 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–2	No function assigned. Fix to "0".		0	0
3	DMITST9 (DMA9 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
4	DMITST8 (DMA8 interrupt request status bit)	1: Interrupt requested		
5	DMITST7 (DMA7 interrupt request status bit)			
6	DMITST6 (DMA6 interrupt request status bit)			
7	DMITST5 (DMA5 interrupt request status bit)			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

The Interrupt Request Status Register helps to know the status of interrupt requests on each channel. If the DMAn interrupt request status bit (n = 0–9) is set to "1", it means that a DMA interrupt request on the corresponding channel has been generated.

(1) DMITSTn (DMAn Interrupt Request Status) bit (n = 0–9)

[Setting the DMAn interrupt request status bit]

This bit is set in hardware, and cannot be set in software.

[Clearing the DMAn interrupt request status bit]

This bit is cleared by writing "0" in software.

Note: • The DMAn interrupt request status bit cannot be cleared by writing "0" to the DMA Interrupt Control Register's "interrupt request bit" included in the Interrupt Controller.

When writing to the DMA Interrupt Request Status Register, make sure only the bits to be cleared are set to "0" and all other bits are set to "1". Those bits that have been set to "1" are unaffected by writing in software and retain the value they had before the write.

DMA0–4 Interrupt Request Mask Register (DM04ITMK) <Address: H'0080 0401>

b8	9	10	11	12	13	14	b15
0	0	0	DMITMK4 0	DMITMK3 0	DMITMK2 0	DMITMK1 0	DMITMK0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	DMITMK4 (DMA4 interrupt request mask bit)	0: Enable interrupt request	R	W
12	DMITMK3 (DMA3 interrupt request mask bit)	1: Mask (disable) interrupt request		
13	DMITMK2 (DMA2 interrupt request mask bit)			
14	DMITMK1 (DMA1 interrupt request mask bit)			
15	DMITMK0 (DMA0 interrupt request mask bit)			

DMA5–9 Interrupt Request Mask Register (DM59ITMK) <Address: H'0080 0409>

b8	9	10	11	12	13	14	b15
0	0	0	DMITMK9 0	DMITMK8 0	DMITMK7 0	DMITMK6 0	DMITMK5 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	DMITMK9 (DMA9 interrupt request mask bit)	0: Enable interrupt request	R	W
12	DMITMK8 (DMA8 interrupt request mask bit)	1: Mask (disable) interrupt request		
13	DMITMK7 (DMA7 interrupt request mask bit)			
14	DMITMK6 (DMA6 interrupt request mask bit)			
15	DMITMK5 (DMA5 interrupt request mask bit)			

The DMA Interrupt Request Mask Register is used to mask interrupt requests on each DMA channel.

(1) DMITMKn (DMAn Interrupt Request Mask) bit (n = 0–9)

Setting the DMAn interrupt request mask bit to "1" masks the interrupt requests on DMAn channel. However, if an interrupt request occurs, the DMAn interrupt request status bit is always set to "1" irrespective of the contents of this mask register.

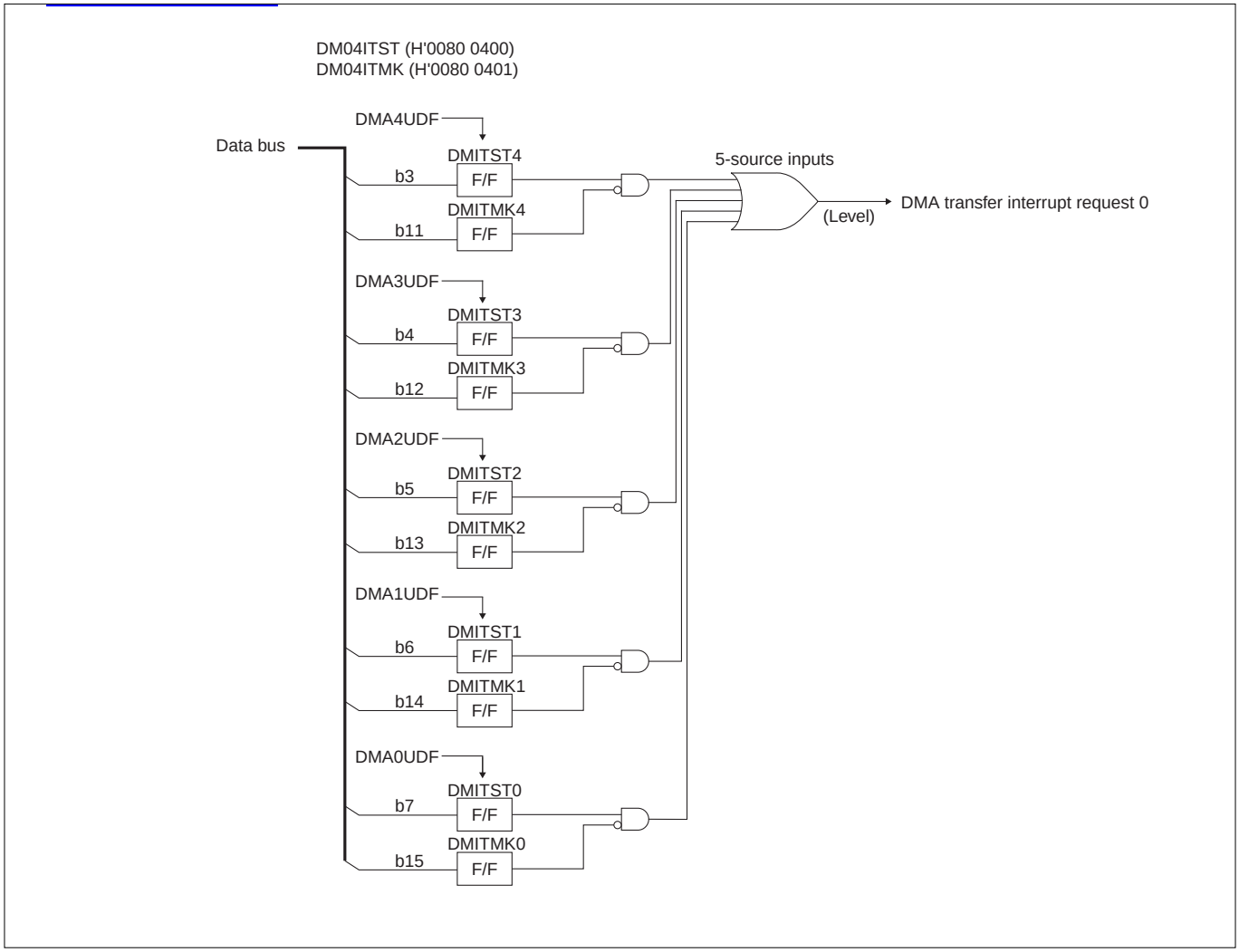


Figure 9.2.3 Block Diagram of DMA Transfer Interrupt Request 0

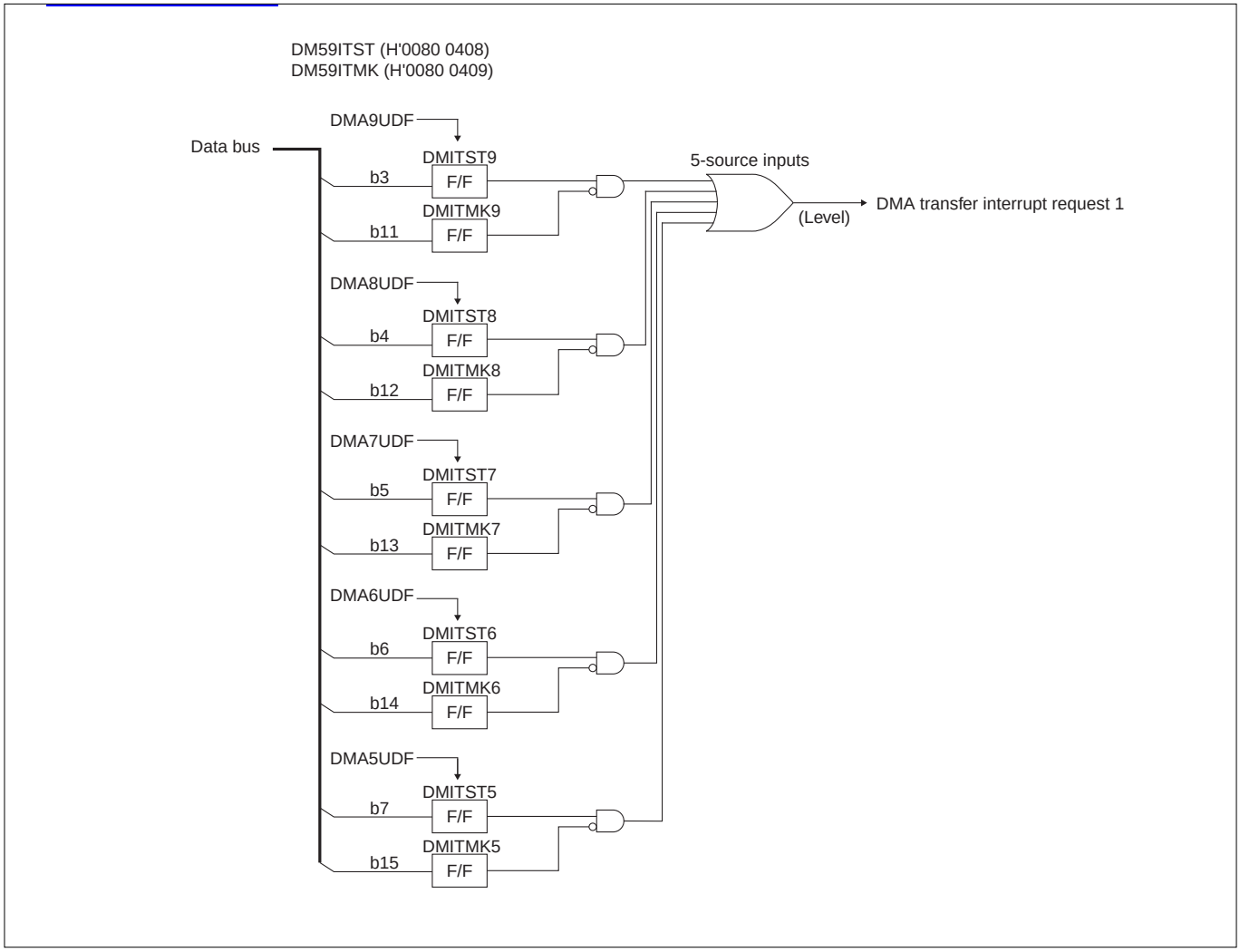


Figure 9.2.4 Block Diagram of DMA Transfer Interrupt Request 1

9.3 Functional Description of the DMAC

9.3.1 DMA Transfer Request Sources

For each DMA channel (channels 0–9), DMA transfer can be requested from two or more sources. There are various causes or sources of DMA transfer request, so that DMA transfer can be started by a request from some internal peripheral I/O, started in software by a program, or can be started upon completion of one transfer or all transfers on another DMA channel (cascade mode).

The causes or sources of DMA transfer requests are selected using the transfer request source select bits REQSLn on each channel (DMAn Channel Control Register bits 2 and 3). The tables below list the causes or sources of DMA transfer requests on each channel.

Table 9.3.1 DMA Transfer Request Sources and Generation Timings on DMA0

REQSL0		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start or one DMA2 transfer completed	When any data is written to the DMA0 Software Request Generation Register (software start) or when one DMA2 transfer is completed (cascade mode)
0	1	A/D0 conversion completed	When A/D0 conversion is completed
1	0	MJT (TIO8_udf)	When MJT TIO8 underflows
1	1	MJT (input event bus 2)	When MJT input event bus 2 signal is generated

Table 9.3.2 DMA Transfer Request Sources and Generation Timings on DMA1

REQSL1		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start	When any data is written to the DMA1 Software Request Generation Register
0	1	MJT (output event bus 0)	When MJT output event bus 0 signal is generated
1	0	Settings inhibited	–
1	1	One DMA0 transfer completed	When one DMA0 transfer is completed (cascade mode)

Table 9.3.3 DMA Transfer Request Sources and Generation Timings on DMA2

REQSL2		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start	When any data is written to the DMA2 Software Request Generation Register
0	1	MJT (output event bus 1)	When MJT output event bus 1 signal is generated
1	0	MJT (TIN18S)	When MJT TIN18 input signal is generated (edge select output)
1	1	One DMA1 transfer completed	When one DMA1 transfer is completed (cascade mode)

Table 9.3.4 DMA Transfer Request Sources and Generation Timings on DMA3

REQSL3		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start	When any data is written to the DMA3 Software Request Generation Register
0	1	SIO0_TXD (transmit buffer empty)	When SIO0 transmit buffer is empty
1	0	SIO1_RXD (reception completed)	When SIO1 reception is completed
1	1	MJT (TIN0S)	When MJT TIN0 input signal is generated (edge select output)

Table 9.3.5 DMA Transfer Request Sources and Generation Timings on DMA4

REQSL4		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start	When any data is written to the DMA4 Software Request Generation Register
0	1	One DMA3 transfer completed	When one DMA3 transfer is completed (cascade mode)
1	0	SIO0_RXD (reception completed)	When SIO0 reception is completed
1	1	MJT (TIN19S)	When MJT TIN19 input signal is generated (edge select output)

Table 9.3.6 DMA Transfer Request Sources and Generation Timings on DMA5

REQSL5		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start or one DMA7 transfer completed	When any data is written to the DMA5 Software Request Generation Register (software start) or when one DMA7 transfer is completed (cascade mode)
0	1	All DMA0 transfers completed	When all DMA0 transfers are completed (cascade mode)
1	0	SIO2_RXD (reception completed)	When SIO2 reception is completed
1	1	MJT (TIN20S)	When MJT TIN20 input signal is generated (edge select output)

Table 9.3.7 DMA Transfer Request Sources and Generation Timings on DMA6

REQSL6		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start	When any data is written to the DMA6 Software Request Generation Register
0	1	SIO1_TXD (transmit buffer empty)	When SIO1 transmit buffer is empty
1	0	CAN (CAN0_S0/S15)	CAN0: when slot 0 transmission failed or slot 15 transmission/reception completed
1	1	One DMA5 transfer completed	When one DMA5 transfer is completed (cascade mode)

Table 9.3.8 DMA Transfer Request Sources and Generation Timings on DMA7

REQSL7		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start	When any data is written to the DMA7 Software Request Generation Register
0	1	SIO2_TXD (transmit buffer empty)	When SIO2 transmit buffer is empty
1	0	CAN (CAN0_S1/S14)	CAN0: when slot 1 transmission failed or slot 14 transmission/reception completed
1	1	One DMA6 transfer completed	When one DMA6 transfer is completed (cascade mode)

Table 9.3.9 DMA Transfer Request Sources and Generation Timings on DMA8

REQSL8		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start	When any data is written to the DMA8 Software Request Generation Register
0	1	MJT (input event bus 0)	When MJT input event bus 0 signal is generated
1	0	SIO3_RXD (reception completed)	When SIO3 reception is completed
1	1	CAN (CAN1_S0/S15)	CAN1: when slot 0 transmission failed or slot 15 transmission/reception completed

Table 9.3.10 DMA Transfer Request Sources and Generation Timings on DMA9

REQSL9		DMA Transfer Request Source	DMA Transfer Request Generation Timing
0	0	Software start	When any data is written to the DMA9 Software Request Generation Register
0	1	SIO3_TXD (transmit buffer empty)	When SIO3 transmit buffer is empty
1	0	CAN (CAN1_S1/S14)	CAN1: when slot 1 transmission failed or slot 14 transmission/reception completed
1	1	One DMA8 transfer completed	When one DMA8 transfer is completed (cascade mode)

9.3.2 DMA Transfer Processing Procedure

Shown below is an example of how to control DMA transfer in cases when performing transfer on DMA0.

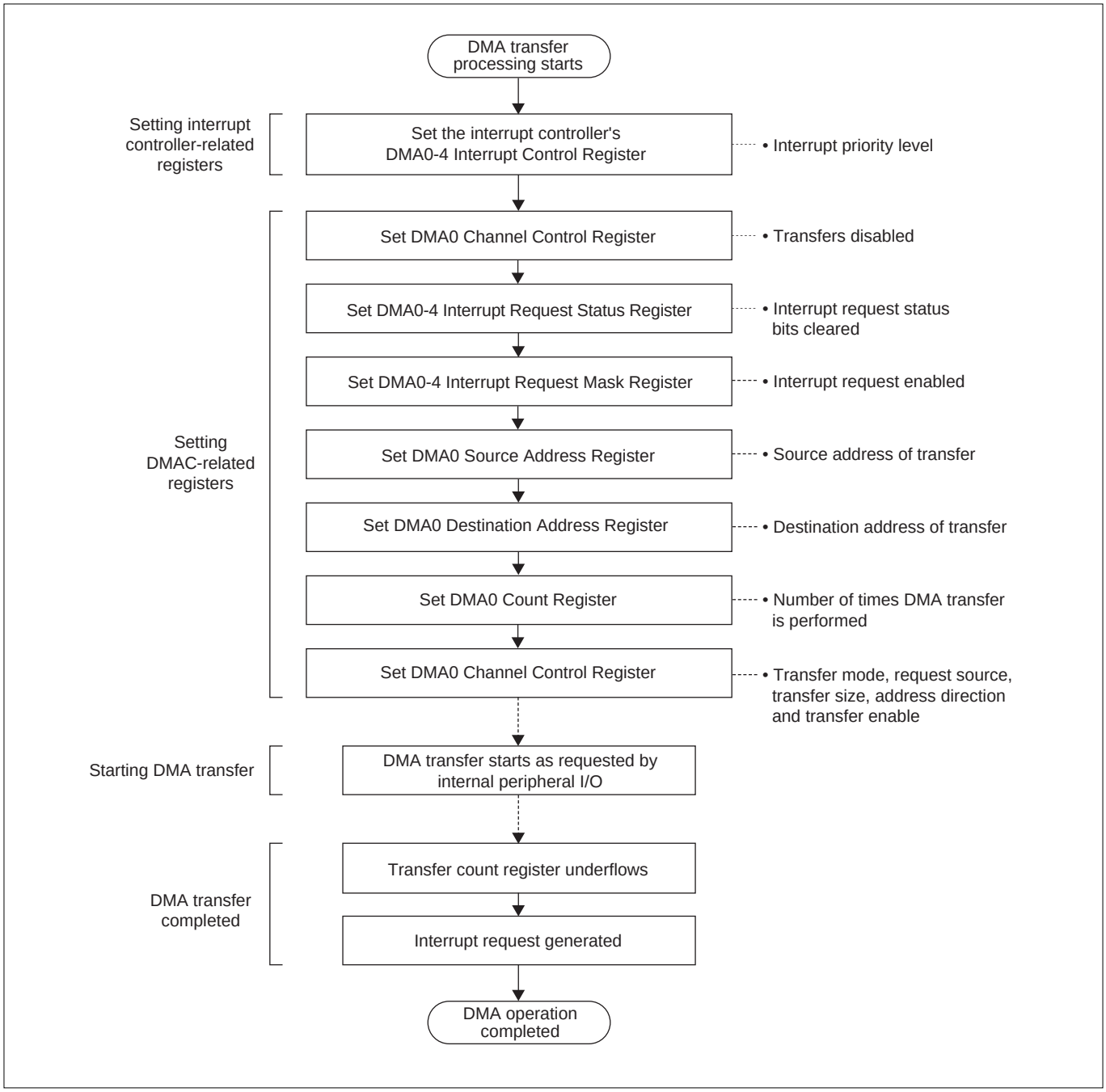


Figure 9.3.1 Example of a DMA Transfer Processing Procedure

9.3.3 Starting DMA

Use the DMAn Channel Control Register (DMnCNT) REQSL (DMA transfer request source select) bit to set the cause or source of DMA transfer request. To enable DMA, set the TENL (DMA transfer enable) bit to "1". DMA transfer begins when the specified cause or source of DMA transfer request becomes effective after setting the TENL (DMA transfer enable) bit to "1".

Note: • If the transfer request source selected by the REQSL (DMA transfer request source select) bit is MJT (TIN input signal), the time required for DMA transfer to begin after detecting the rising or falling or both edges of the TIN input signal is three cycles (150 ns when the internal peripheral clock = 20 MHz) at the shortest. Or, depending on the preceding or following bus usage condition, up to six cycles (300 ns when the internal peripheral clock = 20 MHz) may be required. (However, this applies when the external bus, HOLD and the LOCK instruction all are unused.)
To ensure that changes of the TIN input signal state will be detected correctly, make sure the TIN input signal is held active for a duration of more than 7tc (BCLK)/2. (For details, see Chapter 21 ELECTRICAL CHARACTERISTICS.)

9.3.4 DMA Channel Priority

DMA0 has the highest priority. The priority of this and other channels is shown below.
DMA0 > DMA1 > DMA2 > DMA3 > DMA4 > DMA5 > DMA6 > DMA7 > DMA8 > DMA9
This order of priority is fixed. Channel priority is resolved every transfer cycle (i.e., every three DMA buy cycles), and the channel with the highest priority among those that are requesting a DMA transfer is selected.

9.3.5 Gaining and Releasing Control of the Internal Bus

For any channel, control of the internal bus is gained and released in “single transfer DMA” mode. In single transfer DMA, the DMAC gains control of the internal bus (in one peripheral clock cycle) when DMA transfer request is accepted and after executing one DMA transfer (in one read and one write peripheral clock cycle), returns bus control to the CPU. The diagram below shows the operation in single transfer DMA.

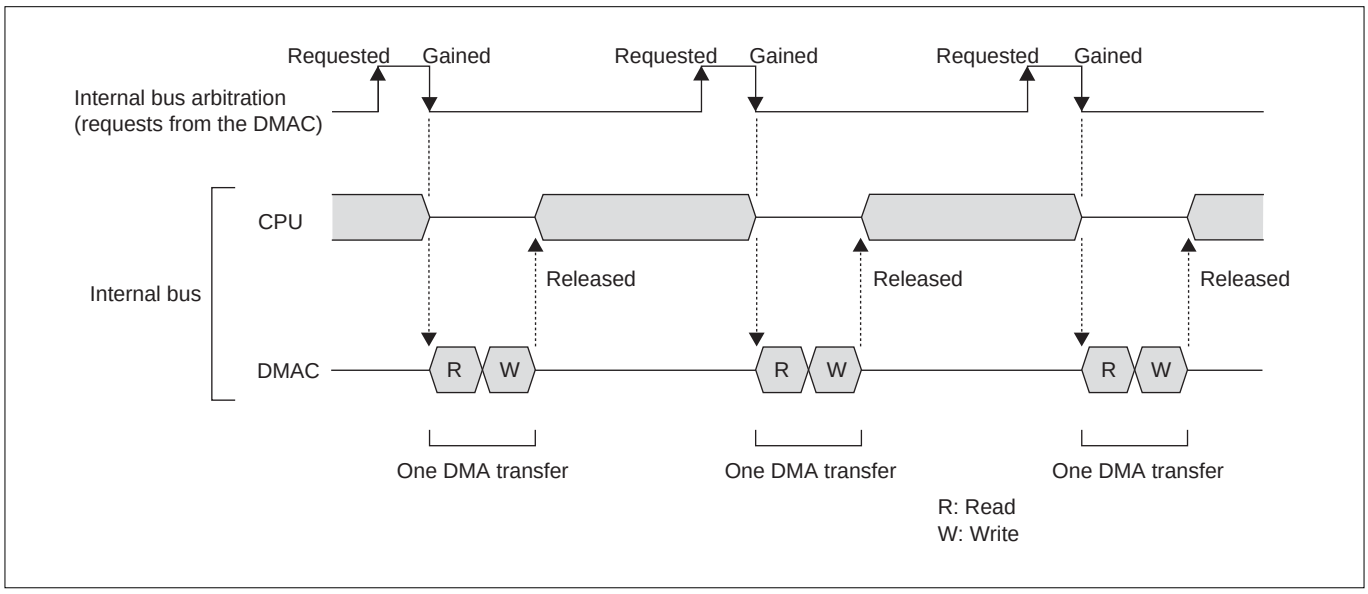


Figure 9.3.2 Gaining and Releasing Control of the Internal Bus

9.3.6 Transfer Units

Use the TSZSL (DMA transfer size select) bit to set for each channel the number of bits (8 or 16 bits) to be transferred in one DMA transfer.

9.3.7 Transfer Counts

Use the DMA Transfer Count Register to set transfer counts for each channel. Transfer can be performed up to 256 times. The value of the DMA Transfer Count Register is decremented by one every time one transfer unit is transferred. In ring buffer mode, the DMA Transfer Count Register operates in free-run mode, with the value set in it ignored.

9.3.8 Address Space

The address space in which data can be transferred by DMA is 64 Kbytes of SFR area or internal RAM space (H'0080 0000 through H'0080 FFFF) for both source and destination. To set the source and destination addresses on each DMA channel, use the DMA Source Address Register and DMA Destination Address Register.

9.3.9 Transfer Operation

(1) Dual-address transfer

Irrespective of the size of transfer unit, data is transferred in two bus cycles, one for source read access and one for destination write access. (The transfer data is taken into the DMAC's internal temporary register before being transferred.)

(2) Bus protocol and bus timing

Because the bus interface is shared with the CPU, DMA transfer is performed with the same bus protocol and the same bus timing as when peripheral modules are accessed by the CPU.

(3) Transfer rate

Transfer is performed using a total of three peripheral clock cycles, one cycle to gain control of the bus and one read and one write cycle to perform one transfer. Therefore, the maximum transfer rate is calculated by the equation below:

$$\text{Maximum transfer rate [bytes per second]} = 2 \text{ bytes} \times \frac{1}{1/f(\text{BCLK}) \times 3 \text{ cycles}}$$

(4) Address count direction and address changes

The direction in which the source and destination addresses are counted as transfer proceeds (“Address fixed” or “Address incremental”) is set for each channel using the SADSL (source address direction select) and DADSL (destination address direction select) bits.
When the transfer size is 16 bits, the address is incremented by two for each DMA transfer performed; when the transfer size is 8 bits, the address is incremented by one.

Table 9.3.11 Address Count Direction and Address Changes

Address Count Direction	Transfer Unit	Address Change for One DMA
Address fixed	8 bits	0
	16 bits	0
Address incremental	8 bits	+1
	16 bits	+2

(5) Transfer count value

The transfer count value is decremented one at a time, irrespective of the size of transfer unit (8 or 16 bits).

(6) Transfer byte positions

When the transfer unit is 8 bits, the LSB of the address register is effective for both source and destination. (Therefore, in addition to data transfers between even addresses or between odd addresses, data may be transferred from even address to odd address or vice versa.)

When the transfer unit is 16 bits, the LSB of the address register (= bit 15) is ignored, and data are always transferred in two bytes aligned to the 16-bit bus.

The diagram below shows the valid byte positions in DMA transfer.

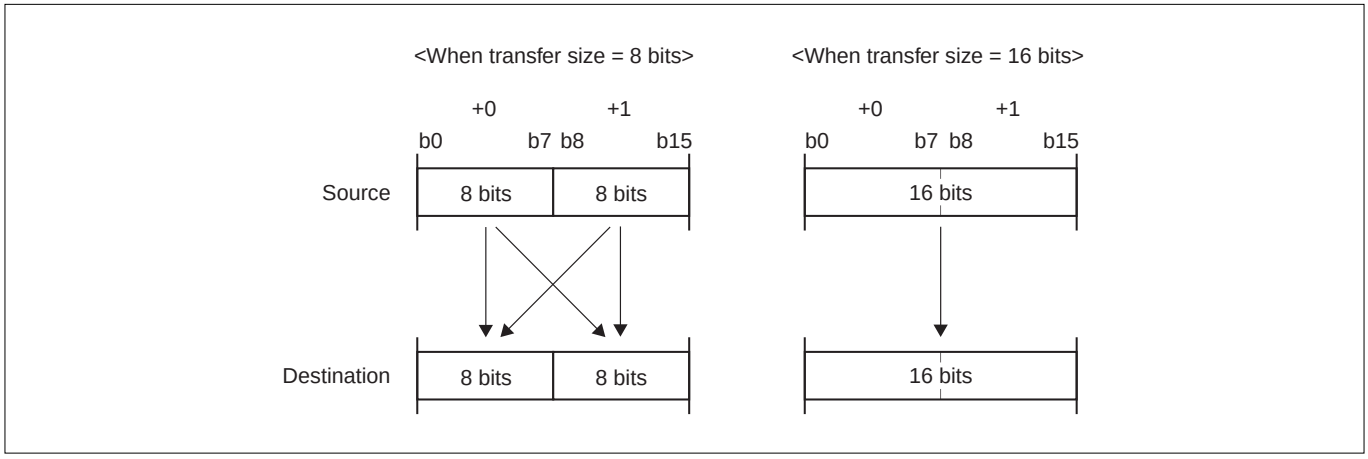


Figure 9.3.3 Transfer Byte Positions

(7) Ring buffer mode

When ring buffer mode is selected, transfer begins from the transfer start address and after performing transfers 32 times, control returns to the transfer start address, from which transfer operation is repeated. In this case, however, the five low-order bits of the ring buffer start address must always be B'00000 (if transfer size = 16 bits, the six low-order bits must be B'000000).

The following describes how addresses are incremented in ring buffer mode.

[1] When the transfer size is 8 bits

The 27 high-order bits of the transfer start address are fixed, and the five low-order bits are incremented by one at a time. When as transfer proceeds the five low-order bits reach B'11111, they are recycled to B'00000 by the next increment operation, thus returning to the start address again.

[2] When the transfer size is 16 bits

The 26 high-order bits of the transfer start address are fixed, and the six low-order bits are incremented by two at a time. When as transfer proceeds the six low-order bits reach B'111110, they are recycled to B'000000 by the next increment operation, thus returning to the start address again.

If the source address has been set to be incremented, it is the source address that recycles to the start address; if the destination address has been set to be incremented, it is the destination address that recycles to the start address. If both source and destination addresses have been set to be incremented, both addresses recycle to the start address. However, the start address on either side must have their five low-order bits initially set to B'00000 (if transfer size = 16 bits, the six low-order bits must be B'000000).

During ring buffer mode, the transfer count register is ignored. Once DMA operation starts, the counter operates in free-run mode, and the transfer continues until the transfer enable bit is cleared to "0" (to disable transfer).

<When transfer size = 8 bits>

Transfer count	Transfer address
1	H'0080 1000
2	H'0080 1001
3	H'0080 1002
⏐	⏐
31	H'0080 101E
32	H'0080 101F
↓	↓
1	H'0080 1000
2	H'0080 1001
⏐	⏐

<When transfer size = 16 bits>

Transfer count	Transfer address
1	H'0080 1000
2	H'0080 1002
3	H'0080 1004
⏐	⏐
31	H'0080 103C
32	H'0080 103E
↓	↓
1	H'0080 1000
2	H'0080 1002
⏐	⏐

Figure 9.3.4 Example of How Addresses Are Incremented in 32-channel Ring Buffer Mode

9.3.10 End of DMA and Interrupt

In normal mode, DMA transfer is terminated by an underflow of the transfer count register. When transfer finishes, the transfer enable bit is cleared to "0" and transfers are thereby disabled. Also, an interrupt request is generated at completion of transfer. However, if interrupt requests on any channel have been masked by the DMA Interrupt Request Mask Register, no interrupt requests are generated on that channel.

During ring buffer mode, the transfer count register operates in free-run mode, and transfer continues until the transfer enable bit is cleared to "0" (to disable transfer). In this case, therefore, no interrupt requests are generated at completion of DMA transfer. Nor are these DMA transfer-completed interrupt requests are generated even when transfer in ring buffer mode is terminated by clearing the transfer enable bit.

9.3.11 Each Register Status after Completion of DMA Transfer

When DMA transfer is completed, the status of the source and destination address registers becomes as follows:

(1) Address fixed

- The values set in the address registers before DMA transfer started remain intact (fixed).

(2) Address incremental

- For 8-bit transfer, the values of the address registers are the last transfer address + 1.
- For 16-bit transfer, the values of the address registers are the last transfer address + 2.

The transfer count register at completion of DMA transfer is in an underflow state (H'FF). Therefore, before another DMA transfer can be performed, the transfer count register must be set newly again, except when trying to perform transfers 256 times (H'FF).

9.4 Notes on the DMAC

- **About writing to the DMAC related registers**

Because DMA transfer involves exchanging data via the internal bus, the DMAC related registers basically can only be accessed for write immediately after reset or when transfer is disabled (transfer enable bit = "0"). When transfer is enabled, do not write to the DMAC related registers, except the DMA transfer enable bit, the transfer request flag and the DMA Transfer Count Register that is protected in hardware. This is a precaution necessary to ensure stable DMA operation.

The table below lists the registers that can or cannot be accessed for write.

Table 9.4.1 DMAC Related Registers That Can or Cannot Be Accessed for Write

Status	Transfer enable bit	Transfer request flag	DMA interrupt related registers	Other DMAC related registers
Transfer enabled	Can be accessed	Can be accessed	Can be accessed	Cannot be accessed
Transfer disabled	Can be accessed	Can be accessed	Can be accessed	Can be accessed

Even for registers that can exceptionally be written to while transfer is enabled, the following conditions must be observed:

(1) DMA Channel Control Register transfer enable bit and transfer request flag

For all other bits than transfer enable bit and transfer request flag in this register, be sure to write the same data that those bits had before the write. Note, however, that only writing "0" is effective for the transfer request flag.

(2) DMA Transfer Count Register

When transfer is enabled, this register is protected in hardware, so that any data rewritten to it is ignored.

(3) Rewriting the DMA source and DMA destination addresses on different channels by DMA transfer

Although this operation means accessing the DMAC related registers while DMA is enabled, there is no problem. Note, however, that no data can be transferred by DMA to the DMAC related registers on the currently active channel itself.

- **Manipulating the DMAC related registers by DMA transfer**

When manipulating the DMAC related registers by means of DMA transfer (e.g., reloading the DMAC related registers with the initial values by DMA transfer), do not write to the DMAC related registers on the currently active channel through that channel. (If this precaution is neglected, device operation cannot be guaranteed.) It is only the DMAC related registers on other channels that can be rewritten by means of DMA transfer. (For example, the DMAn Source Address and DMAn Destination Address Registers on channel 1 can be rewritten by DMA transfer through channel 0.)

- **About the DMA Interrupt Request Status Register**

When clearing the DMA Interrupt Request Status Register, be sure to write "1" to all bits, except those to be cleared. Writing "1" to any bits in this register has no effect, so that they retain the data they had before the write.

• **About the stable operation of DMA transfer**

To ensure the stable operation of DMA transfer, never rewrite the DMAC related registers, except the channel control register's transfer enable bit, unless transfer is disabled. One exception is that even when transfer is enabled, the DMA Source Address and DMA Destination Address Registers can be rewritten by DMA transfer from one channel to another.

CHAPTER 10

MULTIJUNCTION TIMERS

10.1 Outline of Multijunction Timers

The multijunction timers (abbreviated MJT) have input event and output event buses. Therefore, in addition to being used as a single unit, the timers can be internally connected to each other. This capability allows for highly flexible timer configuration, making it possible to meet various application needs. It is because the timers are connected to the internal event buses at multiple points that they are called the “multijunction” timers.

The 32176 has four types of MJT as listed in the table below, providing a total of 37-channel timers.

Table 10.1.1 Outline of MJT

Name	Type	No. of Channels	Description
TOP (Timer Output)	Output-related 16-bit timer (down-counter)	11	One of three output modes can be selected by software. <With correction function> • Single-shot output mode • Delayed single-shot output mode <Without correction function> • Continuous output mode
TIO (Timer Input Output)	Input/output-related 16-bit timer (down-counter)	10	One of three input modes or four output modes can be selected by software. <Input modes> • Measure clear input mode • Measure free-run input mode • Noise processing input mode <Output modes without correction function> • PWM output mode • Single-shot output mode • Delayed single-shot output mode • Continuous output mode
TMS (Timer Measure Small)	Input-related 16-bit timer (up-counter)	8	16-bit input measure timer
TML (Timer Measure Large)	Input-related 32-bit timer (up-counter)	8	32-bit input measure timer

Table 10.1.2 Interrupt Generation Functions of MJT

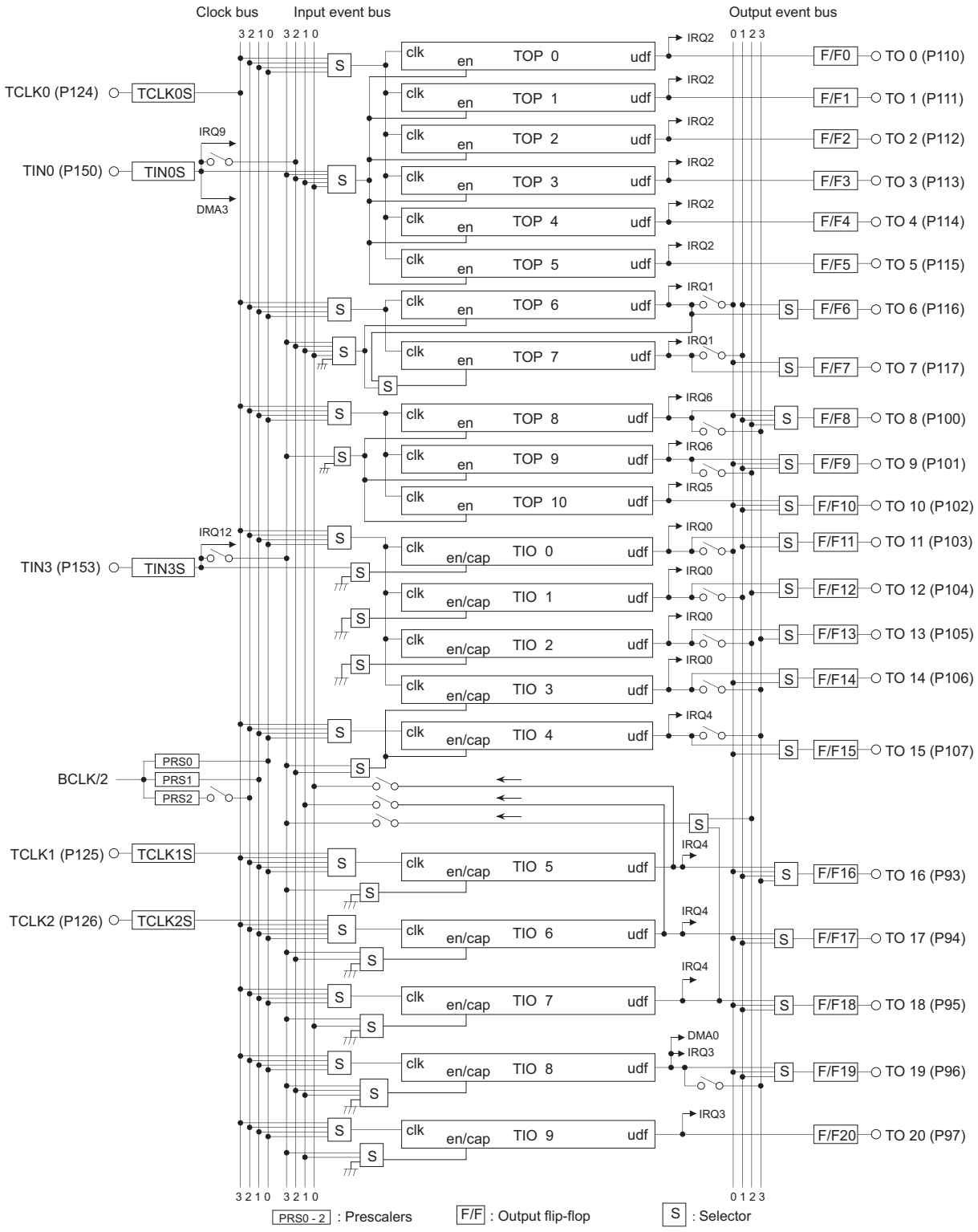
Signal Name	MJT Interrupt Request Source	Source of Interrupt Request	No. of ICU Input Sources
IRQ0	TIO0–3 output	MJT output interrupt 0	4
IRQ1	TOP6, TOP7 output	MJT output interrupt 1	2
IRQ2	TOP0–5 output	MJT output interrupt 2	6
IRQ3	TIO8, TIO9 output	MJT output interrupt 3	2
IRQ4	TIO4–7 output	MJT output interrupt 4	4
IRQ5	TOP10 output	MJT output interrupt 5	1
IRQ6	TOP8, TOP9 output	MJT output interrupt 6	2
IRQ7	TMS0, TMS1 output	MJT output interrupt 7	2
IRQ9	TIN0 input	MJT input interrupt 1	1
IRQ10	TIN16–TIN19 input	MJT input interrupt 2	4
IRQ11	TIN20–TIN23 input	MJT input interrupt 3	4
IRQ12	TIN3 input	MJT input interrupt 4	1

Table 10.1.3 DMA Transfer Request Generation by MJT

Corresponding DMAC Channel No.	DMA Transfer Request Source
DMA0	TIO8_udf
	Input event bus 2
DMA1	Output event bus 0
DMA2	Output event bus 1
	TIN18 input signal (TIN18S)
DMA3	TIN0 input signal (TIN0S)
DMA4	TIN19 input signal (TIN19S)
DMA5	TIN20 input signal (TIN20S)
DMA8	Input event bus 0

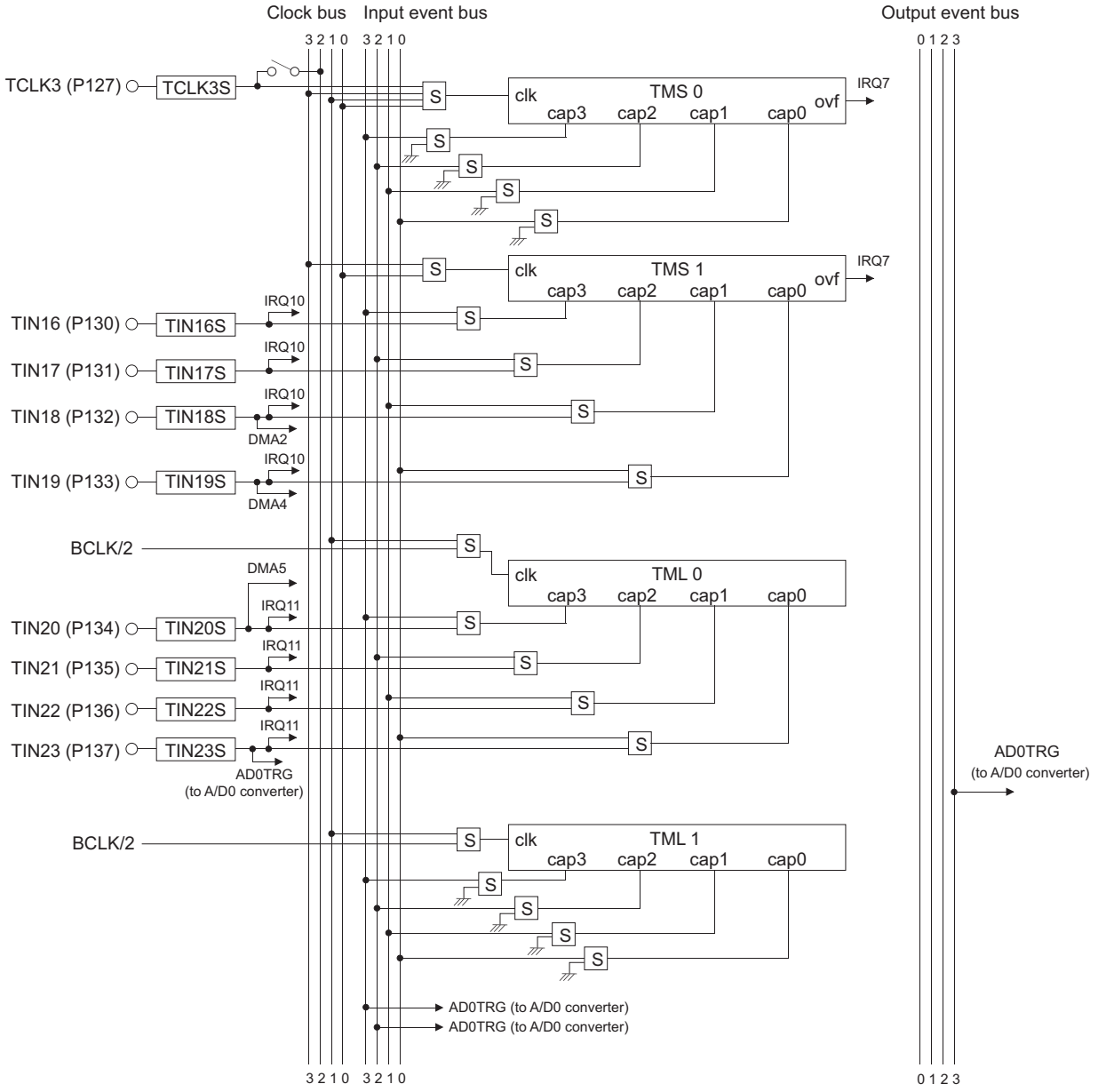
Table 10.1.4 A/D Conversion Start Request by MJT

Signal Name	A/D Conversion Start Request Source	A/D Converter
AD0TRG	Input event bus 2, input event bus 3, output event bus 3, TIN23	Can be input to A/D0 conversion start trigger



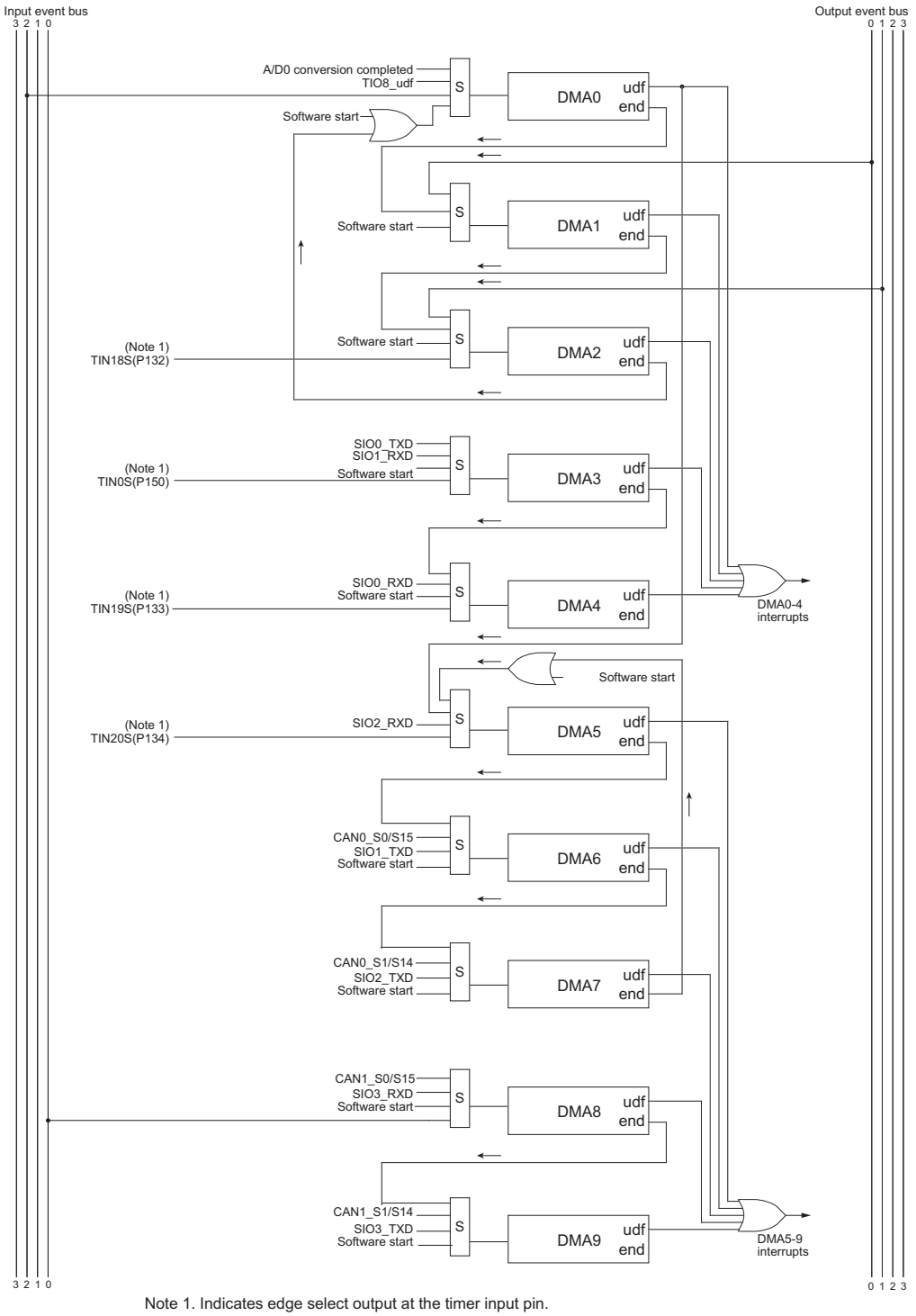
Notes: • IRQ0-7 and IRQ9-12 denote interrupt signals, of which the same number represents the same group of interrupts. (See Table 10.1.2)
• AD0TRG denotes trigger signal to the A/D0 converter. (See Table 10.1.4)

Figure 10.1.1 Block Diagram of MJT (1/3)



Notes: • IRQ0-7 and IRQ9-12 denote interrupt signals, of which the same number represents the same group of interrupts. (See Table 10.1.2)
• AD0TRG denotes trigger signal to the A/D0 converter. (See Table 10.1.4)

Figure 10.1.2 Block Diagram of MJT (2/3)



Note 1. Indicates edge select output at the timer input pin.

Figure 10.1.3 Block Diagram of MJT (3/3)

10.2 Common Units of Multijunction Timers

The common units of MJT include the following:

- Prescaler Unit
- Clock Bus and Input/Output Event Bus Control Unit
- Input Processing Control Unit
- Output Flip-flop Control Unit
- Interrupt Control Unit

10.2.1 MJT Common Unit Register Map

The table below shows a common unit register map of MJT.

MJT Common Unit Register Map

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0200	(Use inhibited area)	Clock Bus & Input Event Bus Control Register (CKIEBCR)	10-13
H'0080 0202	Prescaler Register 0 (PRS0)	Prescaler Register 1 (PRS1)	10-9
H'0080 0204	Prescaler Register 2 (PRS2)	Output Event Bus Control Register (OEBCR)	10-9 10-14
	(Use inhibited area)		
H'0080 0210	TCLK Input Processing Control Register (TCLKCR)		10-17
H'0080 0212	TIN Input Processing Control Register 0 (TINCR0)		10-18
H'0080 0214	(Use inhibited area)		
H'0080 0216	(Use inhibited area)		
H'0080 0218	TIN Input Processing Control Register 3 (TINCR3)		10-19
H'0080 021A	TIN Input Processing Control Register 4 (TINCR4)		10-19
H'0080 021C	(Use inhibited area)		
H'0080 021E	(Use inhibited area)		
H'0080 0220	F/F Source Select Register 0 (FFS0)		10-21
H'0080 0222	(Use inhibited area)	F/F Source Select Register 1 (FFS1)	10-22
H'0080 0224	F/F Protect Register 0 (FFP0)		10-23
H'0080 0226	F/F Data Register 0 (FFD0)		10-24
H'0080 0228	(Use inhibited area)	F/F Protect Register 1 (FFP1)	10-23
H'0080 022A	(Use inhibited area)	F/F Data Register 1 (FFD1)	10-24
	(Use inhibited area)		
H'0080 0230	TOP Interrupt Control Register 0 (TOPIR0)	TOP Interrupt Control Register 1 (TOPIR1)	10-29
H'0080 0232	TOP Interrupt Control Register 2 (TOPIR2)	TOP Interrupt Control Register 3 (TOPIR3)	10-31 10-32
H'0080 0234	TIO Interrupt Control Register 0 (TIOIR0)	TIO Interrupt Control Register 1 (TIOIR1)	10-33 10-34
H'0080 0236	TIO Interrupt Control Register 2 (TIOIR2)	TMS Interrupt Control Register (TMSIR)	10-35 10-36
H'0080 0238	TIN Interrupt Control Register 0 (TINIR0)	TIN Interrupt Control Register 1 (TINIR1)	10-37 10-38
H'0080 023A	(Use inhibited area)		
H'0080 023C	TIN Interrupt Control Register 4 (TINIR4)	TIN Interrupt Control Register 5 (TINIR5)	10-39
H'0080 023E	TIN Interrupt Control Register 6 (TINIR6)	(Use inhibited area)	10-41

10.2.2 Prescaler Unit

The Prescalers PRS0–2 are an 8-bit counter, which generates clocks supplied to each timer (TOP, TIO, TMS and TML) from the internal peripheral clock (BCLK) divided by 2 (10 MHz when f(BCLK) = 20 MHz).
The values of prescaler registers are initialized to H'00 upon exiting the reset state. When the set value of any prescaler register is rewritten, the prescaler starts operating with the new value at the same time it has underflowed.
Values H'00 to H'FF can be set in the prescaler register. The prescaler's divide-by ratio is given by the equation below:

$$\text{Prescaler divide-by ratio} = \frac{1}{\text{prescaler set value} + 1}$$

Prescaler Register 0 (PRS0) <Address: H'0080 0202>
Prescaler Register 1 (PRS1) <Address: H'0080 0203>
Prescaler Register 2 (PRS2) <Address: H'0080 0204>

b0	1	2	3	4	5	6	b7
(b8	9	10	11	12	13	14	b15)
PRS0-PRS2							
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–7 (8–15)	PRS0, PRS2 PRS1 Prescaler	Set the prescaler divide-by value	R	W

Prescaler Registers 0–2 start counting after exiting the reset state.
If the prescaler register is accessed for read during operation, the value written into it, not the current count, is read out.

10.2.3 Clock Bus and Input/Output Event Bus Control Unit

(1) Clock bus

The clock bus is provided for supplying clock to each timer, and is comprised of four lines of clock bus 0–3. Each timer can use these clock bus signals as clock input signals. The table below lists the signals that can be fed into the clock bus.

Table 10.2.1 Acceptable Clock Bus Signals

Clock Bus	Acceptable Signal
3	TCLK0 input
2	Internal prescaler (PRS2) or TCLK3 input
1	Internal prescaler (PRS1)
0	Internal prescaler (PRS0)

(2) Input event bus

The input event bus is provided for supplying a count enable signal or measure capture signal to each timer, and is comprised of four lines of input event bus 0–3. Each timer can use these input event bus signals as enable (or capture) input. Furthermore, they can also be used as request signals to start A/D conversion or DMA transfer.
The table below lists the signals that can be fed into the input event bus.

Table 10.2.2 Connectable (Acceptable) Input Event Bus Signals

Input Event Bus	Connectable (Acceptable) Signal (Note 1)
3	TIN3 input, output event bus 2 or TIO7 underflow signal
2	TIN0 input
1	TIO6 underflow signal
0	TIO5 underflow signal

Note 1: For the destination (output) to which the input event bus signals are connected, see Figure 10.1.1, “Block Diagram of MJT.”

(3) Output event bus

The output event bus has the underflow signal from each timer connected to it, and is comprised of four lines of output event bus 0–3. Output event bus signals are connected to output flip-flops, and output event buses 3, 0 and 1 can be connected to the A/D0 converter, DMA channel 1 and DMA channel 2, respectively. Furthermore, output event bus 2 can be connected to input event bus 3.
The table below lists the signals that can be connected to the output event bus.

Table 10.2.3 Connectable (Acceptable) Output Event Bus Signals

Input Event Bus	Connectable (Acceptable) Signal (Note 1)
3	TOP8, TIO3, TIO4 or TIO8 underflow signal
2	TOP9 or TIO2 underflow signal
1	TOP7 or TIO1 underflow signal
0	TOP6 or TIO0 underflow signal

Note 1: For the destination (output) to which the output event bus signals are connected, see Figure 10.1.1, “Block Diagram of MJT.”

Note that the signals from each timer to the output event bus (and TIO5, 6 signals to the input event bus) are generated with the timing shown in Table 10.2.4, and not the timing at which signals are output from the timer to the output flip-flop.

Table 10.2.4 Timing at Which Signals are Generated to the Output Event Bus by Each Timer

Timer	Mode	Timing at which signals are generated to the output event bus
TOP	Single-shot output mode	When the counter underflows
	Delayed single-shot output mode	When the counter underflows
	Continuous output mode	When the counter underflows
TIO(Note 1)	Measure clear input mode	When the counter underflows
	Measure free-run input mode	When the counter underflows
	Noise processing input mode	When the counter underflows
	PWM output mode	When the counter underflows
	Single-shot output mode	When the counter underflows
	Delayed single-shot output mode	When the counter underflows
	Continuous output mode	When the counter underflows
TMS	(16-bit measure input)	No signals generated
TML	(32-bit measure input)	No signals generated

Note 1: TIO5,6 output an underflow signal to the input event bus.

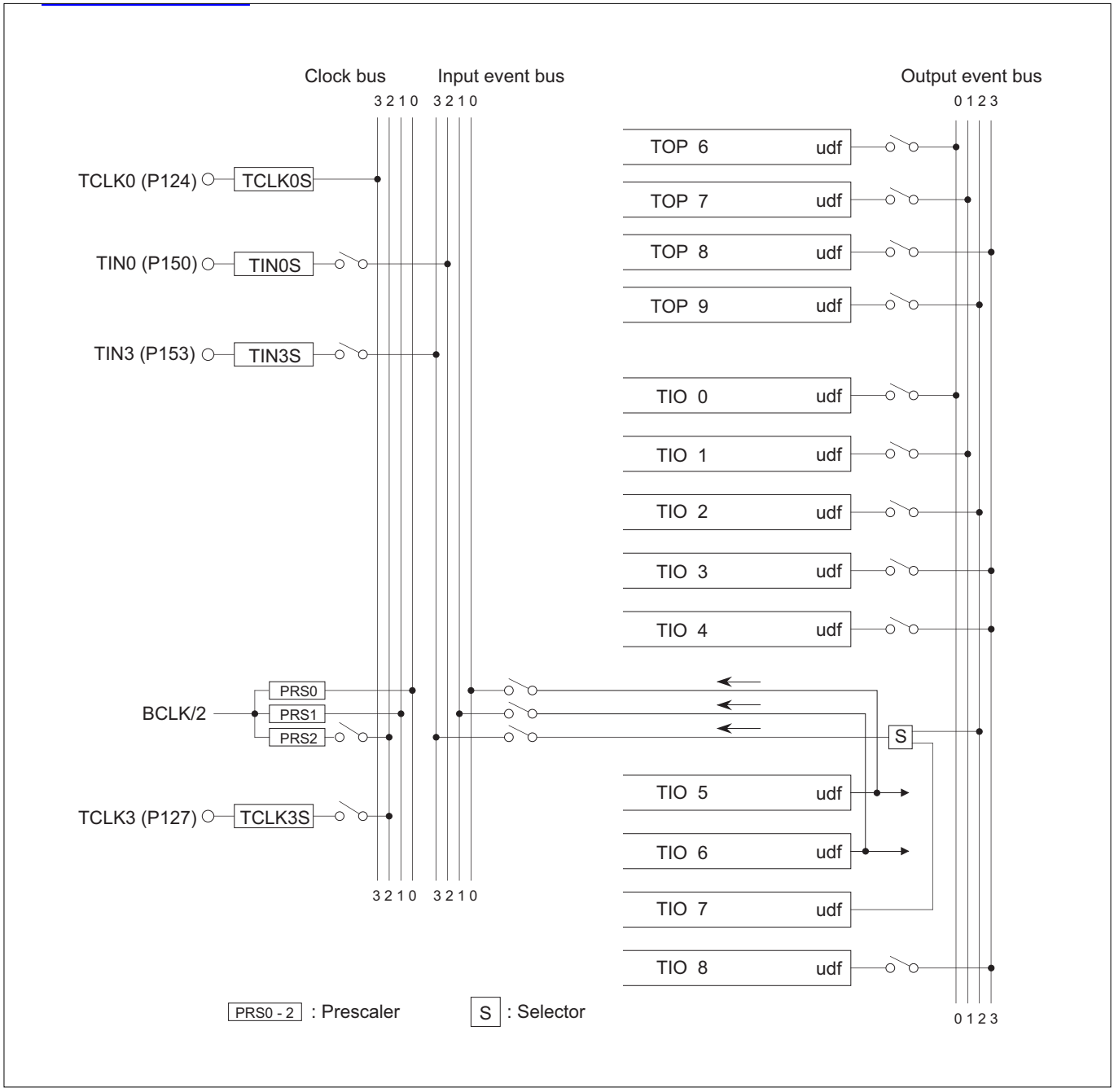


Figure 10.2.1 Conceptual Diagram of the Clock Bus and Input/Output Event Bus

The Clock Bus and Input/Output Event Bus Control Unit has the following registers:

- Clock Bus & Input Event Bus Control Register (CKIEBCR)
- Output Event Bus Control Register (OEBCR)

Clock Bus & Input Event Bus Control Register (CKIEBCR) <Address: H'0080 0201>

b8	9	10	11	12	13	14	b15
IEB3S		IEB2S		IEB1S	IEB0S		CKB2S
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8, 9	IEB3S Input event bus 3 input select bit	0X: Select external input 3 (TIN3) 10: Select output event bus 2 11: Select TIO7 output	R	W
10, 11	IEB2S Input event bus 2 input select bit	00: Select external input 0 (TIN0) 01: Does not use input event bus 2 10: Does not use input event bus 2 11: Does not use input event bus 2	R	W
12	IEB1S Input event bus 1 input select bit	0: Does not use input event bus 1 1: Select TIO6 output	R	W
13	IEB0S Input event bus 0 input select bit	0: Does not use input event bus 0 1: Select TIO5 output	R	W
14	No function assigned. Fix to "0".		0	0
15	CKB2S Clock bus 2 input select bit	0: Select prescaler 2 1: Select external clock 3 (TCLK3)	R	W

The CKIEBCR register is used to select the clock source (external input or prescaler) supplied to the clock bus and the count enable/capture signal (external input or output event bus) supplied to the input event bus.

Output Event Bus Control Register (OEBCR) <Address: H'0080 0205>

b8	9	10	11	12	13	14	b15
OEB3S			OEB2S		OEB1S		OEB0S
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8, 9	OEB3S Output event bus 3 input select bit	00: Select TOP8 output 01: Select TIO3 output 10: Select TIO4 output 11: Select TIO8 output	R	W
10	No function assigned. Fix to "0".		0	0
11	OEB2S Output event bus 2 input select bit	0: Select TOP9 output 1: Select TIO2 output	R	W
12	No function assigned. Fix to "0".		0	0
13	OEB1S Output event bus 1 input select bit	0: Select TOP7 output 1: Select TIO1 output	R	W
14	No function assigned. Fix to "0".		0	0
15	OEB0S Output event bus 0 input select bit	0: Select TOP6 output 1: Select TIO0 output	R	W

The OEBCR register is used to select the timer (TOP or TIO) whose underflow signal is supplied to the output event bus.

10.2.4 Input Processing Control Unit

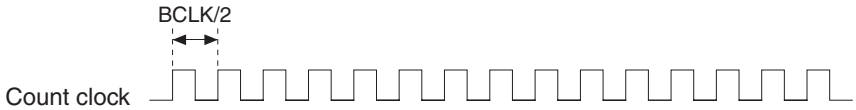
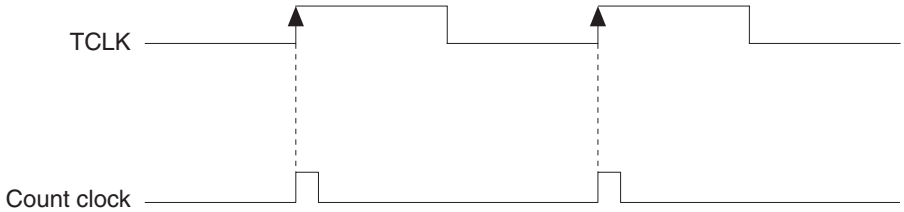
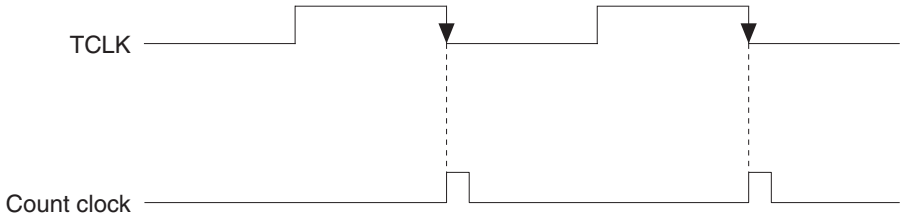
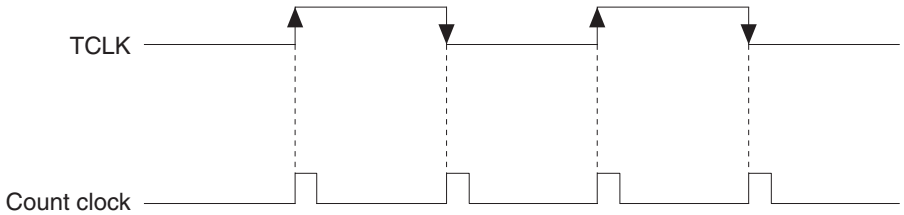
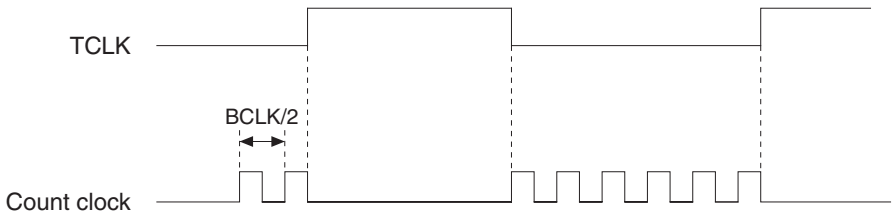
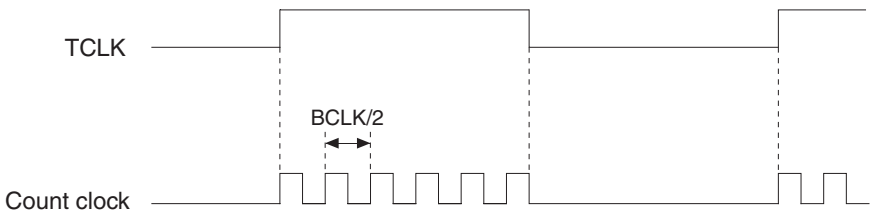
The Input Processing Control Unit processes TCLK and TIN input signals to the MJT. In TCLK input processing, it selects the source of TCLK signal, and for external input, it selects the active edge (rising or falling or both) or level ("H" or "L") of the signal, at which to generate the clock signal supplied to the clock bus.

In TIN input processing, the unit selects the active edge (rising or falling or both) or level ("H" or "L") of the signal, at which to generate the enable, measure or count source signal for each timer or the signal supplied to each event bus.

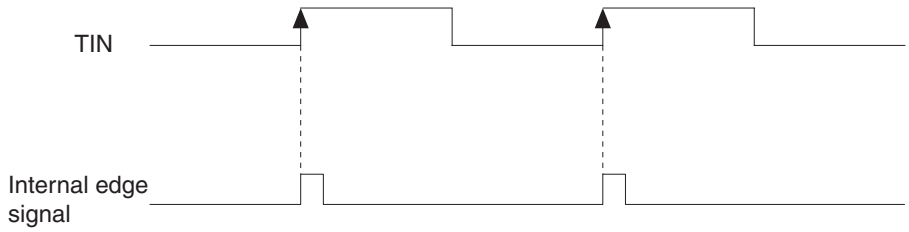
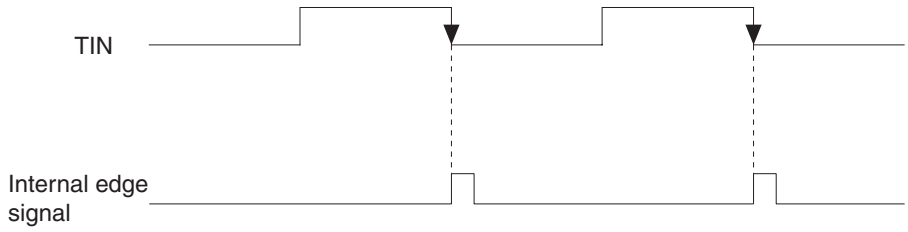
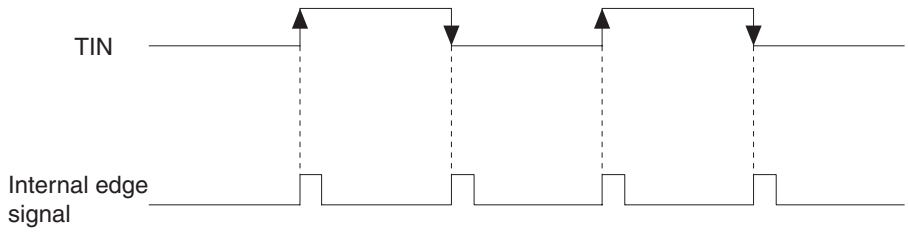
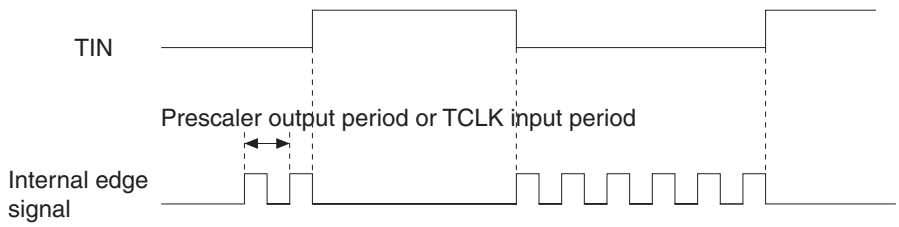
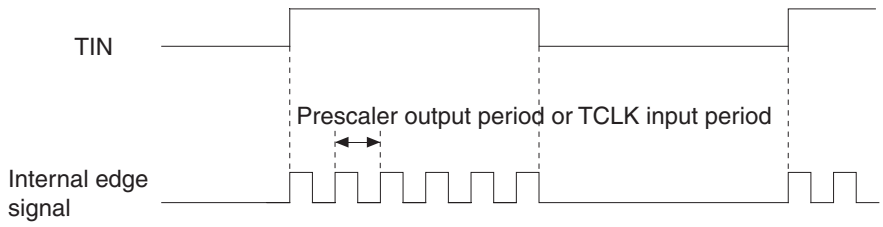
Following input processing registers are included:

- TLCK Input Processing Control Register (TCLKCR)
- TIN Input Processing Control Register 0 (TINCR0)
- TIN Input Processing Control Register 3 (TINCR3)
- TIN Input Processing Control Register 4 (TINCR4)

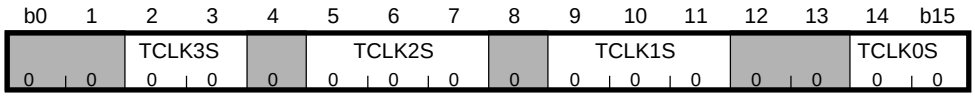
(1) Functions of TCLK Input Processing Control Registers

Item	Function
BCLK/2	
Rising edge	
Falling edge	
Both edges	
L level	
H level	

(2) Functions of TIN Input Processing Control Registers

Item	Function
Rising edge	
Falling edge	
Both edges	
L level	
H level	

TLCK Input Processing Control Register (TCLKCR) <Address: H'0080 0210>



<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0, 1	No function assigned. Fix to "0".		0	0
2, 3	TCLK3S TCLK3 input processing select bit	00: BCLK/2 01: Rising edge 10: Falling edge 11: Both edges	R	W
4	No function assigned. Fix to "0".		0	0
5–7	TCLK2S TCLK2 input processing select bit	000: Disable input 001: Rising edge 010: Falling edge 011: Both edges 100: L level 101: L level 110: H level 111: H level	R	W
8	No function assigned. Fix to "0".		0	0
9–11	TCLK1S TCLK1 input processing select bit	000: Disable input 001: Rising edge 010: Falling edge 011: Both edges 100: L level 101: L level 110: H level 111: H level	R	W
12,13	No function assigned. Fix to "0".		0	0
14,15	TCLK0S TCLK0 input processing select bit	00: BCLK/2 01: Rising edge 10: Falling edge 11: Both edges	R	W

Note: • This register must always be accessed in halfwords.

TIN Input Processing Control Register 0 (TINCR0) <Address: H'0080 0212>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
	TIN4S				TIN3S					TIN2S		TIN1S		TIN0S	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	No function assigned. Fix to "0".		0	0
1–3	TIN4S Reserved bit	Fix to "0".	0	0
4	No function assigned. Fix to "0".		0	0
5–7	TIN3S TIN3 input processing select bit	000: Disable input 001: Rising edge 010: Falling edge 011: Both edges 100: L level 101: L level 110: H level 111: H level	R	W
8, 9	No function assigned. Fix to "0".		0	0
10,11	TIN2S Reserved bit	Fix to "0".	0	0
12,13	TIN1S Reserved bit	Fix to "0".	0	0
14,15	TIN0S TIN0 input processing select bit	00: Disable input 01: Rising edge 10: Falling edge 11: Both edges	R	W

Note: • This register must always be accessed in halfwords.

TIN Input Processing Control Register 3 (TINCR3) <Address: H'0080 0218>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
TIN19S		TIN18S		TIN17S		TIN16S		TIN15S		TIN14S		TIN13S		TIN12S	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0, 1	TIN19S (TIN19 input processing select bit)	00: Disable input	R	W
2, 3	TIN18S (TIN18 input processing select bit)	01: Rising edge		
4, 5	TIN17S (TIN17 input processing select bit)	10: Falling edge		
6, 7	TIN16S (TIN16 input processing select bit)	11: Both edges		
8, 9	TIN15S (Reserved bit)	Fix to "0".	0	0
10, 11	TIN14S (Reserved bit)			
12, 13	TIN13S (Reserved bit)			
14, 15	TIN12S (Reserved bit)			

Note: • This register must always be accessed in halfwords.

TIN Input Processing Control Register 4 (TINCR4) <Address: H'0080 021A>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
TIN33S		TIN32S		TIN31S		TIN30S		TIN23S		TIN22S		TIN21S		TIN20S	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0, 1	TIN33S (Reserved bit)	Fix to "0".	0	0
2, 3	TIN32S (Reserved bit)			
4, 5	TIN31S (Reserved bit)			
6, 7	TIN30S (Reserved bit)			
8, 9	TIN23S (TIN23 input processing select bit)	00: Disable input	R	W
10, 11	TIN22S (TIN22 input processing select bit)	01: Rising edge		
12, 13	TIN21S (TIN21 input processing select bit)	10: Falling edge		
14, 15	TIN20S (TIN20 input processing select bit)	11: Both edges		

Note: • This register must always be accessed in halfwords.

10.2.5 Output Flip-flop Control Unit

The Output Flip-flop Control Unit controls the flip-flops (F/F) provided for each timer. Following flip-flop control registers are included:

- F/F Source Select Register 0 (FFS0)
- F/F Source Select Register 1 (FFS1)
- F/F Protect Register 0 (FFP0)
- F/F Protect Register 1 (FFP1)
- F/F Data Register 0 (FFD0)
- F/F Data Register 1 (FFD1)

The timing at which signals are generated to the output flip-flop by each timer are shown in Table 10.2.5. (Note that this timing is different from one at which signals are output from the timer to the output event bus.)

10.2.5 Timing at Which Signals Are Generated to the Output Flip-Flop by Each Timer

Timer	Mode	Timing at which signals are generated to the output flip-flop
TOP	Single-shot output mode	When counter is enabled or underflows
	Delayed single-shot output mode	When counter underflows
	Continuous output mode	When counter is enabled or underflows
TIO	Measure clear input mode	When counter underflows
	Measure free-run input mode	When counter underflows
	Noise processing input mode	When counter underflows
	PWM output mode	When counter is enabled or underflows
	Single-shot output mode	When counter is enabled or underflows
	Delayed single-shot output mode	When counter underflows
	Continuous output mode	When counter is enabled or underflows
TMS	(16-bit measure input)	No signals generated
TML	(32-bit measure input)	No signals generated

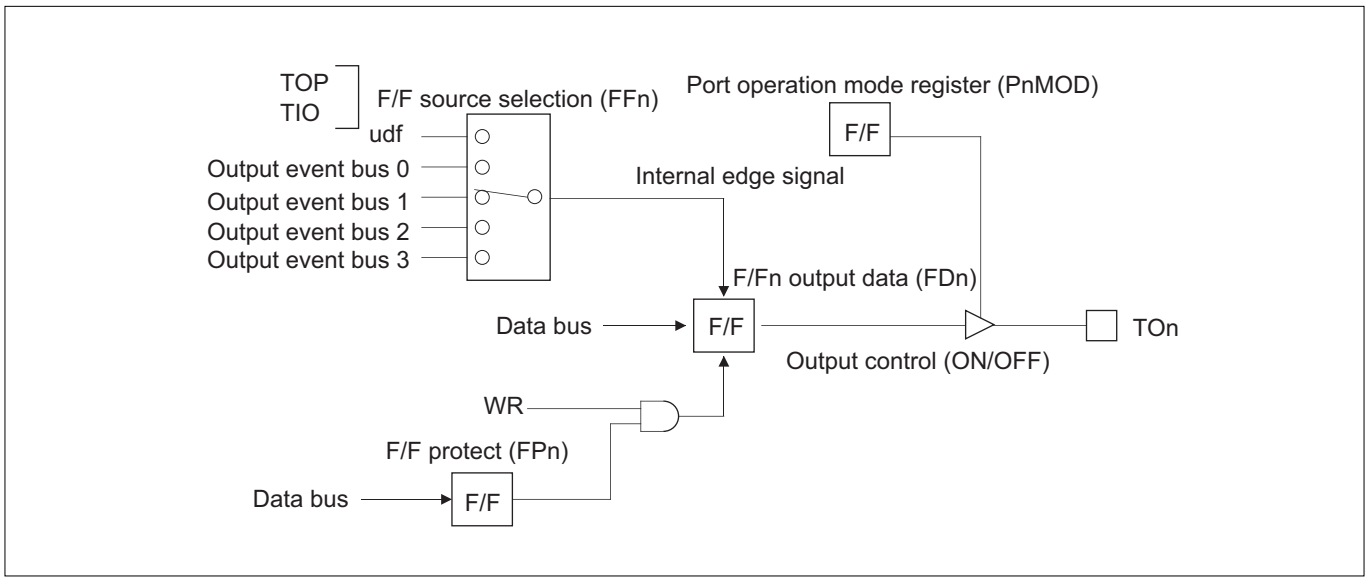


Figure 10.2.2 Configuration of the F/F Output Circuit Table

F/F Source Select Register 0 (FFS0) <Address: H'0080 0220>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
0	0	0	FF15 0	FF14 0	FF13 0	FF12 0	FF11 0	FF10 0	0	FF9 0	0	FF8 0	0	FF7 0	FF6 0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0–2	No function assigned. Fix to "0".		0	0
3	FF15 F/F15 source select bit	0: TIO4 output 1: Output event bus 0	R	W
4	FF14 F/F14 source select bit	0: TIO3 output 1: Output event bus 0	R	W
5	FF13 F/F13 source select bit	0: TIO2 output 1: Output event bus 3	R	W
6	FF12 F/F12 source select bit	0: TIO1 output 1: Output event bus 2	R	W
7	FF11 F/F11 source select bit	0: TIO0 output 1: Output event bus 1	R	W
8, 9	FF10 F/F10 source select bit	00: TOP10 output 01: TOP10 output 10: Output event bus 0 11: Output event bus 1	R	W
10, 11	FF9 F/F9 source select bit	00: TOP9 output 01: TOP9 output 10: Output event bus 0 11: Output event bus 1	R	W
12, 13	FF8 F/F8 source select bit	00: TOP8 output 01: Output event bus 0 10: Output event bus 1 11: Output event bus 2	R	W
14	FF7 F/F7 source select bit	0: TOP7 output 1: Output event bus 0	R	W
15	FF6 F/F6 source select bit	0: TOP6 output 1: Output event bus 1	R	W

Note: • This register must always be accessed in halfwords.

F/F Source Select Register 1 (FFS1) <Address: H'0080 0223>

b8	9	10	11	12	13	14	b15
FF19		FF18		FF17		FF16	
0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
8, 9	FF19 F/F19 source select bit	00: TIO8 output 01: TIO8 output 10: Output event bus 0 11: Output event bus 1	R	W
10, 11	FF18 F/F18 source select bit	00: TIO7 output 01: TIO7 output 10: Output event bus 0 11: Output event bus 1	R	W
12, 13	FF17 F/F17 source select bit	00: TIO6 output 01: TIO6 output 10: Output event bus 0 11: Output event bus 1	R	W
14, 15	FF16 F/F16 source select bit	00: TIO5 output 01: Output event bus 0 10: Output event bus 1 11: Output event bus 3	R	W

These registers select the signal source for each output F/F (flip-flop). This signal source can be chosen to be a signal from the internal output bus or an underflow output from each timer.

F/F Protect Register 0 (FFP0) <Address: H'0080 0224>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
FP15	FP14	FP13	FP12	FP11	FP10	FP9	FP8	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	FP15 (F/F15 protect bit)	0: Enable write to F/F output bit	R	W
1	FP14 (F/F14 protect bit)	1: Disable write to F/F output bit		
2	FP13 (F/F13 protect bit)			
3	FP12 (F/F12 protect bit)			
4	FP11 (F/F11 protect bit)			
5	FP10 (F/F10 protect bit)			
6	FP9 (F/F9 protect bit)			
7	FP8 (F/F8 protect bit)			
8	FP7 (F/F7 protect bit)			
9	FP6 (F/F6 protect bit)			
10	FP5 (F/F5 protect bit)			
11	FP4 (F/F4 protect bit)			
12	FP3 (F/F3 protect bit)			
13	FP2 (F/F2 protect bit)			
14	FP1 (F/F1 protect bit)			
15	FP0 (F/F0 protect bit)			

Note: • This register must always be accessed in halfwords.

F/F Protect Register 1 (FFP1) <Address: H'0080 0229>

b8	9	10	11	12	13	14	b15
			FP20	FP19	FP18	FP17	FP16
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	FP20 (F/F20 protect bit)	0: Enable write to F/F output bit	R	W
12	FP19 (F/F19 protect bit)	1: Disable write to F/F output bit		
13	FP18 (F/F18 protect bit)			
14	FP17 (F/F17 protect bit)			
15	FP16 (F/F16 protect bit)			

These registers enable or disable write to each output F/F (flip-flop). If write to any output F/F is disabled, writing to the corresponding F/F data register has no effect.

F/F Data Register 0 (FFD0) <Address: H'0080 0226>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	FD15 (F/F15 output data bit)	0: F/F output data = 0	R	W
1	FD14 (F/F14 output data bit)	1: F/F output data = 1		
2	FD13 (F/F13 output data bit)			
3	FD12 (F/F12 output data bit)			
4	FD11 (F/F11 output data bit)			
5	FD10 (F/F10 output data bit)			
6	FD9 (F/F9 output data bit)			
7	FD8 (F/F8 output data bit)			
8	FD7 (F/F7 output data bit)			
9	FD6 (F/F6 output data bit)			
10	FD5 (F/F5 output data bit)			
11	FD4 (F/F4 output data bit)			
12	FD3 (F/F3 output data bit)			
13	FD2 (F/F2 output data bit)			
14	FD1 (F/F1 output data bit)			
15	FD0 (F/F0 output data bit)			

Note: • This register must always be accessed in halfwords.

F/F Data Register 1 (FFD1) <Address: H'0080 022B>

b8	9	10	11	12	13	14	b15
			FD20	FD19	FD18	FD17	FD16
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	FD20 (F/F20 output data bit)	0: F/F output data = 0	R	W
12	FD19 (F/F19 output data bit)	1: F/F output data = 1		
13	FD18 (F/F18 output data bit)			
14	FD17 (F/F17 output data bit)			
15	FD16 (F/F16 output data bit)			

These registers are used to set the data for each output F/F (flip-flop). Although the F/F outputs normally change state depending on timer outputs, the F/F outputs can be set to 1 or cleared to 0 as necessary by writing to this register. The F/F data register can only be operated on when the F/F protect register described previously is enabled for write.

10.2.6 Interrupt Control Unit

The Interrupt Control Unit controls the interrupt request signals output to the Interrupt Controller by each timer. Following timer interrupt control registers are provided for each timer:

- TOP Interrupt Control Register 0 (TOPIR0)
- TOP Interrupt Control Register 1 (TOPIR1)
- TOP Interrupt Control Register 2 (TOPIR2)
- TOP Interrupt Control Register 3 (TOPIR3)
- TIO Interrupt Control Register 0 (TIOIR0)
- TIO Interrupt Control Register 1 (TIOIR1)
- TIO Interrupt Control Register 2 (TIOIR2)
- TMS Interrupt Control Register (TMSIR)
- TIN Interrupt Control Register 0 (TINIR0)
- TIN Interrupt Control Register 1 (TINIR1)
- TIN Interrupt Control Register 4 (TINIR4)
- TIN Interrupt Control Register 5 (TINIR5)
- TIN Interrupt Control Register 6 (TINIR6)

For interrupts which have only one interrupt source in the interrupt vector table, no interrupt control registers are included in the timer, and the interrupt status flags are automatically managed within the Interrupt Controller. The relevant timer interrupt is the following.
(For details, see Chapter 5, “Interrupt Controller.”)

- TOP10 MJT Output Interrupt 5 (IRQ5)

For interrupts which have two or more interrupt sources in the interrupt vector table, interrupt control registers are included, with which to control interrupt requests and determine interrupt input. Therefore, the status flags in the Interrupt Controller only serve as a bit to determine interrupt requests from interrupt-enabled sources and cannot be accessed for write.

(1) Interrupt request status bit

This status bit is used to determine whether there is an interrupt request. When an interrupt request occurs, this bit is set in hardware (cannot be set in software). The status bit is cleared by writing "0". Writing "1" has no effect; the bit retains the status it had before the write. Because this status bit is unaffected by the interrupt mask bit, it can be used to inspect the operating status of peripheral functions.
In interrupt handling, make sure that within the grouped interrupt request status, only the status bit for the interrupt request that has been serviced is cleared. If the status bit for any interrupt request that has not been serviced is cleared, the pending interrupt request is cleared simultaneously with its status bit.

(2) Interrupt mask bit

This bit is used to disable unnecessary interrupts within the grouped interrupt. Set this bit to "0" to enable interrupts or "1" to disable interrupts.

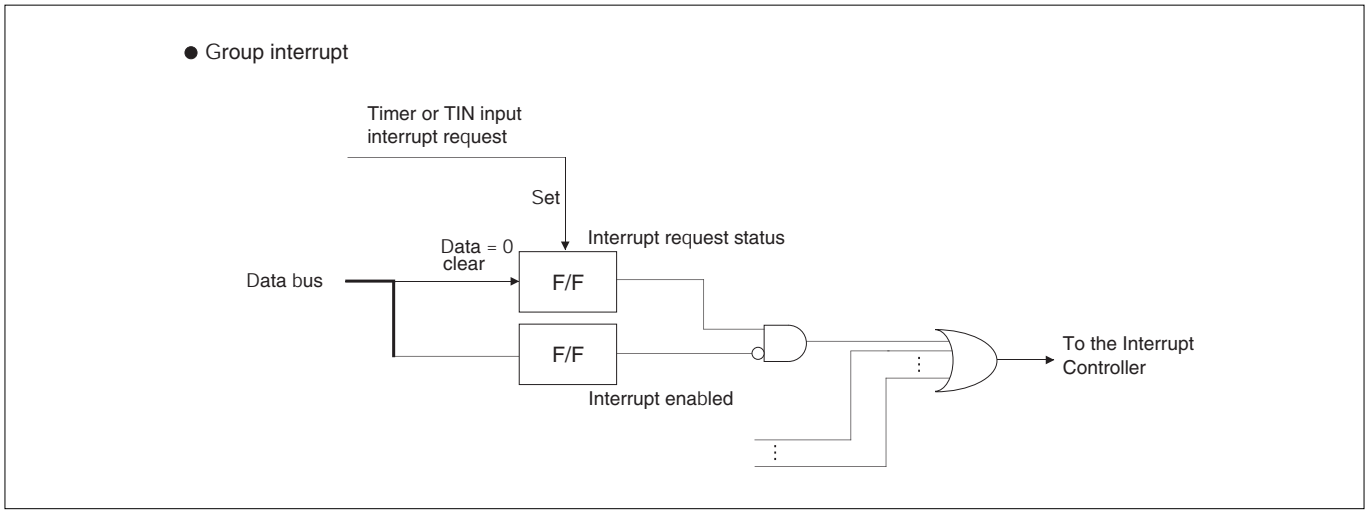
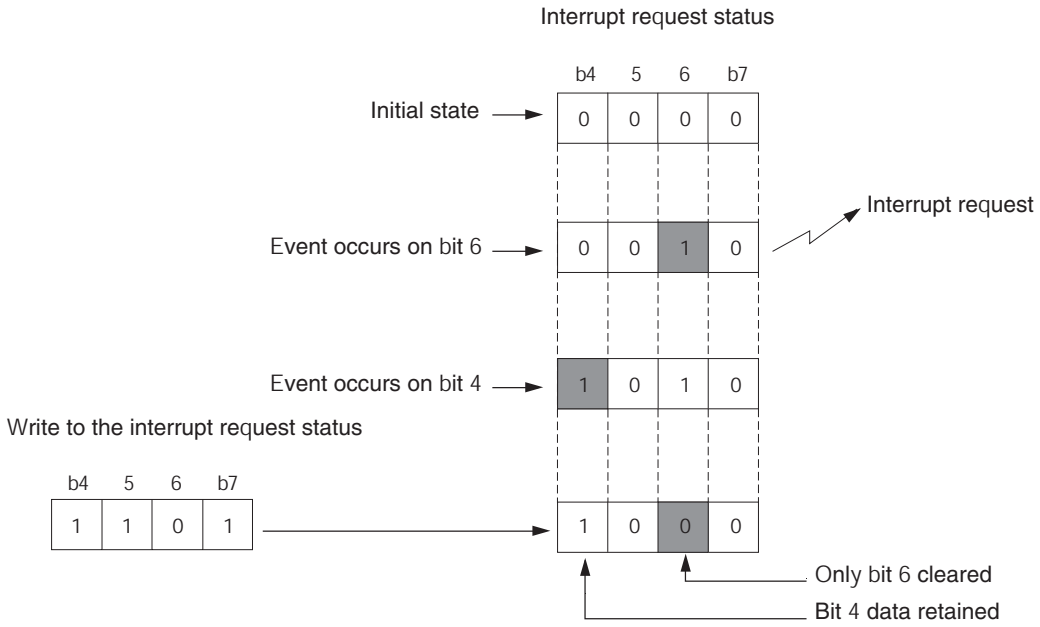


Figure 10.2.3 Interrupt Request Status and Mask Registers

● Example for clearing interrupt request status



● Program example

• To clear the Interrupt Request Status Register (ISTREG) interrupt request status 1, ISTAT1 (0x02 bit)

○ `ISTREG = 0xfd;` /* Clear ISTAT1 (0x02 bit) only */

To clear an interrupt request status, always be sure to write "1" to all other interrupt request status bits. At this time, avoid using a logic operation like the one shown below. Because it requires three step-ISTREG read, logic operation and write, if another interrupt request occurs between the read and write, status may be inadvertently cleared.

× `ISTREG &= 0xfd;` /* Clear ISTAT1 (0x02 bit) only */

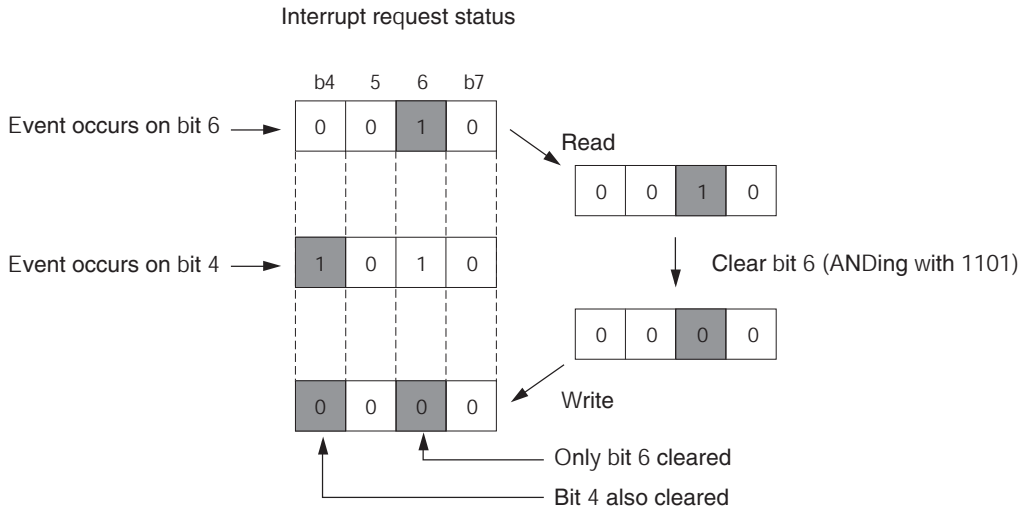


Figure 10.2.4 Example for Clearing Interrupt Request Status

The table below shows the relationship between the interrupt signals generated by multijunction timers and the interrupt sources input to the Interrupt Controller (ICU).

Table 10.2.6 Interrupt Signals Generated by MJT

Signal Name	Generated by	ICU Interrupt Input Source (Note 1)	No. of Input Sources
IRQ0	TIO0, TIO1, TIO2, TIO3	MJT output interrupt 0	4
IRQ1	TOP6, TOP7	MJT output interrupt 1	2
IRQ2	TOP0, TOP1, TOP2, TOP3, TOP4, TOP5	MJT output interrupt 2	6
IRQ3	TIO8, TIO9	MJT output interrupt 3	2
IRQ4	TIO4, TIO5, TIO6, TIO7	MJT output interrupt 4	4
IRQ6	TOP8, TOP9	MJT output interrupt 6	2
IRQ7	TMS0, TMS1	MJT output interrupt 7	2
IRQ9	TIN0	MJT input interrupt 1	1
IRQ10	TIN16, TIN17, TIN18, TIN19	MJT input interrupt 2	4
IRQ11	TIN20, TIN21, TIN22, TIN23	MJT input interrupt 3	4
IRQ12	TIN3	MJT input interrupt 4	1

Note 1: See Chapter 5, “Interrupt Controller (ICU).”

Note: • TOP10 has only one interrupt source in each interrupt group, so that their status and mask registers are nonexistent in the MJT interrupt control registers. (They are controlled directly by the Interrupt Controller.)

TOP Interrupt Control Register 0 (TOPIR0) <Address: H'0080 0230>

b0	1	2	3	4	5	6	b7
0	0	TOPIS5 0	TOPIS4 0	TOPIS3 0	TOPIS2 0	TOPIS1 0	TOPIS0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0, 1	No function assigned. Fix to "0".		0	0
2	TOPIS5 (TOP5 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
3	TOPIS4 (TOP4 interrupt request status bit)	1: Interrupt requested		
4	TOPIS3 (TOP3 interrupt request status bit)			
5	TOPIS2 (TOP2 interrupt request status bit)			
6	TOPIS1 (TOP1 interrupt request status bit)			
7	TOPIS0 (TOP0 interrupt request status bit)			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

TOP Interrupt Control Register 1 (TOPIR1) <Address: H'0080 0231>

b8	9	10	11	12	13	14	b15
0	0	TOPIM5 0	TOPIM4 0	TOPIM3 0	TOPIM2 0	TOPIM1 0	TOPIM0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8, 9	No function assigned. Fix to "0".		0	0
10	TOPIM5 (TOP5 interrupt request mask bit)	0: Enable interrupt request	R	W
11	TOPIM4 (TOP4 interrupt request mask bit)	1: Mask (disable) interrupt request		
12	TOPIM3 (TOP3 interrupt request mask bit)			
13	TOPIM2 (TOP2 interrupt request mask bit)			
14	TOPIM1 (TOP1 interrupt request mask bit)			
15	TOPIM0 (TOP0 interrupt request mask bit)			

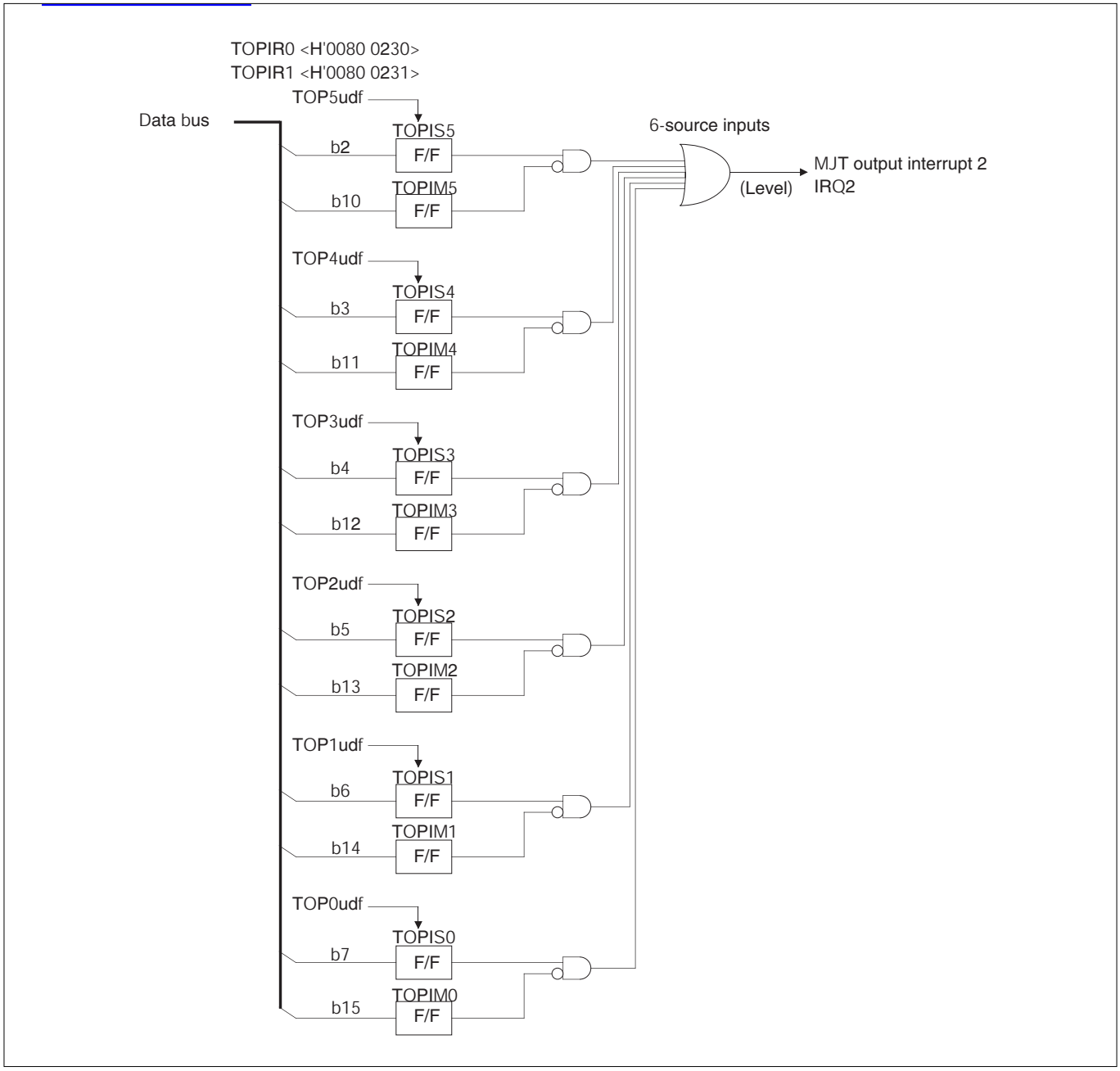


Figure 10.2.5 Block Diagram of MJT Output Interrupt 2

TOP Interrupt Control Register 2 (TOPIR2) <Address: H'0080 0232>

b0	1	2	3	4	5	6	b7
0	0	TOPIS7 0	TOPIS6 0	0	0	TOPIM7 0	TOPIM6 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0, 1	No function assigned. Fix to "0".		0	0
2	TOPIS7 (TOP7 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
3	TOPIS6 (TOP6 interrupt request status bit)	1: Interrupt requested		
4, 5	No function assigned. Fix to "0".		0	0
6	TOPIM7 (TOP7 interrupt request mask bit)	0: Enable interrupt request	R	W
7	TOPIM6 (TOP6 interrupt request mask bit)	1: Mask (disable) interrupt request		

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

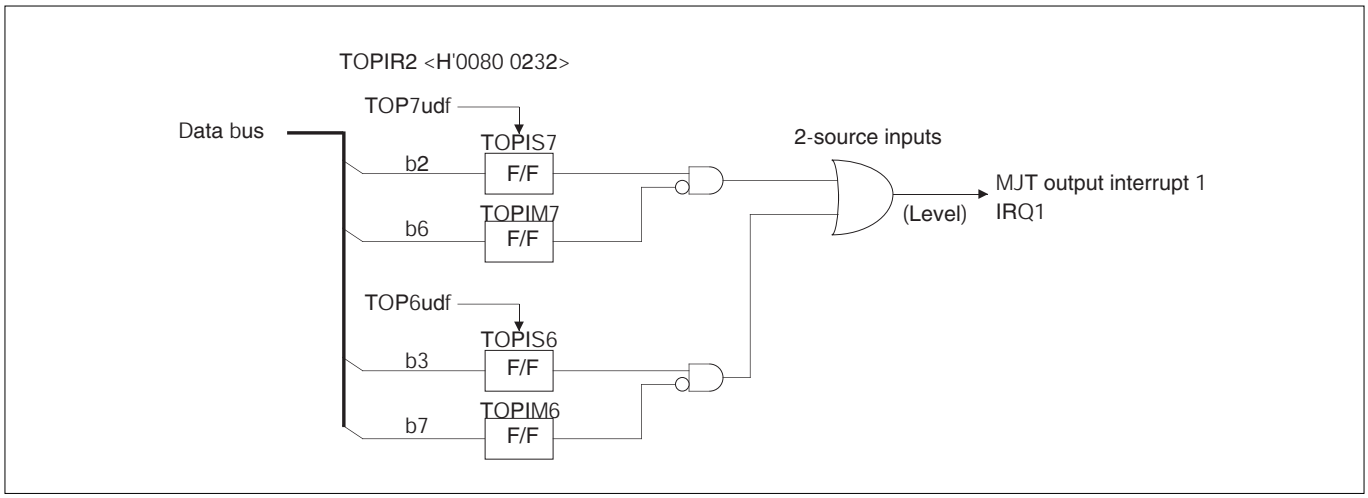


Figure 10.2.6 Block Diagram of MJT Output Interrupt 1

TOP Interrupt Control Register 3 (TOPIR3) <Address: H'0080 0233>

b8	9	10	11	12	13	14	b15
0	0	TOPIS9	TOPIS8	0	0	TOPIM9	TOPIM8
		0	0			0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8,9	No function assigned. Fix to "0".		0	0
10	TOPIS9 (TOP9 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
11	TOPIS8 (TOP8 interrupt request status bit)	1: Interrupt requested		
12,13	No function assigned. Fix to "0".		0	0
14	TOPIM9 (TOP9 interrupt request mask bit)	0: Enable interrupt request	R	W
15	TOPIM8 (TOP8 interrupt request mask bit)	1: Mask (disable) interrupt request		

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

Note: • TOP10 has only one interrupt source in the interrupt group, so that its status and mask registers are nonexistent in the MJT interrupt control registers. (They are controlled directly by the Interrupt Controller.)

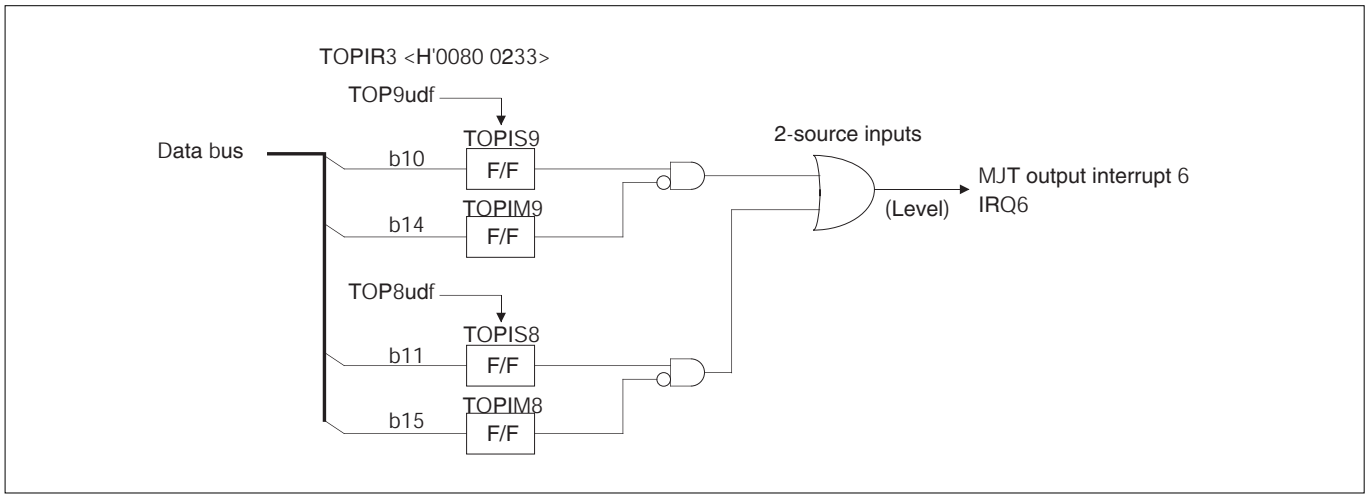


Figure 10.2.7 Block Diagram of MJT Output Interrupt 6

TIO Interrupt Control Register 0 (TIOIR0) <Address: H'0080 0234>

b0	1	2	3	4	5	6	b7
TIOIS3	TIOIS2	TIOIS1	TIOIS0	TIOIM3	TIOIM2	TIOIM1	TIOIM0
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	TIOIS3 (TIO3 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
1	TIOIS2 (TIO2 interrupt request status bit)	1: Interrupt requested		
2	TIOIS1 (TIO1 interrupt request status bit)			
3	TIOIS0 (TIO0 interrupt request status bit)			
4	TIOIM3 (TIO3 interrupt request mask bit)	0: Enable interrupt request	R	W
5	TIOIM2 (TIO2 interrupt request mask bit)	1: Mask (disable) interrupt request		
6	TIOIM1 (TIO1 interrupt request mask bit)			
7	TIOIM0 (TIO0 interrupt request mask bit)			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

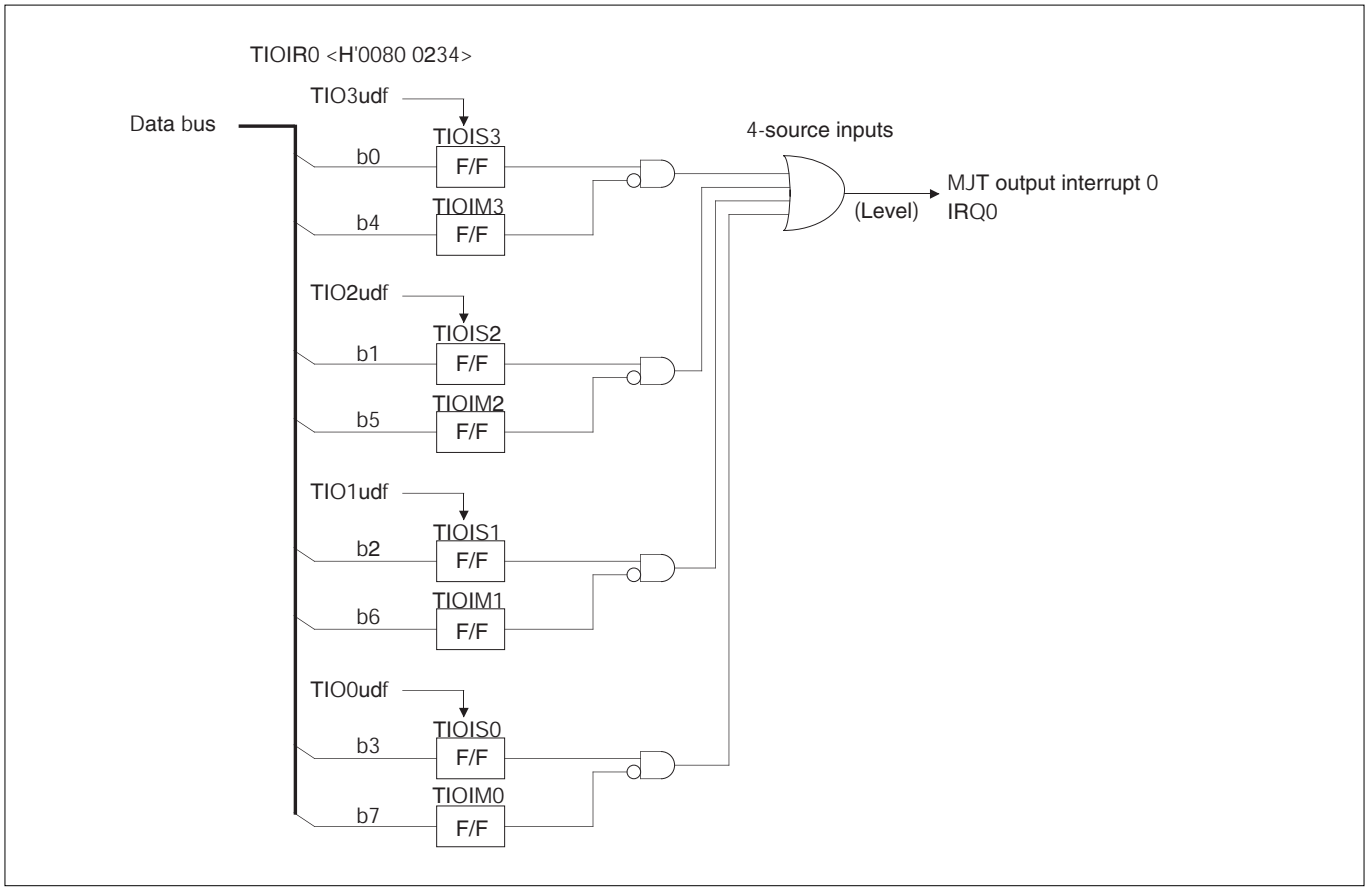


Figure 10.2.8 Block Diagram of MJT Output Interrupt 0

TIO Interrupt Control Register 1 (TIOIR1) <Address: H'0080 0235>

b8	9	10	11	12	13	14	b15
TIOIS7	TIOIS6	TIOIS5	TIOIS4	TIOIM7	TIOIM6	TIOIM5	TIOIM4
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	TIOIS7 (TIO7 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
9	TIOIS6 (TIO6 interrupt request status bit)	1: Interrupt requested		
10	TIOIS5 (TIO5 interrupt request status bit)			
11	TIOIS4 (TIO4 interrupt request status bit)			
12	TIOIM7 (TIO7 interrupt request mask bit)	0: Enable interrupt request	R	W
13	TIOIM6 (TIO6 interrupt request mask bit)	1: Mask (disable) interrupt request		
14	TIOIM5 (TIO5 interrupt request mask bit)			
15	TIOIM4 (TIO4 interrupt request mask bit)			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

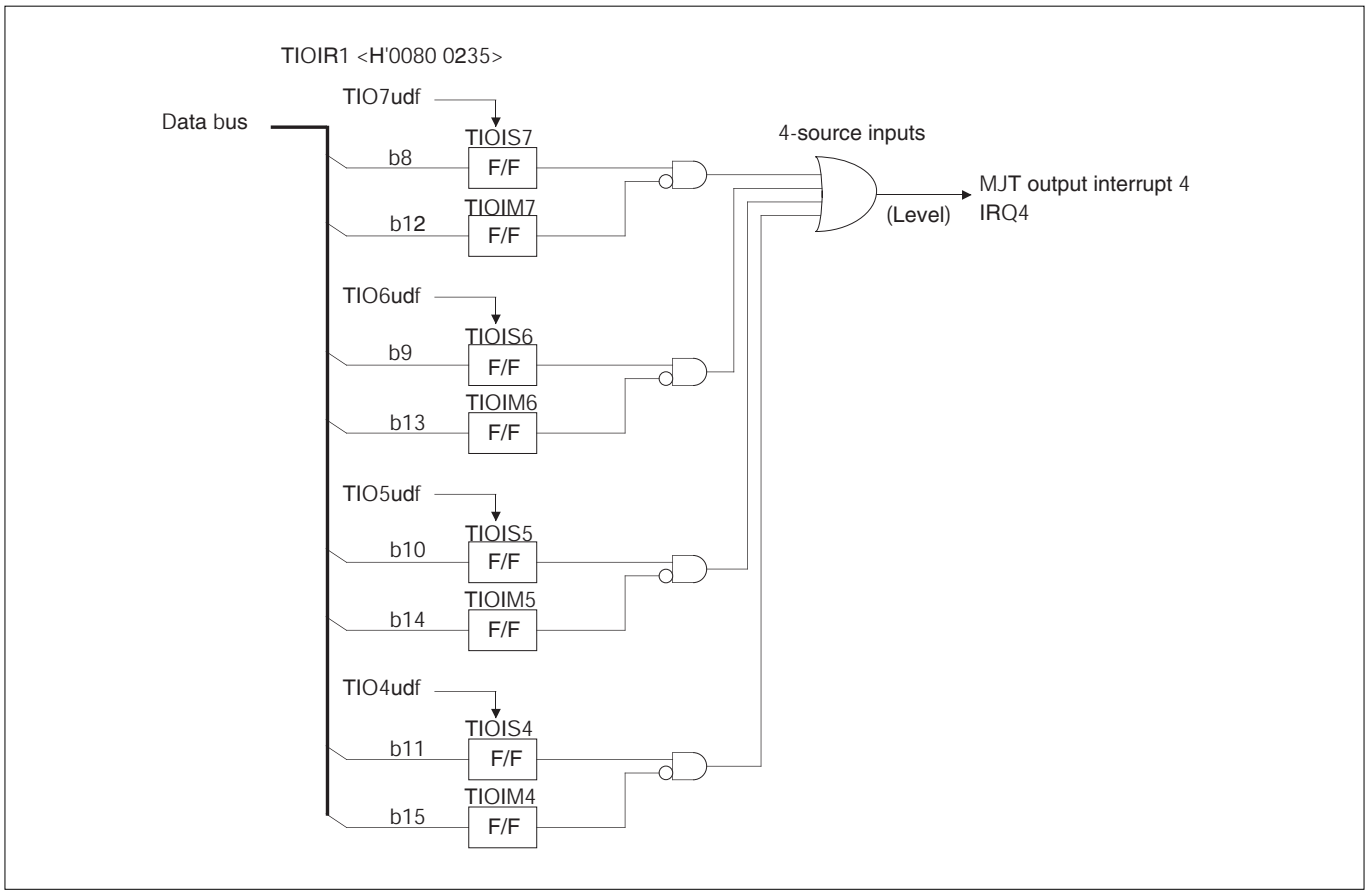


Figure 10.2.9 Block Diagram of MJT Output Interrupt 4

TIO Interrupt Control Register 2 (TIOIR2) <Address: H'0080 0236>

b0	1	2	3	4	5	6	b7
0	0	TIOIS9 0	TIOIS8 0	0	0	TIOIM9 0	TIOIM8 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0, 1	No function assigned. Fix to "0".		0	0
2	TIOIS9 (TIO9 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
3	TIOIS8 (TIO8 interrupt request status bit)	1: Interrupt requested		
4, 5	No function assigned. Fix to "0".		0	0
6	TIOIM9 (TIO9 interrupt request mask bit)	0: Enable interrupt request	R	W
7	TIOIM8 (TIO8 interrupt request mask bit)	1: Mask (disable) interrupt request		

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

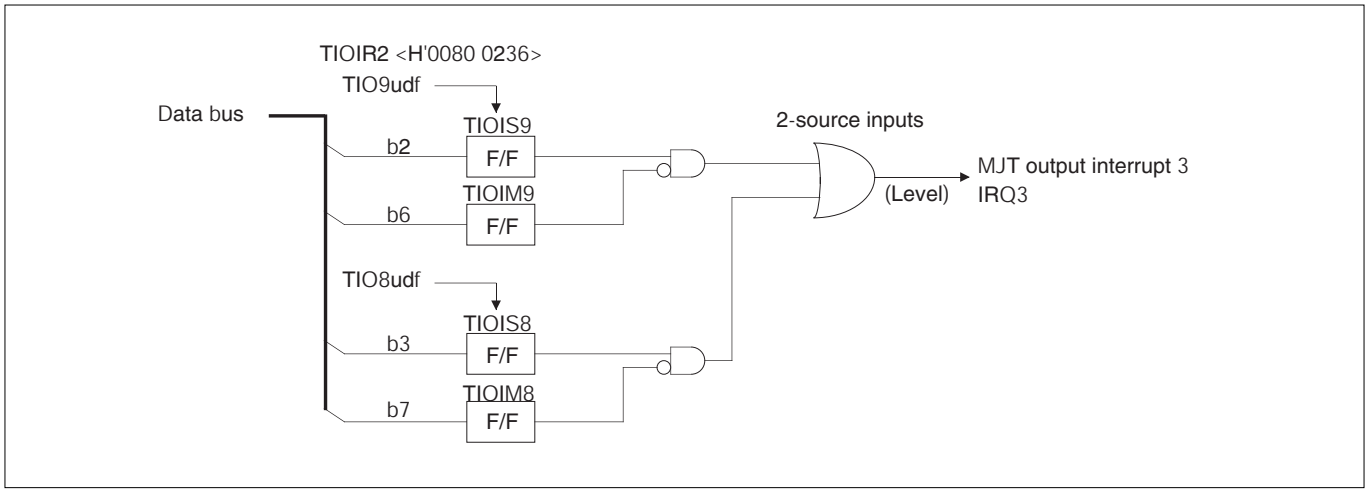


Figure 10.2.10 Block Diagram of MJT Output Interrupt 3

TMS Interrupt Control Register (TMSIR) <Address: H'0080 0237>

b8	9	10	11	12	13	14	b15
0	0	TMSIS1 0	TMSIS0 0	0	0	TMSIM1 0	TMSIM0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8, 9	No function assigned. Fix to "0".		0	0
10	TMSIS1 (TMS1 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
11	TMSIS0 (TMS0 interrupt request status bit)	1: Interrupt requested		
12, 13	No function assigned. Fix to "0".		0	0
14	TMSIM1 (TMS1 interrupt request mask bit)	0: Enable interrupt request	R	W
15	TMSIM0 (TMS0 interrupt request mask bit)	1: Mask (disable) interrupt request		

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

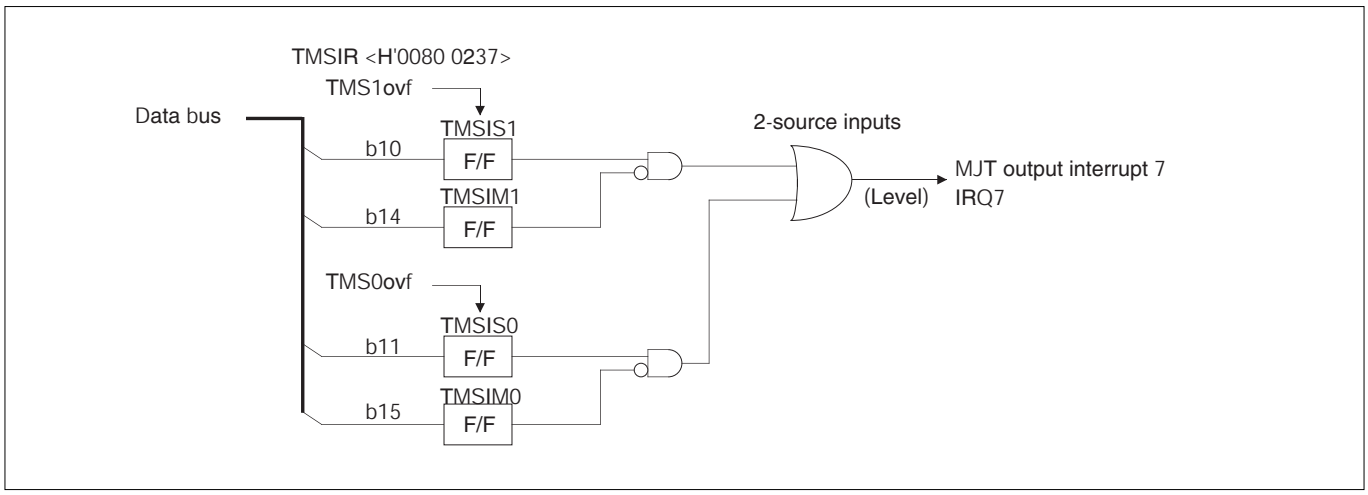


Figure 10.2.11 Block Diagram of MJT Output Interrupt 7

TIN Interrupt Control Register 0 (TINIR0) <Address: H'0080 0238>

b0	1	2	3	4	5	6	b7
0	0	0	TINIS0 0	0	TINIM2 0	TINIM1 0	TINIM0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0-2	No function assigned. Fix to "0".		0	0
3	TINIS0 TIN0 interrupt request status bit	0: Interrupt not requested 1: Interrupt requested	R(Note 1)	
4	No function assigned. Fix to "0".		0	0
5	TINIM2 Reserved bit	Fix to "0".	0	0
6	TINIM1 Reserved bit			
7	TINIM0 TIN0 interrupt request mask bit	0: Enable interrupt request 1: Mask (disable) interrupt request	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

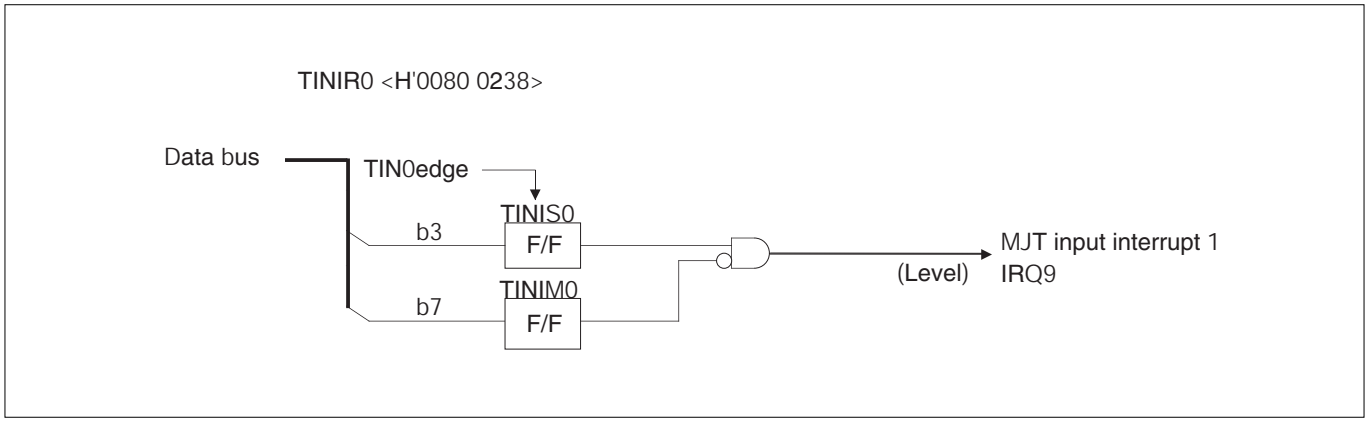


Figure 10.2.12 Block Diagram of MJT Input Interrupt 1

TIN Interrupt Control Register 1 (TINIR1) <Address: H'0080 0239>

b8	9	10	11	12	13	14	b15
0	0	0	TINIS3 0	TINIM6 0	TINIM5 0	TINIM4 0	TINIM3 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8-10	No function assigned. Fix to "0".		0	0
11	TINIS3 TIN3 interrupt request status bit	0: Interrupt not requested 1: Interrupt requested	R(Note 1)	
12	TINIM6 Reserved bit	Fix to "0".	0	0
13	TINIM5 Reserved bit			
14	TINIM4 Reserved bit			
15	TINIM3 TIN3 interrupt request mask bit	0: Enable interrupt request 1: Mask (disable) interrupt request	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

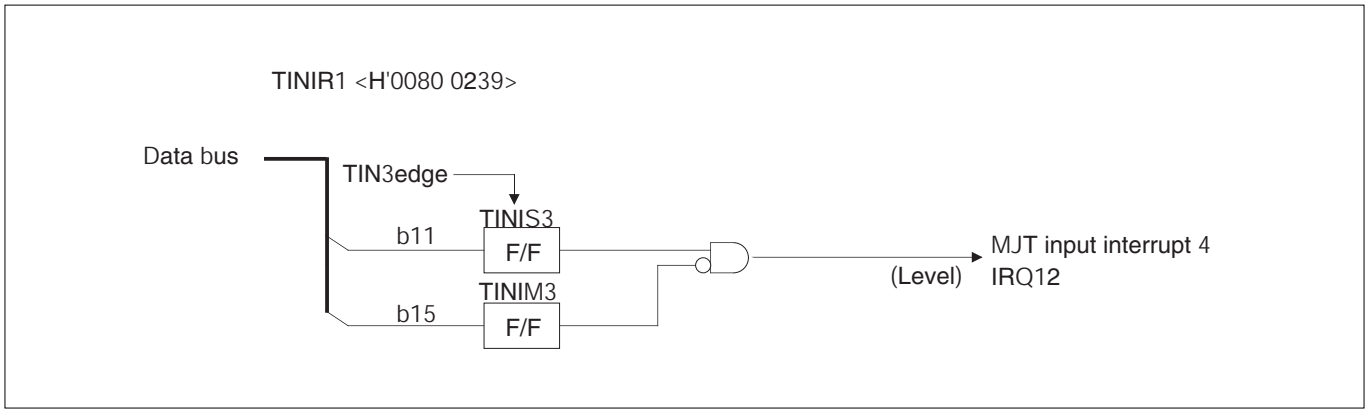


Figure 10.2.13 Block Diagram of MJT Input Interrupt 4

TIN Interrupt Control Register 4 (TINIR4) <Address: H'0080 023C>

b0	1	2	3	4	5	6	b7
TINIS19	TINIS18	TINIS17	TINIS16				
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	TINIS19 (TIN19 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
1	TINIS18 (TIN18 interrupt request status bit)	1: Interrupt requested		
2	TINIS17 (TIN17 interrupt request status bit)			
3	TINIS16 (TIN16 interrupt request status bit)			
4-7	No function assigned. Fix to "0".		0	0

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

TIN Interrupt Control Register 5 (TINIR5) <Address: H'0080 023D>

b8	9	10	11	12	13	14	b15
TINIM19	TINIM18	TINIM17	TINIM16	TINIM15	TINIM14	TINIM13	TINIM12
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	TINIM19 (TIN19 interrupt request mask bit)	0: Enable interrupt request	R	W
9	TINIM18 (TIN18 interrupt request mask bit)	1: Mask (disable) interrupt request		
10	TINIM17 (TIN17 interrupt request mask bit)			
11	TINIM16 (TIN16 interrupt request mask bit)			
12	TINIM15 (Reserved bit)	Fix to "0".	0	0
13	TINIM14 (Reserved bit)			
14	TINIM13 (Reserved bit)			
15	TINIM12 (Reserved bit)			

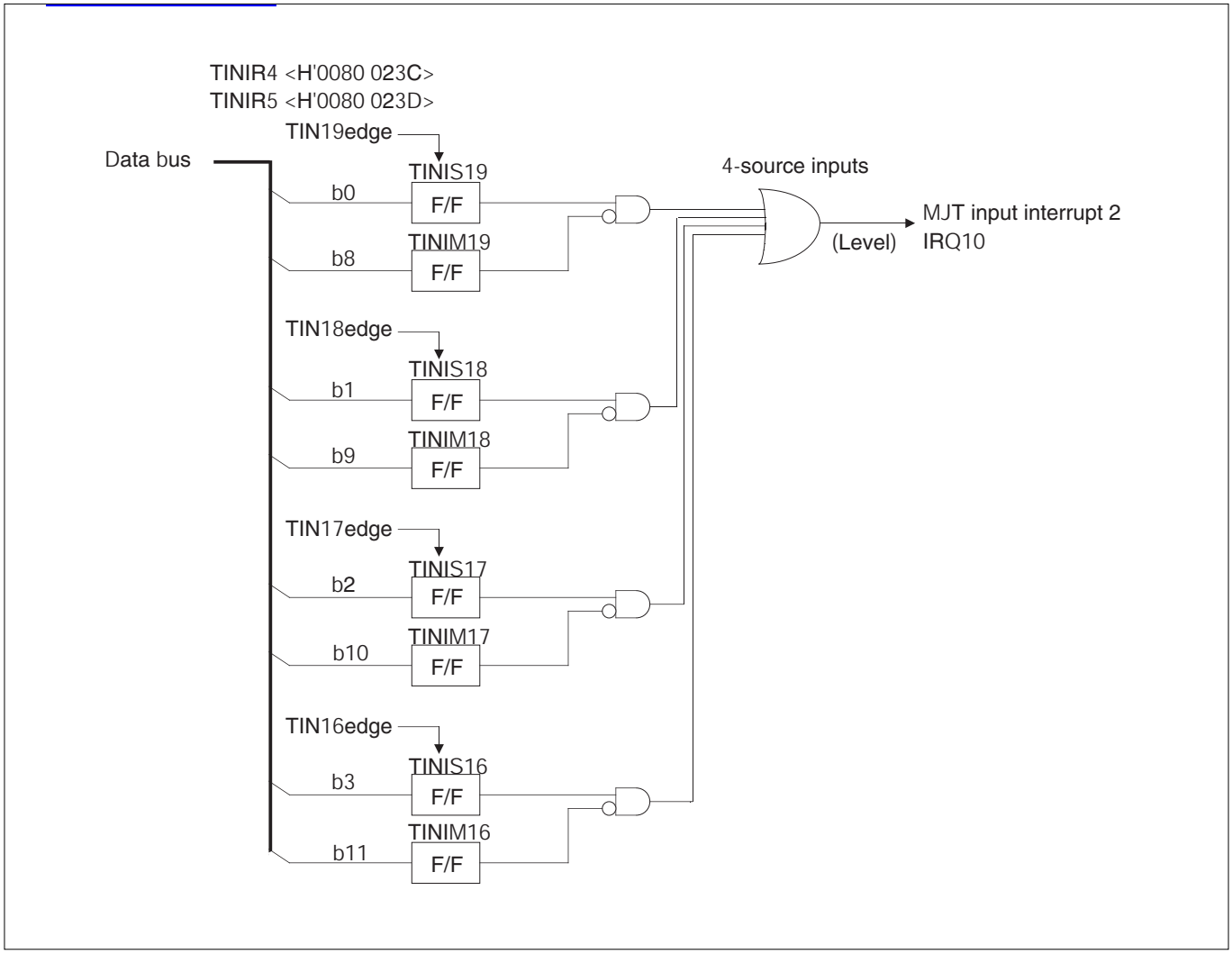


Figure 10.2.14 Block Diagram of MJT Input Interrupt 2

TIN Interrupt Control Register 6 (TINIR6) <Address: H'0080 023E>

b0	1	2	3	4	5	6	b7
TINIS23	TINIS22	TINIS21	TINIS20	TINIM23	TINIM22	TINIM21	TINIM20
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	TINIS23 (TIN23 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
1	TINIS22 (TIN22 interrupt request status bit)	1: Interrupt requested		
2	TINIS21 (TIN21 interrupt request status bit)			
3	TINIS20 (TIN20 interrupt request status bit)			
4	TINIM23 (TIN23 interrupt request mask bit)	0: Enable interrupt request	R	W
5	TINIM22 (TIN22 interrupt request mask bit)	1: Mask (disable) interrupt request		
6	TINIM21 (TIN21 interrupt request mask bit)			
7	TINIM20 (TIN20 interrupt request mask bit)			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

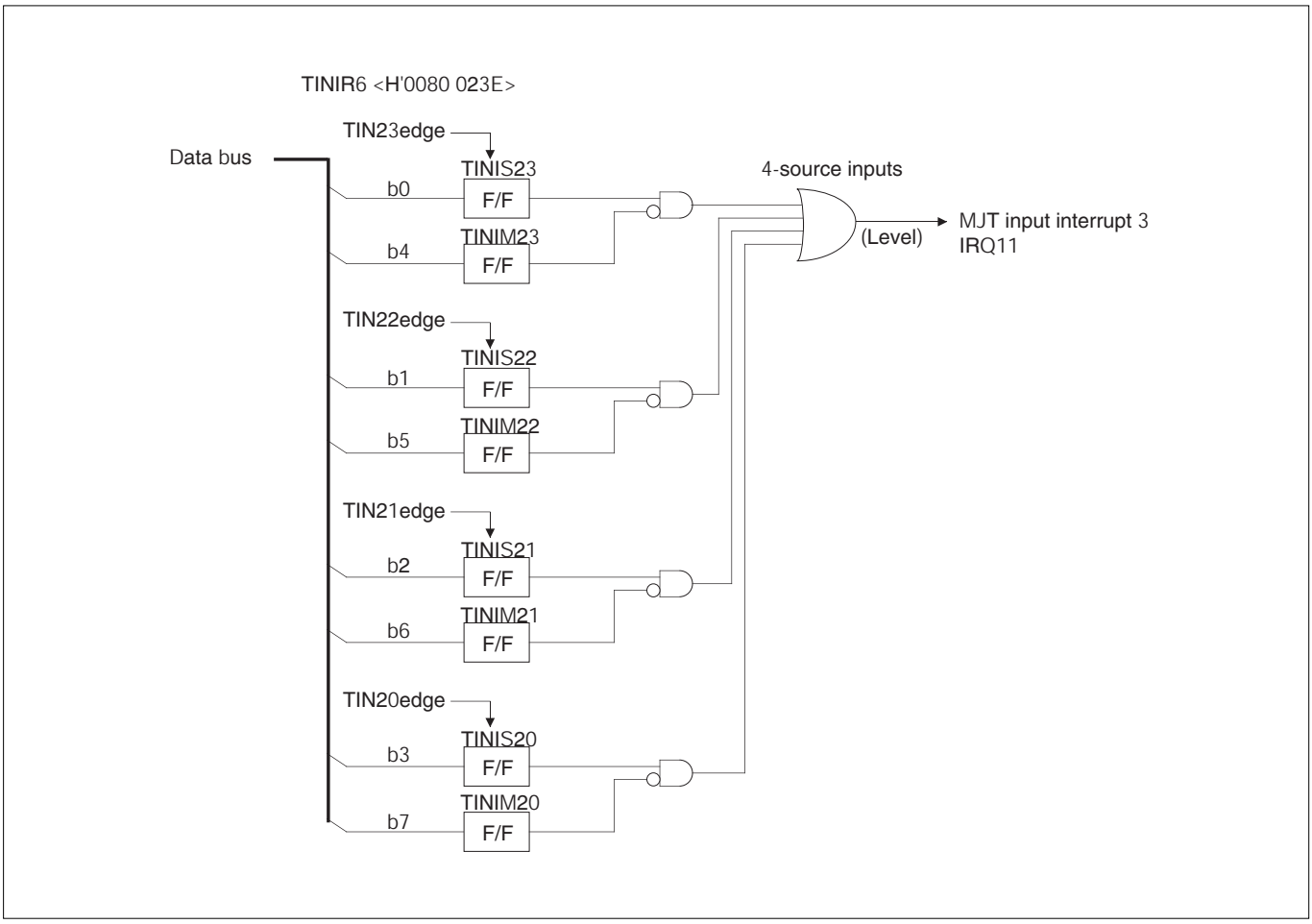


Figure 10.2.15 Block Diagram of MJT Input Interrupt 3

10.3 TOP (Output-Related 16-Bit Timer)

10.3.1 Outline of TOP

TOP (Timer OutPut) is an output-related 16-bit timer, whose operation mode can be selected from the following by mode switching in software:

- Single-shot output mode
- Delayed single-shot output mode
- Continuous output mode

Table 10.3.1 below shows specifications of TOP. Figure 10.3.1 shows a block diagram of TOP.

Table 10.3.1 Specifications of TOP (Output-Related 16-Bit Timer)

Item	Specification
Number of channels	11 channels
Counter	16-bit down-counter
Reload register	16-bit reload register
Correction register	16-bit correction register
Timer startup	Started by writing to the enable bit in software or enabled by external input (rising or falling edge or both)
Mode switching	<With correction function> • Single-shot output mode • Delayed single-shot output mode <Without correction function> • Continuous output mode
Interrupt request generation	Can be generated by a counter underflow

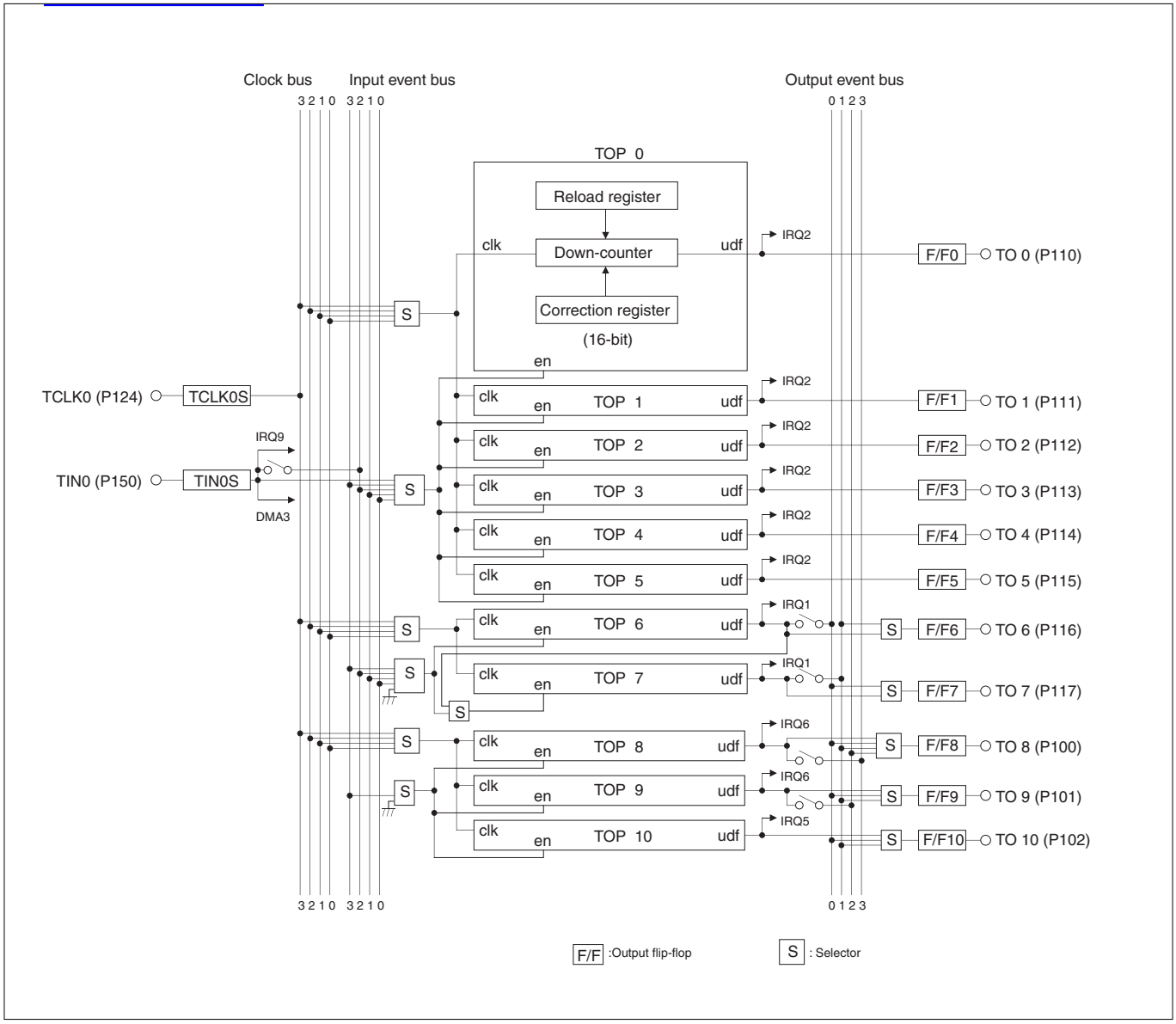


Figure 10.3.1 Block Diagram of TOP (Output-Related 16-Bit Timer)

10.3.2 Outline of Each Mode of TOP

Each mode of TOP is outlined below. For each TOP channel, only one of the following modes can be selected.

(1) Single-shot output mode

In single-shot output mode, the timer generates a pulse in width of "reload register set value +1" only once and then stops.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the reload register, the counter is loaded with the content of "the reload register -1" and starts counting synchronously with the count clock at the next cycle. The counter counts down and stops when it underflows after reaching the minimum count.

The F/F output waveform in single-shot output mode is inverted at enable and upon underflow, generating a single-shot pulse waveform in width of "reload register set value + 1" only once. An interrupt request can be generated when the counter underflows. The counter value is "setting value of reload register +1."

(2) Delayed single-shot output mode

In delayed single-shot output mode, the timer generates a pulse in width of "reload register set value + 1" after a finite time equal to "counter set value +1" only once and then stops.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the counter and reload register, it starts counting down from the counter's set value synchronously with the count clock. The next cycle after first time counter underflow, it is loaded with "the reload register -1" and continues counting down. The counter stops when it underflows next time.

The F/F output waveform in delayed single-shot output mode is inverted when the counter underflows first time and next, generating a single-shot pulse waveform in width of "reload register set value +1" after a finite time equal to "first set value of counter +1" only once.
An interrupt request can be generated when the counter underflows first time and next.
The effective counter value is "counter set value + 1" or "reload register set value +1."

(3) Continuous output mode

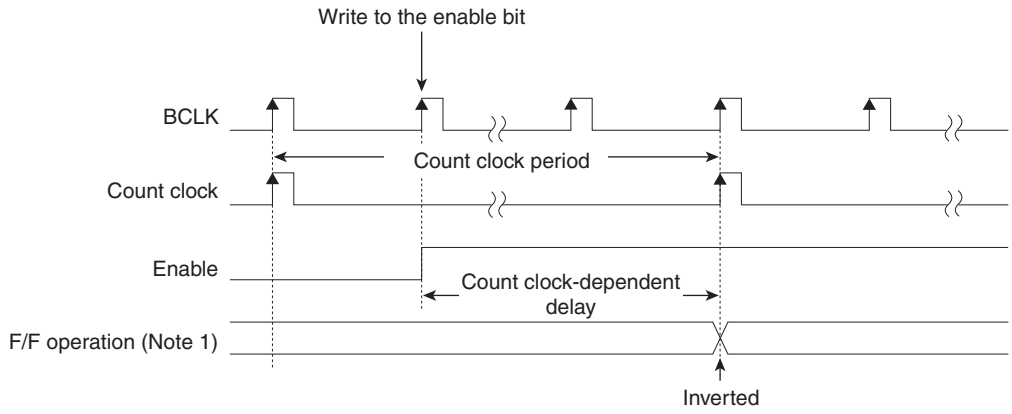
In continuous output mode, the timer counts down starting from the set value of the counter and when the counter underflows, it is loaded with the reload register value. Thereafter, this operation is repeated each time the counter underflows, thus generating consecutive pulses, invert in width of "reload register set value +1."

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the counter and reload register, it starts counting down from the counter's set value synchronously with the count clock and when the minimum count is reached, generates an underflow.
At the cycle after this underflow, the counter to be loaded with the content of "the reload register -1" and start counting over again. Thereafter, this operation is repeated each time an underflow occurs. To stop the counter, disable count by writing to the enable bit in software.

The F/F output waveform in continuous output mode is inverted at startup and upon underflow, generating a waveform of consecutive pulses until the timer stops counting. An interrupt request can be generated each time the counter underflows.
The effective counter value is "counter set value + 1" or "reload register set value +1."

<Count clock-dependent delay>

- Because the timer operates synchronously with the count clock, up to one count clock-dependent delay is generated by the time when the timer actually starts operating after writing to the enable bit. In operation mode where the F/F output is inverted when the timer is enabled, there is also a count clock-dependent delay before the F/F output is inverted.



Note 1: This applies to the case where F/F output is inverted when the timer is enabled.

Figure 10.3.2 Count Clock Dependent Delay

10.3.3 TOP Related Register Map

Shown below is a TOP related register map.

TOP Related Register Map (1/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0240	TOP0 Counter (TOP0CT)		10-53
H'0080 0242	TOP0 Reload Register (TOP0RL)		10-54
H'0080 0244	(Use inhibited area)		
H'0080 0246	TOP0 Correction Register (TOP0CC)		10-55
|	(Use inhibited area)		
H'0080 0250	TOP1 Counter (TOP1CT)		10-53
H'0080 0252	TOP1 Reload Register (TOP1RL)		10-54
H'0080 0254	(Use inhibited area)		
H'0080 0256	TOP1 Correction Register (TOP1CC)		10-55
|	(Use inhibited area)		
H'0080 0260	TOP2 Counter (TOP2CT)		10-53
H'0080 0262	TOP2 Reload Register (TOP2RL)		10-54
H'0080 0264	(Use inhibited area)		
H'0080 0266	TOP2 Correction Register (TOP2CC)		10-55
|	(Use inhibited area)		
H'0080 0270	TOP3 Counter (TOP3CT)		10-53
H'0080 0272	TOP3 Reload Register (TOP3RL)		10-54
H'0080 0274	(Use inhibited area)		
H'0080 0276	TOP3 Correction Register (TOP3CC)		10-55
|	(Use inhibited area)		
H'0080 0280	TOP4 Counter (TOP4CT)		10-53
H'0080 0282	TOP4 Reload Register (TOP4RL)		10-54
H'0080 0284	(Use inhibited area)		
H'0080 0286	TOP4 Correction Register (TOP4CC)		10-55
|	(Use inhibited area)		
H'0080 0290	TOP5 Counter (TOP5CT)		10-53
H'0080 0292	TOP5 Reload Register (TOP5RL)		10-54
H'0080 0294	(Use inhibited area)		
H'0080 0296	TOP5 Correction Register (TOP5CC)		10-55
H'0080 0298	(Use inhibited area)		

TOP Related Register Map (2/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 029A	TOP0–5 Control Register 0 (TOP05CR0)		10-49
H'0080 029C	(Use inhibited area)	TOP0–5 Control Register 1 (TOP05CR1)	10-49
H'0080 029E	(Use inhibited area)		
H'0080 02A0	TOP6 Counter (TOP6CT)		10-53
H'0080 02A2	TOP6 Reload Register (TOP6RL)		10-54
H'0080 02A4	(Use inhibited area)		
H'0080 02A6	TOP6 Correction Register (TOP6CC)		10-55
H'0080 02A8	(Use inhibited area)		
H'0080 02AA	TOP6,7 Control Register (TOP67CR)		10-51
⋮	(Use inhibited area)		
H'0080 02B0	TOP7 Counter (TOP7CT)		10-53
H'0080 02B2	TOP7 Reload Register (TOP7RL)		10-54
H'0080 02B4	(Use inhibited area)		
H'0080 02B6	TOP7 Correction Register (TOP7CC)		10-55
⋮	(Use inhibited area)		
H'0080 02C0	TOP8 Counter (TOP8CT)		10-53
H'0080 02C2	TOP8 Reload Register (TOP8RL)		10-54
H'0080 02C4	(Use inhibited area)		
H'0080 02C6	TOP8 Correction Register (TOP8CC)		10-55
⋮	(Use inhibited area)		
H'0080 02D0	TOP9 Counter (TOP9CT)		10-53
H'0080 02D2	TOP9 Reload Register (TOP9RL)		10-54
H'0080 02D4	(Use inhibited area)		
H'0080 02D6	TOP9 Correction Register (TOP9CC)		10-55
⋮	(Use inhibited area)		
H'0080 02E0	TOP10 Counter (TOP10CT)		10-53
H'0080 02E2	TOP10 Reload Register (TOP10RL)		10-54
H'0080 02E4	(Use inhibited area)		
H'0080 02E6	TOP10 Correction Register (TOP10CC)		10-55
H'0080 02E8	(Use inhibited area)		
H'0080 02EA	TOP8–10 Control Register (TOP810CR)		10-52
⋮	(Use inhibited area)		
H'0080 02FA	TOP0-10 External Enable Permit Register (TOPEEN)		10-56
H'0080 02FC	TOP0-10 Enable Protect Register (TOPPRO)		10-56
H'0080 02FE	TOP0-10 Count Enable Register (TOPCEN)		10-57

10.3.4 TOP Control Registers

The TOP control registers are used to select operation modes of TOP0–10 (single-shot output, delayed single-shot output or continuous output mode), as well as select the count enable and count clock sources. Following TOP control registers are provided for each timer group.

- TOP0–5 Control Register 0 (TOP05CR0)
- TOP0–5 Control Register 1 (TOP05CR1)
- TOP6,7 Control Register (TOP67CR)
- TOP8–10 Control Register (TOP810CR)

TOP0–5 Control Register 0 (TOP05CR0) <Address: H'0080 029A>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
TOP3M		TOP2M		TOP1M		TOP0M			TOP05ENS					TOP05CKS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0, 1	TOP3M (TOP3 operation mode select bit)	00: Single-shot output mode	R	W
2, 3	TOP2M (TOP2 operation mode select bit)	01: Delayed single-shot output mode		
4, 5	TOP1M (TOP1 operation mode select bit)	10: Continuous output mode		
6, 7	TOP0M (TOP0 operation mode select bit)	11: Continuous output mode		
8	No function assigned. Fix to "0".		0	0
9–11	TOP05ENS TOP0–5 enable source select bit	000: External TIN0 input 001: External TIN0 input 010: External TIN0 input 011: External TIN0 input 100: Input event bus 0 101: Input event bus 1 110: Input event bus 2 111: Input event bus 3	R	W
12, 13	No function assigned. Fix to "0".		0	0
14, 15	TOP05CKS TOP0–5 clock source select bit	00: Clock bus 0 01: Clock bus 1 10: Clock bus 2 11: Clock bus 3	R	W

Notes: • This register must always be accessed in halfwords.
• Operation mode can only be set or changed while the counter is inactive.

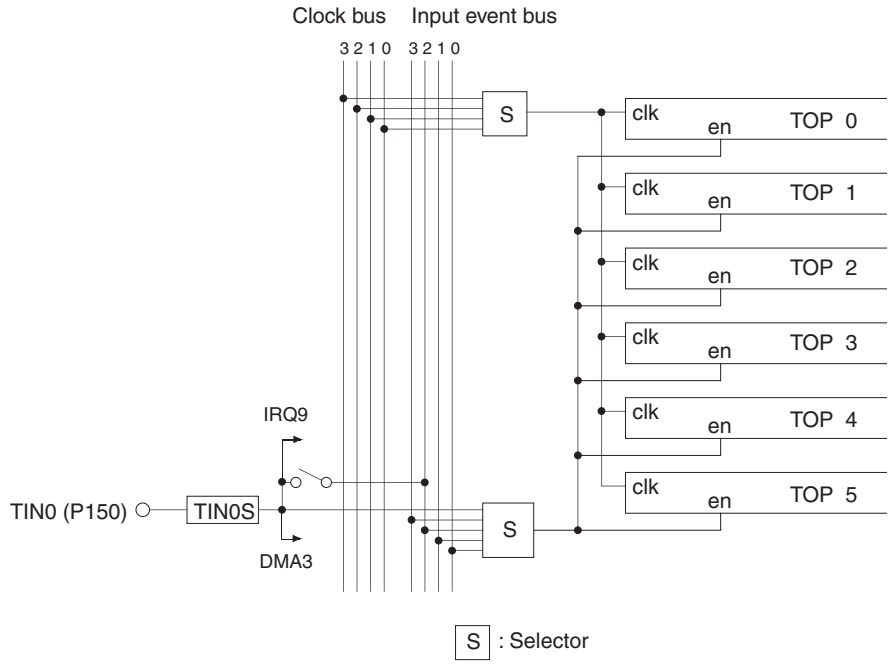
TOP0–5 Control Register 1 (TOP05CR1) <Address: H'0080 029D>

b8	9	10	11	12	13	14	b15
				TOP5M		TOP4M	
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–11	No function assigned. Fix to "0".		0	0
12, 13	TOP5M (TOP5 operation mode select bit)	00: Single-shot output mode	R	W
14, 15	TOP4M (TOP4 operation mode select bit)	01: Delayed single-shot output mode 10: Continuous output mode 11: Continuous output mode		

Note: • Operation mode can only be set or changed while the counter is inactive.



Note: • This diagram only illustrates TOP control registers and is partly omitted.

Figure 10.3.3 Outline Diagram of TOP0–5 Clock and Enable Inputs

TOP6,7 Control Register (TOP67CR) <Address: H'0080 02AA>

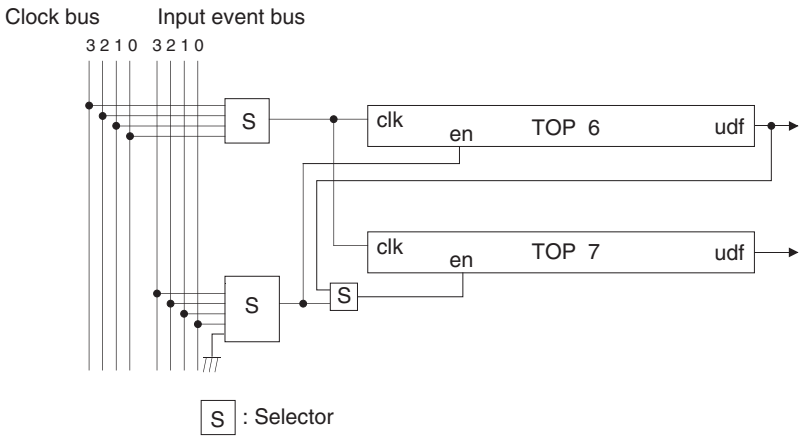
b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
	TOP7 ENS	TOP7M				TOP6M			TOP67ENS					TOP67CKS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	No function assigned. Fix to "0".		0	0
1	TOP7ENS TOP7 enable source select bit	0: Result selected by TOP67ENS bit 1: TOP6 output	R	W
2, 3	TOP7M TOP7 operation mode select bit	00: Single-shot output mode 01: Delayed single-shot output mode 10: Continuous output mode 11: Continuous output mode	R	W
4, 5	No function assigned. Fix to "0".		0	0
6, 7	TOP6M TOP6 operation mode select bit	00: Single-shot output mode 01: Delayed single-shot output mode 10: Continuous output mode 11: Continuous output mode	R	W
8	No function assigned. Fix to "0".		0	0
9–11	TOP67ENS TOP6, TOP7 enable source select bit (Note 1)	000: Does not select enable source 001: Does not select enable source 010: Does not select enable source 011: Does not select enable source 100: Input event bus 0 101: Input event bus 1 110: Input event bus 2 111: Input event bus 3	R	W
12, 13	No function assigned. Fix to "0".		0	0
14, 15	TOP67CKS TOP6, TOP7 clock source select bit	00: Clock bus 0 01: Clock bus 1 10: Clock bus 2 11: Clock bus 3	R	W

Note 1: This register must always be accessed in halfwords.

Note : • Operation mode can only be set or changed while the counter is inactive.



Note: • This diagram only illustrates TOP control registers and is partly omitted.

Figure 10.3.4 Outline Diagram of TOP6, TOP7 Clock and Enable Inputs

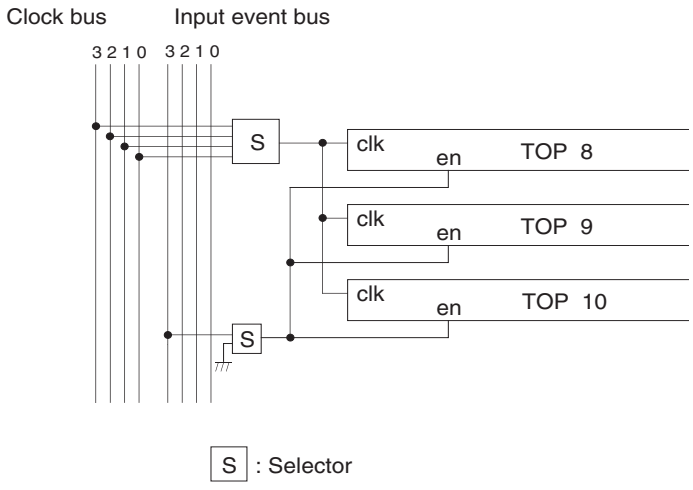
TOP8–10 Control Register (TOP810CR) <Address: H'0080 02EA>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
0	0	TOP10M 0	0	TOP9M 0	0	TOP8M 0	0	0	0	0	TOP810ENS 0	0	0	TOP810CKS 0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0, 1	No function assigned. Fix to "0".		0	0
2, 3	TOP10M TOP10 operation mode select bit	00: Single-shot output mode 01: Delayed single-shot output mode	R	W
4, 5	TOP9M TOP9 operation mode select bit	10: Continuous output mode 11: Continuous output mode		
6, 7	TOP8M TOP8 operation mode select bit			
8–10	No function assigned. Fix to "0".		0	0
11	TOP810ENS TOP8–10 enable source select bit	0: Does not select enable source 1: Input event bus 3	R	W
12, 13	No function assigned. Fix to "0".		0	0
14, 15	TOP810CKS TOP8–10 clock source select bit	00: Clock bus 0 01: Clock bus 1 10: Clock bus 2 11: Clock bus 3	R	W

Notes: • This register must always be accessed in halfwords.
• Operation mode can only be set or changed while the counter is inactive.

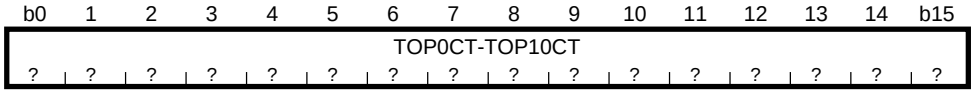


Note: • This diagram only illustrates TOP control registers and is partly omitted.

Figure 10.3.5 Outline Diagram of TOP8–10 Clock and Enable Inputs

10.3.5 TOP Counters (TOP0CT–TOP10CT)

TOP0 Counter (TOP0CT)	<Address: H'0080 0240>
TOP1 Counter (TOP1CT)	<Address: H'0080 0250>
TOP2 Counter (TOP2CT)	<Address: H'0080 0260>
TOP3 Counter (TOP3CT)	<Address: H'0080 0270>
TOP4 Counter (TOP4CT)	<Address: H'0080 0280>
TOP5 Counter (TOP5CT)	<Address: H'0080 0290>
TOP6 Counter (TOP6CT)	<Address: H'0080 02A0>
TOP7 Counter (TOP7CT)	<Address: H'0080 02B0>
TOP8 Counter (TOP8CT)	<Address: H'0080 02C0>
TOP9 Counter (TOP9CT)	<Address: H'0080 02D0>
TOP10 Counter (TOP10CT)	<Address: H'0080 02E0>



<Upon exiting reset: Undefined>

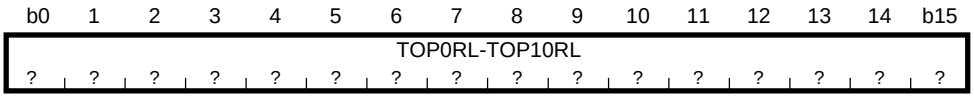
b	Bit Name	Function	R	W
0–15	TOP0CT–TOP10CT	16-bit counter value	R	W

Note: • These registers must always be accessed in halfwords.

The TOP counters are a 16-bit down-counter. After the timer is enabled (by writing to the enable bit in software or by external input), the counter starts counting synchronously with the count clock.

10.3.6 TOP Reload Registers (TOP0RL–TOP10RL)

TOP0 Reload Register (TOP0RL)	<Address: H'0080 0242>
TOP1 Reload Register (TOP1RL)	<Address: H'0080 0252>
TOP2 Reload Register (TOP2RL)	<Address: H'0080 0262>
TOP3 Reload Register (TOP3RL)	<Address: H'0080 0272>
TOP4 Reload Register (TOP4RL)	<Address: H'0080 0282>
TOP5 Reload Register (TOP5RL)	<Address: H'0080 0292>
TOP6 Reload Register (TOP6RL)	<Address: H'0080 02A2>
TOP7 Reload Register (TOP7RL)	<Address: H'0080 02B2>
TOP8 Reload Register (TOP8RL)	<Address: H'0080 02C2>
TOP9 Reload Register (TOP9RL)	<Address: H'0080 02D2>
TOP10 Reload Register (TOP10RL)	<Address: H'0080 02E2>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	TOP0RL–TOP10RL	16-bit reload register value	R	W

Note: • These registers must always be accessed in halfwords.

The TOP reload registers are used to load data into the TOP counter registers (TOP0CT–TOP10CT). The content of "the reload register -1" is loaded into the counter synchronously with the count clock at the following timing:

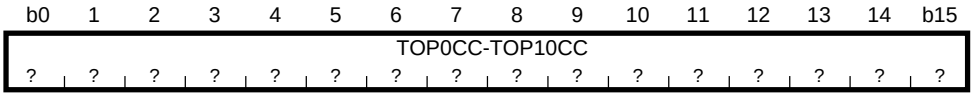
- At the next cycle when the counter is enabled in single-shot output mode
- At the next cycle when the counter underflowed in delayed single-shot or continuous output mode

Simply because data is written to the reload register does not mean that the data is loaded into the counter. The counter is loaded with data in only the above cases.

Note that reloading of data after an underflow is performed synchronously with a clock pulse at which the counter underflowed.

10.3.7 TOP Correction Registers (TOP0CC–TOP10CC)

TOP0 Correction Register (TOP0CC)	<Address: H'0080 0246>
TOP1 Correction Register (TOP1CC)	<Address: H'0080 0256>
TOP2 Correction Register (TOP2CC)	<Address: H'0080 0266>
TOP3 Correction Register (TOP3CC)	<Address: H'0080 0276>
TOP4 Correction Register (TOP4CC)	<Address: H'0080 0286>
TOP5 Correction Register (TOP5CC)	<Address: H'0080 0296>
TOP6 Correction Register (TOP6CC)	<Address: H'0080 02A6>
TOP7 Correction Register (TOP7CC)	<Address: H'0080 02B6>
TOP8 Correction Register (TOP8CC)	<Address: H'0080 02C6>
TOP9 Correction Register (TOP9CC)	<Address: H'0080 02D6>
TOP10 Correction Register (TOP10CC)	<Address: H'0080 02E6>



(Acceptable range of values: +32767 to –32768)

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	TOP0CC–TOP10CC	16-bit correction register value	R	W

Note: • These registers must always be accessed in halfwords.

The TOP correction registers are used to correct the TOP counter value by adding or subtracting in the middle of operation. To increase or reduce the counter value, write to this correction register a value by which the counter value is to be increased or reduced from its initial set value. To add, write the value to be added to the correction register directly as is. To subtract, write the 2's complement of the value to be subtracted to the correction register.

The counter is corrected synchronously with a clock pulse next to one at which the correction value was written to the TOP correction register. If the counter is corrected this way, note that because one down count in that clock period is canceled, the counter value actually is corrected by (correction register value + 1). For example, if the initial counter value is 10 and the value 3 is written to the correction register when the counter has counted down to 5, then the counter counts a total of 15 before it underflows.

10.3.8 TOP Enable Control Registers

TOP0-10 External Enable Permit Register (TOPEEN) <Address: H'0080 02FA>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
0	0	0	0	0	TOP10EEN 0	TOP9EEN 0	TOP8EEN 0	TOP7EEN 0	TOP6EEN 0	TOP5EEN 0	TOP4EEN 0	TOP3EEN 0	TOP2EEN 0	TOP1EEN 0	TOP0EEN 0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0–4	No function assigned. Fix to "0".		0	0
5	TOP10EEN (TOP10 external enable permit bit)	0: Disable external enable	R	W
6	TOP9EEN (TOP9 external enable permit bit)	1: Enable external enable		
7	TOP8EEN (TOP8 external enable permit bit)			
8	TOP7EEN (TOP7 external enable permit bit)			
9	TOP6EEN (TOP6 external enable permit bit)			
10	TOP5EEN (TOP5 external enable permit bit)			
11	TOP4EEN (TOP4 external enable permit bit)			
12	TOP3EEN (TOP3 external enable permit bit)			
13	TOP2EEN (TOP2 external enable permit bit)			
14	TOP1EEN (TOP1 external enable permit bit)			
15	TOP0EEN (TOP0 external enable permit bit)			

Note: • This register must always be accessed in halfwords.

The TOP0-10 External Enable Permit Register controls enable operation on TOP counters from external devices by enabling or disabling it.

TOP0-10 Enable Protect Register (TOPPRO) <Address: H'0080 02FC>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
0	0	0	0	0	TOP10PRO 0	TOP9PRO 0	TOP8PRO 0	TOP7PRO 0	TOP6PRO 0	TOP5PRO 0	TOP4PRO 0	TOP3PRO 0	TOP2PRO 0	TOP1PRO 0	TOP0PRO 0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0–4	No function assigned. Fix to "0".		0	0
5	TOP10PRO (TOP10 enable protect bit)	0: Enable for rewriting	R	W
6	TOP9PRO (TOP9 enable protect bit)	1: Protect against rewriting		
7	TOP8PRO (TOP8 enable protect bit)			
8	TOP7PRO (TOP7 enable protect bit)			
9	TOP6PRO (TOP6 enable protect bit)			
10	TOP5PRO (TOP5 enable protect bit)			
11	TOP4PRO (TOP4 enable protect bit)			
12	TOP3PRO (TOP3 enable protect bit)			
13	TOP2PRO (TOP2 enable protect bit)			
14	TOP1PRO (TOP1 enable protect bit)			
15	TOP0PRO (TOP0 enable protect bit)			

Note: • This register must always be accessed in halfwords.

The TOP0-10 Enable Protect Register controls rewriting of the TOP0-10 count enable bit by enabling for or protecting it against rewriting.

TOP0-10 Count Enable Register (TOPCEN) <Address: H'0080 02FE>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
0	0	0	0	0	TOP10CEN 0	TOP9CEN 0	TOP8CEN 0	TOP7CEN 0	TOP6CEN 0	TOP5CEN 0	TOP4CEN 0	TOP3CEN 0	TOP2CEN 0	TOP1CEN 0	TOP0CEN 0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0–4	No function assigned. Fix to "0".		0	0
5	TOP10CEN (TOP10 count enable bit)	0: Stop counting	R	W
6	TOP9CEN (TOP9 count enable bit)	1: Enable counting		
7	TOP8CEN (TOP8 count enable bit)			
8	TOP7CEN (TOP7 count enable bit)			
9	TOP6CEN (TOP6 count enable bit)			
10	TOP5CEN (TOP5 count enable bit)			
11	TOP4CEN (TOP4 count enable bit)			
12	TOP3CEN (TOP3 count enable bit)			
13	TOP2CEN (TOP2 count enable bit)			
14	TOP1CEN (TOP1 count enable bit)			
15	TOP0CEN (TOP0 count enable bit)			

Note: • This register must always be accessed in halfwords.

The TOP0-10 Count Enable Register controls operation of TOP counters. To enable any TOP counter in software, enable its corresponding enable protect bit for write and set the count enable bit by writing "1". To stop any TOP counter, enable its corresponding enable protect bit for write and reset the count enable bit by writing "0".

In all but continuous output mode, when the counter stops due to occurrence of an underflow, the count enable bit is automatically reset to "0". Therefore, the TOP0-10 Count Enable Register when accessed for read serves as a status register indicating whether the counter is operating or idle.

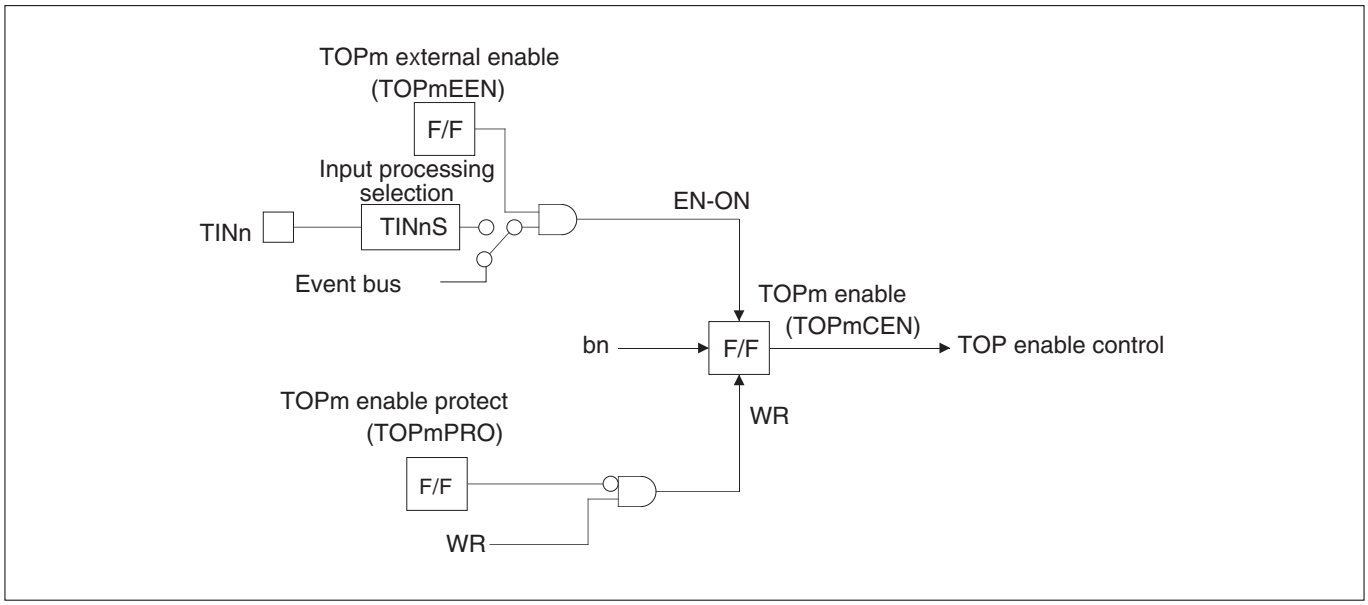


Figure 10.3.6 Configuration of the TOP Enable Circuit

10.3.9 Operation in TOP Single-shot Output Mode (with Correction Function)

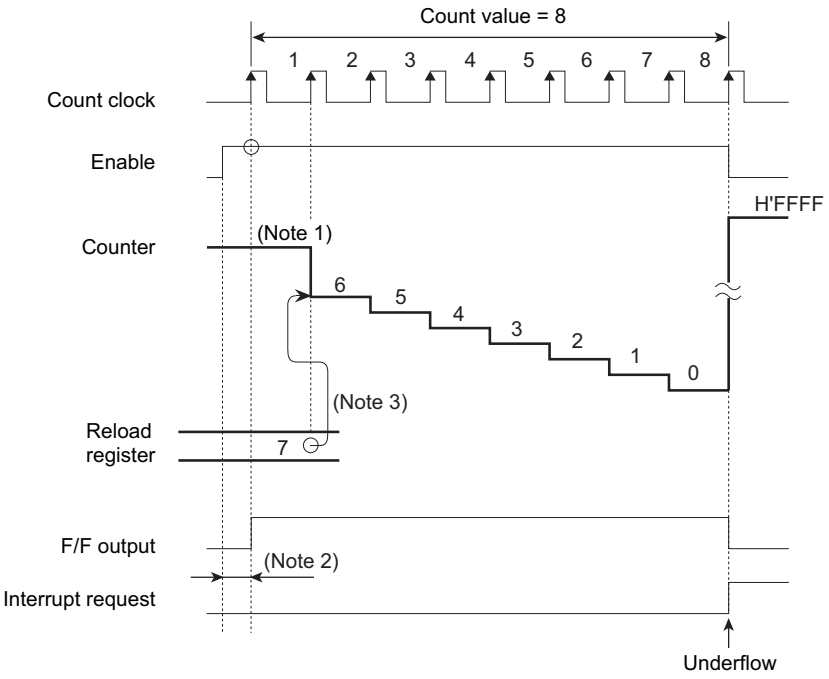
(1) Outline of TOP single-shot output mode

In single-shot output mode, the timer generates a pulse in width of "reload register set value + 1" only once and then stops.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the reload register, at the next cycle the counter is loaded with the content of "the reload register -1" and starts counting synchronously with the count clock. The counter counts down and stops when it underflows after reaching the minimum count.

The F/F output waveform in single-shot output mode is inverted (F/F output levels change from "L" to "H" or vice versa) at startup and upon underflow, generating a single-shot pulse waveform in width of "reload register set value + 1" only once. An interrupt request can be generated when the counter underflows. The count value is "reload register set value + 1."

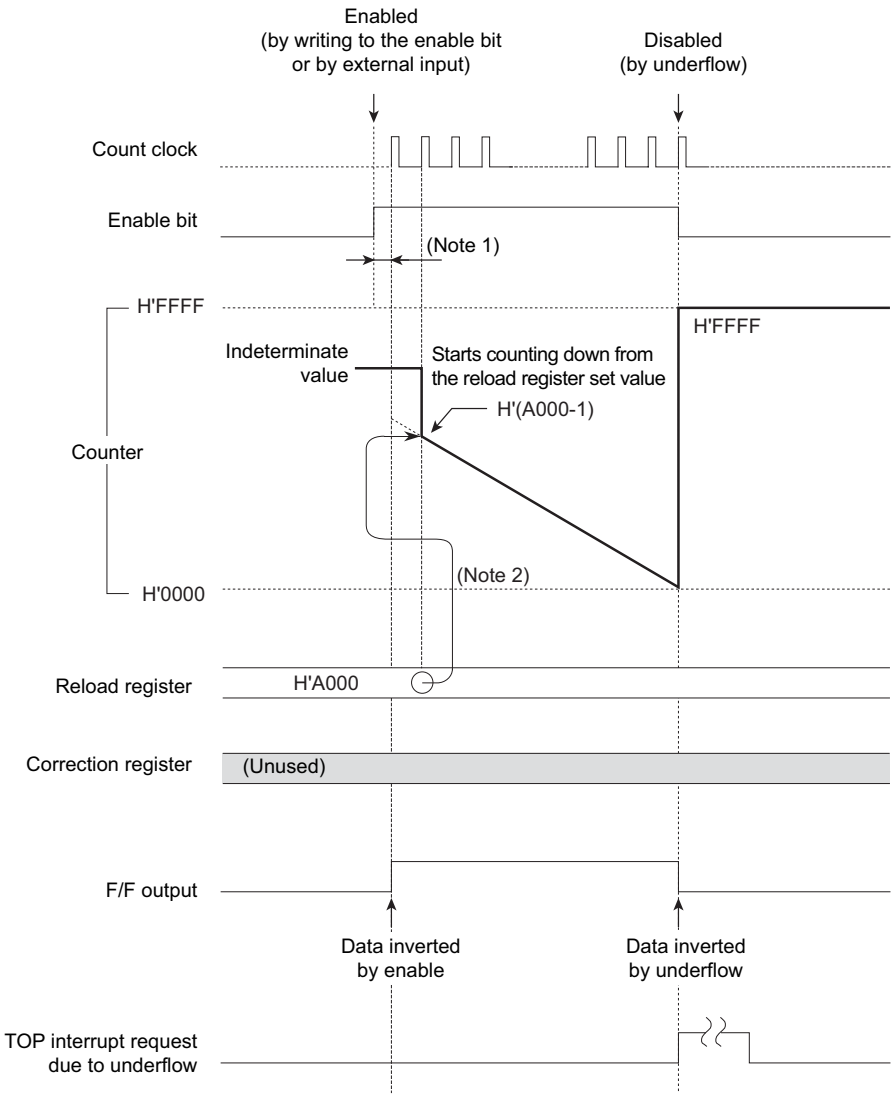
For example, if the initial reload register value is 7, then the count value is 8.



Note 1: What actually is seen in the cycle immediately during enable is the previous counter value, and not 7.
Note 2: A count clock dependent delay is included before F/F output changes state after the timer is enabled.
Note 3: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.3.7 Example of Counting in TOP Single-shot Output Mode

In the example below, the reload register is initially set to H'A000. (The initial counter value can be undefined, and does not have to be specific.) When the timer starts, the value that "the reload register -1" is loaded into the counter, letting it start counting. Thereafter, it continues counting down until it underflows after reaching the minimum count.



Note 1: A count clock dependent delay is included before F/F output changes state after the timer is enabled.
Note 2: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

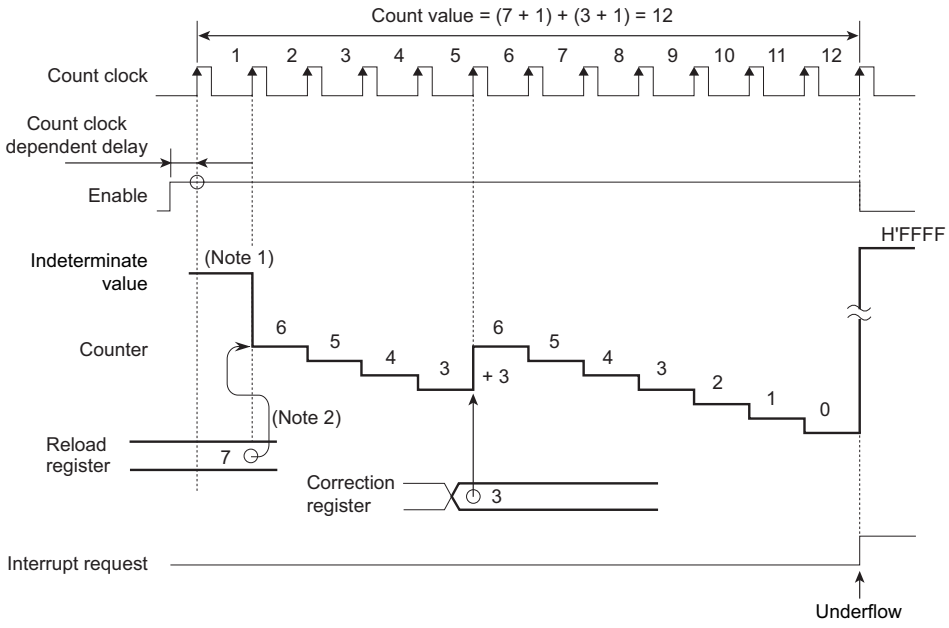
Figure 10.3.8 Typical Operation in TOP Single-shot Output Mode

(2) Correction function of TOP single-shot output mode

To change the counter value while in progress, write to the TOP correction register a value by which the counter value is to be increased or reduced from its initial set value. To add, write the value to be added to the correction register directly as is. To subtract, write the 2's complement of the value to be subtracted to the correction register.

The counter is corrected synchronously with a count clock pulse next to one at which the correction value was written to the TOP correction register. If the counter is corrected this way, note that because one down count in that clock period is canceled, the counter value actually is corrected by (correction register value + 1).

For example, if the initial counter value is 7 and the value 3 is written to the correction register when the counter has counted down to 3, then the counter counts a total of 12 before it underflows.

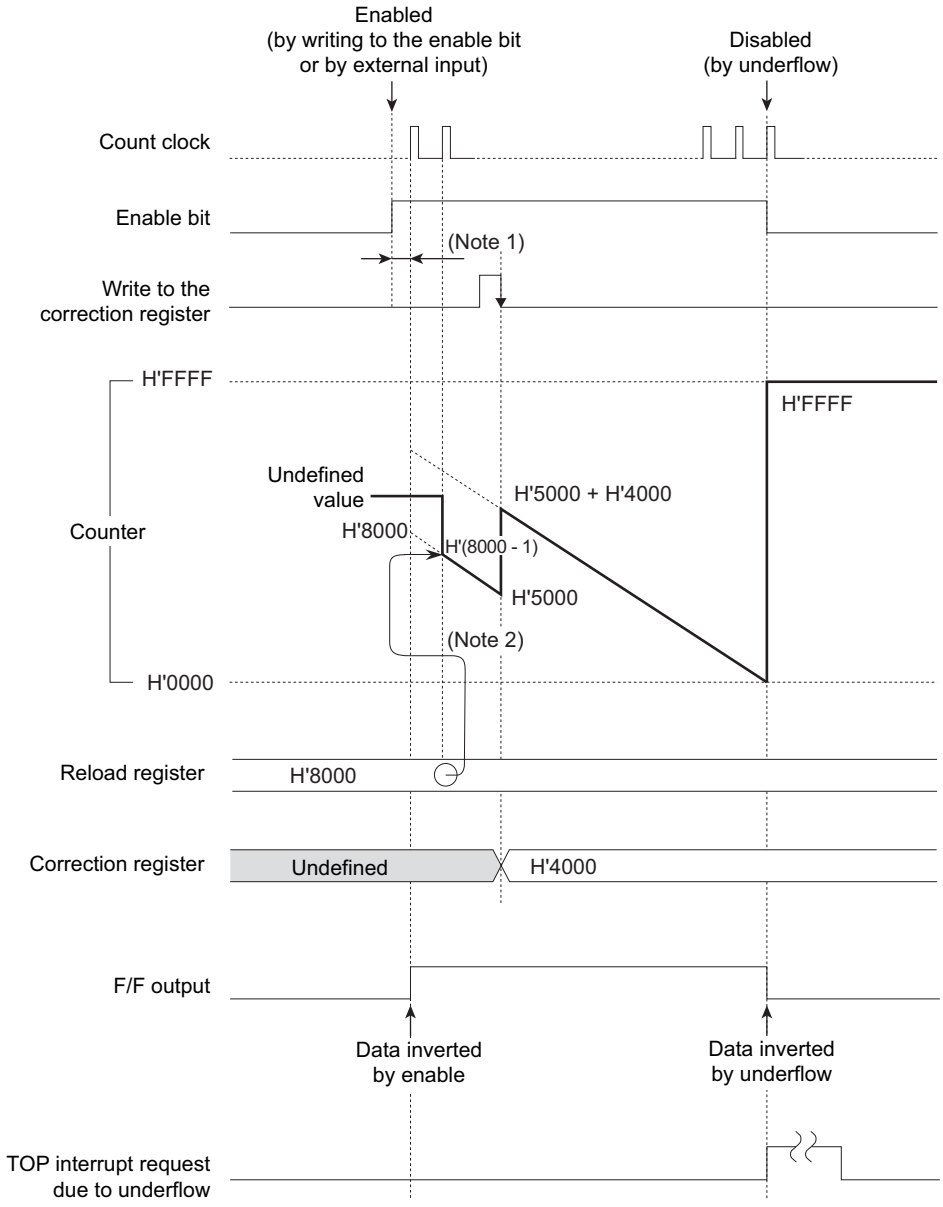


Note 1: What actually is seen in the cycle immediately after enable is the previous counter value, and not 7.
Note 2: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.3.9 Example of Counting in TOP Single-shot Output Mode When Count is Corrected

When writing to the correction register, be careful not to cause the counter to overflow. Even if the counter overflows due to correction of counts, no interrupt requests are generated for reasons of an overflow.

In the example below, the reload register is initially set to H'8000. When the timer starts, the value that "the reload register -1" is loaded into the counter, letting it start counting down. In the diagram below, the value H'4000 is written to the correction register when the counter has counted down to H'5000. As a result of this correction, the count has been increased to H'9000, so that the counter counts a total of (H'8000 + 1 + H'4000 + 1) before it stops.



Note 1: What actually is seen in the cycle immediately after enable is the previous counter value, and not 7.
Note 2: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

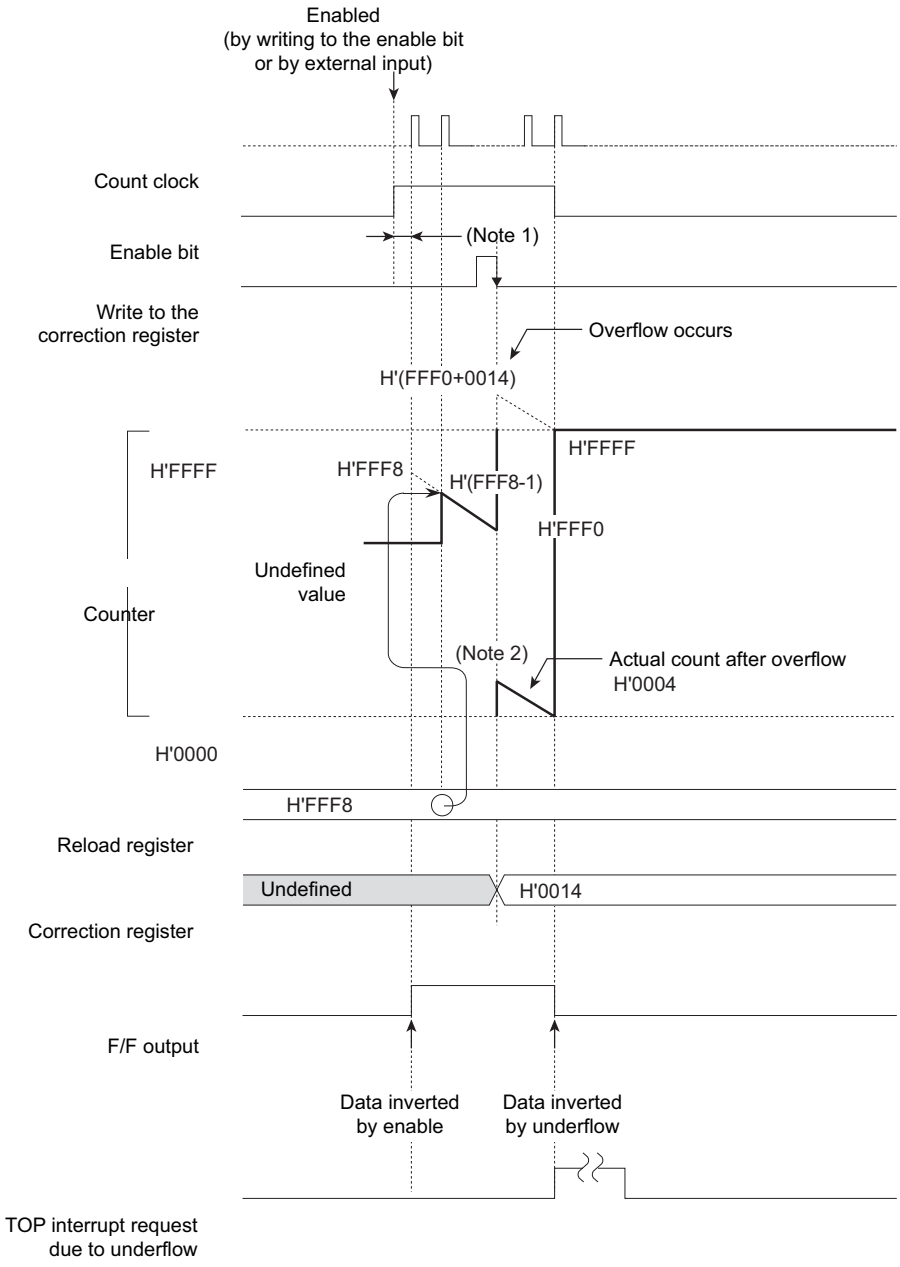
Figure 10.3.10 Typical Operation in TOP Single-shot Output Mode When Count is Corrected

(3) Precautions on using TOP single-shot output mode

The following describes precautions to be observed when using TOP single-shot output mode.

- If the counter stops due to an underflow in the same clock period as the timer is enabled by external input, the former has priority so that the counter stops.
- If the counter stops due to an underflow in the same clock period as count is enabled by writing to the enable bit, the latter has priority so that count is enabled.
- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- Because the timer operates synchronously with the count clock, a count clock-dependent delay is included before starting F/F operation after the timer is enabled.
- When writing to the correction register, be careful not to cause the counter to overflow. Even if the counter overflows due to correction of counts, no interrupt requests are generated for reasons of an overflow. Therefore, if the counter underflows in the subsequent down-count after an overflow, a false interrupt request is generated for an underflow that includes the overflowed count.

In the example below, the reload register is initially set to H'FFF8. When the timer starts, the value that "the reload register -1" is loaded into the counter, letting it start counting down. In the diagram below, the value H'0014 is written to the correction register when the counter has counted down to H'FFF0. As a result of this correction, the count overflows to H'0004 and the counter fails to count correctly. Also, an interrupt request is generated for an erroneous overflowed count.



Note 1: A count clock dependent delay is included before F/F output changes state after the timer is enabled.
Note 2: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.3.11 Example of an Operation in TOP Single-shot Output Mode Where Count Overflows Due to Correction

10.3.10 Operation in TOP Delayed Single-shot Output Mode (with Correction Function)

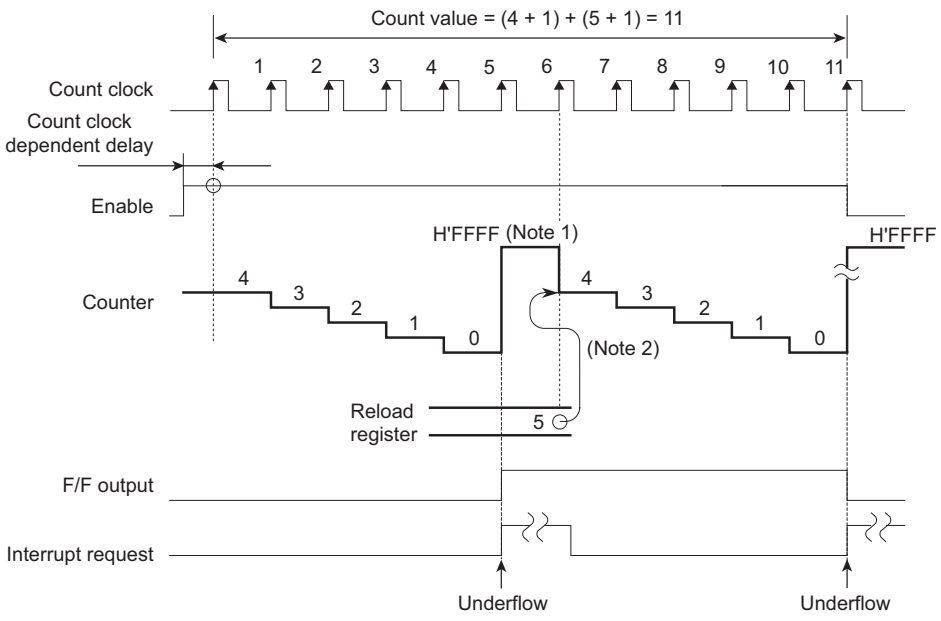
(1) Outline of TOP delayed single-shot output mode

In delayed single-shot output mode, the timer generates a pulse in width of "reload register set value + 1" after a finite time equal to "counter set value + 1" only once and then stops.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the counter and reload register, it starts counting down from the counter's set value synchronously with the count clock. At the cycle after the first time the counter underflows, it is loaded with the value that "the reload register -1" and continues counting down. The counter stops when it underflows next time.

The F/F output waveform in delayed single-shot output mode is inverted (F/F output level changes from "L" to "H" or vice versa) when the counter underflows first time and next, generating a single-shot pulse waveform in width of "reload register set value + 1" after a finite time equal to "first set value of counter + 1" only once.
An interrupt request can be generated when the counter underflows first time and next.
The "counter set value + 1" and "reload register set value + 1" are effective as count values.

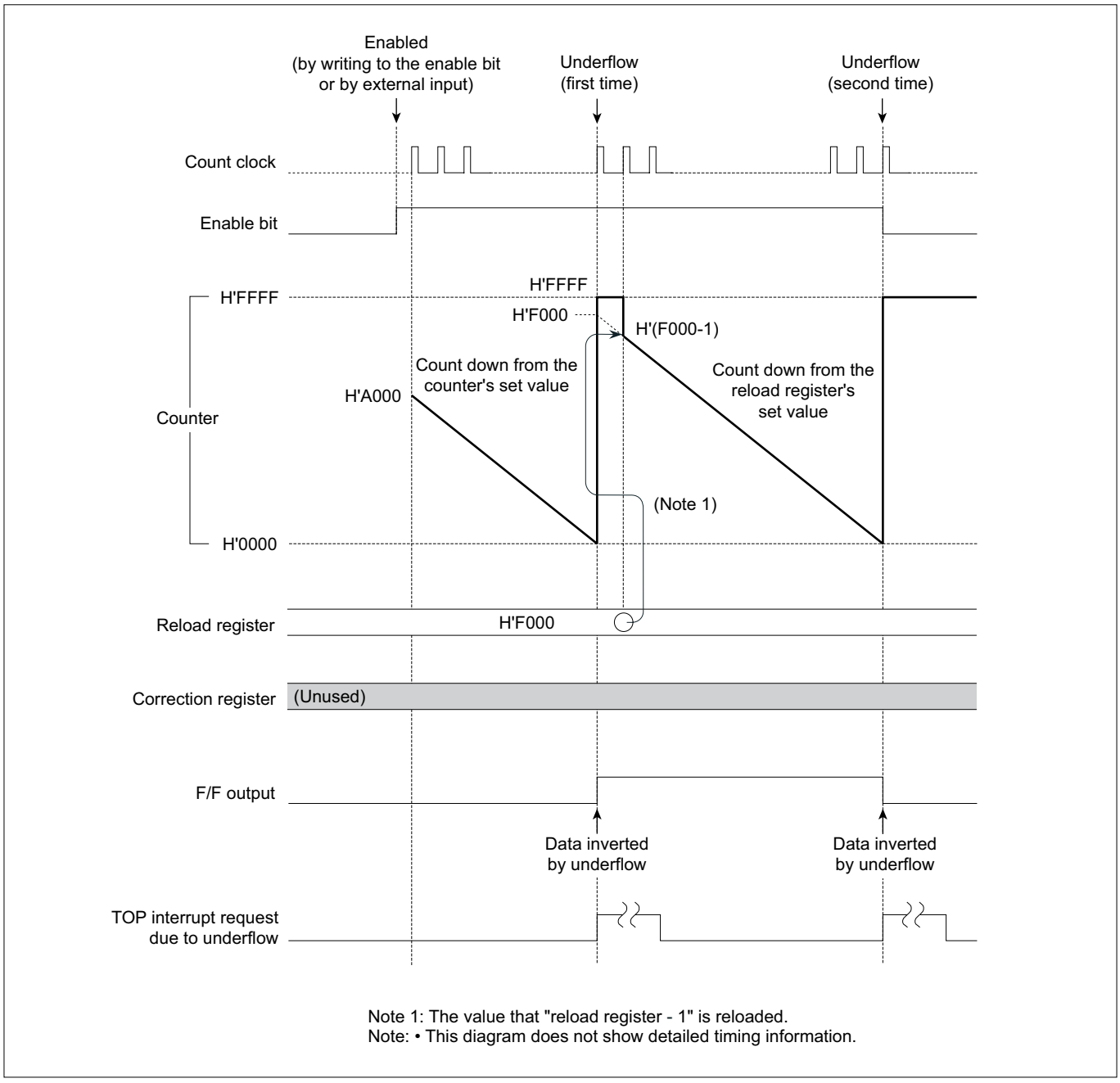
For example, if the initial counter value is 4 and the initial reload register value is 5, then the timer operates as shown below.



Note 1: What actually is seen in the cycle immediately during underflow is H'FFFF(underflow value), and not 5.
Note 2: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.3.12 Example of Counting in TOP Delayed Single-shot Output Mode

In the example below, the counter and the reload register are initially set to H'A000 and H'F000, respectively. When the timer is enabled, the counter starts counting down at the cycle after it underflows, the counter is loaded with the content of "the reload register -1" and continues counting down. The counter stops when it underflows second time.



Note 1: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

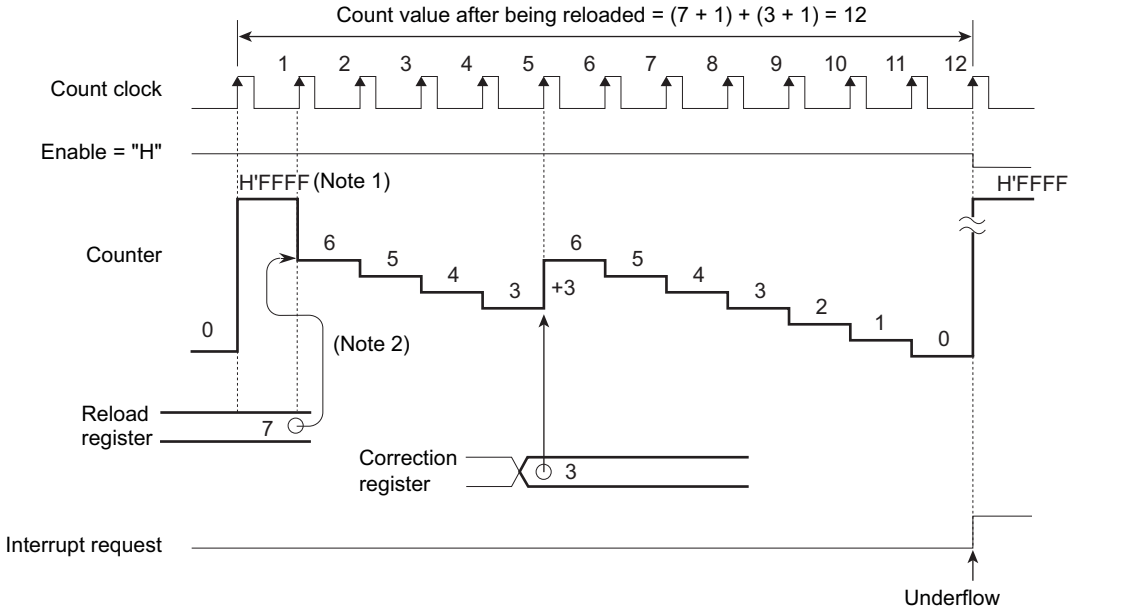
Figure 10.3.13 Typical Operation in TOP Delayed Single-shot Output Mode

(2) Correction function of TOP delayed single-shot output mode

To change the counter value while in progress, write to the TOP correction register a value by which the counter value is to be increased or reduced from its initial set value. To add, write the value to be added to the correction register directly as is. To subtract, write the 2's complement of the value to be subtracted to the correction register.

The counter is corrected synchronously with a count clock pulse next to one at which the correction value was written to the TOP correction register. If the counter is corrected this way, note that because one down count in that clock period is canceled, the counter value actually is corrected by (correction register value + 1).

For example, if the reload register value is 7 and the value 3 is written to the correction register when the counter has counted down to 3 after being reloaded, then the counter counts a total of 12 after being reloaded before it underflows.

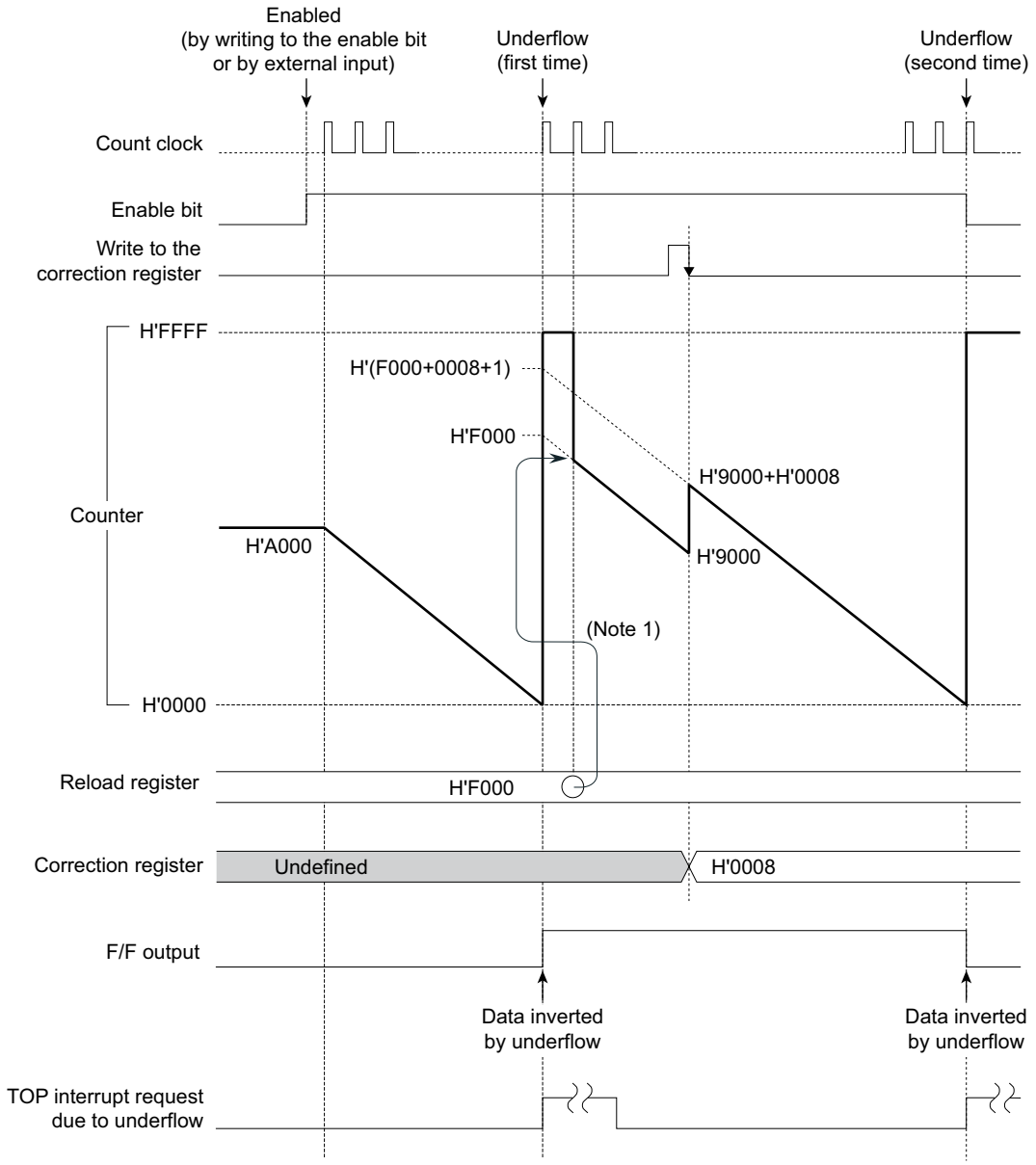


Note 1: What actually is seen in the cycle immediately during underflow is H'FFFF(the underflow value), and not 7.
Note 2: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.3.14 Example of Counting in TOP Delayed Single-shot Output Mode When Count is Corrected

When writing to the correction register, be careful not to cause the counter to overflow. Even if the counter overflows due to correction of counts, no interrupt requests are generated for reasons of an overflow.

In the example below, the counter and the reload register are initially set to H'A000 and H'F000, respectively. When the timer is enabled, the counter starts counting down and at the cycle after the first underflow, the counter is loaded with the content of "the reload register -1" and continues counting down. In the diagram below, the value H'0008 is written to the correction register when the counter has counted down to H'9000. As a result of this correction, the counter has its count value increased to H'9008 and counts (H'F000 + 1 + H'0008 + 1) after the first underflow before it stops.



Note 1: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.3.15 Typical Operation in TOP Delayed Single-shot Output Mode when Count is Corrected

(3) Precautions on using TOP delayed single-shot output mode

The following describes precautions to be observed when using TOP delayed single-shot output mode.

- If the counter stops due to an underflow in the same clock period as the timer is enabled by external input, the former has priority so that the counter stops.
- If the counter stops due to an underflow in the same clock period as count is enabled by writing to the enable bit, the latter has priority so that count is enabled.
- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- Even if the counter overflows due to correction of counts, no interrupt requests are generated for reasons of an overflow. Therefore, if the counter underflows in the subsequent down-count after an overflow, a false interrupt request is generated for an underflow that includes the overflowed count.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF. The reload reads the value that "the reload register -1" into the counter at the timing of the counter clock after the underflow.

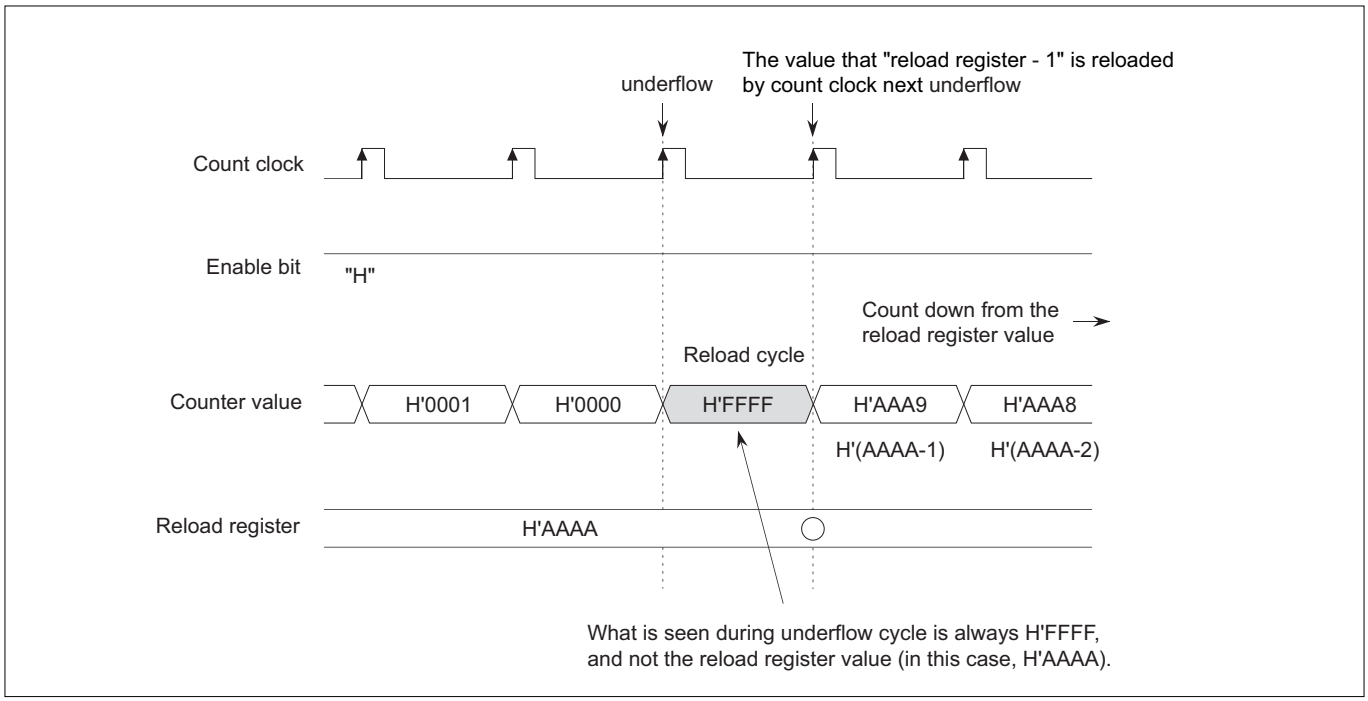


Figure 10.3.16 Counter Value Immediately after Underflow

10.3.11 Operation in TOP Continuous Output Mode (without Correction Function)

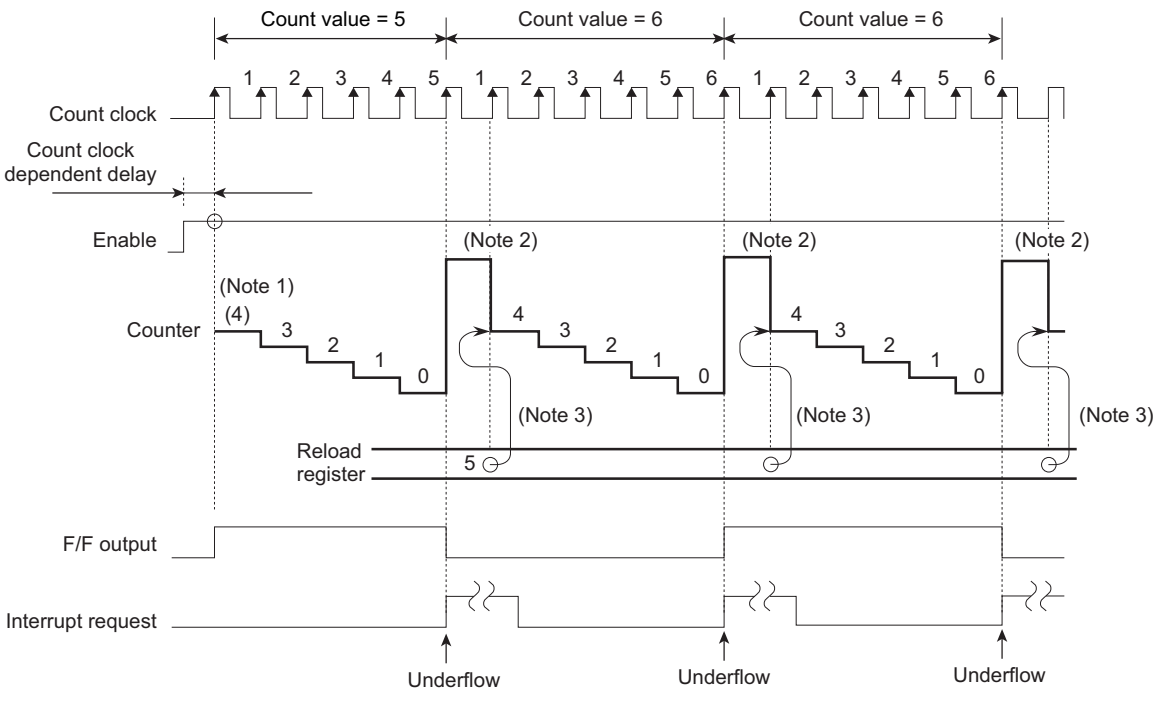
(1) Outline of TOP continuous output mode

In continuous output mode, the timer counts down starting from the set value of the counter and at the cycle after the counter underflows, it is loaded with the value that "the reload register -1." Thereafter, this operation is repeated each time the counter underflows, thus generating consecutive pulses whose waveform is inverted in width of "reload register set value + 1."

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the counter and reload register, it starts counting down from the counter's set value synchronously with the count clock and when the minimum count is reached, generates an underflow. At the cycle after this underflow, the counter to be loaded with the content of "the reload register -1" and start counting over again. Thereafter, this operation is repeated each time an underflow occurs. To stop the counter, disable count by writing to the enable bit in software.

The F/F output waveform in continuous output mode is inverted (F/F output level changes from "L" to "H" or vice versa) at startup and upon underflow, generating a waveform of consecutive pulses until the timer stops counting. An interrupt request can be generated each time the counter underflows.
The "counter set value + 1" and "reload register set value + 1" are effective as count values.

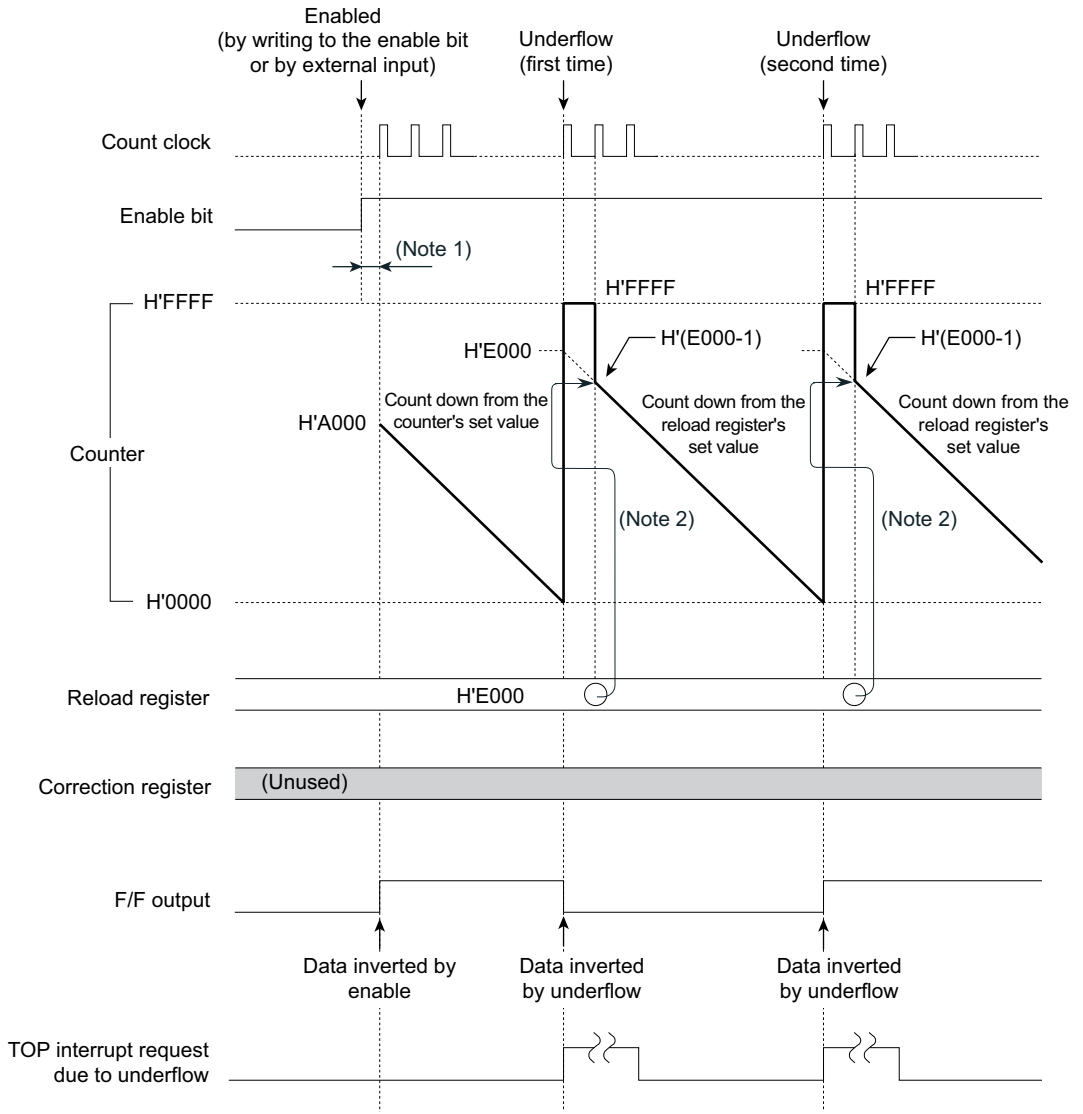
For example, if the initial counter value is 4 and the initial reload register value is 5, then the timer operates as shown below.



Note 1: What actually is seen in the cycle immediately during enable is the previous counter value, and not 4.
Note 2: What actually is seen in the cycle immediately during underflow is H'FFFF (underflow value), and not 5.
Note 3: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.3.17 Example of Counting in TOP Continuous Output Mode

In the example below, the counter and the reload register are initially set to H'A000 and H'E000, respectively. When the timer is enabled, the counter starts counting down and when it underflows after reaching the minimum count, the counter is loaded with the content of "the reload register -1" and continues counting down. However, the timing for reloading is at the cycle after underflow.



Note 1: A count clock dependent delay is included before F/F output changes state after the timer is enabled.
Note 2: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.3.18 Typical Operation in TOP Continuous Output Mode

(2) Precautions on using TOP continuous output mode

The following describes precautions to be observed when using TOP continuous output mode.

- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF. The reload reads the value that "the reload register -1" into the counter at the timing of the counter clock after the underflow.
- Because the timer operates synchronously with the count clock, a count clock-dependent delay is included before F/F output is inverted after the timer is enabled.

10.4 TIO (Input/Output-Related 16-Bit Timer)

10.4.1 Outline of TIO

TIO (Timer Input/Output) is an input/output-related 16-bit timer, whose operation mode can be selected from the following by mode switching in software, one at a time:

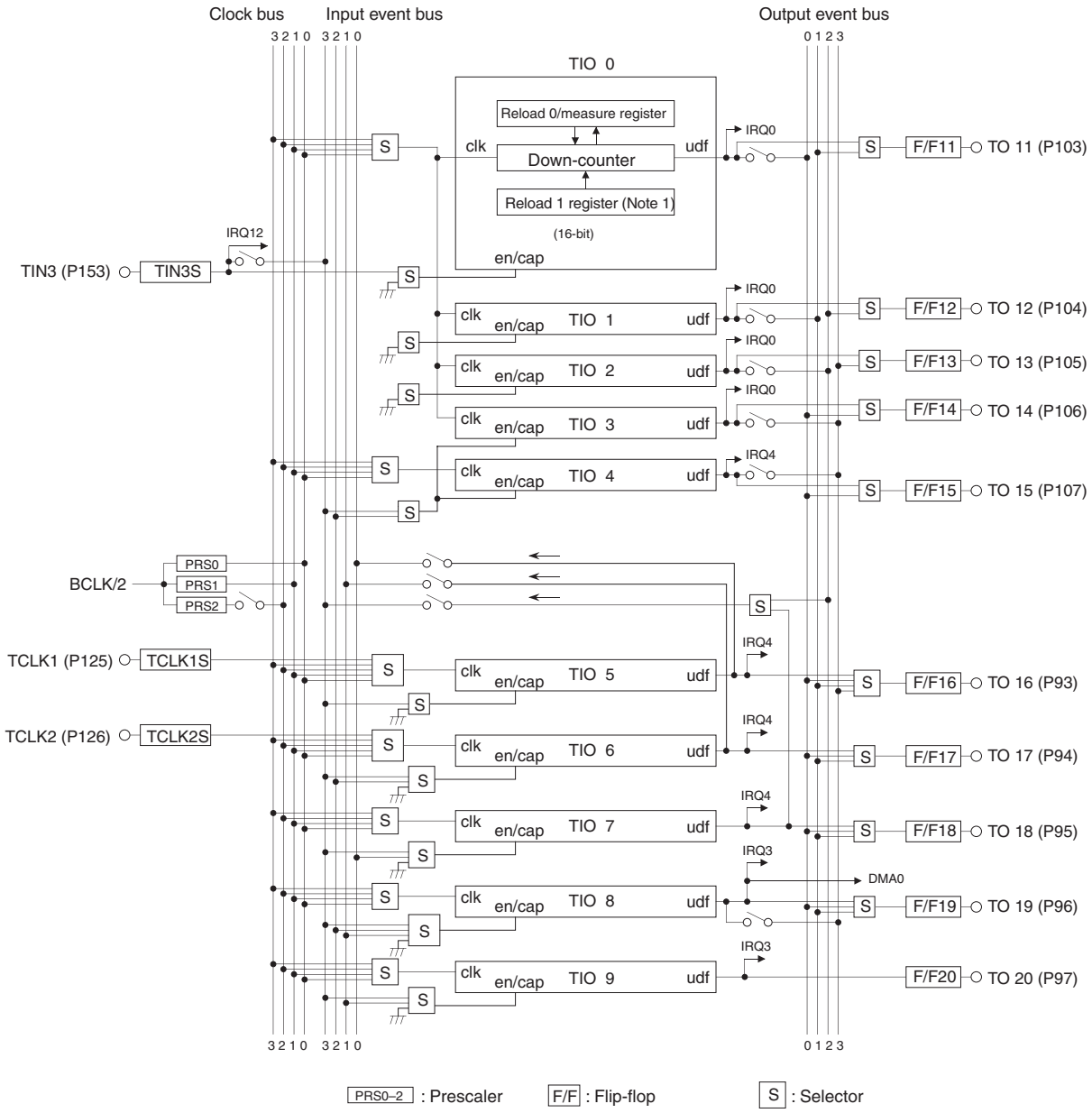
<Input modes>
- Measure clear input mode
- Measure free-run input mode
- Noise processing input mode

<Output modes without correction function>
- PWM output mode
- Single-shot output mode
- Delayed single-shot output mode
- Continuous output mode

The table below shows specifications of TIO. The diagram in the next page shows a block diagram of TIO.

Table 10.4.1 Specifications of TIO (Input/Output-Related 16-Bit Timer)

Item	Specification
Number of channels	10 channels
Counter	16-bit down-counter
Reload register	16-bit reload register
Measure register	16-bit capture register
Timer startup	Started by writing to the enable bit in software or enabled by external input (rising or falling or both edges or "H" or "L" level)
Mode switching	<Input modes> • Measure clear input mode • Measure free-run input mode • Noise processing input mode <Output modes without correction function> • PWM output mode • Single-shot output mode • Delayed single-shot output mode • Continuous output mode
Interrupt request generation	Can be generated by a counter underflow
DMA transfer request generation	Can be generated by a counter underflow (for only the TIO8)



Note 1: The reload 1 register is used in only PWM output mode.

Figure 10.4.1 Block Diagram of TIO (Input/Output-Related 16-Bit Timer)

10.4.2 Outline of Each Mode of TIO

Each mode of TIO is outlined below. For each TIO channel, only one of the following modes can be selected.

(1) Measure clear/free-run input modes

In measure clear/free-run input modes, the timer is used to measure a duration of time from when the counter starts counting till when an external capture signal is entered. And also it is possible to generate both an interrupt requested by underflow at the counter or execution of measurement operation and a DMA transfer request (for only the TIO8) upon underflow of the counter.

After the timer is enabled (by writing to the enable bit in software), the counter starts counting down synchronously with the count clock. When a capture signal is entered from an external device, the counter value at that point in time is written into a register called the “measure register.”

In measure clear input mode, the counter value is initialized to H’FFFF upon capture, from which the counter starts counting down again. The counter returns to H'FFFF upon underflow, from which it starts counting down. Furthermore, when it underflows goes back to H'FFFF and continues down counting. In measure free-run input mode, the counter continues counting down even after capture. The counter returns to H’FFFF upon underflow, from which it starts counting down again.

To stop the counter, disable count by writing to the enable bit in software.

(2) Noise processing input mode

In noise processing input mode, the timer is used to detect that the input signal remained in the same state for over a predetermined time.

In noise processing input mode, a "H" or "L" level on external input activates the counter and if the input signal remains in the same state for over a predetermined time before the counter underflows, the counter generates an interrupt request before stopping. If the valid-level signal being applied turns to an invalid level before the counter underflows, the counter temporarily stops counting and at the next cycle when a valid-level signal is entered again, the counter is reloaded with the value that "the reload register -1" and restarts counting.

The timer stops at the same time the counter underflows or count is disabled by writing to the enable bit. Furthermore, it is possible to generate an interrupt request and a DMA transfer request (for only the TIO8) upon underflow of the counter.

(3) PWM output mode (without correction function)

In PWM output mode, the timer uses two reload registers to generate a waveform with a given duty cycle.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the initial values in the reload 0 and reload 1 registers, the counter is loaded with the value that "the reload 0 register -1" and starts counting down synchronously with the count clock at the next cycle. The next cycle after the first time the counter underflows, it is loaded with the value that "the reload 1 register -1" and continues counting. Thereafter, the counter is loaded with the reload 0 and reload 1 register values alternately each time an underflow occurs. The effective counter value is "reload 0 register set value +1" or "reload 1 register set value +1."
The timer stops at the same time count is disabled by writing to the enable bit (and not in synchronism with PWM output period).

The F/F output waveform in PWM output mode is inverted when the counter starts counting and each time it underflows.

Furthermore, it is possible to generate an interrupt request at even-numbered occurrences of underflow after the counter is enabled and a DMA transfer request (for only the TIO8) every time the counter underflows.
In addition, PWM output mode of TIO does not have function of correction.

(4) Single-shot output mode (without correction function)

In single-shot output mode, the timer generates a pulse in width of "reload 0 register set value + 1" only once and then stops.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the reload 0 register, the counter is loaded with the value that "the reload 0 register -1" and starts counting synchronously with the count clock at the next cycle. The counter counts down and when the minimum count is reached, stops upon underflow.

The F/F output waveform in single-shot output mode is inverted at startup and upon underflow, generating a single-shot pulse waveform in width of "reload 0 register set value + 1" only once.
Furthermore, it is possible to generate an interrupt request and a DMA transfer request (for only the TI08) upon underflow of the counter.

(5) Delayed single-shot output mode (without correction function)

In delayed single-shot output mode, the timer generates a pulse in width of "reload 0 register set value + 1" after a finite time equal to "counter set value + 1" only once and then stops.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the counter and reload 0 register, it starts counting down from the counter's set value synchronously with the count clock. The next cycle after the first time the counter underflows, it is loaded with the value that "the reload 0 register -1" and continues counting down. The counter stops when it underflows next time.

The F/F output waveform in delayed single-shot output mode is inverted when the counter underflows first time and next, generating a single-shot pulse waveform in width of "reload 0 register set value + 1" after a finite time equal to "first set value of counter + 1" only once.
Furthermore, it is possible to generate an interrupt request and a DMA transfer request (for only the TI08) upon the first and next underflows of the counter.

(6) Continuous output mode (without correction function)

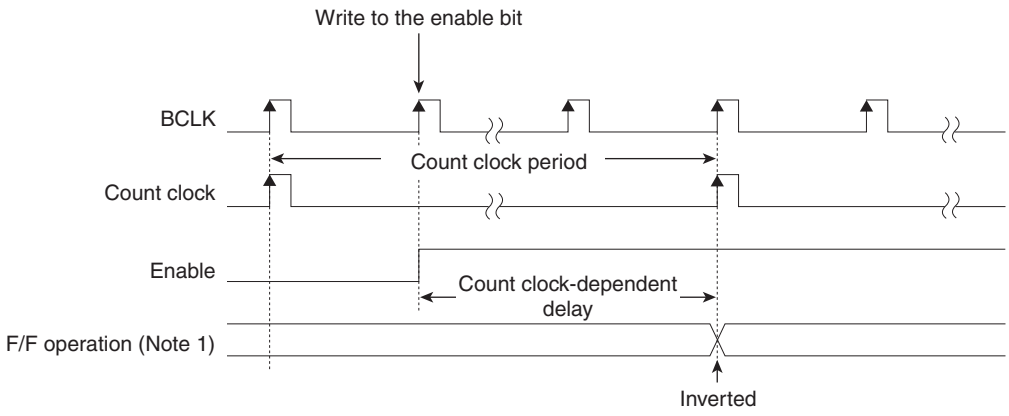
In continuous output mode, the timer counts down starting from the set value of the counter and the next cycle after the counter underflows, it is loaded with the value that "the reload 0 register -1." Thereafter, this operation is repeated each time the counter underflows, thus generating consecutive pulses in width of "reload 0 register set value + 1."

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the counter and reload 0 register, it starts counting down from the counter's set value synchronously with the count clock and when the minimum count is reached, generates an underflow. This underflow causes the counter to be loaded with the content of the reload 0 register and start counting over again. Thereafter, this operation is repeated each time an underflow occurs. To stop the counter, disable count by writing to the enable bit in software. The timing for reloading to counter is the cycle after underflow.

The F/F output waveform in continuous output mode is inverted at startup and upon underflow, generating a waveform of consecutive pulses until the timer stops counting.
Furthermore, it is possible to generate an interrupt request and a DMA transfer request (for only the TI08) each time the counter underflows.

<Count clock-dependent delay>

- Because the timer operates synchronously with the count clock, there is a count clock-dependent delay from when the timer is enabled till when it actually starts operating. In operation mode where the F/F output is inverted when the timer is enabled, the F/F output is inverted synchronously with the count clock.



Note 1: This applies to the case where F/F output is inverted when the timer is enabled.

Figure 10.4.2 Count Clock Dependent Delay

10.4.3 TIO Related Register Map

Shown below is a TIO related register map.

TIO Related Register Map (1/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0300	TIO0 Counter (TIO0CT)		10-87
H'0080 0302	(Use inhibited area)		
H'0080 0304	TIO0 Reload 1 Register (TIO0RL1)		10-89
H'0080 0306	TIO0 Reload 0/Measure Register (TIO0RL0)		10-88
	(Use inhibited area)		
H'0080 0310	TIO1 Counter (TIO1CT)		10-87
H'0080 0312	(Use inhibited area)		
H'0080 0314	TIO1 Reload 1 Register (TIO1RL1)		10-89
H'0080 0316	TIO1 Reload 0/Measure Register (TIO1RL0)		10-88
H'0080 0318	(Use inhibited area)		
H'0080 031A	TIO0–3 Control Register 0 (TIO03CR0)		10-80
H'0080 031C	(Use inhibited area)	TIO0–3 Control Register 1 (TIO03CR1)	10-81
H'0080 031E	(Use inhibited area)		
H'0080 0320	TIO2 Counter (TIO2CT)		10-87
H'0080 0322	(Use inhibited area)		
H'0080 0324	TIO2 Reload 1 Register (TIO2RL1)		10-89
H'0080 0326	TIO2 Reload 0/Measure Register (TIO2RL0)		10-88
	(Use inhibited area)		
H'0080 0330	TIO3 Counter (TIO3CT)		10-87
H'0080 0332	(Use inhibited area)		
H'0080 0334	TIO3 Reload 1 Register (TIO3RL1)		10-89
H'0080 0336	TIO3 Reload 0/Measure Register (TIO3RL0)		10-88
	(Use inhibited area)		
H'0080 0340	TIO4 Counter (TIO4CT)		10-87
H'0080 0342	(Use inhibited area)		
H'0080 0344	TIO4 Reload 1 Register (TIO4RL1)		10-89
H'0080 0346	TIO4 Reload 0/Measure Register (TIO4RL0)		10-88
H'0080 0348	(Use inhibited area)		
H'0080 034A	TIO4 Control Register (TIO4CR)	TIO5 Control Register (TIO5CR)	10-82 10-84
	(Use inhibited area)		
H'0080 0350	TIO5 Counter (TIO5CT)		10-87
H'0080 0352	(Use inhibited area)		

TIO Related Register Map (2/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0354	TIO5 Reload 1 Register (TIO5RL1)		10-89
H'0080 0356	TIO5 Reload 0/Measure Register (TIO5RL0)		10-88
	(Use inhibited area)		
H'0080 0360	TIO6 Counter (TIO6CT)		10-87
H'0080 0362	(Use inhibited area)		
H'0080 0364	TIO6 Reload 1 Register (TIO6RL1)		10-89
H'0080 0366	TIO6 Reload 0/Measure Register (TIO6RL0)		10-88
H'0080 0368	(Use inhibited area)		
H'0080 036A	TIO6 Control Register (TIO6CR)	TIO7 Control Register (TIO7CR)	10-85 10-86
	(Use inhibited area)		
H'0080 0370	TIO7 Counter (TIO7CT)		10-87
H'0080 0372	(Use inhibited area)		
H'0080 0374	TIO7 Reload 1 Register (TIO7RL1)		10-89
H'0080 0376	TIO7 Reload 0/Measure Register (TIO7RL0)		10-88
	(Use inhibited area)		
H'0080 0380	TIO8 Counter (TIO8CT)		10-87
H'0080 0382	(Use inhibited area)		
H'0080 0384	TIO8 Reload 1 Register (TIO8RL1)		10-89
H'0080 0386	TIO8 Reload 0/Measure Register (TIO8RL0)		10-88
H'0080 0388	(Use inhibited area)		
H'0080 038A	TIO8 Control Register (TIO8CR)	TIO9 Control Register (TIO9CR)	10-86 10-87
	(Use inhibited area)		
H'0080 0390	TIO9 Counter (TIO9CT)		10-87
H'0080 0392	(Use inhibited area)		
H'0080 0394	TIO9 Reload 1 Register (TIO9RL1)		10-89
H'0080 0396	TIO9 Reload 0/Measure Register (TIO9RL0)		10-88
	(Use inhibited area)		
H'0080 03BC	TIO0-9 Enable Protect Register (TIOPRO)		10-90
H'0080 03BE	TIO0-9 Count Enable Register (TIOCEN)		10-91

10.4.4 TIO Control Registers

The TIO control registers are used to select operation modes of TIO0–9 (measure input, noise processing input, PWM output, single-shot output, delayed single-shot output or continuous output mode), as well as select the count enable and count clock sources.
Following TIO control registers are provided for each timer group.

- TIO0–3 Control Register 0 (TIO03CR0)
- TIO0–3 Control Register 1 (TIO03CR1)
- TIO4 Control Register (TIO4CR)
- TIO5 Control Register (TIO5CR)
- TIO6 Control Register (TIO6CR)
- TIO7 Control Register (TIO7CR)
- TIO8 Control Register (TIO8CR)
- TIO9 Control Register (TIO9CR)

TIO0–3 Control Register 0 (TIO03CR0) <Address: H'0080 031A>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
TIO3EEN	TIO3M			TIO2ENS	TIO2M			TIO1ENS	TIO1M			TIO0ENS	TIO0M		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	TIO3EEN (Note 1) TIO3 external input enable bit	0: Disable external input 1: Enable external input	R	W
1–3	TIO3M TIO3 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Noise processing input mode 111: Noise processing input mode	R	W
4	TIO2ENS (Reserved bit)	Fix to "0"	0	0
5–7	TIO2M TIO2 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Use inhibited 111: Use inhibited	R	W
8	TIO1ENS (Reserved bit)	Fix to "0"	0	0
9–11	TIO1M TIO1 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Use inhibited 111: Use inhibited	R	W
12	TIO0ENS TIO0 enable/measure input source select bit	0: Does not use enable/measure input source 1: External input TIN3	R	W
13–15	TIO0M TIO0 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Noise processing input mode 111: Noise processing input mode	R	W

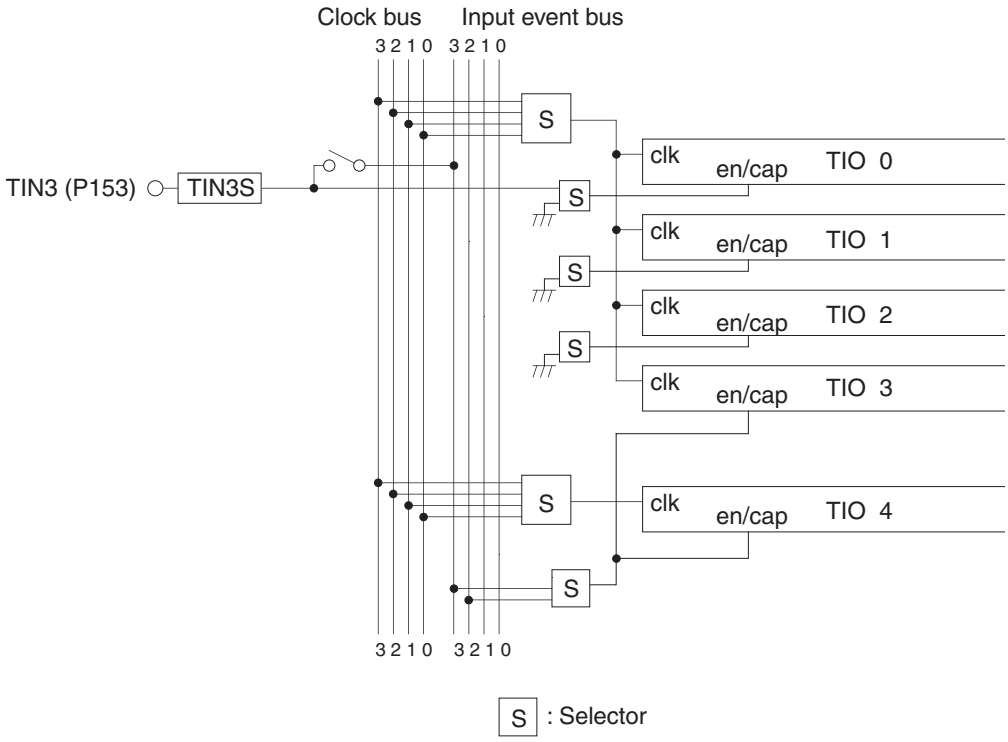
Note 1: During measure free-run/clear input mode, even if this bit is set to "0" (external input disabled), when a capture signal is entered from an external device, the counter value at that point in time is written into the measure register. In measure clear input mode, however, if this bit = "0" (external input disabled), the counter value is not initialized (H'FFFF) upon capture and, therefore, this bit should be set to "1" (external input enabled).

Notes: • This register must always be accessed in halfwords.

• Operation mode can only be set or changed while the counter is inactive.

• To select TIO3 enable/measure input sources, use the TIO4 Control Register TIO34ENS (TIO3, TIO4 enable/measure input source select) bits.

• TIO1 and TIO2 do not have the capture function during measure free-run/clear input mode.



Note: • This diagram only illustrates TIO control registers and is partly omitted.

Figure 10.4.3 Outline Diagram of TIO0–4 Clock and Enable Inputs

TIO0–3 Control Register 1 (TIO03CR1) <Address: H'0080 031D>

b8	9	10	11	12	13	14	b15
0	0	0	0	0	0	TIO03CKS 0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–13	No function assigned. Fix to "0".		0	0
14, 15	TIO03CKS TIO0–3 clock source select bit	00: Clock bus 0 01: Clock bus 1 10: Clock bus 2 11: Clock bus 3	R	W

TIO4 Control Register (TIO4CR) <Address: H'0080 034A>

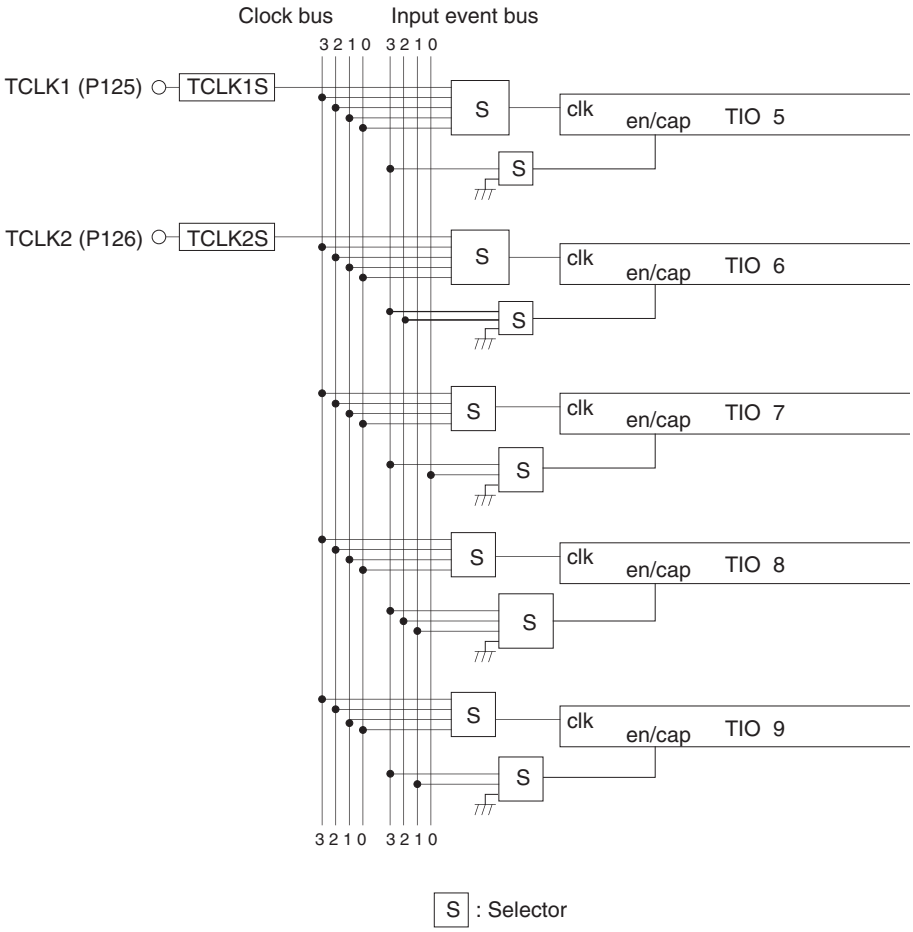
b0	1	2	3	4	5	6	b7
TIO4CKS		TIO4EEN	TIO34ENS		TIO4M		
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0, 1	TIO4CKS TIO4 clock source select bit	00: Clock bus 0 01: Clock bus 1 10: Clock bus 2 11: Clock bus 3	R	W
2	TIO4EEN (Note 1) TIO4 external input enable bit	0: Disable external input 1: Enable external input	R	W
3, 4	TIO34ENS TIO3,4 enable/measure input source select bit	00: Does not use enable/measure input source 01: Does not use enable/measure input source 10: Input event bus 2 11: Input event bus 3	R	W
5–7	TIO4M TIO4 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Noise processing input mode 111: Noise processing input mode	R	W

Note 1: During measure free-run/clear input mode, even if this bit is set to "0" (external input disabled), when a capture signal is entered from an external device, the counter value at that point in time is written into the measure register. In measure clear input mode, however, if this bit = "0" (external input disabled), the counter value is not initialized (H'FFFF) upon capture and, therefore, this bit should be set to "1" (external input enabled).

Note: • Operation mode can only be set or changed while the counter is inactive.



Note: • This diagram only illustrates TIO control registers and is partly omitted.

Figure 10.4.4 Outline Diagram of TIO5–9 Clock and Enable Inputs

TIO5 Control Register (TIO5CR) <Address: H'0080 034B>

b8	9	10	11	12	13	14	b15
TIO5CKS			TIO5ENS		TIO5M		
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	TIO5CKS TIO5 clock source select bit	000: External input TCLK1 001: External input TCLK1 010: External input TCLK1 011: External input TCLK1 100: Clock bus 0 101: Clock bus 1 110: Clock bus 2 111: Clock bus 3	R	W
11, 12	TIO5ENS TIO5 enable/measure input source select bit	00: Does not use enable/measure input source 01: Does not use enable/measure input source 10: Does not use enable/measure input source 11: Input event bus 3	R	W
13–15	TIO5M TIO5 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Noise processing input mode 111: Noise processing input mode	R	W

Note: • Operation mode can only be set or changed while the counter is inactive.

TIO6 Control Register (TIO6CR) <Address: H'0080 036A>

b0	1	2	3	4	5	6	b7
TIO6CKS			TIO6ENS		TIO6M		
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–2	TIO6CKS TIO6 clock source select bit	000: External input TCLK2 001: External input TCLK2 010: External input TCLK2 011: External input TCLK2 100: Clock bus 0 101: Clock bus 1 110: Clock bus 2 111: Clock bus 3	R	W
3, 4	TIO6ENS TIO6 enable/measure input source select bit	00: Does not use enable/measure input source 01: Does not use enable/measure input source 10: Input event bus 2 11: Input event bus 3	R	W
5–7	TIO6M TIO6 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Noise processing input mode 111: Noise processing input mode	R	W

Note: • Operation mode can only be set or changed while the counter is inactive.

TIO7 Control Register (TIO7CR) <Address: H'0080 036B>

b8	9	10	11	12	13	14	b15
	TIO7CKS		TIO7ENS		TIO7M		
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	No function assigned. Fix to "0".		0	0
9, 10	TIO7CKS TIO7 clock source select bit	00: Clock bus 0 01: Clock bus 1 10: Clock bus 2 11: Clock bus 3	R	W
11, 12	TIO7ENS TIO7 enable/measure input source select bit	00: Does not use enable/measure input source 01: Does not use enable/measure input source 10: Input event bus 0 11: Input event bus 3	R	W
13–15	TIO7M TIO7 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Noise processing input mode 111: Noise processing input mode	R	W

Note: • Operation mode can only be set or changed while the counter is inactive.

TIO8 Control Register (TIO8CR) <Address: H'0080 038A>

b0	1	2	3	4	5	6	b7
TIO8CKS		TIO8ENS			TIO8M		
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0, 1	TIO8CKS TIO8 clock source select bit	00: Clock bus 0 01: Clock bus 1 10: Clock bus 2 11: Clock bus 3	R	W
2–4	TIO8ENS TIO8 enable/measure input source select bit	000: Does not use enable/measure input source 001: Does not use enable/measure input source 010: Does not use enable/measure input source 011: Does not use enable/measure input source 100: Does not use enable/measure input source 101: Input event bus 1 110: Input event bus 2 111: Input event bus 3	R	W
5–7	TIO8M TIO8 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Noise processing input mode 111: Noise processing input mode	R	W

Note: • Operation mode can only be set or changed while the counter is inactive.

TIO9 Control Register (TIO9CR) <Address: H'0080 038B>

b8	9	10	11	12	13	14	b15
	TIO9CKS		TIO9ENS		TIO9M		
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	No function assigned. Fix to "0".		0	–
9, 10	TIO9CKS TIO9 clock source select bit	00: Clock bus 0 01: Clock bus 1 10: Clock bus 2 11: Clock bus 3	R	W
11, 12	TIO9ENS TIO9 enable/measure input source select bit	00: Does not use enable/measure input source 01: Does not use enable/measure input source 10: Input event bus 1 11: Input event bus 3	R	W
13–15	TIO9M TIO9 operation mode select bit	000: Single-shot output mode 001: Delayed single-shot output mode 010: Continuous output mode 011: PWM output mode 100: Measure clear input mode 101: Measure free-run input mode 110: Noise processing input mode 111: Noise processing input mode	R	W

Note: • Operation mode can only be set or changed while the counter is inactive.

10.4.5 TIO Counters (TIO0CT–TIO9CT)

TIO0 Counter (TIO0CT) <Address: H'0080 0300>
TIO1 Counter (TIO1CT) <Address: H'0080 0310>
TIO2 Counter (TIO2CT) <Address: H'0080 0320>
TIO3 Counter (TIO3CT) <Address: H'0080 0330>
TIO4 Counter (TIO4CT) <Address: H'0080 0340>
TIO5 Counter (TIO5CT) <Address: H'0080 0350>
TIO6 Counter (TIO6CT) <Address: H'0080 0360>
TIO7 Counter (TIO7CT) <Address: H'0080 0370>
TIO8 Counter (TIO8CT) <Address: H'0080 0380>
TIO9 Counter (TIO9CT) <Address: H'0080 0390>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
TIO0CT-TIO9CT															
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

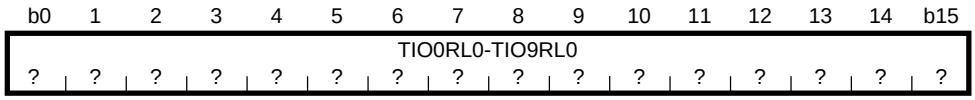
b	Bit Name	Function	R	W
0–15	TIO0CT–TIO9CT	16-bit counter value	R(Note 1)	

Note 1: Protected against write during PWM output mode.
Note: • These registers must always be accessed in halfwords.

The TIO counter is a 16-bit down-counter. After the timer is enabled (by writing to the enable bit in software or by external input), the counter starts counting synchronously with the count clock.
These counters are protected against write during PWM output mode.

10.4.6 TIO Reload 0/ Measure Registers (TIO0RL0–TIO9RL0)

TIO0 Reload 0/ Measure Register (TIO0RL0)	<Address: H'0080 0306>
TIO1 Reload 0/ Measure Register (TIO1RL0)	<Address: H'0080 0316>
TIO2 Reload 0/ Measure Register (TIO2RL0)	<Address: H'0080 0326>
TIO3 Reload 0/ Measure Register (TIO3RL0)	<Address: H'0080 0336>
TIO4 Reload 0/ Measure Register (TIO4RL0)	<Address: H'0080 0346>
TIO5 Reload 0/ Measure Register (TIO5RL0)	<Address: H'0080 0356>
TIO6 Reload 0/ Measure Register (TIO6RL0)	<Address: H'0080 0366>
TIO7 Reload 0/ Measure Register (TIO7RL0)	<Address: H'0080 0376>
TIO8 Reload 0/ Measure Register (TIO8RL0)	<Address: H'0080 0386>
TIO9 Reload 0/ Measure Register (TIO9RL0)	<Address: H'0080 0396>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	TIO0RL0–TIO9RL0	16-bit reload register value	R(Note 1)	

Note 1: These registers are protected against write during measure input mode.

Note: • These registers must always be accessed in halfwords.

The TIO Reload 0/ Measure Registers serve dual purposes as a register for reloading data into the TIO Counter Registers (TIO0CT–TIO9CT) and as a measure register during measure input mode. These registers are protected against write during measure input mode.

The content of "the reload 0 register -1" is loaded into the counter synchronously with the count clock at the following timing:

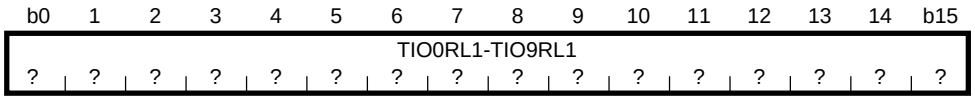
- At the next cycle when after the counter started counting in noise processing input mode, the input signal is inverted and a valid-level signal is entered again before the counter underflows
- At the next cycle when the counter is enabled in single-shot output mode
- At the next cycle when the counter underflowed in delayed single-shot output or continuous output mode
- At the next cycle when the counter is enabled in PWM output mode and when the counter value set by the reload 1 register underflowed

Simply because data is written to the reload 0 register does not mean that the data is loaded into the counter.

If this register is used as a measure register, the counter value is latched into that measure register by event input.

10.4.7 TIO Reload 1 Registers (TIO0RL1–TIO9RL1)

TIO0 Reload 1 Register (TIO0RL1)	<Address: H'0080 0304>
TIO1 Reload 1 Register (TIO1RL1)	<Address: H'0080 0314>
TIO2 Reload 1 Register (TIO2RL1)	<Address: H'0080 0324>
TIO3 Reload 1 Register (TIO3RL1)	<Address: H'0080 0334>
TIO4 Reload 1 Register (TIO4RL1)	<Address: H'0080 0344>
TIO5 Reload 1 Register (TIO5RL1)	<Address: H'0080 0354>
TIO6 Reload 1 Register (TIO6RL1)	<Address: H'0080 0364>
TIO7 Reload 1 Register (TIO7RL1)	<Address: H'0080 0374>
TIO8 Reload 1 Register (TIO8RL1)	<Address: H'0080 0384>
TIO9 Reload 1 Register (TIO9RL1)	<Address: H'0080 0394>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	TIO0RL1–TIO9RL1	16-bit reload register value	R	W

Note: • These registers must always be accessed in halfwords.

The TIO Reload 1 Registers are used to reload data into the TIO Counter Registers (TIO0CT–TIO9CT).

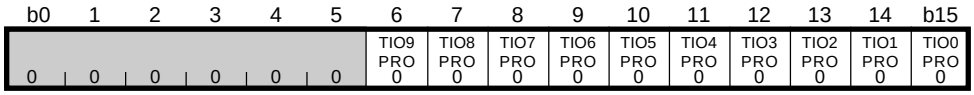
The content of "the reload 1 register -1" is loaded into the counter counting synchronously with the count clock at the following timing:

- At the next cycle when the count value set by the reload 0 register underflowed in PWM output mode

Simply because data is written to the reload 1 register does not mean that the data is loaded into the counter.

10.4.8 TIO Enable Control Registers

TIO0-9 Enable Protect Register (TIOPRO) <Address: H'0080 03BC>



<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0–5	No function assigned. Fix to "0".		0	0
6	TIO9PRO (TIO9 enable protect bit)	0: Enable rewrite	R	W
7	TIO8PRO (TIO8 enable protect bit)	1: Disable rewrite		
8	TIO7PRO (TIO7 enable protect bit)			
9	TIO6PRO (TIO6 enable protect bit)			
10	TIO5PRO (TIO5 enable protect bit)			
11	TIO4PRO (TIO4 enable protect bit)			
12	TIO3PRO (TIO3 enable protect bit)			
13	TIO2PRO (TIO2 enable protect bit)			
14	TIO1PRO (TIO1 enable protect bit)			
15	TIO0PRO (TIO0 enable protect bit)			

Note: • This register must always be accessed in halfwords.

The TIO0-9 Enable Protect Register controls rewriting of the TIO count enable bit described in the next page by enabling or disabling it.

TIO0-9 Count Enable Register (TIOCEN) <Address: H'0080 03BE>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
0	0	0	0	0	0	TIO9 CEN 0	TIO8 CEN 0	TIO7 CEN 0	TIO6 CEN 0	TIO5 CEN 0	TIO4 CEN 0	TIO3 CEN 0	TIO2 CEN 0	TIO1 CEN 0	TIO0 CEN 0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0–5	No function assigned. Fix to "0".		0	0
6	TIO9CEN (TIO9 count enable bit)	0: Stop count	R	W
7	TIO8CEN (TIO8 count enable bit)	1: Enable count		
8	TIO7CEN (TIO7 count enable bit)			
9	TIO6CEN (TIO6 count enable bit)			
10	TIO5CEN (TIO5 count enable bit)			
11	TIO4CEN (TIO4 count enable bit)			
12	TIO3CEN (TIO3 count enable bit)			
13	TIO2CEN (TIO2 count enable bit)			
14	TIO1CEN (TIO1 count enable bit)			
15	TIO0CEN (TIO0 count enable bit)			

Note: • This register must always be accessed in halfwords

The TIO0-9 Count Enable Register controls operation of the TIO counters. To enable any TIO counter in software, enable its corresponding enable protect bit for write and set the count enable bit by writing "1". To stop any TIO counter, enable its corresponding enable protect bit for write and reset the count enable bit by writing "0".

In all but continuous output mode, when the counter stops due to occurrence of an underflow, the count enable bit is automatically reset to "0". Therefore, the TIO0-9 Count Enable Register when accessed for read serves as a status register indicating whether the counter is operating or idle.

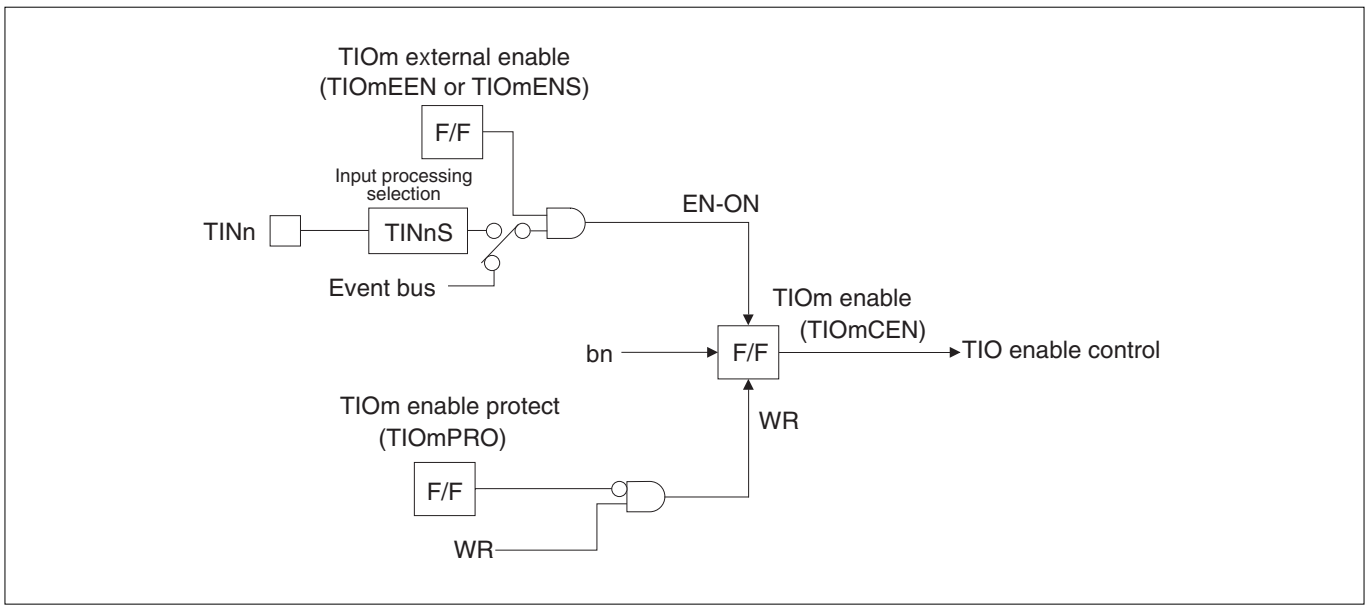


Figure 10.4.5 Configuration of the TIO Enable Circuit

10.4.9 Operation in TIO Measure Free-Run/Clear Input Modes

(1) Outline of TIO measure free-run/clear input modes

In measure free-run/clear input modes, the timer is used to measure a duration of time from when the counter starts counting till when an external capture signal is entered. It is possible to generate an interrupt request upon underflow of the counter or execution of measurement operation and a DMA transfer request (for only the TIO8) upon underflow of the counter.

After the timer is enabled (by writing to the enable bit in software), the counter starts counting down synchronously with the count clock. When a capture signal is entered from an external device, the counter value at that point in time is written into a register called the “measure register.”

In measure clear input mode, the counter value is initialized to H’FFFF upon capture, from which the counter starts counting down again. The counter returns to H'FFFF upon underflow, from which starts counting down.
In measure free-run input mode, the counter continues counting down even after capture. The counter returns to H’FFFF upon underflow, from which it starts counting down again.

To stop the counter, disable count by writing to the enable bit in software.

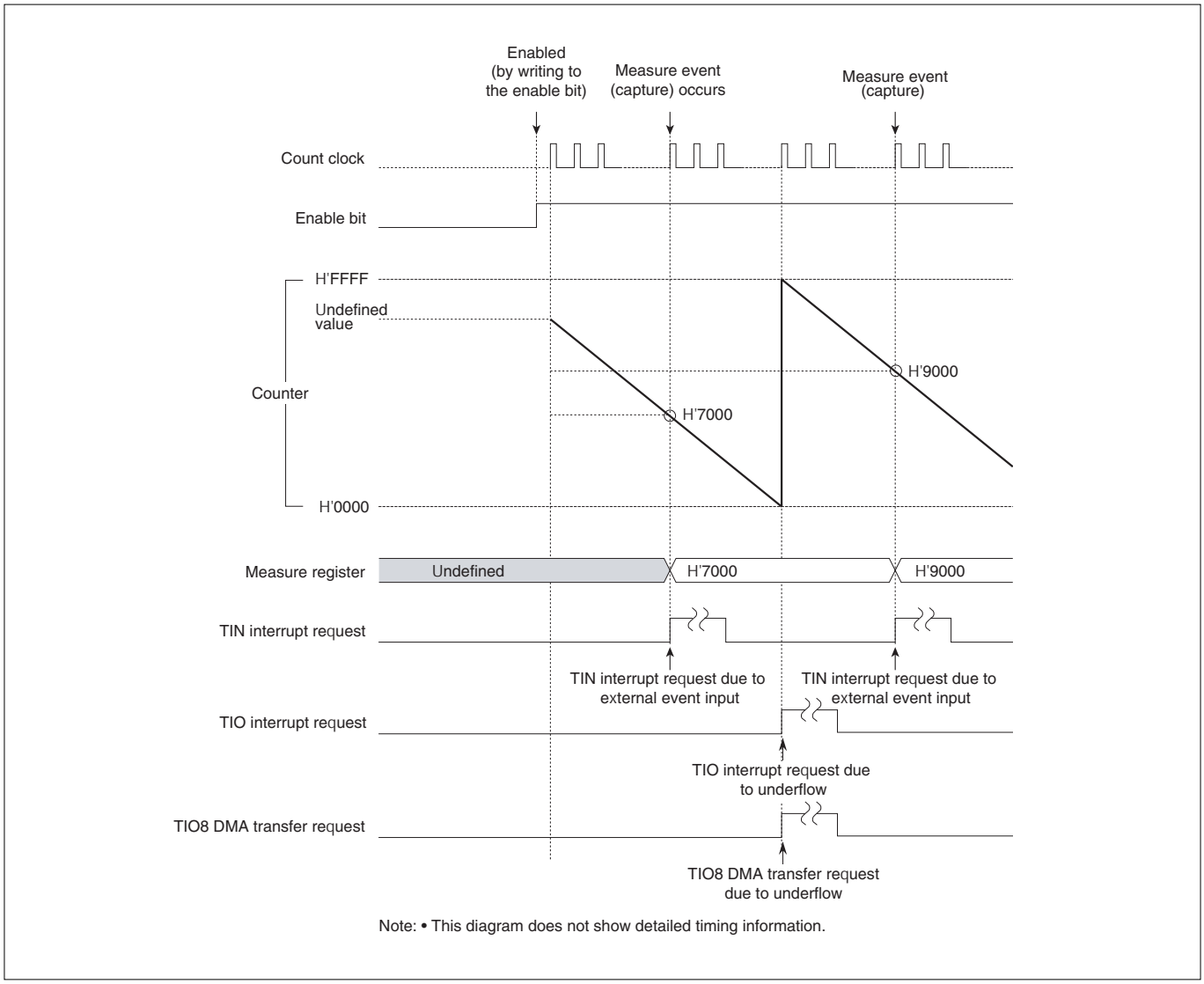
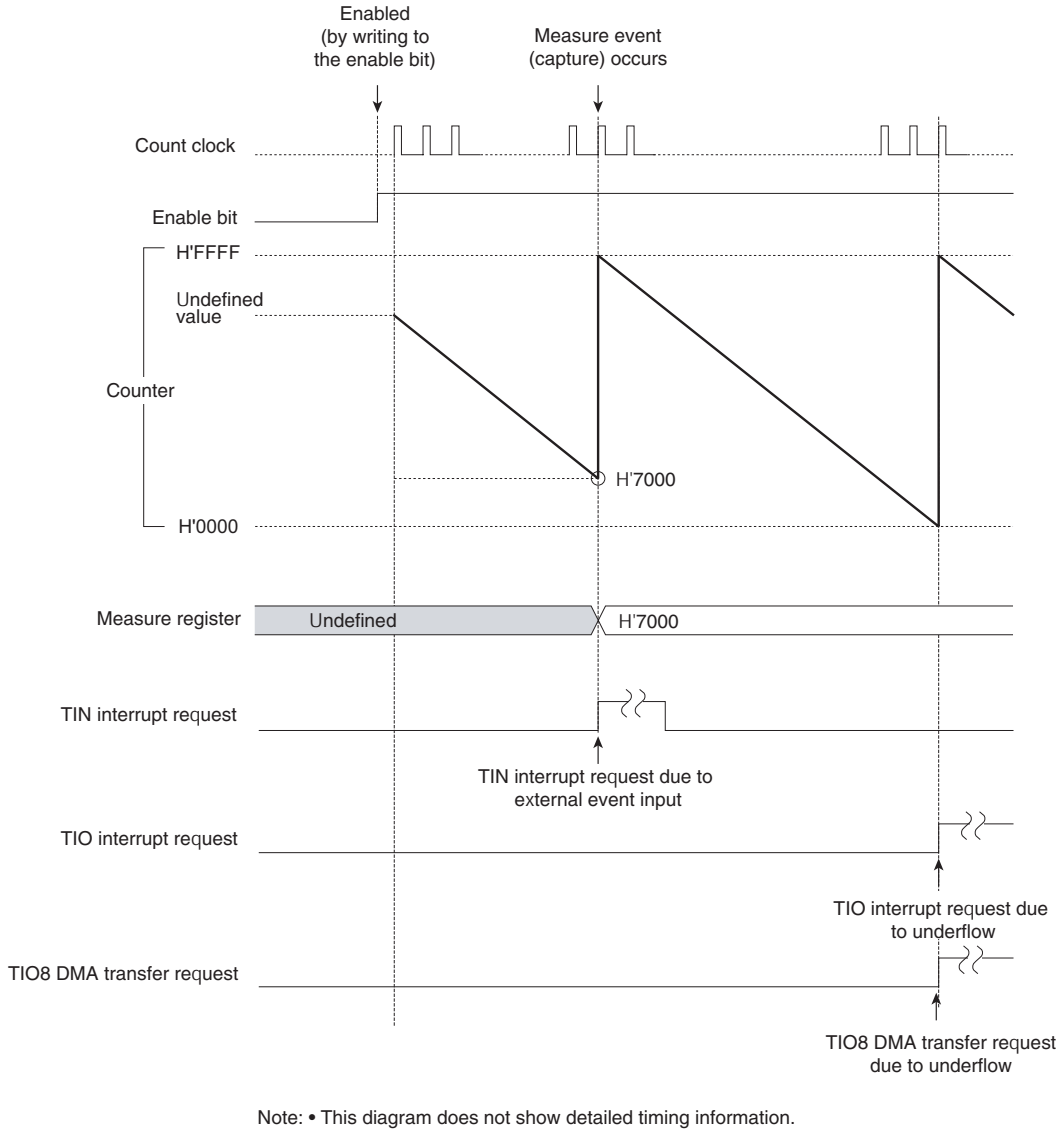


Figure 10.4.6 Typical Operation in Measure Free-Run Input Mode



Note: • This diagram does not show detailed timing information.

Figure 10.4.7 Typical Operation in Measure Clear Input Mode

(2) Precautions on using TIO measure free-run/clear input modes

The following describes precautions to be observed when using TIO measure free-run/clear input modes.

- If measure event input and write to the counter occur in the same clock period, the write value is set in the counter while at the same time latched into the measure register.

10.4.10 Operation in TIO Noise Processing Input Mode

In noise processing input mode, the timer is used to detect that the input signal remained in the same state for over a predetermined time.

In noise processing input mode, a "H" or "L" level on external input activates the counter and if the input signal remains in the same state for over a predetermined time before the counter underflows, the counter generates an interrupt request before stopping. If the valid-level signal being applied turns to an invalid level before the counter underflows, the counter temporarily stops counting and at the next cycle after a valid-level signal is entered again, the counter is reloaded with the value that "reload register -1" and restarts counting, synchronously with the count clock. The effective count width is "reload 0 register set value + 1."

The timer stops at the same time the counter underflows or count is disabled by writing to the enable bit. Furthermore, it is possible to generate an interrupt request and a DMA transfer request (for only the TI08) upon underflow of the counter.

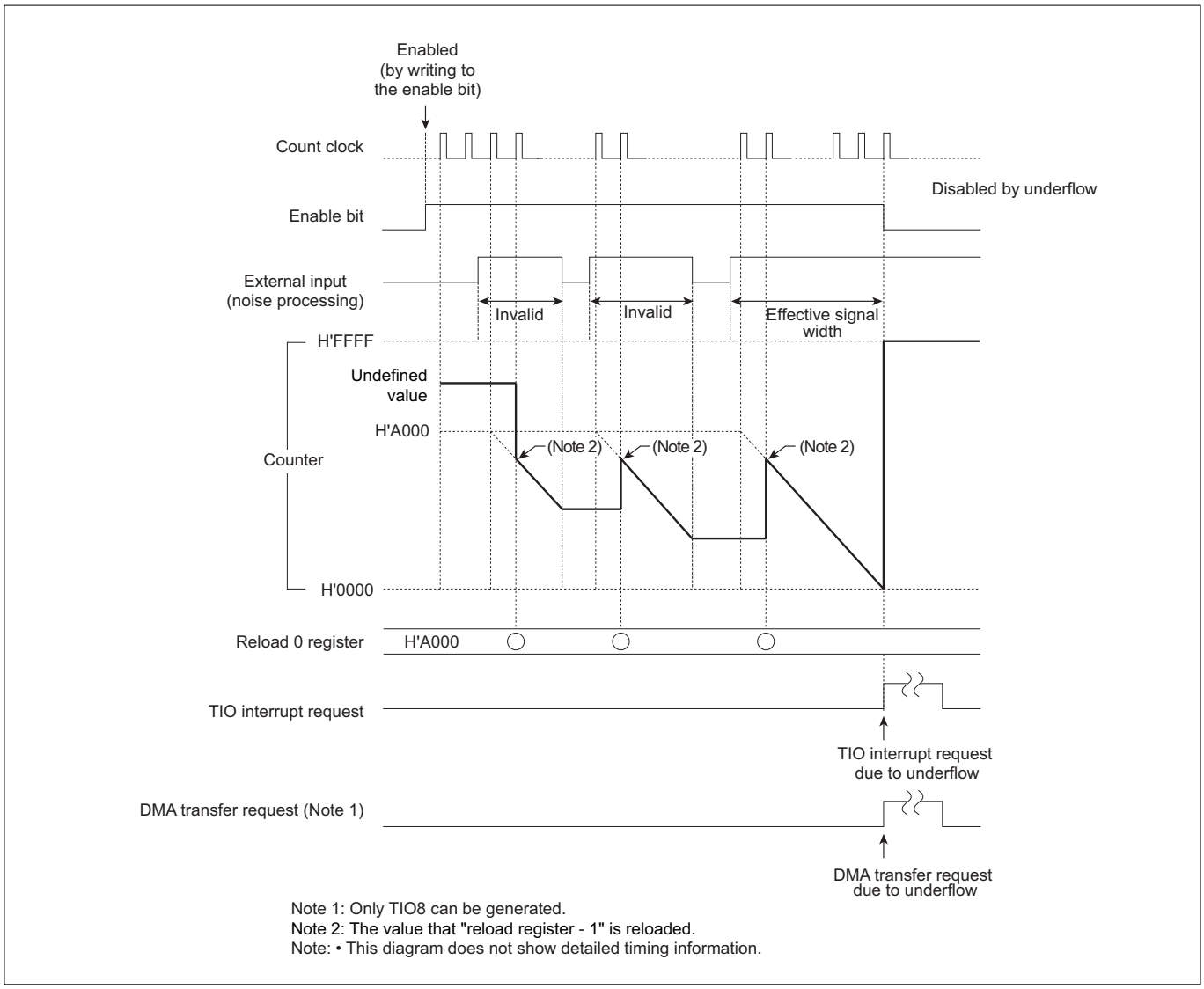


Figure 10.4.8 Typical Operation in Noise Processing Input Mode

10.4.11 Operation in TIO PWM Output Mode

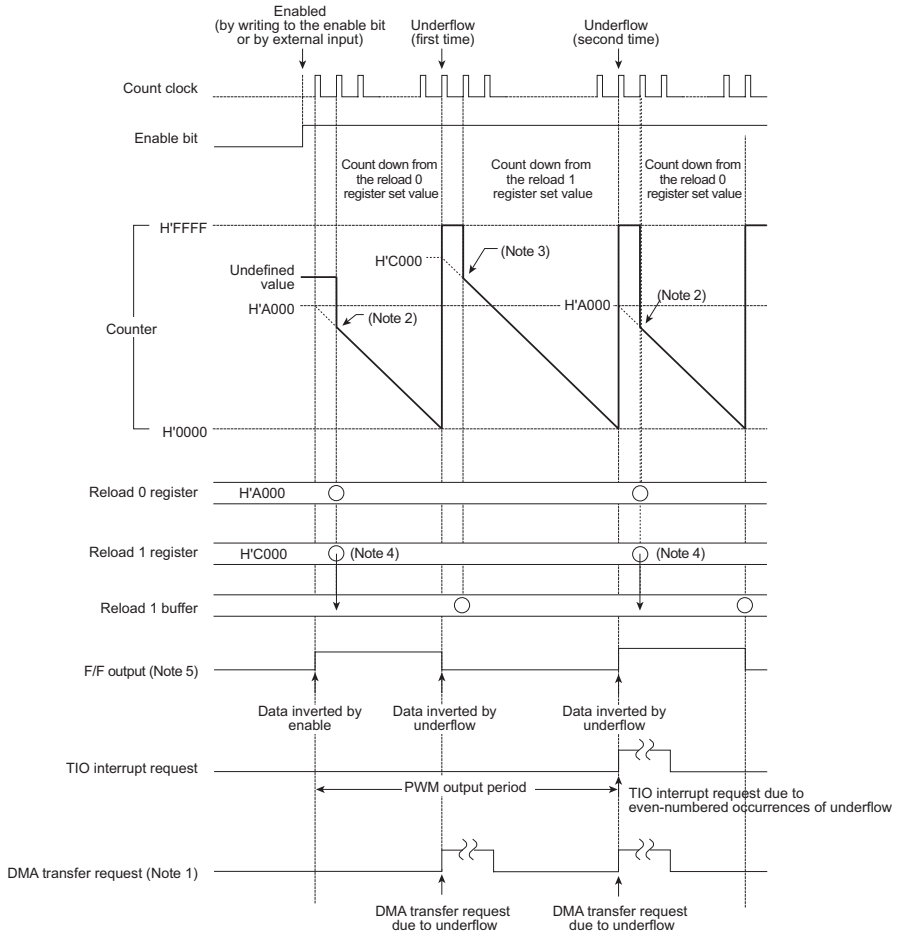
(1) Outline of TIO PWM output mode

In PWM output mode, the timer uses two reload registers to generate a waveform with a given duty cycle.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the initial values in the reload 0 and reload 1 registers, the counter is loaded with the value that "the reload 0 register -1" and starts counting down synchronously with the count clock at the next cycle. At the cycle after the first time the counter underflows, it is loaded with the value that "the reload 1 register -1" and continues counting. Thereafter, the counter is loaded with the reload 0 and reload 1 register values alternately each time an underflow occurs. The "reload 0 register set value +1" and "reload 1 register set value +1" respectively are effective as count values. The timer stops at the same time count is disabled by writing to the enable bit (and not in synchronism with PWM output period).

The F/F output waveform in PWM output mode is inverted (F/F output level changes from "L" to "H" or vice versa) when the counter starts counting and each time it underflows.
Furthermore, it is possible to generate an interrupt request at even-numbered occurrences of underflow after the counter is enabled and a DMA transfer request (for only the TI08) every time the counter underflows.

Note that TIO's PWM output mode does not have the count correction function.



Note 1: Only TIO8 can be generated.
Note 2: The value that "reload 0 register - 1" is reloaded.
Note 3: The value that "reload 1 buffer - 1" is reloaded.
Note 4: When reload0 is reloaded after updating reload0 register, reload 1 buffer is tranferd.
Note 5: Updating of reload 0 and reload 1 during timer operation does not effect PWM waveform that is outputting at present.
Updating is reflected at the next PWM period after updating reload 0 register.
Note: • This diagram does not show detailed timing information.

Figure 10.4.9 Typical Operation in PWM Output Mode

(2) Reload register updates in TIO PWM output mode

In PWM output mode, when the timer remains idle, the reload 0 and reload 1 registers are updated at the same time data are written to the respective registers. But when the timer is operating, the reload 1 register is updated at the reloading the updated reload 0 register by updating the reload 0 register. However, if the reload 0 and reload 1 registers are accessed for read, the read values are always the data that have been written to the respective registers.

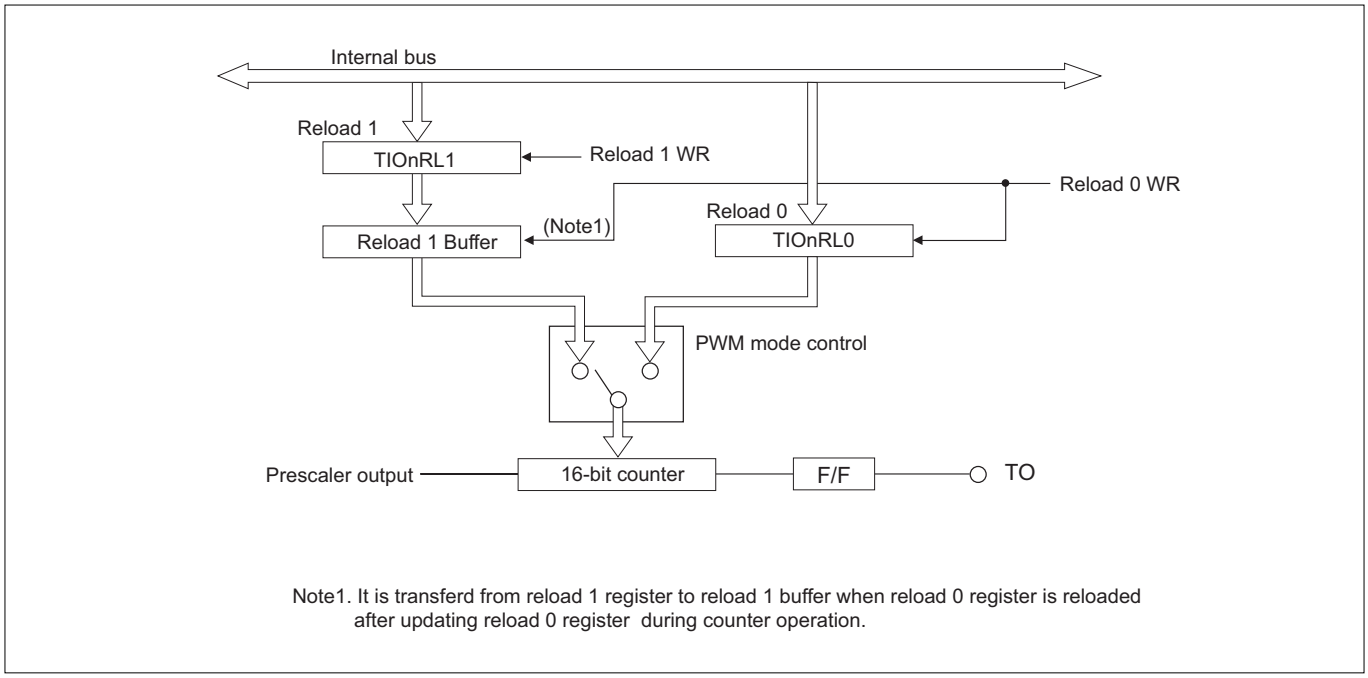


Figure 10.4.10 PWM Circuit Diagram

To rewrite the reload 0 and reload 1 registers while the timer is operating, rewrite the reload 1 register first and then the reload 0 register. That way, the reload 0 and reload 1 registers both are updated synchronously with PWM period, from which the timer starts operating.
This operation can normally be performed collectively by accessing 32-bit addresses beginning with the reload 1 register address wordwise. (Data are automatically written to the reload 1 and then the reload 0 registers in succession.)

Note also that if the reload 0 and reload 1 registers are accessed for read, the read values are always the data that have been written to the respective registers, and not the reload values being actually used.

When altering PWM period by rewriting the reload registers, if the PWM period terminates before the CPU finishes writing to reload 0, the PWM period is not altered in the current session and the data written to the register is reflected in the next period.

When operating in the PWM output mode, writing the reload 0 register and reload 1 register more than twice within the PWM period and meet the following conditions at the same time, the PWM waveform is output with the value that the last time written reload 0 register and finally written reload 1 register.

Condition 1: Start writing reload 0 register after latching the reload 0 register PWM period of the old PWM output period.

Condition 2: Rewrite reload 1 register before latching PWM period of the new PWM output period and start writing reload 0 register after latching PWM period.

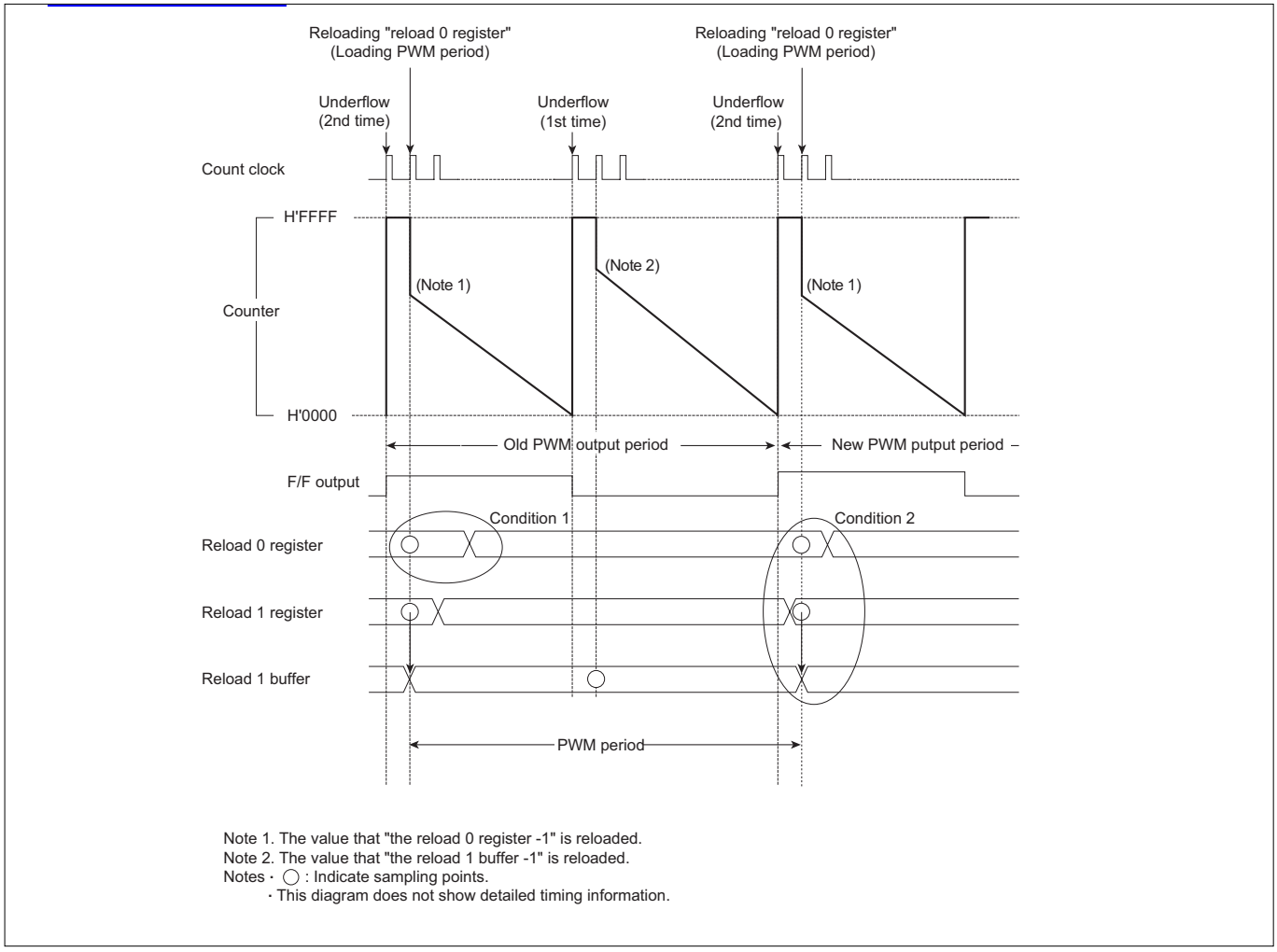


Figure 10.4.11 Update timing of PWM period

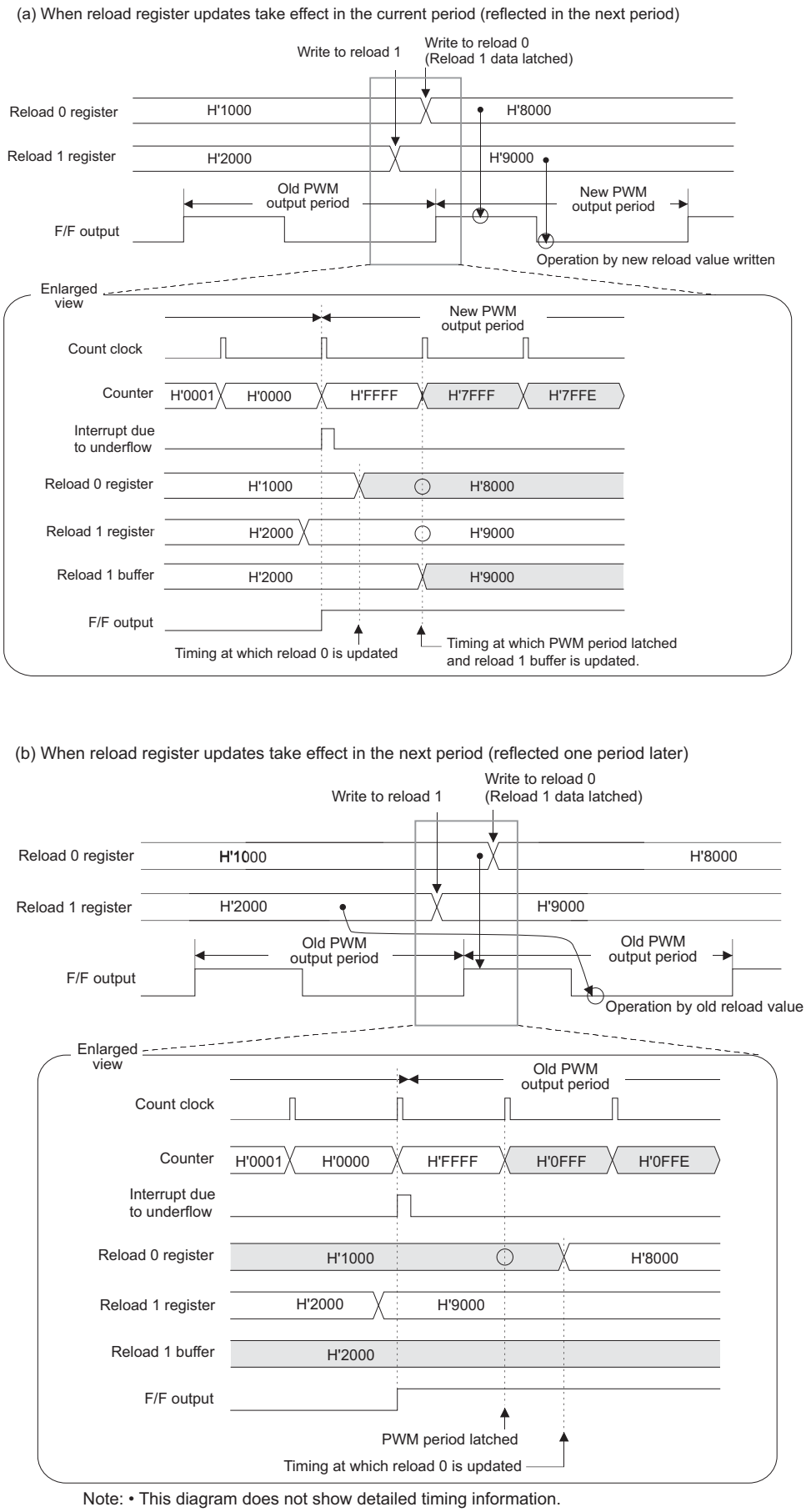
To update PWM period correctly, take either one of the following measures.

- Identify the completion timing of PWM period by reading counter value at writing reload 1 register and reload 0 register, and then start writing reload 1 register and reload 0 register without crossing PWM period.
- When writing to reload 1 register and reload 0 register by using interruption, set the prescaler value of counter as small as possible. By doing this, write to reload 1 register and reload 0 register later than the counter to be H'FFFF in the PWM period.
- Writing reload 1 register and reload 0 register is performed under the period, less than one time per PWM period. (Extend the reload register's rewrite period against PWM period.)

(3) Precautions on using TIO PWM output mode

The following describes precautions to be observed when using TIO PWM output mode.

- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF but changes to "reload value – 1" at the next clock timing.
- Because the timer operates synchronously with the count clock, up to one count clock-dependent delay is generated before F/F output is inverted after writing to the enable bit.



Note: • This diagram does not show detailed timing information.

Figure 10.4.12 Reload 0 and Reload 1 Register Updates in PWM Output Mode

10.4.12 Operation in TIO Single-shot Output Mode (without Correction Function)

(1) Outline of TIO single-shot output mode

In single-shot output mode, the timer generates a pulse in width of "reload 0 register set value + 1" only once and then stops.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the reload 0 register, the counter is loaded with the content of "the reload 0 register -1" and starts counting synchronously with the count clock at the next cycle. The counter counts down and when the minimum count is reached, stops upon underflow.

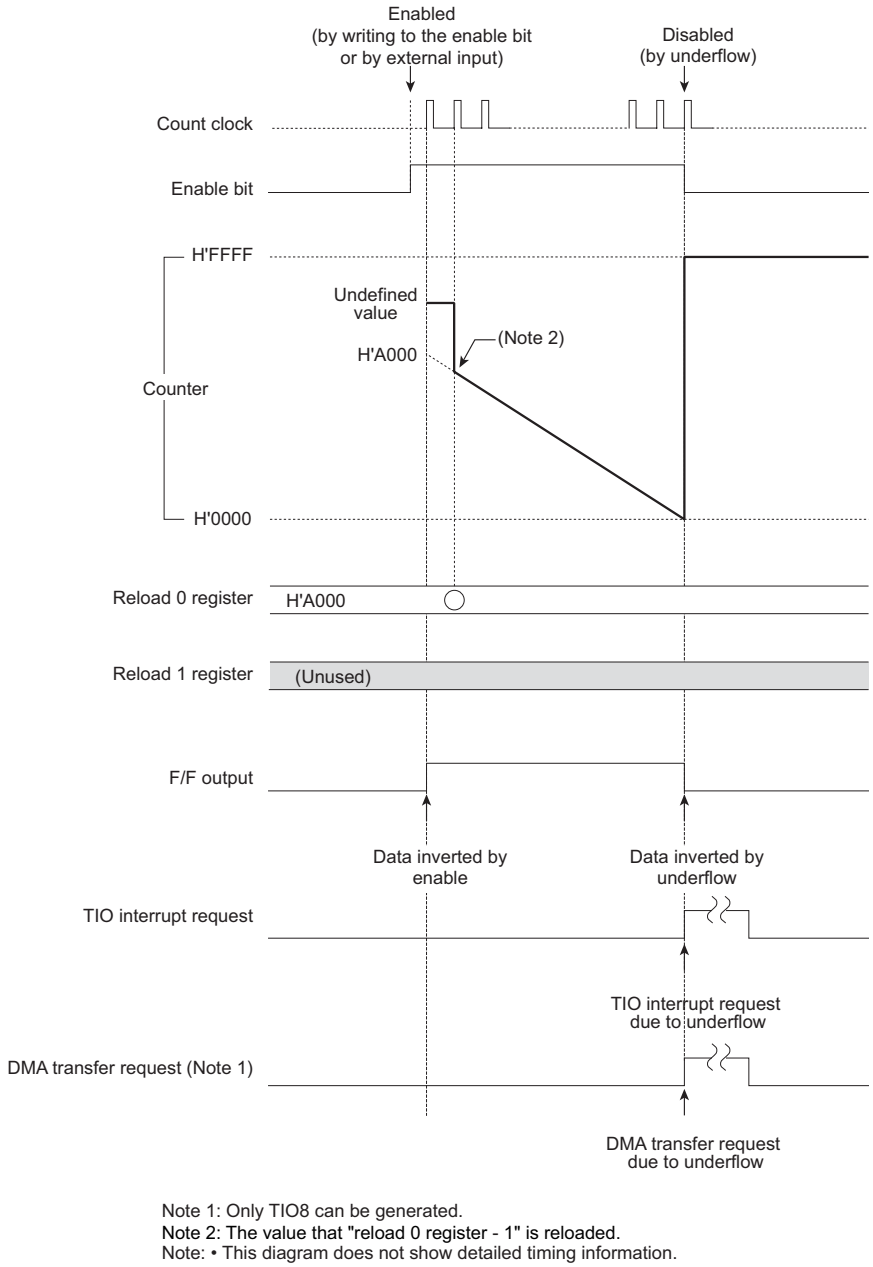
The F/F output waveform in single-shot output mode is inverted (F/F output level changes from "L" to "H" or vice versa) at startup and upon underflow, generating a single-shot pulse waveform in width of "reload 0 register set value + 1" only once.
Furthermore, it is possible to generate an interrupt request and a DMA transfer request (for only the TI08) upon underflow of the counter.

The count value is "reload 0 register set value + 1." (For counting operation, see also Section 10.3.9, “Operation of TOP Single-shot Output Mode.”)

(2) Precautions on using TIO single-shot output mode

The following describes precautions to be observed when using TIO single-shot output mode.

- If the counter stops due to an underflow in the same clock period as the timer is enabled by external input, the former has priority so that the counter stops.
- If the counter stops due to an underflow in the same clock period as count is enabled by writing to the enable bit, the latter has priority so that count is enabled.
- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- Because the timer operates synchronously with the count clock, up to one count clock-dependent delay is generated before F/F output is inverted after writing to the enable bit.



Note 1: Only TIO8 can be generated.
Note 2: The value that "reload 0 register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.4.13 Typical Operation in TIO Single-shot Output Mode (without Correction Function)

10.4.13 Operation in TIO Delayed Single-shot Output Mode (without Correction Function)

(1) Outline of TIO delayed single-shot output mode

In delayed single-shot output mode, the timer generates a pulse in width of "reload 0 register set value + 1" after a finite time equal to (counter set value + 1) only once and then stops.

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the counter and reload 0 register, it starts counting down from the counter's set value synchronously with the count clock. At the cycle after the first time the counter underflows, it is loaded with the value that "the reload 0 register -1" and continues counting down. The counter stops when it underflows next time.

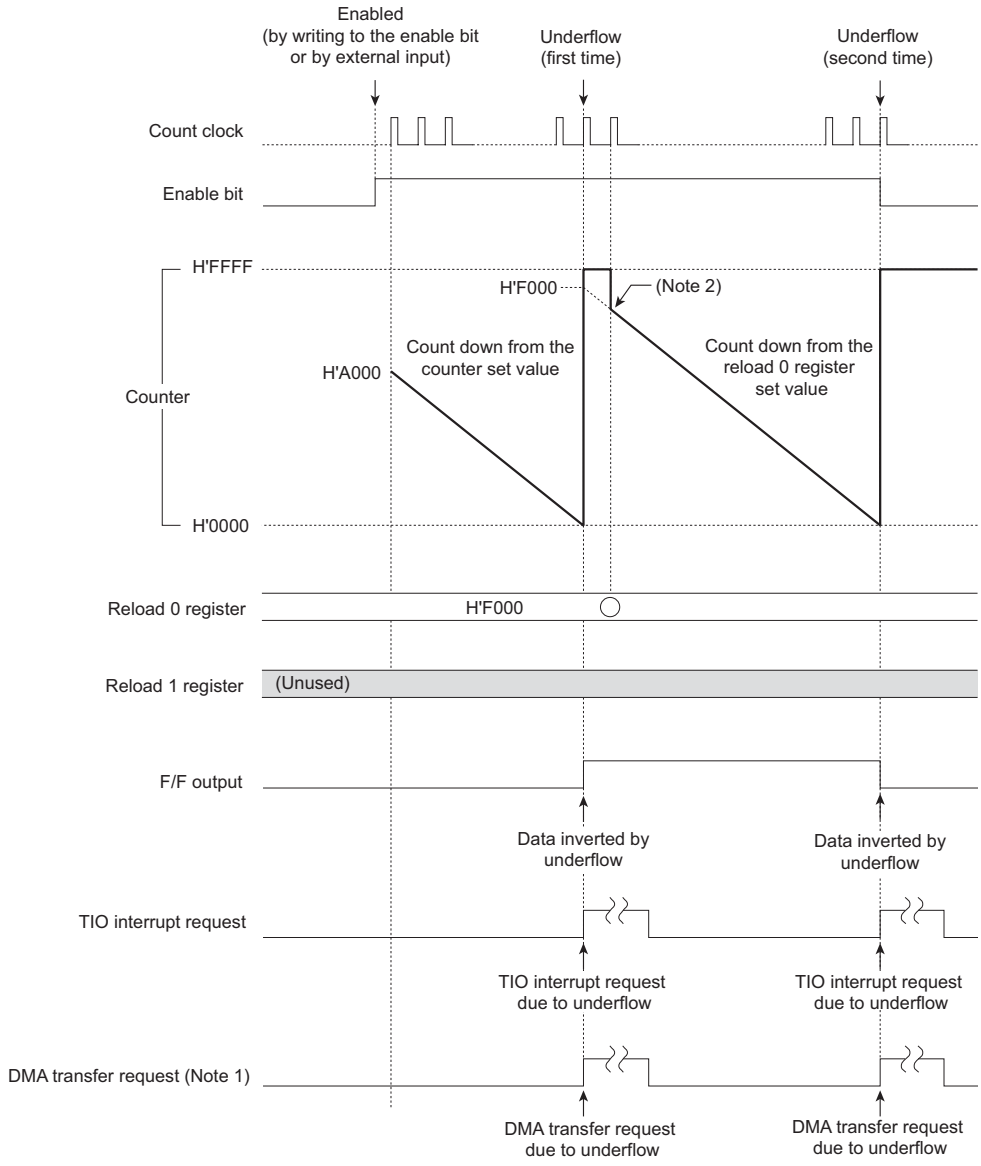
The F/F output waveform in delayed single-shot output mode is inverted (F/F output level changes from "L" to "H" or vice versa) when the counter underflows first time and next, generating a single-shot pulse waveform in width of "reload 0 register set value + 1" after a finite time equal to "first set value of counter + 1" only once.
Furthermore, it is possible to generate an interrupt request and a DMA transfer request (for only the TI08) upon the first and next underflows of the counter.

The "counter set value + 1" and "reload 0 register set value + 1" are effective as count values. (For counting operation, see also Section 10.3.10, "Operation of TOP Delayed Single-shot Output Mode.")

(2) Precautions on using TIO delayed single-shot output mode

The following describes precautions to be observed when using TIO delayed single-shot output mode.

- If the counter stops due to an underflow in the same clock period as the timer is enabled by external input, the former has priority so that the counter stops.
- If the counter stops due to an underflow in the same clock period as count is enabled by writing to the enable bit, the latter has priority so that count is enabled.
- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF. The reload reads the value that "the reload register -1" into the counter at the timing of the counter clock after the underflow.



Note 1: Only TIO8 can be generated.
Note 2: The value that "reload 0 register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.4.14 Typical Operation in TIO Delayed Single-shot Output Mode (without Correction Function)

10.4.14 Operation in TIO Continuous Output Mode (without Correction Function)

(1) Outline of TIO continuous output mode

In continuous output mode, the timer counts down starting from the set value of the counter and the next cycle when the counter underflows, it is loaded with the value that "the reload 0 register -1." Thereafter, this operation is repeated each time the counter underflows, thus generating consecutive pulses whose waveform is inverted in width of "reload 0 register set value + 1."

When the timer is enabled (by writing to the enable bit in software or by external input) after setting the counter and reload 0 register, it starts counting down from the counter's set value synchronously with the count clock and when the minimum count is reached, generates an underflow. The cycle after this underflow causes the counter to be loaded with the content of "the reload 0 register -1" and start counting over again. Thereafter, this operation is repeated each time an underflow occurs. To stop the counter, disable count by writing to the enable bit in software. The timing for reloading to counter is the cycle after underflow.

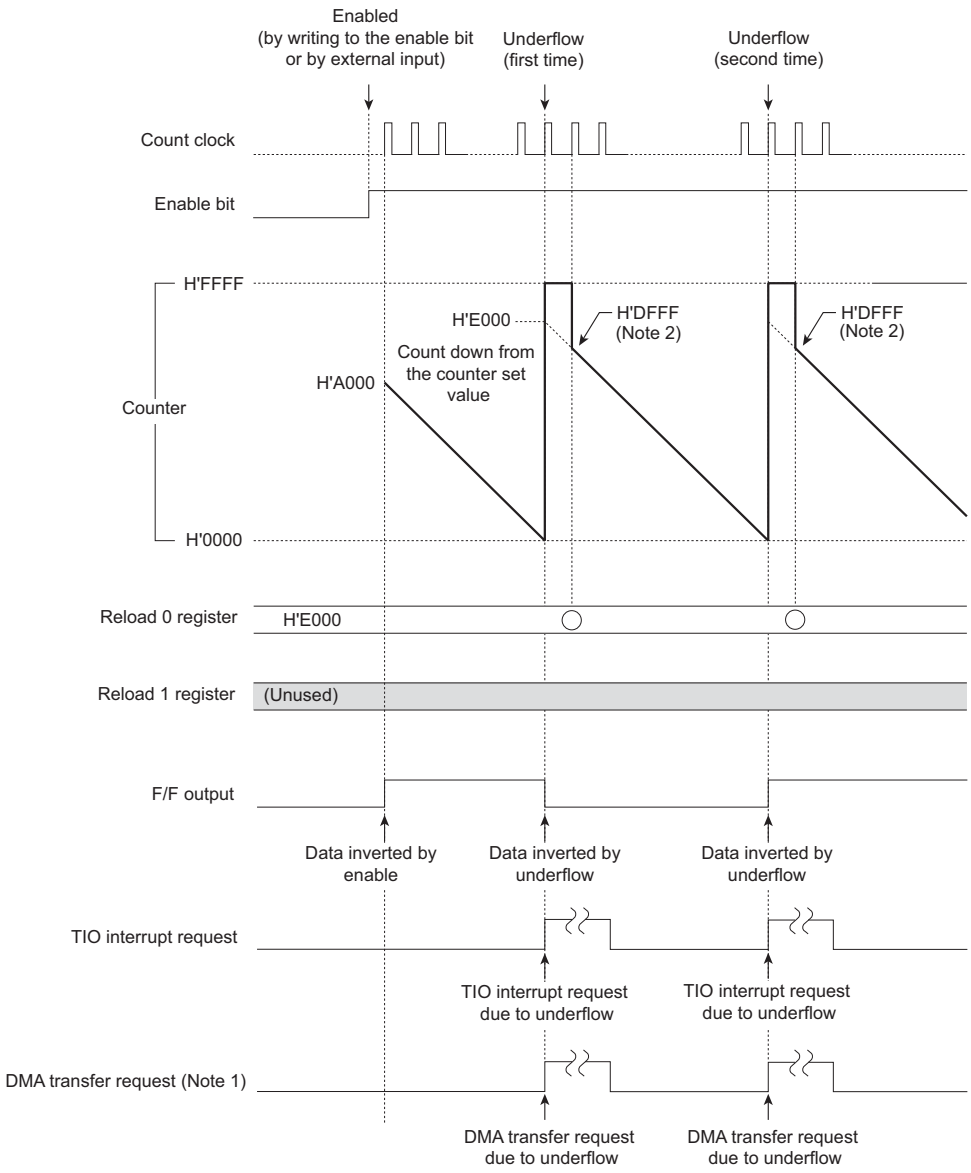
The F/F output waveform in continuous output mode is inverted (F/F output level changes from "L" to "H" or vice versa) at startup and upon underflow, generating a waveform of consecutive pulses until the timer stops counting.
Furthermore, it is possible to generate an interrupt request and a DMA transfer request (for only the TI08) each time the counter underflows.

The "counter set value + 1" and "reload 0 register set value + 1" are effective as count values. (For counting operation, see also Section 10.3.11, "Operation of TOP Continuous Output Mode.")

(2) Precautions on using TIO continuous output mode

The following describes precautions to be observed when using TIO continuous output mode.

- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF but changes to "reload register value – 1" at the next count clock timing.
- Because the timer operates synchronously with the count clock, up to one count clock-dependent delay is generated before F/F output is inverted after writing to the enable bit.



Note 1: Only TIO8 can be generated.
Note 2: The value that "reload 0 register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Figure 10.4.15 Typical Operation in TIO Continuous Output Mode (without Correction Function)

10.5 TMS (Input-Related 16-Bit Timer)

10.5.1 Outline of TMS

TMS (Timer Measure Small) is an input-related 16-bit timer capable of measuring input pulses in two circuit blocks comprising a total of eight channels.
The table below shows specifications of TMS. Figure 10.5.1 shows a block diagram of TMS.

Table 10.5.1 Specifications of TMS (Input-Related 16-Bit Timer)

Item	Specification
Number of channels	8 channels (2 circuit blocks consisting of 4 channels each, 8 channels in total)
Counter	16-bit up-counter × 2
Measure register	16-bit measure register × 8
Timer startup	Started by writing to the enable bit in software
Interrupt request generation	Can be generated by a counter overflow

10.5.2 Outline of TMS Operation

In TMS, when the timer is enabled (by writing to the enable bit in software), the counter starts operating. The counter is a 16-bit up-counter, where the counter value is latched into each measure register when a measure signal is entered from an external device.
The counter stops counting at the same time count is disabled by writing to the enable bit in software.
TIN and TMS interrupt requests can be generated by external measure signal input and counter overflow, respectively (however, TMS0 does not have a TIN interrupt).

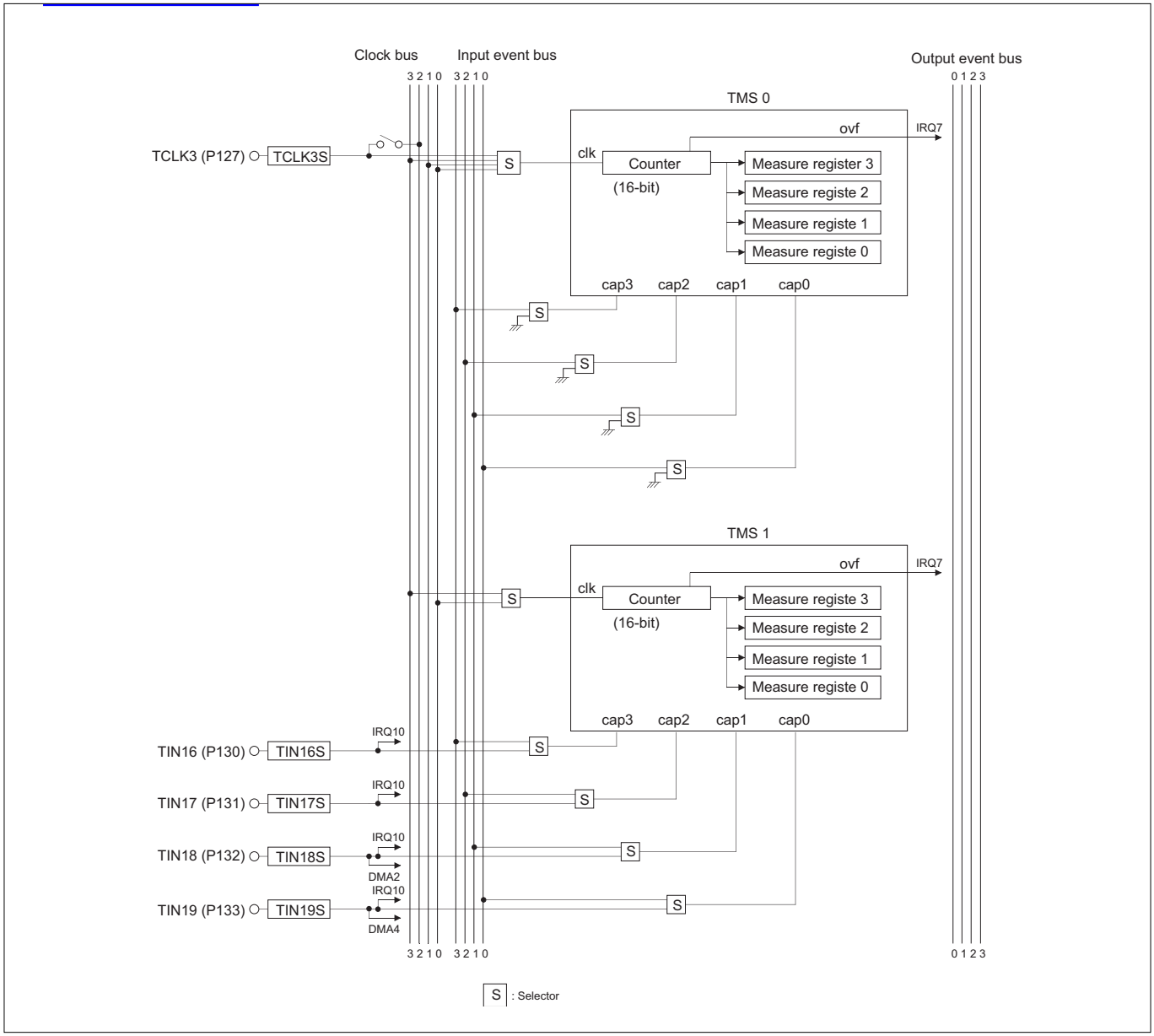


Figure 10.5.1 Block Diagram of TMS (Input-Related 16-Bit Timer)

<Count clock-dependent delay>

- Because the timer operates synchronously with the count clock, there is a count clock-dependent delay from when the timer is enabled till when it actually starts operating.

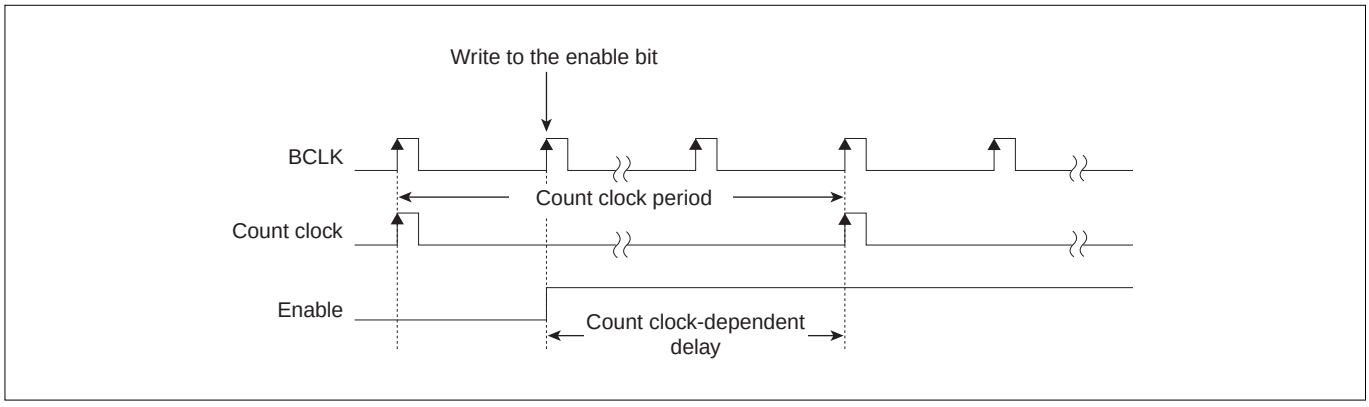


Figure 10.5.2 Count Clock-Dependent Delay

10.5.3 TMS Related Register Map

Shown below is a TMS related register map.

TMS Related Register Map

Address	+0 address b0 – b7	+1 address b8 – b15	See pages
H'0080 03C0	TMS0 Counter (TMS0CT)		10-109
H'0080 03C2	TMS0 Measure 3 Register (TMS0MR3)		10-109
H'0080 03C4	TMS0 Measure 2 Register (TMS0MR2)		10-109
H'0080 03C6	TMS0 Measure 1 Register (TMS0MR1)		10-109
H'0080 03C8	TMS0 Measure 0 Register (TMS0MR0)		10-109
H'0080 03CA	TMS0 Control Register (TMS0CR)	TMS1 Control Register (TMS1CR)	10-108
	(Use inhibited area)		
H'0080 03D0	TMS1 Counter (TMS1CT)		10-109
H'0080 03D2	TMS1 Measure 3 Register (TMS1MR3)		10-109
H'0080 03D4	TMS1 Measure 2 Register (TMS1MR2)		10-109
H'0080 03D6	TMS1 Measure 1 Register (TMS1MR1)		10-109
H'0080 03D8	TMS1 Measure 0 Register (TMS1MR0)		10-109

10.5.4 TMS Control Registers

The TMS control registers are used to select TMS0/1 input events and count clock sources, as well as control count enable. Following two TMS control registers are included:

- TMS0 Control Register (TMS0CR)
- TMS1 Control Register (TMS1CR)

TMS0 Control Register (TMS0CR) <Address: H'0080 03CA>

b0	1	2	3	4	5	6	b7
TMS0SS0	TMS0SS1	TMS0SS2	TMS0SS3	TMS0CKS			TMS0CEN
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	TMS0SS0 TMS0 measure 0 source select bit	0: Does not use measure source 1: Input event bus 0	R	W
1	TMS0SS1 TMS0 measure 1 source select bit	0: Does not use measure source 1: Input event bus 1	R	W
2	TMS0SS2 TMS0 measure 2 source select bit	0: Does not use measure source 1: Input event bus 2	R	W
3	TMS0SS3 TMS0 measure 3 source select bit	0: Does not use measure source 1: Input event bus 3	R	W
4, 5	TMS0CKS TMS0 clock source select bit	00: External input TCLK3 01: Clock bus 0 10: Clock bus 1 11: Clock bus 3	R	W
6	No function assigned. Fix to "0".		0	0
7	TMS0CEN TMS0 count enable bit	0: Stop count 1: Start count	R	W

TMS1 Control Register (TMS1CR) <Address: H'0080 03CB>

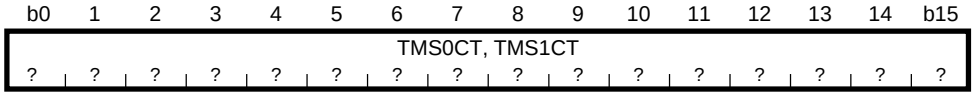
b8	9	10	11	12	13	14	b15
TMS1SS0	TMS1SS1	TMS1SS2	TMS1SS3		TMS1CKS		TMS1CEN
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	TMS1SS0 TMS1 measure 0 source select bit	0: External input TIN19 1: Input event bus 0	R	W
9	TMS1SS1 TMS1 measure 1 source select bit	0: External input TIN18 1: Input event bus 1	R	W
10	TMS1SS2 TMS1 measure 2 source select bit	0: External input TIN17 1: Input event bus 2	R	W
11	TMS1SS3 TMS1 measure 3 source select bit	0: External input TIN16 1: Input event bus 3	R	W
12	No function assigned. Fix to "0".		0	0
13	TMS1CKS TMS1 clock source select bit	0: Clock bus 0 1: Clock bus 3	R	W
14	No function assigned. Fix to "0".		0	0
15	TMS1CEN TMS1 count enable bit	0: Stop count 1: Start count	R	W

10.5.5 TMS Counters (TMS0CT, TMS1CT)

TMS0 Counter (TMS0CT) <Address: H'0080 03C0>
TMS1 Counter (TMS1CT) <Address: H'0080 03D0>



<Upon exiting reset: Undefined>

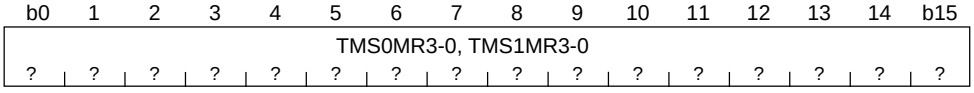
b	Bit Name	Function	R	W
0–15	TMS0CT, TMS1CT	16-bit counter value	R	W

Note: • These registers must always be accessed in halfwords.

The TMS counter is a 16-bit up-counter, which starts counting when the timer is enabled (by writing to the enable bit in software). The counters can be read during operation.

10.5.6 TMS Measure Registers (TMS0MR3–0, TMS1MR3–0)

TMS0 Measure 3 Register (TMS0MR3) <Address: H'0080 03C2>
TMS0 Measure 2 Register (TMS0MR2) <Address: H'0080 03C4>
TMS0 Measure 1 Register (TMS0MR1) <Address: H'0080 03C6>
TMS0 Measure 0 Register (TMS0MR0) <Address: H'0080 03C8>
TMS1 Measure 3 Register (TMS1MR3) <Address: H'0080 03D2>
TMS1 Measure 2 Register (TMS1MR2) <Address: H'0080 03D4>
TMS1 Measure 1 Register (TMS1MR1) <Address: H'0080 03D6>
TMS1 Measure 0 Register (TMS1MR0) <Address: H'0080 03D8>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	TMS0MR3-TMS0MR0 TMS1MR3-TMS1MR0	16-bit counter value	R	–

Notes: • This register is a read-only register.
• This register can be accessed in either byte or halfword.

The TMS measure registers are used to latch counter contents upon event input. The TMS measure registers are a read-only register.

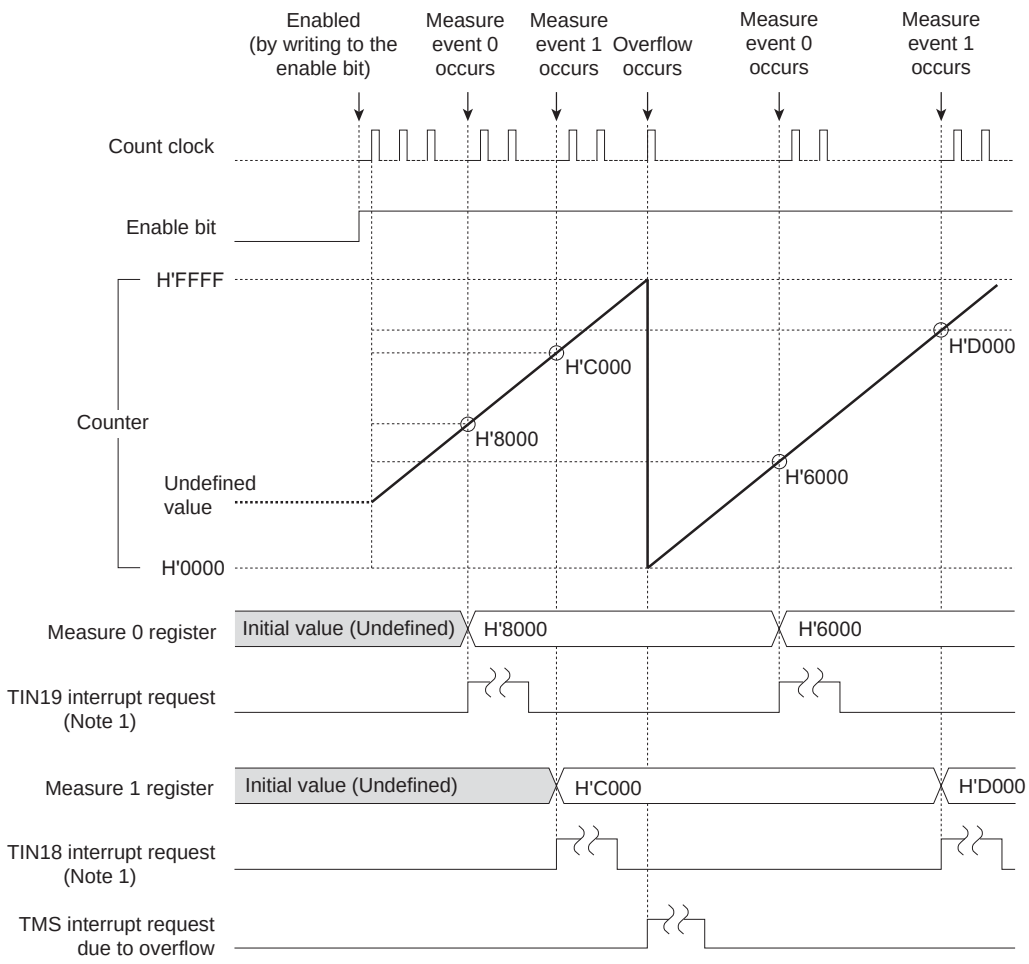
10.5.7 Operation of TMS Measure Input

(1) Outline of TMS measure input

In TMS measure input, the timer starts counting up when it is enabled (by writing to the enable bit in software). Then when event input to TMS is detected while the timer is operating, the counter value is latched into measure registers 0–3.

The timer stops counting at the same time count is disabled by writing to the enable bit.

A TIN interrupt request can be generated by measure signal input from an external device (TMS1 alone has a TIN interrupt. TMS0 does not have a TIN interrupt.) A TMS interrupt request can be generated when the counter overflows.



Note 1: TMS1 alone has a TIN interrupt. (TMS0 does not have a TIN interrupt.)
Note: • This diagram does not show detailed timing information.

Figure 10.5.3 Typical Operation of TMS Measure Input

(2) Precautions on using TMS measure input

The following describes precautions to be observed when using TMS measure input.

- If measure event input and write to the counter occur in the same clock period, the write value is set in the counter while at the same time latched into the measure register.

10.6 TML (Input-Related 32-Bit Timer)

10.6.1 Outline of TML

TML (Timer Measure Large) is an input-related 32-bit timer capable of measuring input pulses in two circuit blocks comprising a total of eight channels.
The table below shows specifications of TML. Figure 10.6.1 shows a block diagram of TML.

Table 10.6.1 Specifications of TML (Input-Related 32-Bit Timer)

Item	Specification
Number of channels	8 channels (2 circuit blocks consisting of 4 channels each, 8 channels in total)
Input clock	BCLK/2 (10.0 MHz when f(BCLK) = 20 MHz) or clock bus 1 input
Counter	32-bit up-counter × 2
Measure register	32-bit measure register × 8
Timer startup	Start counting after exiting the reset state

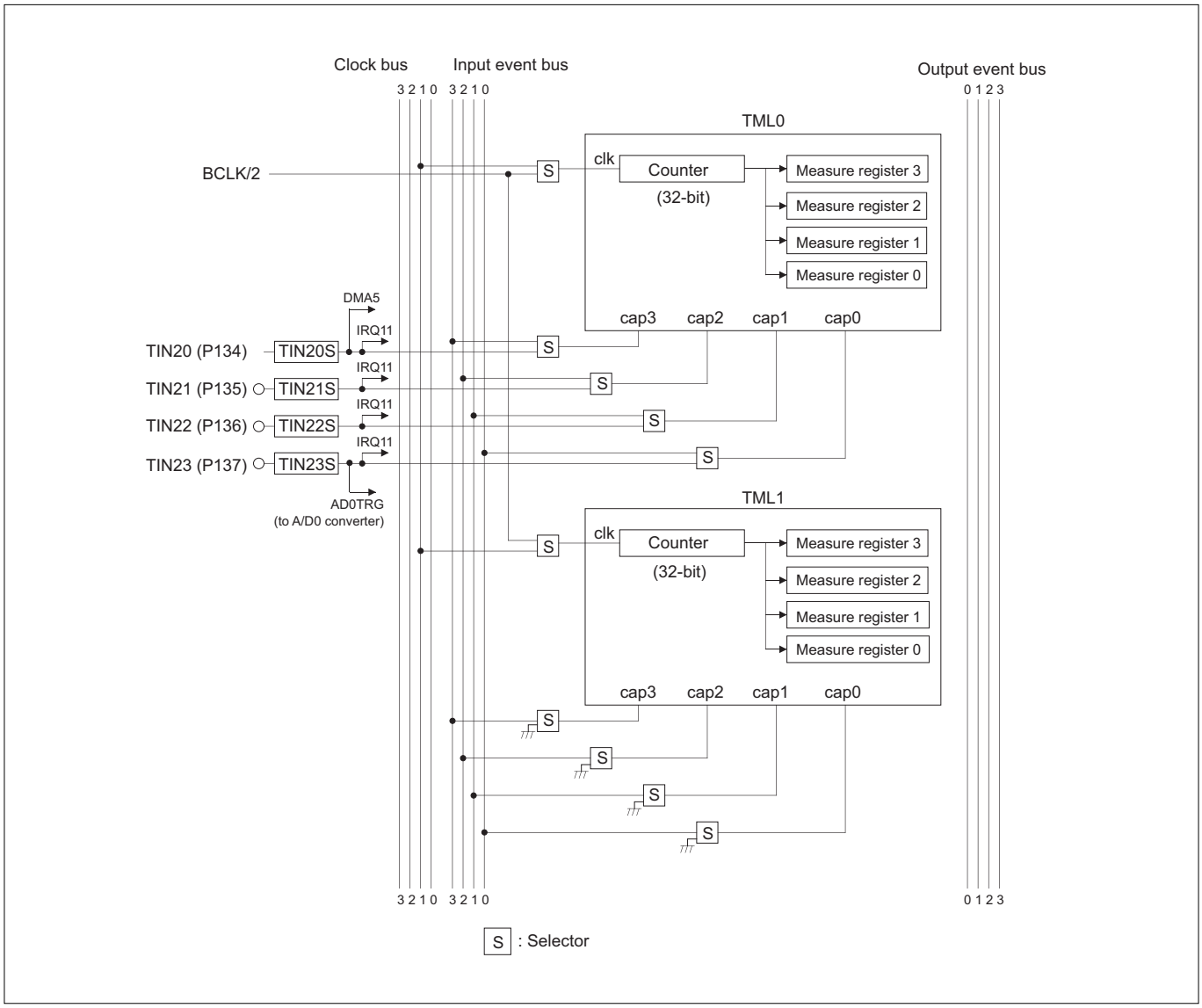


Figure 10.6.1 Block Diagram of TML (Input-Related 32-Bit Timer)

10.6.2 Outline of TML Operation

In TML, the timer starts counting upon deassertion of the reset input signal. The counter included in the timer is a 32-bit up-counter, where when a measure event signal is entered from an external device, the counter value at that point in time is stored in each 32-bit measure register.

When the reset input signal is deasserted, the counter starts operating with a BCLK/2, and cannot be stopped once it has started. The counter is idle only when the microcomputer remains reset.

A TIN interrupt request can be generated by external measure signal input (TML0 alone has a TIN interrupt. TML1 does not have a TIN interrupt.) However, no TML counter overflow interrupts are available.

10.6.3 TML Related Register Map

Shown below is a TML related register map.

TML Related Register Map

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 03E0	TML0 Counter (TML0CTH)	(Upper)	10-114
H'0080 03E2	TML0 Counter (TML0CTL)	(Lower)	10-114
	(Use inhibited area)		
H'0080 03EA	(Use inhibited area)	TML0 Control Register (TML0CR)	10-113
	(Use inhibited area)		
H'0080 03F0	TML0 Measure 3 Register (TML0MR3H)	(Upper)	10-115
H'0080 03F2	TML0 Measure 3 Register (TML0MR3L)	(Lower)	10-115
H'0080 03F4	TML0 Measure 2 Register (TML0MR2H)	(Upper)	10-115
H'0080 03F6	TML0 Measure 2 Register (TML0MR2L)	(Lower)	10-115
H'0080 03F8	TML0 Measure 1 Register (TML0MR1H)	(Upper)	10-115
H'0080 03FA	TML0 Measure 1 Register (TML0MR1L)	(Lower)	10-115
H'0080 03FC	TML0 Measure 0 Register (TML0MR0H)	(Upper)	10-115
H'0080 03FE	TML0 Measure 0 Register (TML0MR0L)	(Lower)	10-115
	(Use inhibited area)		
H'0080 0FE0	TML1 Counter (TML1CTH)	(Upper)	10-114
H'0080 0FE2	TML1 Counter (TML1CTL)	(Lower)	10-114
	(Use inhibited area)		
H'0080 0FEA	(Use inhibited area)	TML1 Control Register (TML1CR)	10-113
	(Use inhibited area)		
H'0080 0FF0	TML1 Measure 3 Register (TML1MR3H)	(Upper)	10-116
H'0080 0FF2	TML1 Measure 3 Register (TML1MR3L)	(Lower)	10-116
H'0080 0FF4	TML1 Measure 2 Register (TML1MR2H)	(Upper)	10-116
H'0080 0FF6	TML1 Measure 2 Register (TML1MR2L)	(Lower)	10-116
H'0080 0FF8	TML1 Measure 1 Register (TML1MR1H)	(Upper)	10-116
H'0080 0FFA	TML1 Measure 1 Register (TML1MR1L)	(Lower)	10-116
H'0080 0FFC	TML1 Measure 0 Register (TML1MR0H)	(Upper)	10-116
H'0080 0FFE	TML1 Measure 0 Register (TML1MR0L)	(Lower)	10-116

10.6.4 TML Control Registers

TML0 Control Register (TML0CR) <Address: H'0080 03EB>

b8	9	10	11	12	13	14	b15
TML0SS0	TML0SS1	TML0SS2	TML0SS3				TML0CKS
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	TML0SS0 TML0 measure 0 source select bit	0: External input TIN23 1: Input event bus 0	R	W
9	TML0SS1 TML0 measure 1 source select bit	0: External input TIN22 1: Input event bus 1	R	W
10	TML0SS2 TML0 measure 2 source select bit	0: External input TIN21 1: Input event bus 2	R	W
11	TML0SS3 TML0 measure 3 source select bit	0: External input TIN20 1: Input event bus 3	R	W
12–14	No function assigned. Fix to "0".		0	0
15	TML0CKS (Note 1) TML0 clock source select bit	0: BCLK/2 1: Clock bus 1	R	W

Note 1: The counter can only be written normally when BCLK/2 is used as the clock source for the counter. If the selected clock source is not BCLK/2, do not write to the counter because it cannot be written normally.

TML1 Control Register (TML1CR) <Address: H'0080 0FEB>

b8	9	10	11	12	13	14	b15
TML1SS0	TML1SS1	TML1SS2	TML1SS3				TML1CKS
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

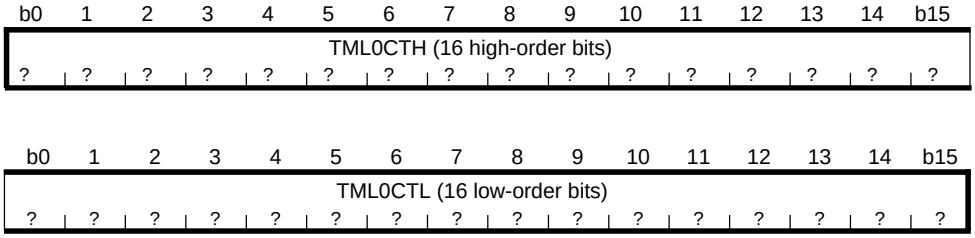
b	Bit Name	Function	R	W
8	TML1SS0 TML1 measure 0 source select bit	0: Does not use measure source 1: Input event bus 0	R	W
9	TML1SS1 TML1 measure 1 source select bit	0: Does not use measure source 1: Input event bus 1	R	W
10	TML1SS2 TML1 measure 2 source select bit	0: Does not use measure source 1: Input event bus 2	R	W
11	TML1SS3 TML1 measure 3 source select bit	0: Does not use measure source 1: Input event bus 3	R	W
12–14	No function assigned. Fix to "0".		0	0
15	TML1CKS (Note 1) TML1 clock source select bit	0: BCLK/2 1: Clock bus 1	R	W

Note 1: The counter can only be written normally when BCLK/2 is used as the clock source for the counter. If the selected clock source is not BCLK/2, do not write to the counter because it cannot be written normally.

The TML control register is used to select TML input event and count clock.

10.6.5 TML Counters

TML0 Counter (Upper) (TML0CTH) <Address: H'0080 03E0>
TML0 Counter (Lower) (TML0CTL) <Address: H'0080 03E2>



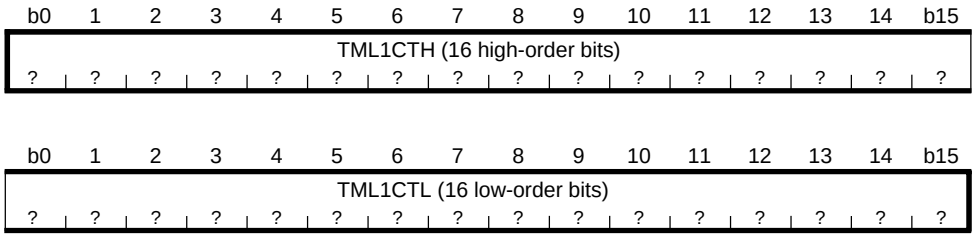
<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	TML0CTH	32-bit counter value (16 high-order bits)	R	W
	TML0CTL	32-bit counter value (16 low-order bits)		

Note: • This register must always be accessed wordwise (in 32 bits) beginning with the address of the TML0CTH.

The TML0 counter is a 32-bit up-counter, which starts counting upon deassertion of the reset input signal. The TML0CTH accommodates the 16 high-order bits of the 32-bit counter, and the TML0CTL accommodates the 16 low-order bits. The counters can be read during operation.

TML1 Counter (Upper) (TML1CTH) <Address: H'0080 0FE0>
TML1 Counter (Lower) (TML1CTL) <Address: H'0080 0FE2>



<Upon exiting reset: Undefined>

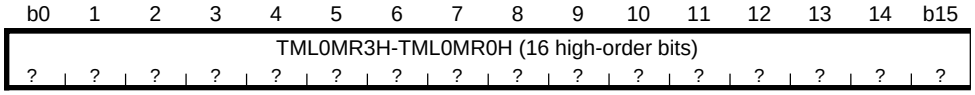
b	Bit Name	Function	R	W
0–15	TML1CTH	32-bit counter value (16 high-order bits)	R	W
	TML1CTL	32-bit counter value (16 low-order bits)		

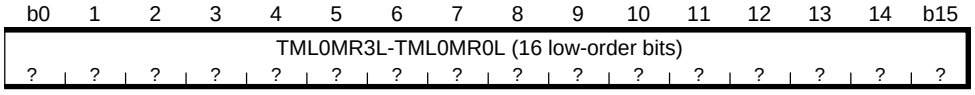
Note: • This register must always be accessed wordwise (in 32 bits) beginning with the address of the TML1CTH.

The TML1 counter is a 32-bit up-counter, which starts counting upon deassertion of the reset input signal. The TML1CTH accommodates the 16 high-order bits of the 32-bit counter, and the TML1CTL accommodates the 16 low-order bits. The counters can be read during operation.

10.6.6 TML Measure Registers

TML0 Measure 3 Register (TML0MR3H)	<Address: H'0080 03F0>
TML0 Measure 3 Register (TML0MR3L)	<Address: H'0080 03F2>
TML0 Measure 2 Register (TML0MR2H)	<Address: H'0080 03F4>
TML0 Measure 2 Register (TML0MR2L)	<Address: H'0080 03F6>
TML0 Measure 1 Register (TML0MR1H)	<Address: H'0080 03F8>
TML0 Measure 1 Register (TML0MR1L)	<Address: H'0080 03FA>
TML0 Measure 0 Register (TML0MR0H)	<Address: H'0080 03FC>
TML0 Measure 0 Register (TML0MR0L)	<Address: H'0080 03FE>





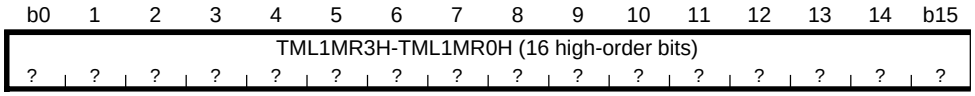
<Upon exiting reset: Undefined>

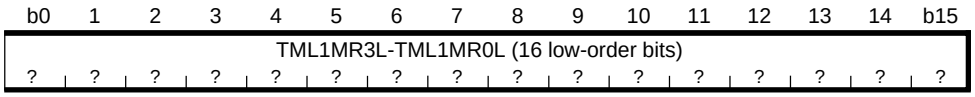
b	Bit Name	Function	R	W
0–15	TML0MR3H–0H	32-bit measure register value (16 high-order bits)	R	–
	TML0MR3L–0L	32-bit measure register value (16 low-order bits)		

Notes: • These registers are a read-only register.
• These registers must always be accessed wordwise (in 32 bits) beginning with the word boundary.

The TML0 measure register is used to latch the counter content upon event input. The TML0 measure register consists of 32 bits, which TML0MR3H–0H and TML0MR3L–0L are 16 high-order bits and 16 low-order bits, respectively. The TML0 measure registers can only be read, and cannot be written to. The register must always be accessed wordwise beginning with the word boundary.

TML1 Measure 3 Register (TML1MR3H)	<Address: H'0080 0FF0>
TML1 Measure 3 Register (TML1MR3L)	<Address: H'0080 0FF2>
TML1 Measure 2 Register (TML1MR2H)	<Address: H'0080 0FF4>
TML1 Measure 2 Register (TML1MR2L)	<Address: H'0080 0FF6>
TML1 Measure 1 Register (TML1MR1H)	<Address: H'0080 0FF8>
TML1 Measure 1 Register (TML1MR1L)	<Address: H'0080 0FFA>
TML1 Measure 0 Register (TML1MR0H)	<Address: H'0080 0FFC>
TML1 Measure 0 Register (TML1MR0L)	<Address: H'0080 0FFE>





<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	TML1MR3H–0H	32-bit measure register value (16 high-order bits)	R	–
	TML1MR3L–0L	32-bit measure register value (16 low-order bits)		

Notes: • These registers are a read-only register.
• These registers must always be accessed wordwise (in 32 bits) beginning with the word boundary.

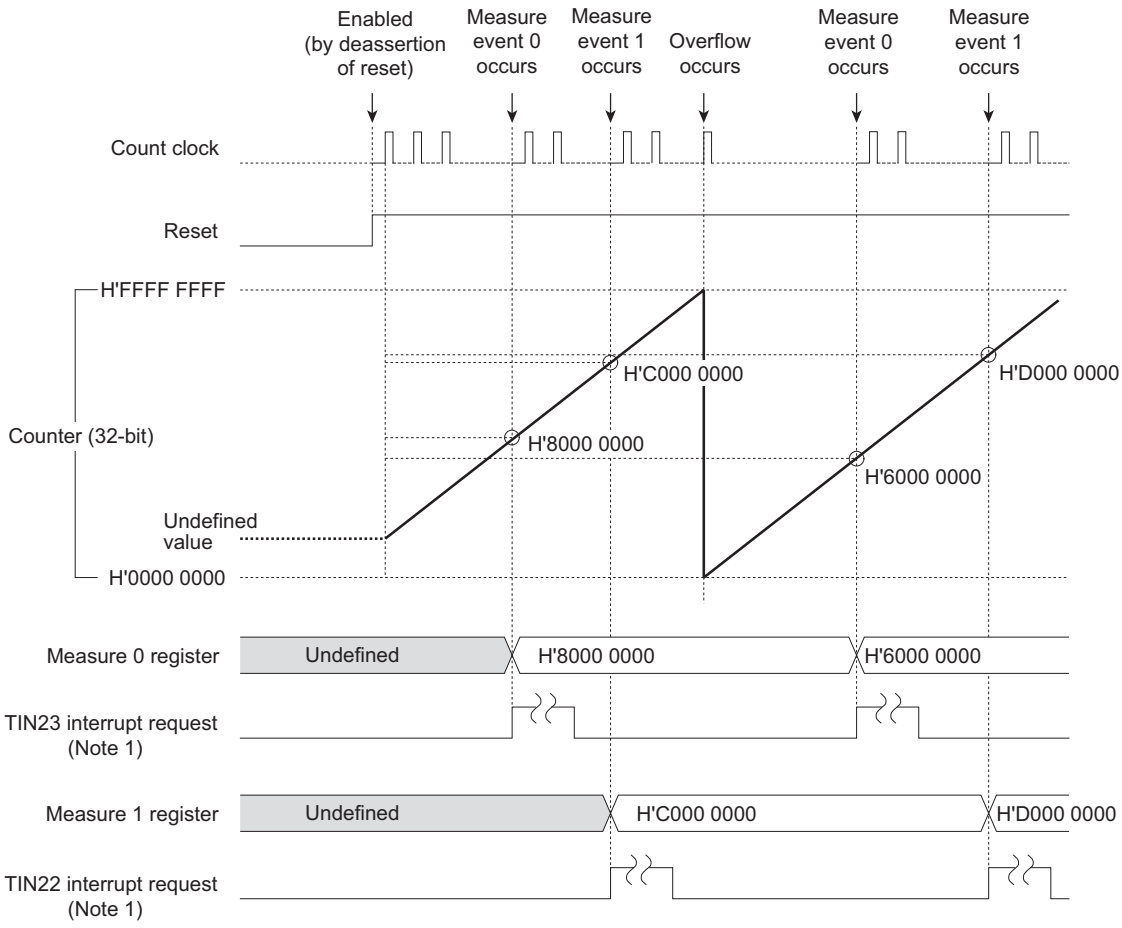
The TML1 measure register is used to latch the counter content upon event input. The TML1 measure register consists of 32 bits, which TML1MR3H–0H and TML1MR3L–0L are 16 high-order bits and 16 low-order bits, respectively. The TML1 measure registers can only be read, and cannot be written to. The register must always be accessed wordwise beginning with the word boundary.

10.6.7 Operation of TML Measure Input

(1) Outline of TML measure input

In TML measure input, the counter starts counting up when the reset input signal is deasserted. Upon event input to measure registers 0–3, the counter value is latched into each measure register.

A TIN interrupt request can be generated by measure signal input from an external device (TML0 alone has a TIN interrupt. TML1 does not have a TIN interrupt.) However, no TML counter overflow interrupts are available.



Note 1: TML0 alone has a TIN interrupt. (TML1 does not have a TIN interrupt.)
Note: • This diagram does not show detailed timing information.

Figure 10.6.2 Typical Operation of TML Measure Input

(2) Precautions on using TML measure input

The following describes precautions to be observed when using TML measure input.

- If measure event input and write to the counter occur in the same clock period, the write value is set in the counter, whereas the up-count value (before being rewritten) is latched into the measure register.
- If clock bus 1 is selected and any clock other than BCLK/2 is used for the timer, the counter cannot be written normally. Therefore, when using any clock other than BCLK/2, do not write to the counter.
- If clock bus 1 is selected and any clock other than BCLK/2 is used for the timer, the value captured into the measure register is one count larger the counter value. During the count clock to BCLK/2 period interval, however, the captured value is exactly the counter value.

The diagram below shows the relationship between counter operation and the valid data that can be captured.

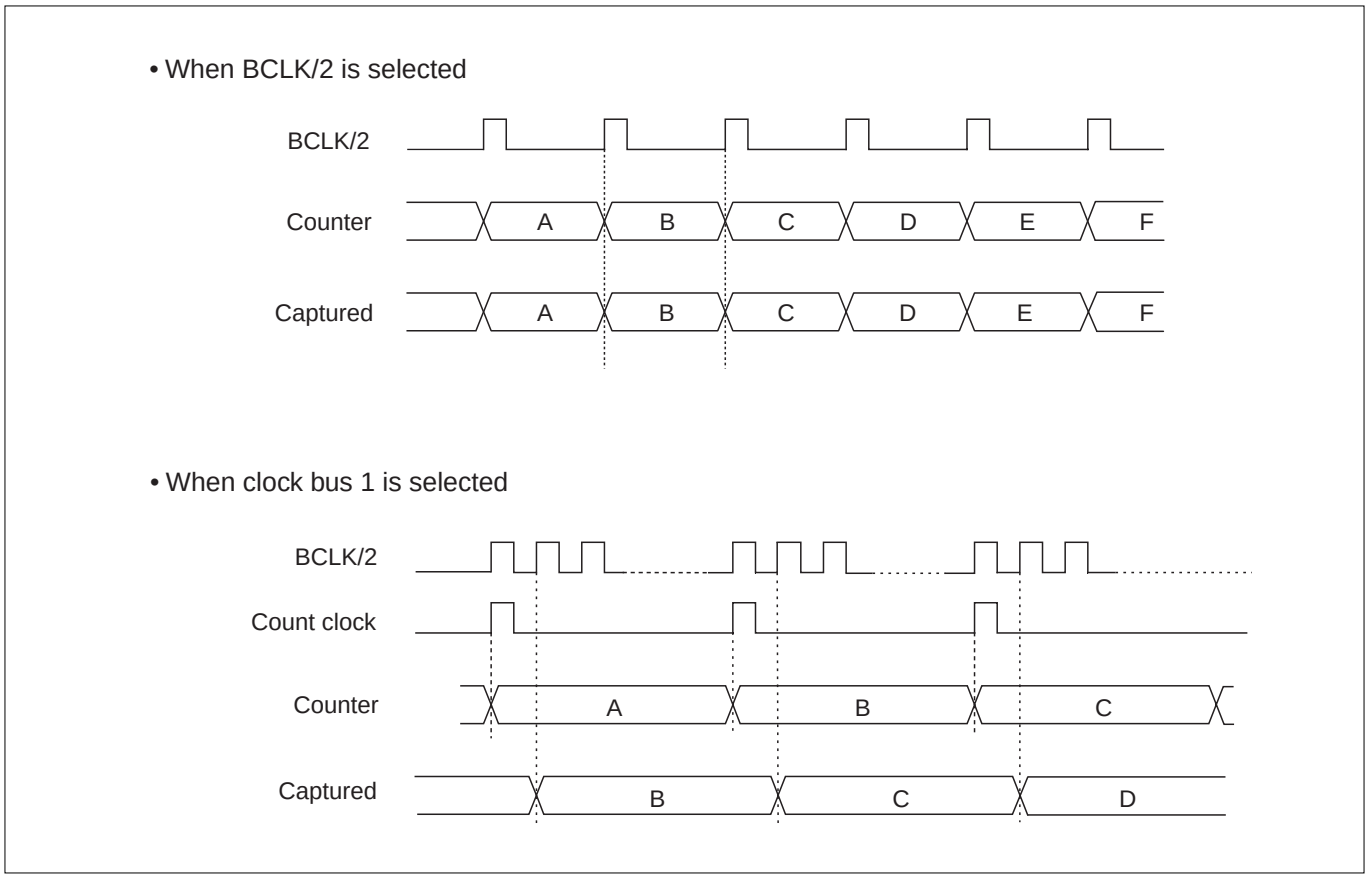


Figure 10.6.3 Mistimed Counter Value and the Captured Value

CHAPTER 11

A/D CONVERTER

11.1 Outline of A/D Converter

The 32176 contains 10-bit resolution A/D Converter of the successive approximation type. The A/D converter has 16 analog input pins (channels) AD0IN0–AD0IN15. In addition to performing conversion individually on each channel, the A/D Converter can perform conversion successively on all of N channels (N = 1–16) as a single group. The conversion result can be read out in either 10 or 8 bits.

There are following conversion and operation modes for the A/D conversion:

(1) Conversion Modes

- A/D conversion mode : Ordinary mode in which analog input voltages are converted into digital quantities.
- Comparator mode (Note 1): A mode in which analog input voltage is compared with a preset comparison voltage to find only the relative magnitude of two quantities. (Useful in only single operation mode)

(2) Operation Modes

- Single mode : Analog input voltage on one channel is A/D converted once or comparated (Note 1) with a given quantity.
- Scan mode : Analog input voltages on two or more selected channels (in N channel units, N = 1–16) are sequentially A/D converted.
  Single-shot scan mode : Scan operation is performed for one cycle.
  Continuous scan mode : Scan operation is repeatedly until stopped.

(3) Special Operation Modes

- Forcible single mode execution during scan mode : Conversion is forcibly executed in single mode (comparator mode) during scan operation.
- Scan mode start after single mode execution : Scan operation is started subsequently after executing conversion in single mode.
- Conversion restart : A/D conversion being executed in single or scan mode is restarted.

(4) Sample-and-Hold Function

The analog input voltage is sampled when starting A/D conversion, and A/D conversion is performed on the sampled voltage. This function can be enabled or disabled as necessary.

(5) A/D Disconnection Detection Assist Function

To suppress influences of the analog input voltage leakage from any preceding channel during scan mode operation, a function is incorporated that helps to fix the electric charge on the chopper amp capacitor to the given state (AVCC0 or AVSS0) before starting A/D conversion. This function provides a sure and reliable means of detecting a disconnection in the wiring patterns connecting to the analog input pins.

(6) Inflow Current Bypass Circuit

If an overvoltage or negative voltage is applied to any analog input channel which is currently inactive, a current flows into or out of the analog input channel currently being A/D converted via the internal circuit, causing the conversion accuracy to degrade. To solve this problem, the A/D Converter incorporates a circuit that bypasses such inflow current. This circuit is always enabled.

(7) Conversion Speed

The A/D conversion and comparate speed can be selected from a total of four speeds available: slow mode (normal or double speed) and fast mode (normal or double speed). The normal speed and double speed in slow mode are compatible with the 32170 group of Renesas microcomputers.

(8) Interrupt Request and DMA Transfer Request Generation Functions

An A/D conversion interrupt or DMA transfer request can be generated each time A/D conversion or comparate operation in single mode is completed, as well as when a single-shot scan operation or one cycle of continuous scan operation is completed.

Note 1: To discriminate between the comparison performed internally by the successive approximation-type A/D Converter and that performed in comparator mode using the same A/D Converter as a comparator, the comparison in comparator mode is referred to in this manual as “comparate.”

Table 11.1.1 outlines the A/D Converter and Figure 11.1.1 shows block diagram of A/D Converter.

Table 11.1.1 Outline of the A/D Converter

Item	Description					
Analog input	16 channels					
A/D conversion method	Successive approximation method					
Resolution	10 bits (Conversion result can be read out in either 8 or 10 bits)					
Absolute accuracy (Note 1) Conditions: Ta = 25°C, AVCC0 = 5.12 V, VREF0 = 5.12 V	When sample-and-hold disabled or normal sample-and-hold enabled	Slow mode	Normal speed	±2LSB		
			Double speed	±2LSB		
		Fast mode	Normal speed	±3LSB		
			Double speed	±3LSB		
	When fast sample-and-hold enabled	Slow mode	Normal speed	±3LSB		
			Double speed	±3LSB		
		Fast mode	Normal speed	±3LSB		
			Double speed	±8LSB		
Conversion mode	A/D conversion mode and comparator mode					
Operation mode	Single mode, single-shot scan mode and continuous scan mode					
Conversion start trigger	Software start	Started by setting the A/D conversion start bit to "1"				
	Hardware start	A/D0 Converter	MJT (input event bus 2), MJT (input event bus 3), MJT (output event bus 3) and MJT (TIN23S)			
Conversion speed BCLK: Internal peripheral clock	During single mode (• When sample-and-hold disabled • When normal sample-and-hold enabled)	Slow mode	Normal speed	299BCLK	14.95µs (Note 2)	
			Double speed	173BCLK	8.65µs	
		Fast mode	Normal speed	131BCLK	6.55µs	
			Double speed	89BCLK	4.45µs	
	During single mode (When fast sample-and-hold enabled)	Slow mode	Normal speed	191BCLK	9.55µs	
			Double speed	101BCLK	5.05µs	
		Fast mode	Normal speed	95BCLK	4.75µs	
			Double speed	53BCLK	2.65µs	
	During comparator mode	Slow mode	Normal speed	47BCLK	2.35µs	
			Double speed	29BCLK	1.45µs	
		Fast mode	Normal speed	23BCLK	1.15µs	
			Double speed	17BCLK	0.85µs	
Sample-and-hold function	Sample-and-hold function can be enabled or disabled as necessary.					
A/D disconnection detection assist function	Influences of the analog input voltage leakage from any preceding channel during scan mode operation are suppressed.					
Interrupt request generation function	Generated when A/D conversion (single mode operation, single-shot scan operation or one cycle of continuous operation) or comparate operation is completed					
DMA transfer request generation function	Generated when A/D conversion (single mode operation, single-shot scan operation or one cycle of continuous operation) or comparate operation is completed					

Note 1: The conversion accuracy stipulated here refers to that of the microcomputer alone, with influences of the power supply wiring and noise on the board not taken into account.

Note 2: This indicates the conversion time when f(BCLK) = 20 MHz (1 BCLK = 50 ns).

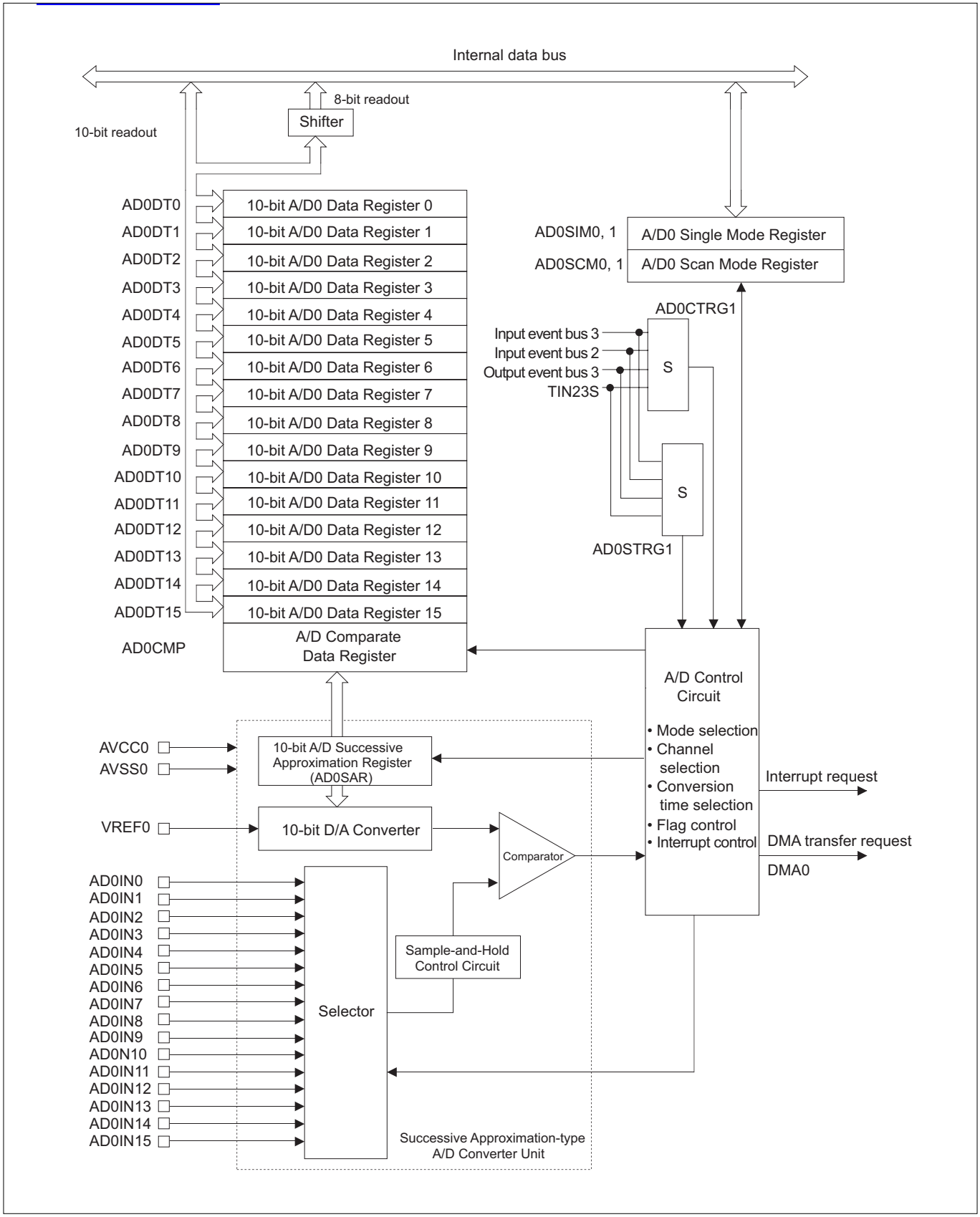


Figure 11.1.1 Block Diagram of the A/D0 Converter

11.1.1 Conversion Modes

The A/D Converter has two conversion modes: “A/D Conversion mode” and “Comparator mode.”

(1) A/D Conversion Mode

In A/D conversion mode, the analog input voltage on a specified channel is A/D converted.
In single mode, A/D conversion is performed on a channel selected by the A/D Single Mode Register 1 analog input pin select bit.
In scan mode, A/D conversion is performed on channels selected by A/D Scan Mode Register 1 according to settings of A/D Scan Mode Register 0.
The conversion result is stored in each channel’s corresponding 10-bit A/D Data Register. There is also an 8-bit A/D Data Register for each channel, from which 8-bit A/D conversion results can be read out.
An A/D conversion interrupt or DMA transfer request can be generated when A/D conversion in single mode is completed, as well as when one cycle of scan loop in scan mode is completed.

(2) Comparator Mode

In comparator mode, the analog input voltage on a specified channel is “comparated” (compared) with the successive approximation register value, and the result (relative magnitude of two values) is returned to a flag.
The channel to be comparated is selected using the A/D Single Mode Register 1 analog input pin select bit.
The result of comparate operation is flagged ("0" or "1") by setting the A/D Comparate Data Register bit that corresponds to the selected channel.
An A/D conversion interrupt or DMA transfer request can be generated when comparate operation is completed.

11.1.2 Operation Modes

There are two operation modes for the A/D Converter: “Single mode” and “Scan mode.” When comparator mode is selected as A/D conversion mode, only single mode can be used.

(1) Single Mode

In single mode, the analog input voltage on one selected channel is A/D converted or comparated once. An A/D conversion interrupt or DMA transfer request can be generated when A/D conversion or comparate operation is completed.

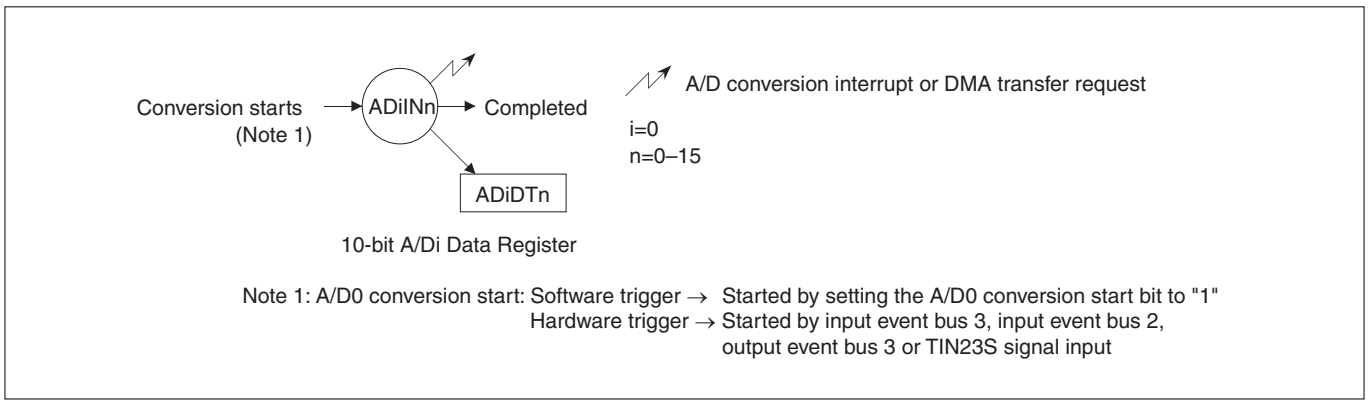


Figure 11.1.2 Operation in Single Mode (A/D Conversion)

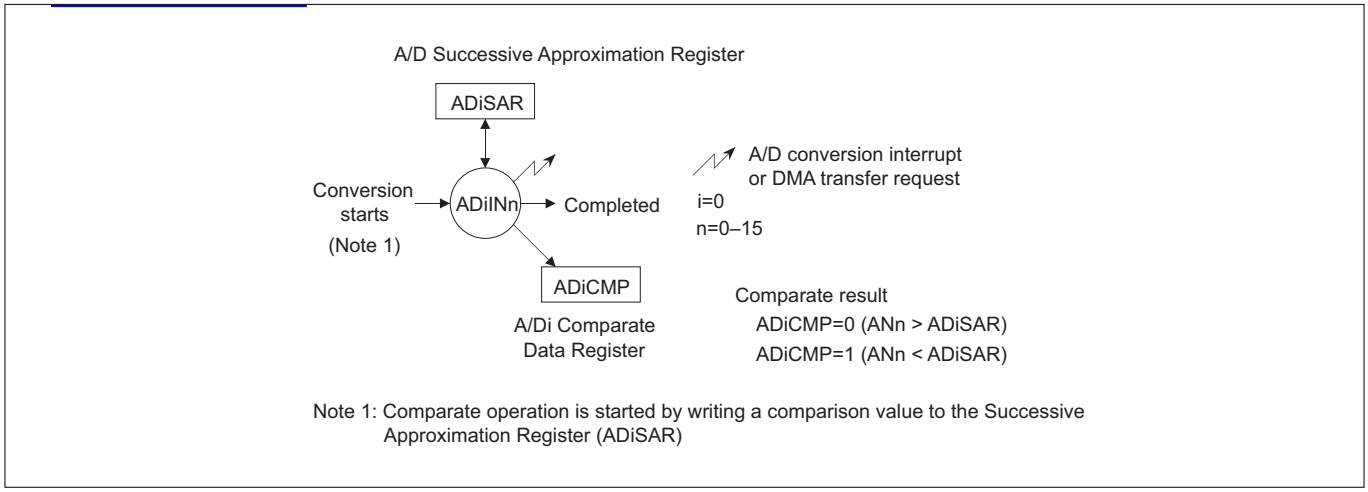


Figure 11.1.3 Operation in Single Mode (Comparate)

(2) Scan Mode

In scan mode, the analog input voltages from channel 0 to the channel selected by the A/D Scan Mode Register 1 scan loop select bit (channels 0–15) are sequentially A/D converted.
There are two types of scan mode: “Single-shot scan mode” in which A/D conversion is completed after performing one cycle of scan operation, and “Continuous scan mode” in which scan operation is continued until halted by setting the A/D scan mode register 0’s A/D conversion stop bit to "1".

These types of scan mode are selected using A/D Scan Mode Register 0. The channels to be scanned are selected using A/D Scan Mode Register 1. The selected channels are scanned sequentially beginning with channel 0.

An A/D conversion interrupt or DMA transfer request can be generated when one cycle of scan operation is completed.

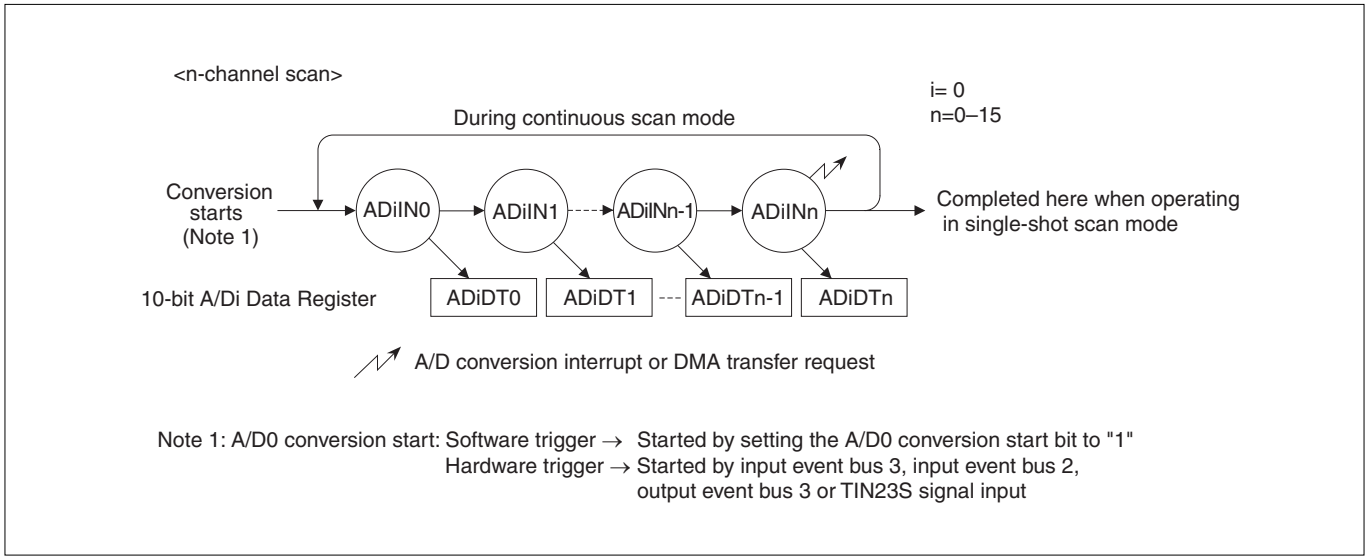


Figure 11.1.4 Operation of A/D Conversion in Scan Mode

Table 11.1.2 Registers in Which Scan Mode A/D Conversion Results Are Stored

Scan Mode Register 1 channel selection	Selected channels for single-shot scan	Selected channels for continuous scan	A/D conversion result storage register
B'0000:0 (ADiIN0)	ADiIN0	ADiIN0	10-bit A/Di Data Register 0
	Completed	ADiIN0	10-bit A/Di Data Register 0
		⋮ (Repeated until forcibly terminated)	⋮
B'0001:1 (ADiIN1)	ADiIN0	ADiIN0	10-bit A/Di Data Register 0
	ADiIN1	ADiIN1	10-bit A/Di Data Register 1
	Completed	ADiIN0	10-bit A/Di Data Register 0
		⋮ (Repeated until forcibly terminated)	⋮
B'0010:2 (ADiIN2)	ADiIN0	ADiIN0	10-bit A/Di Data Register 0
	ADiIN1	ADiIN1	10-bit A/Di Data Register 1
	ADiIN2	ADiIN2	10-bit A/Di Data Register 2
	Completed	ADiIN0	10-bit A/Di Data Register 0
		⋮ (Repeated until forcibly terminated)	⋮
B'0011:3 (ADiIN3)	ADiIN0	ADiIN0	10-bit A/Di Data Register 0
	ADiIN1	ADiIN1	10-bit A/Di Data Register 1
	ADiIN2	ADiIN2	10-bit A/Di Data Register 2
	ADiIN3	ADiIN3	10-bit A/Di Data Register 3
	Completed	ADiIN0	10-bit A/Di Data Register 0
		⋮ (Repeated until forcibly terminated)	⋮
B'XXXX:n (ADiINn)	ADiIN0	ADiIN0	10-bit A/Di Data Register 0
	ADiIN1	ADiIN1	10-bit A/Di Data Register 1
	ADiIN2	ADiIN2	10-bit A/Di Data Register 2
n≤15		⋮	⋮
	ADiINn	ADiINn	10-bit A/Di Data Register n
	Completed	ADiIN0	10-bit A/Di Data Register 0
		⋮ (Repeated until forcibly terminated)	⋮

(i=0)

11.1.3 Special Operation Modes

(1) Forcible single mode execution during scan mode

In this special operation mode, single mode conversion (A/D conversion or comparate) is forcibly executed on a specified channel during scan mode operation. For A/D conversion mode, the conversion result is stored in the 10-bit A/D Data Register corresponding to the specified channel, whereas for comparate mode, the conversion result is stored in the 10-bit A/D Comparate Data Register. When the A/D conversion or comparate operation on a specified channel finishes, scan mode A/D conversion is restarted from where it was canceled during scan operation.

To start single mode conversion during scan mode operation in software, choose a software trigger using the A/D Single Mode Register 0 A/D conversion start trigger select bit. Then, for A/D conversion, set the said register's A/D conversion start bit to "1". For comparate mode, write a comparison value to the A/D Successive Approximation Register (AD0SAR) during scan mode operation.

To start single mode conversion during scan mode operation in hardware, choose a hardware trigger using the A/D Single Mode Register 0 A/D conversion start trigger select bit. Then enter the hardware trigger selected with the said register.

An A/D conversion interrupt or DMA transfer request can be generated when conversion on a specified channel or one cycle of scan operation is completed.

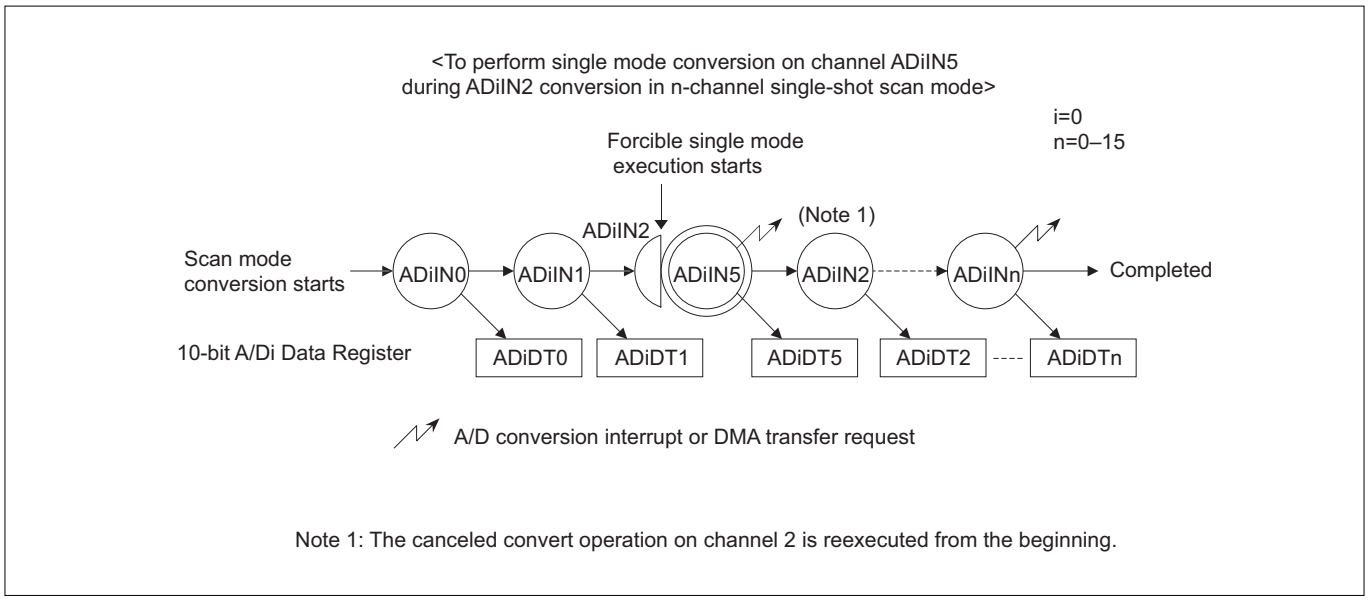


Figure 11.1.5 Forcible Single Mode Execution during Scan Mode

(2) Scan mode start after single mode execution

In this special operation mode, scan operation is started subsequently after executing single mode conversion (A/D conversion or comparate).

To start this mode in software, choose a software trigger using the A/D Scan Mode Register 0 A/D conversion start trigger select bit. Then set the said register's A/D conversion start bit to "1" during single mode conversion operation.

To start this mode in hardware, choose a hardware trigger using the A/D Scan Mode Register 0 A/D conversion start trigger select bit. Then enter the hardware trigger selected with the said register during single mode conversion operation.

If a hardware trigger is selected using the A/D conversion start trigger select bit in both A/D Single Mode Register 0 and A/D Scan Mode Register 0 and the selected hardware triggers are entered, the A/D Converter first performs single mode conversion and then scan mode conversion in succession.

An A/D conversion interrupt or DMA transfer request can be generated when single mode conversion on a specified channel or one cycle of scan operation is completed.

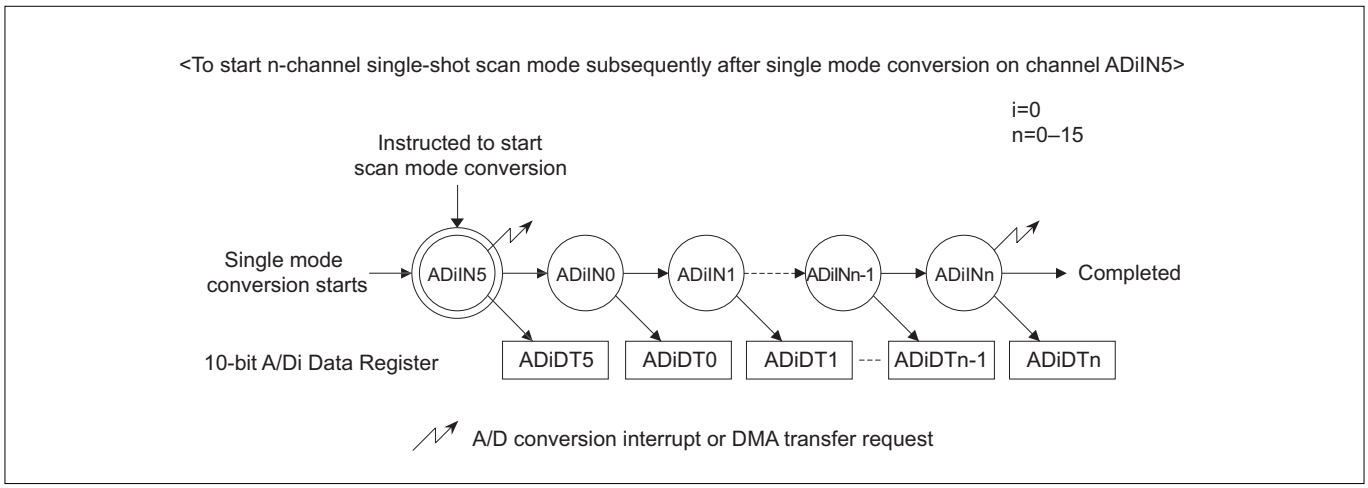


Figure 11.1.6 Scan Mode Start after Single Mode Execution

(3) Conversion restart

In this special operation mode, operation being executed in single or scan mode is stopped in the middle and reexecuted from the beginning.

When in single mode, set the A/D Single Mode Register 0 A/D conversion start bit to "1" again or enter a hardware trigger during A/D conversion or comparate operation, and the operation being executed is re-started over again.

When in scan mode, set the A/D Scan Mode Register 0 A/D conversion start bit to "1" again or enter a hardware trigger signal during scan operation, and the channel being converted is canceled and A/D conversion is performed from channel 0 over again.

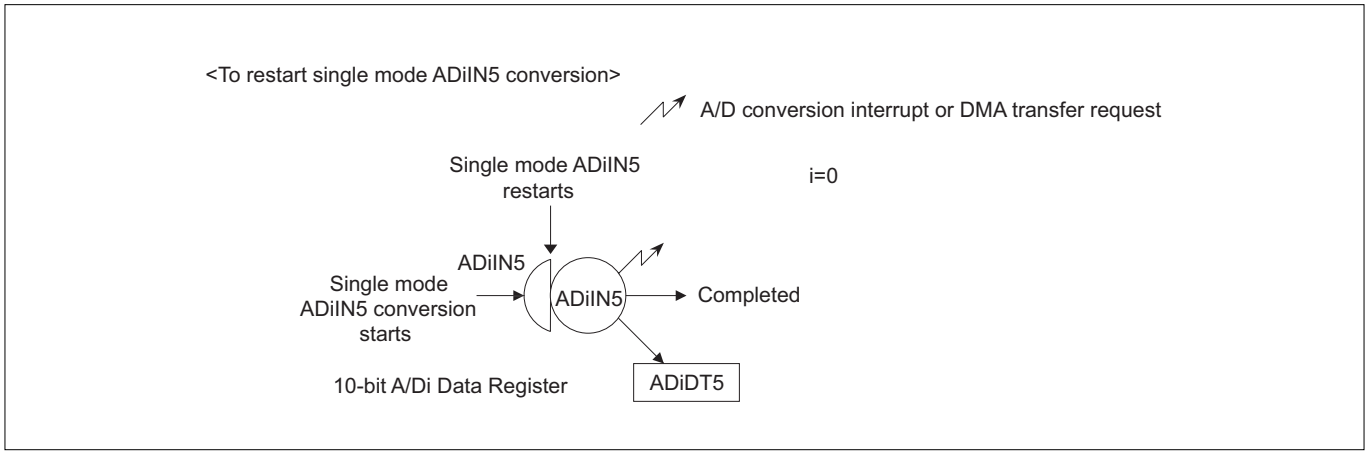


Figure 11.1.7 Conversion Restart during Single Mode Operation

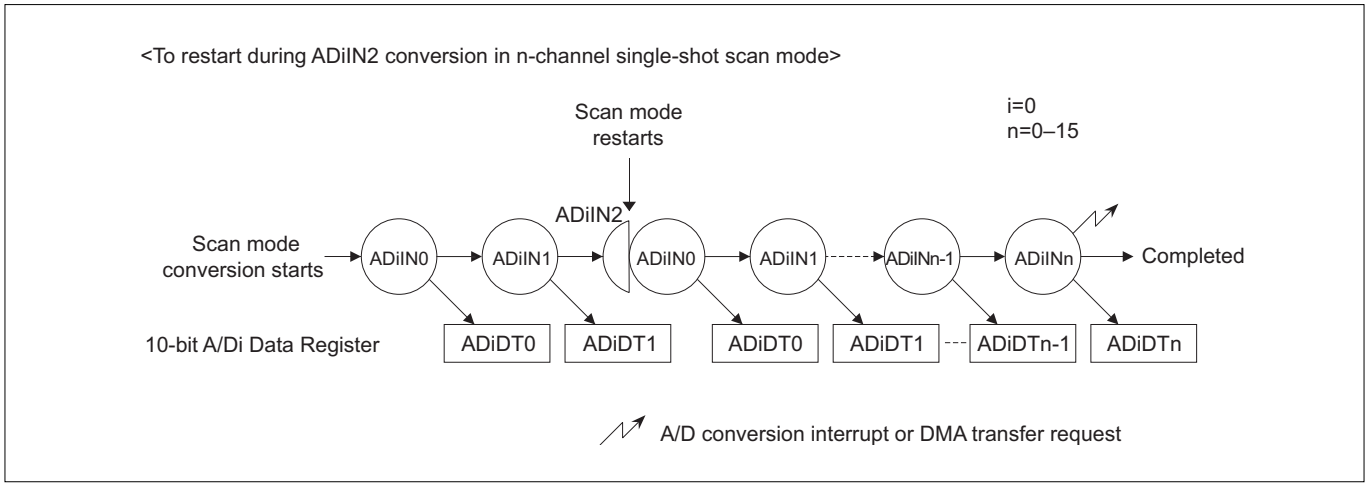


Figure 11.1.8 Conversion Restart during Scan Operation

11.1.4 A/D Converter Interrupt and DMA Transfer Requests

The A/D Converter can generate an A/D conversion interrupt or DMA transfer request each time A/D conversion, comparate operation, single-shot scan or one cycle of continuous scan mode is completed. The A/D Single Mode Register 0 and A/D Scan Mode Register 0 are used to select between A/D conversion interrupt and DMA transfer requests.

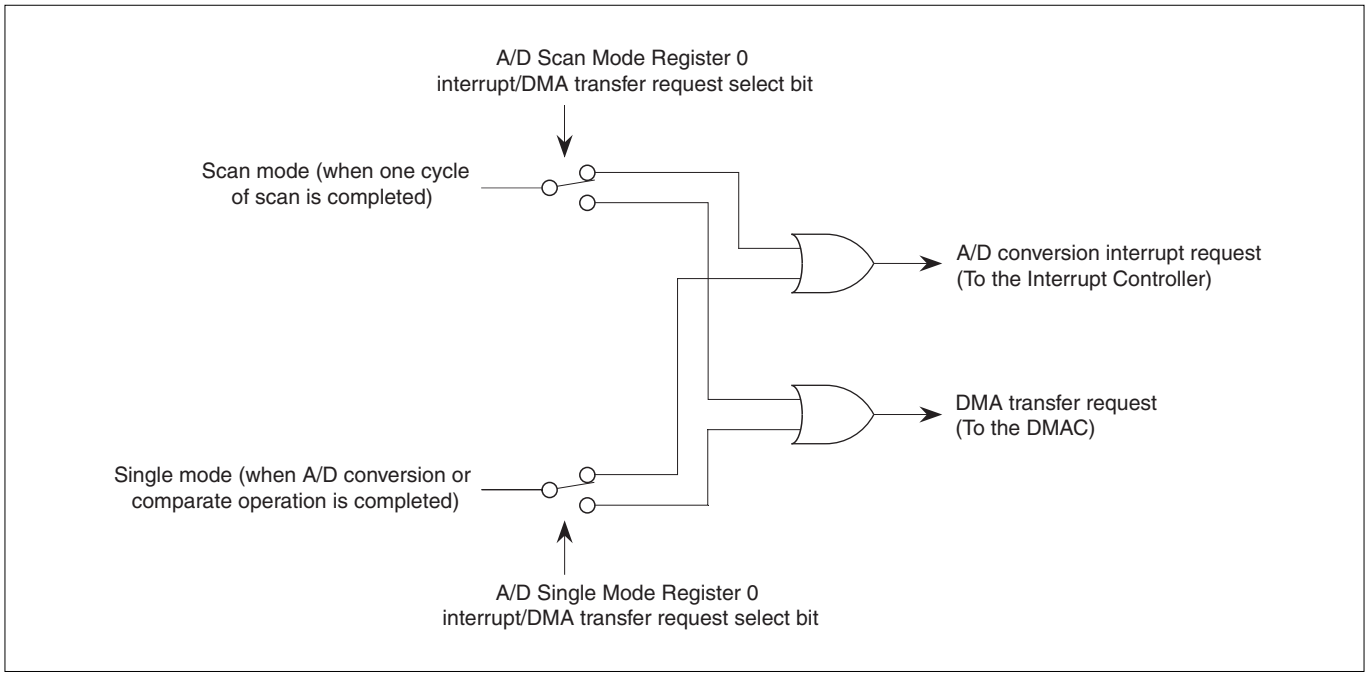


Figure 11.1.9 Selecting between Interrupt and DMA Transfer Requests

11.1.5 Sample-and-Hold Function

The analog input voltage that was sampled immediately after A/D conversion started is held on, and A/D conversion is performed on that seized voltage.

The A/D conversion time in “normal” sample-and-hold mode is the same as in conventional A/D conversion mode of the 32170, etc. The A/D conversion time in “fast” sample-and-hold mode is significantly short, allowing to obtain conversion results more quickly than ever.

11.2 A/D Converter Related Registers

Shown below is an A/D converter related register map.

A/D Converter Related Register Map (1/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0080	A/D0 Single Mode Register 0 (AD0SIM0)	A/D0 Single Mode Register 1 (AD0SIM1)	11-14 11-16
H'0080 0082	(Use inhibited area)		
H'0080 0084	A/D0 Scan Mode Register 0 (AD0SCM0)	A/D0 Scan Mode Register 1 (AD0SCM1)	11-18 11-20
H'0080 0086	A/D0 Disconnection Detection Assist Function Control Register (AD0DDACR)	A/D0 Conversion Speed Control Register (AD0CVSCR)	11-23 11-22
H'0080 0088	A/D0 Successive Approximation Register (AD0SAR)		11-27
H'0080 008A	A/D0 Disconnection Detection Assist Method Select Register (AD0DDASEL)		11-24
H'0080 008C	A/D0 Comparate Data Register (AD0CMP)		11-28
H'0080 008E	(Use inhibited area)		
H'0080 0090	10-bit A/D0 Data Register 0 (AD0DT0)		11-29
H'0080 0092	10-bit A/D0 Data Register 1 (AD0DT1)		11-29
H'0080 0094	10-bit A/D0 Data Register 2 (AD0DT2)		11-29
H'0080 0096	10-bit A/D0 Data Register 3 (AD0DT3)		11-29
H'0080 0098	10-bit A/D0 Data Register 4 (AD0DT4)		11-29
H'0080 009A	10-bit A/D0 Data Register 5 (AD0DT5)		11-29
H'0080 009C	10-bit A/D0 Data Register 6 (AD0DT6)		11-29
H'0080 009E	10-bit A/D0 Data Register 7 (AD0DT7)		11-29
H'0080 00A0	10-bit A/D0 Data Register 8 (AD0DT8)		11-29
H'0080 00A2	10-bit A/D0 Data Register 9 (AD0DT9)		11-29
H'0080 00A4	10-bit A/D0 Data Register 10 (AD0DT10)		11-29
H'0080 00A6	10-bit A/D0 Data Register 11 (AD0DT11)		11-29
H'0080 00A8	10-bit A/D0 Data Register 12 (AD0DT12)		11-29
H'0080 00AA	10-bit A/D0 Data Register 13 (AD0DT13)		11-29
H'0080 00AC	10-bit A/D0 Data Register 14 (AD0DT14)		11-29
H'0080 00AE	10-bit A/D0 Data Register 15 (AD0DT15)		11-29
H'0080 00D0	(Use inhibited area)	8-bit A/D0 Data Register 0 (AD08DT0)	11-30
H'0080 00D2	(Use inhibited area)	8-bit A/D0 Data Register 1 (AD08DT1)	11-30
H'0080 00D4	(Use inhibited area)	8-bit A/D0 Data Register 2 (AD08DT2)	11-30
H'0080 00D6	(Use inhibited area)	8-bit A/D0 Data Register 3 (AD08DT3)	11-30
H'0080 00D8	(Use inhibited area)	8-bit A/D0 Data Register 4 (AD08DT4)	11-30
H'0080 00DA	(Use inhibited area)	8-bit A/D0 Data Register 5 (AD08DT5)	11-30
H'0080 00DC	(Use inhibited area)	8-bit A/D0 Data Register 6 (AD08DT6)	11-30

A/D Converter Related Register Map (2/2)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 00DE	(Use inhibited area)	8-bit A/D0 Data Register 7 (AD08DT7)	11-30
H'0080 00E0	(Use inhibited area)	8-bit A/D0 Data Register 8 (AD08DT8)	11-30
H'0080 00E2	(Use inhibited area)	8-bit A/D0 Data Register 9 (AD08DT9)	11-30
H'0080 00E4	(Use inhibited area)	8-bit A/D0 Data Register 10 (AD08DT10)	11-30
H'0080 00E6	(Use inhibited area)	8-bit A/D0 Data Register 11 (AD08DT11)	11-30
H'0080 00E8	(Use inhibited area)	8-bit A/D0 Data Register 12 (AD08DT12)	11-30
H'0080 00EA	(Use inhibited area)	8-bit A/D0 Data Register 13 (AD08DT13)	11-30
H'0080 00EC	(Use inhibited area)	8-bit A/D0 Data Register 14 (AD08DT14)	11-30
H'0080 00EE	(Use inhibited area)	8-bit A/D0 Data Register 15 (AD08DT15)	11-30

11.2.1 A/D Single Mode Register 0

A/D0 Single Mode Register 0 (AD0SIM0) <Address: H'0080 0080>

b0	1	2	3	4	5	6	b7
ADSTRG1		ADSTRG0	ADSSEL	ADSREQ	ADSCMP	ADSSTP	ADSSTT
0	0	0	0	0	1	0	0

<Upon exiting reset: H'04>

b	Bit Name	Function	R	W
0	ADSTRG1 (Note 1) A/D hardware trigger select 1 bit	Bits 0 and 2 are used to select an A/D hardware trigger. b0 b2 0 0 : Input event bus 2 0 1 : Input event bus 3 1 0 : Output event bus 3 1 1 : TIN23S signal	R	W
1	No function assigned. Fix to "0".		0	0
2	ADSTRG0 (Note 1) A/D hardware trigger select 0 bit	Bits 0 and 2 are used to select an A/D hardware trigger. (See the column for bit 0.)	R	W
3	ADSSEL A/D conversion start trigger select bit	0: Software trigger 1: Hardware trigger (Note 2)	R	W
4	ADSREQ A/D Interrupt/DMA transfer request select bit	0: A/D conversion interrupt request 1: DMA transfer request	R	W
5	ADSCMP A/D conversion/comparate completed bit	0: A/D conversion/comparate in progress 1: A/D conversion/comparate completed	R	–
6	ADSSTP A/D conversion stop bit	0: No operation 1: Stop A/D conversion	0	W
7	ADSSTT A/D conversion start bit	0: No operation 1:Start A/D conversion	0	W

Note 1: Two bits—bit 0 (A/D hardware trigger select 1) and bit 2 (A/D hardware trigger select 0)—are used to select an A/D hardware trigger.

Note 2: During comparator mode, hardware triggers, if any selected, are ignored and operation is started by a software trigger.

A/D Single Mode Register 0 is used to control operation of the A/D Converter during single mode (including "Forcible single mode execution during scan mode").

(1) ADSTRG (A/D Hardware Trigger Select) bits (Bits 0 and 2)

These bits select a hardware trigger when A/D conversion by the A/D Converter is to be started in hardware. Select one from the following hardware trigger sources:

A/D0 Converter: Input event bus 2
Input event bus 3
Output event bus 3
TIN23 edge select output

The contents of these bits are ignored if a software trigger is selected by ADSSEL (A/D conversion start trigger select bit).

(2) ADSSEL (A/D Conversion Start Trigger Select) bit (Bit 3)

This bit selects whether to use a software or hardware trigger to start A/D conversion during single mode. If a software trigger is selected, A/D conversion is started by setting the ADSSTT (A/D conversion start) bit to "1". If a hardware trigger is selected, A/D conversion is started by the trigger source selected with the ADSTRG (hardware trigger select) bits.

(3) ADSREQ (A/D Interrupt Request/DMA Transfer Request Select) bit (Bit 4)

This bit selects whether to request an A/D conversion interrupt or a DMA transfer when single mode operation (A/D conversion or comparate) is completed. If neither an interrupt nor a DMA transfer are used, choose to request an A/D conversion interrupt and use the A/D Conversion Interrupt Control Register of the Interrupt Controller (ICU) to mask the interrupt request, or choose to request a DMA transfer and use the DMA Channel Control Register to disable DMA transfers to be performed upon completion of A/D conversion.

(4) ADSCMP (A/D Conversion/Comparate Completed) bit (Bit 5)

This is a read-only bit, whose value when exiting the reset state is "1". This bit is "0" when the A/D Converter is performing single mode operation (A/D conversion or comparate) and is set to "1" when the operation finishes.
This bit is also set to "1" when A/D conversion or comparate operation is forcibly terminated by setting the ADSSTP (A/D conversion stop) bit to "1" during A/D conversion or comparate operation.

(5) ADSSTP (A/D Conversion Stop) bit (Bit 6)

Setting this bit to "1" while the A/D Converter is performing single mode operation (A/D conversion or comparate) causes the operation being performed to stop. Manipulation of this bit is ignored while single mode operation is idle or scan mode operation is under way.
Operation stops immediately after writing to this bit. If the A/D Successive Approximation Register is read after being stopped, the content read from the register is the value in the middle of conversion (not transferred to the A/D Data Register).
If the A/D conversion start bit and A/D conversion stop bit are set to "1" at the same time, the A/D conversion stop bit has priority.
If this bit is set to "1" when performing single mode operation in special mode “Forcible single mode execution during scan mode,” only single mode conversion stops and scan mode operation restarts.

(6) ADSSTT (A/D Conversion Start) bit (Bit 7)

If this bit is set to "1" when a software trigger has been selected with the ADSSEL (A/D conversion start trigger select) bit, the A/D Converter starts A/D conversion.
If the A/D conversion start bit and A/D conversion stop bit are set to "1" at the same time, the A/D conversion stop bit has priority.
If this bit is set to "1" again while performing single mode conversion, special operation mode “Conversion restart” is turned on, so that single mode conversion restarts.
If this bit is set to "1" again while performing A/D conversion in scan mode, special operation mode “Forcible single mode execution during scan mode” is turned on, so that the channel being converted in scan mode is canceled and single mode conversion is performed. When the single mode conversion finishes, scan mode A/D conversion restarts beginning with the canceled channel.

11.2.2 A/D Single Mode Register 1

A/D0 Single Mode Register 1 (AD0SIM1) <Address: H'0080 0081>

b8	9	10	11	12	13	14	b15
ADSMSL	ADSSPD	ADSSHSL	ADSSHSPD	ANSEL			
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function		R	W
8	ADSMSL A/D conversion mode select bit	0: A/D0 conversion mode 1: Comparator mode		R	W
9	ADSSPD (Note 1) A/D conversion speed select bit	0: Normal speed 1: Double speed		R	W
10	ADSSHSL A/D conversion method select bit	0: Disable sample-and-hold 1: Enable sample-and-hold		R	W
11	ADSSHSPD (Note 2) A/D sample-and-hold conversion speed select bit	0: Normal sample-and-hold 1: Fast sample-and-hold		R	W
12–15	ANSEL A/D analog input pin select bit	0000 : Select ADiIN0 0001 : Select ADiIN1 0010 : Select ADiIN2 0011 : Select ADiIN3 0100 : Select ADiIN4 0101 : Select ADiIN5 0110 : Select ADiIN6 0111 : Select ADiIN7 1000 : Select ADiIN8 1001 : Select ADiIN9 1010 : Select ADiIN10 1011 : Select ADiIN11 1100 : Select ADiIN12 1101 : Select ADiIN13 1110 : Select ADiIN14 1111 : Select ADiIN15	(i = 0)	R	W

Note 1: The A/D conversion speed is determined by a combination of ADSSPD, ADSSHSL and ADSSHSPD bits and the A/D Conversion Speed Control Register ADCVSD bit.

Note 2: Setting of this bit is effective when the sample-and-hold function is enabled by ADSSHSL bit.

A/D Single Mode Register 1 is used to select operation mode, conversion speed and analog input pins when the A/D Converter is operating in single mode.

(1) ADSMSL (A/D Conversion Mode Select) bit (Bit 8)

This bit selects A/D conversion mode when the A/D Converter is operating in single mode. Setting this bit to "0" selects A/D conversion mode, and setting this bit to "1" selects comparator mode.

(2) ADSSPD (A/D Conversion Speed Select) bit (Bit 9)

This bit selects the A/D conversion speed when the A/D Converter is operating in single mode. Setting this bit to "0" selects normal speed, and setting this bit to "1" selects double speed.

(3) ADSSHSL (A/D Conversion Method Select) bit (Bit 10)

This bit enables or disables the sample-and-hold function when the A/D Converter is operating in single mode. Setting this bit to "0" disables the sample-and-hold function, and setting this bit to "1" enables the sample-and-hold function.
Setting of this bit has no effect if comparator mode is selected with the ADSMSL (A/D conversion mode select) bit.

(4) ADSSHSPD (A/D Sample-and-Hold Speed Select) bit (Bit 11)

When the A/D Converter's sample-and-hold function is enabled, this bit selects a conversion speed. When this bit is "0", the conversion speed is the same as normal A/D conversion speed. When this bit is "1", conversion is performed at a speed faster than normal A/D conversion speed.
Setting of this bit has no effect if the sample-and-hold function is disabled by setting the ADSSHSL (A/D conversion method select) bit to "0".
For details about the conversion time, see Section 11.3.4, "Calculating the A/D Conversion Time."

(5) ANSEL (A/D Analog Input Pin Select) bits (Bits 12–15)

These bits select the analog input pins when the A/D Converter is operating in single mode. A/D conversion or comparate operation is performed on the channels selected with these bits. If these bits are accessed for read, the value written to them is read out.

Notes: • If either A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIM1) or A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as sample-and-hold enabled, both single and scan mode operate under sample-and-hold enabled mode.

• If either A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIMI1) or A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as sample-and-hold enabled, and A/D sample-and-hold conversion speed select (ADSSHSPD) bit in A/D0 Single Mode Register 1 (AD0SM1) or A/D sample-and-hold conversion speed select (ADCSHSPD) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as fast sample-and-hold, both single and scan mode operate in fast sample-and-hold conversion spped.

• To use single or scan mode under sample-and-hold enabled mode, make sure that A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIM1), A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1), A/D sample-and-hold conversion spped select (ADSSHSPD)bit in A/D0 Single Mode Register 1 (AD0SIM1) and A/D sample-and-hold conversion speed select (ADCSHSPD) bit in A/D0 Scan Mode Register 1 (AD0SCM1) are set in the same value.

11.2.3 A/D Scan Mode Register 0

A/D0 Scan Mode Register 0 (AD0SCM0) <Address: H'0080 0084>

b0	1	2	3	4	5	6	b7
ADCTRG1	ADCMSL	ADCTRG0	ADCSEL	ADCREQ	ADCCMP	ADCSTP	ADCSTT
0	0	0	0	0	1	0	0

<Upon exiting reset: H'04>

b	Bit Name	Function	R	W
0	ADCTRG1 (Note 1) A/D hardware trigger select 1 bit	Bits 0 and 2 are used to select an A/D hardware trigger. b0 b2 0 0 : Input event bus 2 0 1 : Input event bus 3 1 0 : Output event bus 3 1 1 : TIN23S signal	R	W
1	ADCMSL A/D scan mode select bit	0: Single-shot mode 1: Continuous mode	R	W
2	ADCTRG0 A/D hardware trigger select 0 bit	Bits 0 and 2 are used to select an A/D hardware trigger. (See the column for bit 0.)	R	W
3	ADCSEL A/D conversion start trigger select bit	0: Software trigger 1: Hardware trigger	R	W
4	ADCREQ A/D Interrupt/DMA transfer request select bit	0: A/D conversion interrupt request 1: DMA transfer request	R	W
5	ADCCMP A/D conversion completed bit	0: A/D conversion in progress 1: A/D conversion completed	R	–
6	ADCSTP A/D conversion stop bit	0: No operation 1: Stop A/D conversion	0	W
7	ADCSTT A/D conversion start bit	0: No operation 1: Start A/D conversion	0	W

Note 1: Two bits—bit 0 (A/D hardware trigger select 1) and bit 2 (A/D hardware trigger select 0)—are used to select an A/D hardware trigger.

A/D Scan Mode Register 0 is used to control operation of the A/D Converter during scan mode.

(1) ADCTRG (A/D Hardware Trigger Select) bits (Bits 0 and 2)

These bits select a hardware trigger when A/D conversion by the A/D Converter is to be started in hardware. Select one from the following hardware trigger sources:

A/D0 Converter: Input event bus 2
Input event bus 3
Output event bus 3
TIN23 edge select output

The contents of these bits are ignored if a software trigger is selected by ADCSEL (A/D conversion start trigger select bit).

(2) ADCMSL (A/D Scan Mode Select) bit (Bit 1)

This bit selects scan mode of the A/D Converter between single-shot scan and continuous scan.
Setting this bit to "0" selects single-shot scan mode, where the channels selected with the ANSCAN (A/D scan loop select) bits are sequentially A/D converted and when A/D conversion on all selected channels is completed, the conversion operation stops.
Setting this bit to "1" selects continuous scan mode, where after operation in single-shot scan mode finishes, A/D conversion is reexecuted beginning with the first channel and continued until stopped by setting the ADCSTP (A/D conversion stop) bit to "1".

(3) ADCSEL (A/D Conversion Start Trigger Select) bit (Bit 3)

This bit selects whether to use a software or hardware trigger to start A/D conversion during scan mode. If a software trigger is selected, A/D conversion is started by setting the ADCSTT (A/D conversion start) bit to "1". If a hardware trigger is selected, A/D conversion is started by the trigger source selected with the ADCTRG (hardware trigger select) bits.

(4) ADCREQ (A/D Interrupt Request/DMA Transfer Request Select) bit (Bit 4)

This bit selects whether to request an A/D conversion interrupt or a DMA transfer when one cycle of scan mode operation is completed. If neither an interrupt nor a DMA transfer are used, choose to request an A/D conversion interrupt and use the A/D Conversion Interrupt Control Register of the Interrupt Controller (ICU) to mask the interrupt request, or choose to request a DMA transfer and use the DMA Channel Control Register to disable DMA transfers to be performed upon completion of A/D conversion.

(5) ADCCMP (A/D Conversion Completed) bit (Bit 5)

This is a read-only bit, whose value when exiting the reset state is "1". This bit is "0" when the A/D Converter is performing scan mode A/D conversion and is set to "1" when single-shot scan mode finishes or continuous scan mode is stopped by setting the ADCSTP (A/D conversion stop) bit to "1".

(6) ADCSTP (A/D Conversion Stop) bit (Bit 6)

Setting this bit to "1" while the A/D Converter is performing scan mode A/D conversion causes the operation being performed to stop. This bit is effective only for scan mode operation, and does not affect single mode operation even when single and scan modes both are active during special operation mode.
Operation stops immediately after writing to this bit, and the A/D conversion being performed on any channel is aborted in the middle, without transferring the result to the A/D data register.
If the A/D conversion start bit and A/D conversion stop bit are set to "1" at the same time, the A/D conversion stop bit has priority.

(7) ADCSTT (A/D Conversion Start) bit (Bit 7)

This bit is used to start scan mode operation of the A/D Converter in software. Only when a software trigger has been selected with the ADCSEL (A/D conversion start trigger select) bit, setting this bit to "1" causes A/D conversion to start.
If the A/D conversion start bit and A/D conversion stop bit are set to "1" at the same time, the A/D conversion stop bit has priority.
If this bit is set to "1" again while performing scan mode conversion, special operation mode “Conversion restart” is turned on, so that scan mode operation is restarted using the contents set by A/D Scan Mode Registers 0 and 1.
If this bit is set to "1" again while performing A/D conversion in single mode, special operation mode “Scan mode start after single mode execution” is turned on, so that scan mode operation starts subsequently after single mode has finished.

11.2.4 A/D Scan Mode Register 1

A/D0 Scan Mode Register 1 (AD0SCM1) <Address: H'0080 0085>

b8	9	10	11	12	13	14	b15
0	ADCSPD 0	ADCSHSL 0	ADCSHSPD 0	ANSCAN 0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	No function assigned. Fix to "0".		0	0
9	ADCSPD (Note 1) A/D conversion speed select bit	0: Normal speed 1: Double speed	R	W
10	ADCSHSL A/D conversion method select bit	0: Disable sample-and-hold 1: Enable sample-and-hold	R	W
11	ADCSHSPD (Note 2) A/D sample-and-hold conversion speed select bit	0: Normal sample-and-hold 1: Fast sample-and-hold	R	W
12–15	ANSCAN A/D scan loop select bit	<For write> 'B0000–1111 (channels 0–15) <For read during conversion> (i = 0) 0000: Converting ADiIN0 0001: Converting ADiIN1 0010: Converting ADiIN2 0011: Converting ADiIN3 0100: Converting ADiIN4 0101: Converting ADiIN5 0110: Converting ADiIN6 0111: Converting ADiIN7 1000: Converting ADiIN8 1001: Converting ADiIN9 1010: Converting ADiIN10 1011: Converting ADiIN11 1100: Converting ADiIN12 1101: Converting ADiIN13 1110: Converting ADiIN14 1111: Converting ADiIN15	R	W

Note 1: The A/D conversion speed is determined by a combination of ADCSPD, ADCSHSL and ADCSHSPD bits and the A/D Conversion Speed Control Register ADCVSD bit.

Note 2: Setting of this bit is effective when the sample-and-hold function is enabled by ADCSHSL bit.

A/D Scan Mode Register 1 is used to select operation mode, conversion speed and scan loop when the A/D Converter is operating in scan mode. The channels selected with the scan loop select bit are scanned sequentially beginning with channel 0 (n-channel scan).

(1) ADCSPD (A/D Conversion Speed Select) bit (Bit 9)

This bit selects an A/D conversion speed when the A/D Converter is operating in scan mode. Setting this bit to "0" selects normal speed, and setting this bit to "1" selects double speed.

(2) ADCSHSL (A/D Conversion Method Select) bit (Bit 10)

This bit enables or disables the sample-and-hold function when the A/D Converter is operating in scan mode. Setting this bit to "0" disables the sample-and-hold function, and setting this bit to "1" enables the sample-and-hold function.

(3) ADCSHSPD (A/D Sample-and-Hold Conversion Speed Select) bit (Bit 11)

When the A/D Converter's sample-and-hold function is enabled, this bit selects a conversion speed. When this bit is "0", the conversion speed is the same as normal A/D conversion speed. When this bit is "1", conversion is performed at a speed faster than normal A/D conversion speed.
Setting of this bit has no effect if the sample-and-hold function is disabled by setting the ADCSHSL (A/D conversion method select) bit to "0".
For details about the conversion time, see Section 11.3.4, "Calculating the A/D Conversion Time."

(4) ANSCAN (A/D Scan Loop Select) bits (Bits 12–15)

The ANSCAN (A/D scan loop select) bits set the channels to be scanned during scan mode of the A/D Converter.
The ANSCAN (A/D scan loop select) bits when accessed for read during scan operation serve as a status register indicating the channel being scanned.
The value read from these bits during single mode is always B'0000.
When accessed for read after scan operation in single-shot mode is completed, the value read from these bits indicates the channel whose A/D conversion has been finished last.
If A/D conversion is stopped by setting A/D Scan Mode Register 0 ADCSTP (A/D conversion stop) bit to "1" while executing scan mode, the value read from these bits indicates the channel whose A/D conversion has been canceled.
Also, if read during single mode conversion of special operation mode "Forcible single mode execution during scan mode," the value of these bits indicates the channel whose A/D conversion has been canceled in the middle of scan.

Notes: • If either A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIM1) or A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as sample-and-hold enabled, both single and scan mode operate under sample-and-hold enabled mode.

• If either A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIMI1) or A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as sample-and-hold enabled, and A/D sample-and-hold conversion speed select (ADSSHSPD) bit in A/D0 Single Mode Register 1 (AD0SM1) or A/D sample-and-hold conversion speed select (ADCSHSPD) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as fast sample-and-hold, both single and scan mode operate in fast sample-and-hold conversion spped.

• To use single or scan mode under sample-and-hold enabled mode, make sure that A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIM1), A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1), A/D sample-and-hold conversion spped select (ADSSHSPD)bit in A/D0 Single Mode Register 1 (AD0SIM1) and A/D sample-and-hold conversion speed select (ADCSHSPD) bit in A/D0 Scan Mode Register 1 (AD0SCM1) are set in the same value.

11.2.5 A/D Conversion Speed Control Register

A/D0 Conversion Speed Control Register (AD0CVSCR) <Address: H'0080 0087>

b8	9	10	11	12	13	14	b15
0	0	0	0	0	0	0	ADCVSD 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–14	No function assigned. Fix to "0".		0	0
15	ADCVSD (Note 1) A/D conversion speed control bit	0: Slow mode 1: Fast mode	R	W

Note 1: The A/D conversion speed is determined by a combination of ADCVSD bit and A/D Single Mode Register 1's relevant bit during single mode, or a combination of ADCVSD bit and A/D Scan Mode Register 1's relevant bit during scan mode.

The A/D Conversion Speed Control Register controls the A/D conversion speed during single and scan modes of the A/D Converter. The A/D conversion speed is determined in combination with A/D Single Mode Register 1's conversion speed select bit (Double/Normal).

11.2.6 A/D Disconnection Detection Assist Function Control Register

A/D0 Disconnection Detection Assist Function Control Register (AD0DDACR) <Address: H'0080 0086>

b0	1	2	3	4	5	6	b7
0	0	0	0	0	0	0	ADDDAEN 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–6	No function assigned. Fix to "0".		0	0
7	ADDDAEN (Note 1) A/D disconnection detection assist function enable bit	0: Disable A/D disconnection detection assist function 1: Enable A/D disconnection detection assist function	R	W

Note 1: For the A/D disconnection detection assist function to be enabled, the conversion start state (discharge or precharge) must be set using the A/D disconnection detection assist method select register after setting the ADDDAEN bit to "1".

The A/D Disconnection Detection Assist Function Control Register is used to enable or disable the content of the A/D Disconnection Detection Assist Method Select Register.

Note: • If any analog input wiring is disconnected, the conversion result varies depending on the circuits fitted external to the chip. This function must be fully evaluated in the actual application system before it can be used.

11.2.7 A/D Disconnection Detection Assist Method Select Register

A/D0 Disconnection Detection Assist Method Select Register (AD0DDASEL) <Address: H'0080 008A>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
ADDDA SEL0 ?	ADDDA SEL1 ?	ADDDA SEL2 ?	ADDDA SEL3 ?	ADDDA SEL4 ?	ADDDA SEL5 ?	ADDDA SEL6 ?	ADDDA SEL7 ?	ADDDA SEL8 ?	ADDDA SEL9 ?	ADDDA SEL10 ?	ADDDA SEL11 ?	ADDDA SEL12 ?	ADDDA SEL13 ?	ADDDA SEL14 ?	ADDDA SEL15 ?

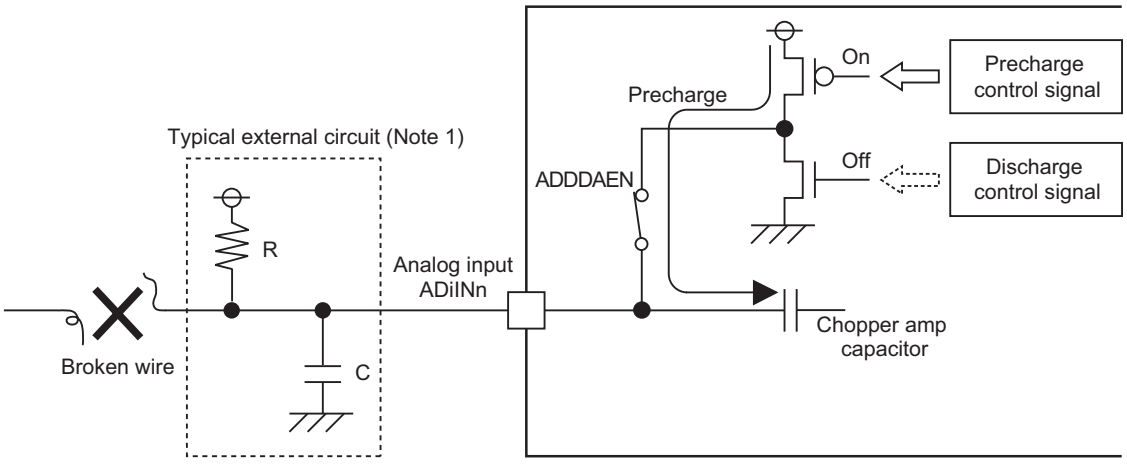
<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0	ADDDASEL0 Channel 0 disconnection detection assist method select bit	0: Discharge before conversion 1: Precharge before conversion	R	W
1	ADDDASEL1 Channel 1 disconnection detection assist method select bit			
2	ADDDASEL2 Channel 2 disconnection detection assist method select bit			
3	ADDDASEL3 Channel 3 disconnection detection assist method select bit			
4	ADDDASEL4 Channel 4 disconnection detection assist method select bit			
5	ADDDASEL5 Channel 5 disconnection detection assist method select bit			
6	ADDDASEL6 Channel 6 disconnection detection assist method select bit			
7	ADDDASEL7 Channel 7 disconnection detection assist method select bit			
8	ADDDASEL8 Channel 8 disconnection detection assist method select bit			
9	ADDDASEL9 Channel 9 disconnection detection assist method select bit			
10	ADDDASEL10 Channel 10 disconnection detection assist method select bit			
11	ADDDASEL11 Channel 11 disconnection detection assist method select bit			
12	ADDDASEL12 Channel 12 disconnection detection assist method select bit			
13	ADDDASEL13 Channel 13 disconnection detection assist method select bit			
14	ADDDASEL14 Channel 14 disconnection detection assist method select bit			
15	ADDDASEL15 Channel 15 disconnection detection assist method select bit			

Notes: • This register must always be accessed in halfwords.
• For these bits to be enabled, the ADDDAEN bit (A/D Disconnection Detection Assist Function Control Register bit 7) must be set to "1" before setting these bits.

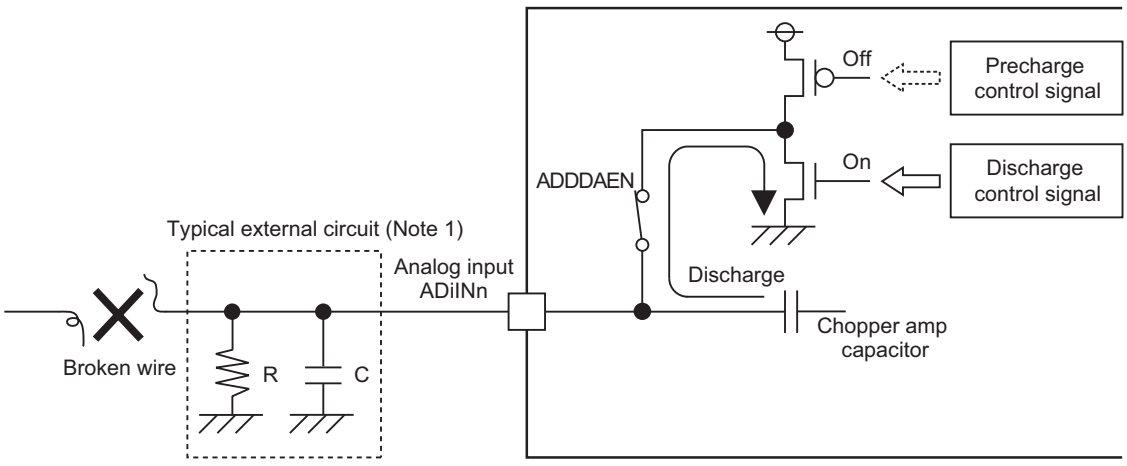
In order to prevent the A/D conversion result from being affected by the analog input voltage leakage from any preceding channel, the A/D Disconnection Detection Assist Method Select Register is used to control the conversion start state by selecting whether to discharge or precharge the chopper amp capacitor before starting regular conversion operation.

Figure 11.2.1 shows an example of A/D disconnection detection assist method in which the conversion start state is set to the AVCC0 side (i.e., precharge before conversion is selected). Figure 11.2.2 shows an example of A/D disconnection detection assist method in which the conversion start state is set to the AVSS0 side (i.e., discharge before conversion is selected).



Note 1: In case of broken wire, the conversion result varies with external circuits.
Therefore, careful evaluation is required before this function can be used.

Figure 11.2.1 Example of A/D Disconnection Detection on AVCC0 Side (Precharge Before Conversion Selected)



Note 1: In case of broken wire, the conversion result varies with external circuits.
Therefore, careful evaluation is required before this function can be used.

Figure 11.2.2 Example of A/D Disconnection Detection on AVSS0 Side (Discharge Before Conversion Selected)

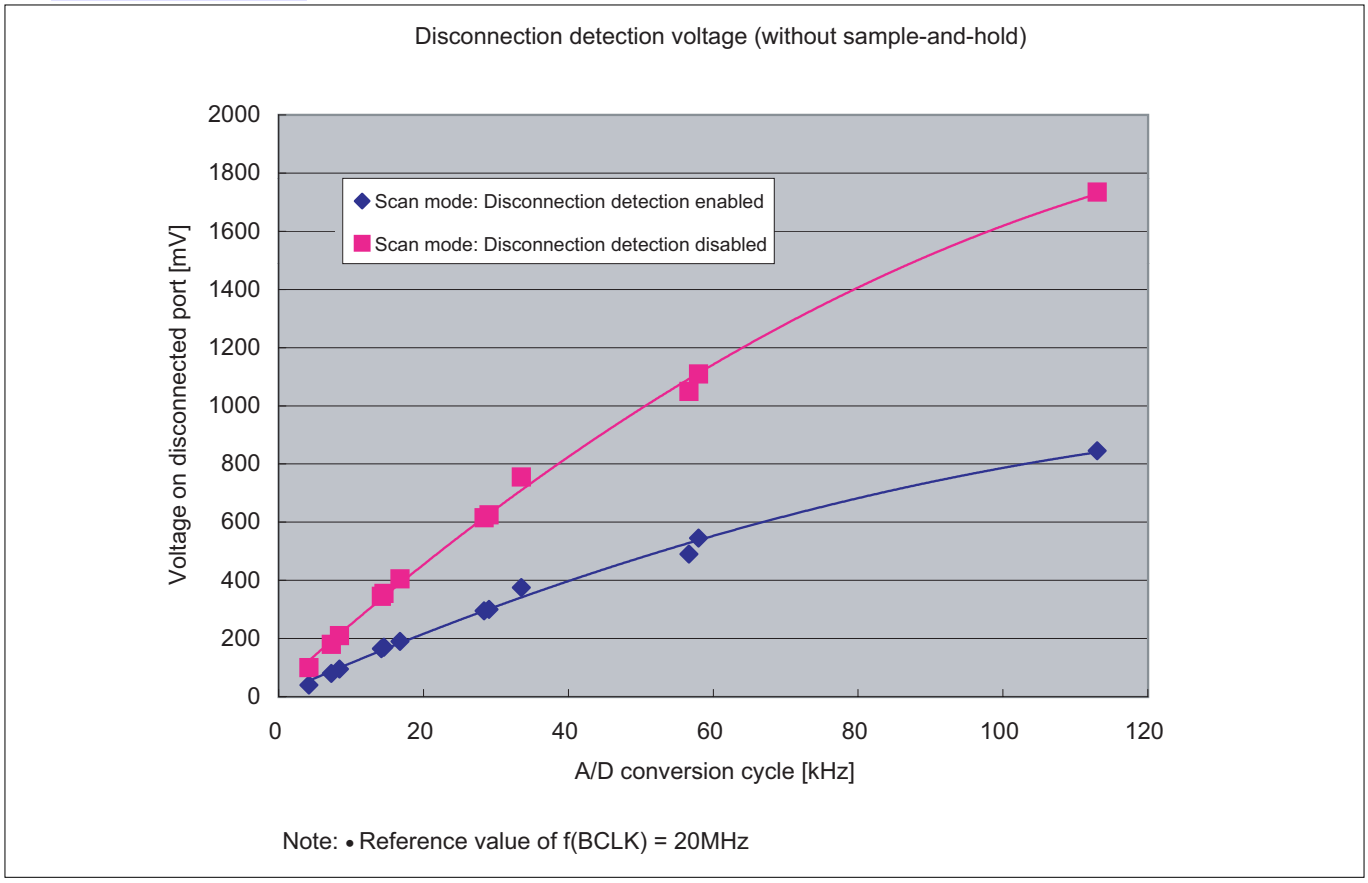


Figure 11.2.3 A/D Disconnection Detection Assist Data (when Discharge Before Conversion Selected)

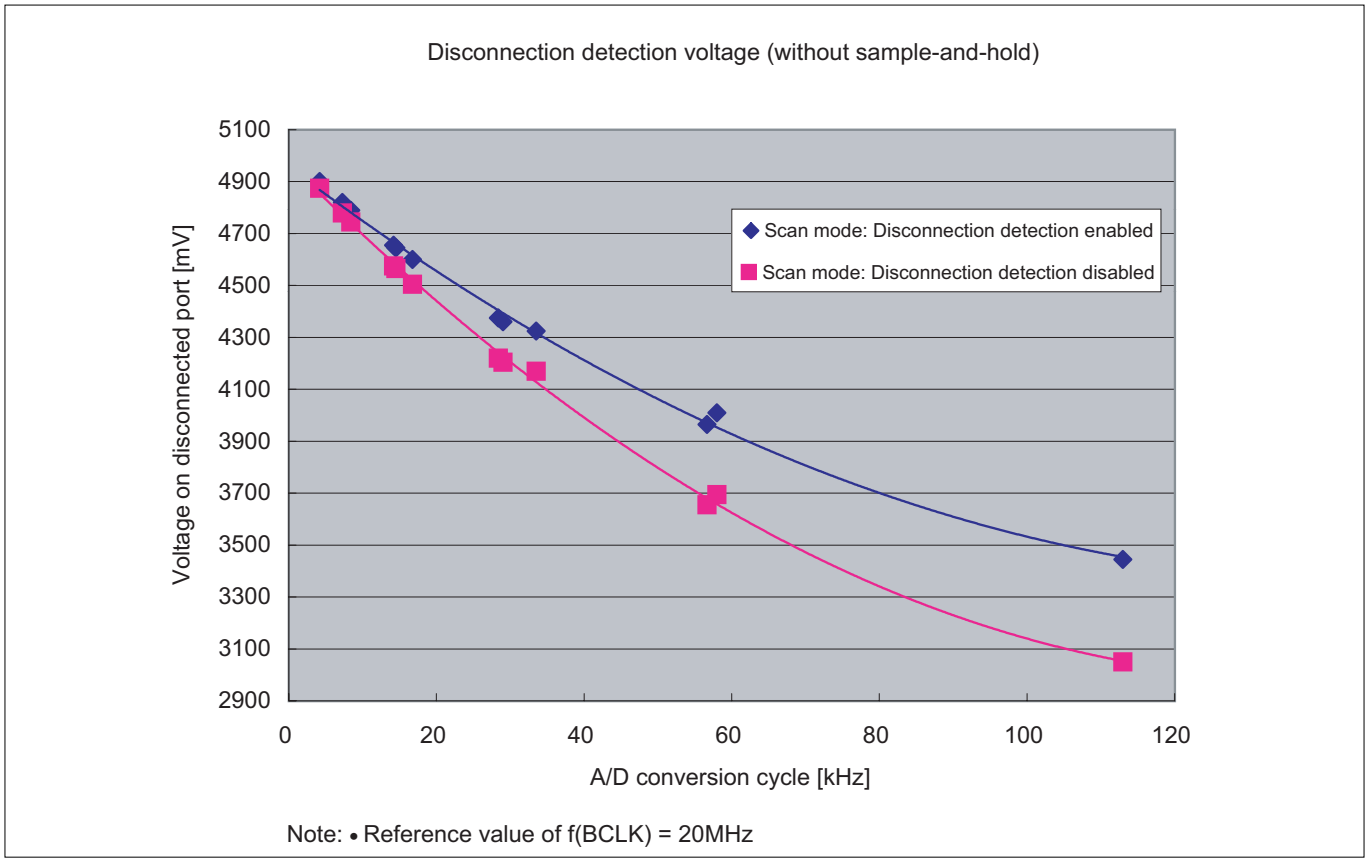


Figure 11.2.4 A/D Disconnection Detection Assist Data (when Precharge Before Conversion Selected)

11.2.8 A/D Successive Approximation Register

A/D0 Successive Approximation Register(AD0SAR) <Address: H'0080 0088>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
						ADSAR									
0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–5	No function assigned. Fix to "0".		0	0
6–15	ADSAR A/D successive approximation value/comparison value	• A/D successive approximation value (A/D conversion mode) • Comparison value (comparator mode)	R	W

Note: • This register must always be accessed in halfwords.

The A/D Successive Approximation Register (ADSAR) is used to read the conversion result of the A/D Converter when operating in A/D conversion mode or write a comparison value when operating in comparator mode.

In A/D conversion mode, the successive approximation method is used to perform A/D conversion. With this method, the reference voltage VREF0 and analog input voltages are sequentially compared bitwise beginning with the high-order bit, and the comparison result is set in the A/D Successive Approximation Register (ADSAR) bits 6–15. When the A/D conversion has finished, the value of this register is transferred to the 10-bit A/D Data Register (ADDTn) corresponding to each converted channel. When this register is accessed for read in the middle of A/D conversion, the value read from the register indicates the intermediate result of conversion.

In comparator mode, this register is used to write a comparison value (the voltage with which to "comparate"). Simultaneously with a write to this register, the A/D Converter starts comparing the voltage on the analog input pin selected with A/D Single Mode Register 1 and the value written in this register. After comparate operation, the result is stored in the A/D Comparate Data Register (ADCMP).

Use the calculation formula shown below to find the comparison value to be written to the A/D Successive Approximation Register (ADSAR) during comparator mode.

$$\text{Comparison value} = \text{H'3FF} \times \frac{\text{Comparate comparison voltage [V]}}{\text{VREF0 input voltage [V]}}$$

11.2.9 A/D Comparate Data Register

A/D0 Comparate Data Register (AD0CMP) <Address: H'0080 008C>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
AD CMP0 ?	AD CMP1 ?	AD CMP2 ?	AD CMP3 ?	AD CMP4 ?	AD CMP5 ?	AD CMP6 ?	AD CMP7 ?	AD CMP8 ?	AD CMP9 ?	AD CMP10 ?	AD CMP11 ?	AD CMP12 ?	AD CMP13 ?	AD CMP14 ?	AD CMP15 ?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	ADCMP0–ADCMP15 (Note 1) A/D comparate result flag	0: Analog input voltage > comparison voltage 1: Analog input voltage < comparison voltage	R	–

Note 1: During comparator mode, the bits in this register correspond one for one to channels 0–15.

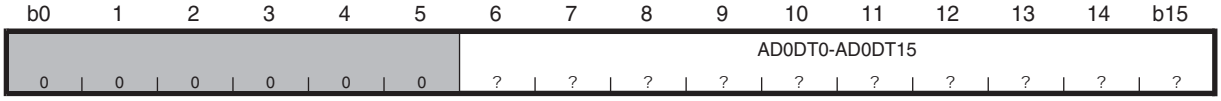
Note: • This register must always be accessed in halfwords.

When comparator mode is selected using the A/D Single Mode Register 1 ADSMSL (A/D conversion mode select) bit, the selected analog input voltage is compared with the value written to the A/D Successive Approximation Register and the result is stored in the corresponding bit of this comparate data register.

The bit or flag in this register is "0" when analog input voltage > comparison voltage, or "1" when analog input voltage < comparison voltage.

11.2.10 10-bit A/D Data Registers

10-bit A/D0 Data Register 0(AD0DT0)	<Address: H'0080 0090>
10-bit A/D0 Data Register 1(AD0DT1)	<Address: H'0080 0092>
10-bit A/D0 Data Register 2(AD0DT2)	<Address: H'0080 0094>
10-bit A/D0 Data Register 3(AD0DT3)	<Address: H'0080 0096>
10-bit A/D0 Data Register 4(AD0DT4)	<Address: H'0080 0098>
10-bit A/D0 Data Register 5(AD0DT5)	<Address: H'0080 009A>
10-bit A/D0 Data Register 6(AD0DT6)	<Address: H'0080 009C>
10-bit A/D0 Data Register 7(AD0DT7)	<Address: H'0080 009E>
10-bit A/D0 Data Register 8(AD0DT8)	<Address: H'0080 00A0>
10-bit A/D0 Data Register 9(AD0DT9)	<Address: H'0080 00A2>
10-bit A/D0 Data Register 10(AD0DT10)	<Address: H'0080 00A4>
10-bit A/D0 Data Register 11(AD0DT11)	<Address: H'0080 00A6>
10-bit A/D0 Data Register 12(AD0DT12)	<Address: H'0080 00A8>
10-bit A/D0 Data Register 13(AD0DT13)	<Address: H'0080 00AA>
10-bit A/D0 Data Register 14(AD0DT14)	<Address: H'0080 00AC>
10-bit A/D0 Data Register 15(AD0DT15)	<Address: H'0080 00AE>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–5	No function assigned.		0	–
6–15	AD0DT0-AD0DT15 10-bit A/D data	10-bit A/D conversion result	R	–

Note: • These registers must always be accessed in halfwords.

During single mode, the 10-bit A/D Data Registers are used to store the result of A/D conversion performed on each corresponding channel.

During single-shot or continuous scan mode, the content of the A/D Successive Approximation Register is transferred to the 10-bit A/D Data Register for the corresponding channel when A/D conversion on each channel has finished. Each 10-bit A/D Data Register retains the last conversion result until they receive the next conversion result transferred, allowing the content to be read out at any time.

11.2.11 8-bit A/D Data Registers

8-bit A/D0 Data Register 0(AD08DT0)	<Address: H'0080 00D1>
8-bit A/D0 Data Register 1(AD08DT1)	<Address: H'0080 00D3>
8-bit A/D0 Data Register 2(AD08DT2)	<Address: H'0080 00D5>
8-bit A/D0 Data Register 3(AD08DT3)	<Address: H'0080 00D7>
8-bit A/D0 Data Register 4(AD08DT4)	<Address: H'0080 00D9>
8-bit A/D0 Data Register 5(AD08DT5)	<Address: H'0080 00DB>
8-bit A/D0 Data Register 6(AD08DT6)	<Address: H'0080 00DD>
8-bit A/D0 Data Register 7(AD08DT7)	<Address: H'0080 00DF>
8-bit A/D0 Data Register 8(AD08DT8)	<Address: H'0080 00E1>
8-bit A/D0 Data Register 9(AD08DT9)	<Address: H'0080 00E3>
8-bit A/D0 Data Register 10(AD08DT10)	<Address: H'0080 00E5>
8-bit A/D0 Data Register 11(AD08DT11)	<Address: H'0080 00E7>
8-bit A/D0 Data Register 12(AD08DT12)	<Address: H'0080 00E9>
8-bit A/D0 Data Register 13(AD08DT13)	<Address: H'0080 00EB>
8-bit A/D0 Data Register 14(AD08DT14)	<Address: H'0080 00ED>
8-bit A/D0 Data Register 15(AD08DT15)	<Address: H'0080 00EF>

b8	9	10	11	12	13	14	b15
AD08DT0-AD08DT15							
?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–15	AD08DT0-AD08DT15 8-bit A/D data	8-bit A/D conversion result	R	–

The A/D data register is used to store the 8-bit conversion data for the A/D converter.

During single mode, the 8-bit A/D Data Registers store the result of A/D conversion performed on each corresponding channel.
During single-shot or continuous scan mode, the content of the A/D Successive Approximation Register is transferred to the 8-bit A/D Data Register for the corresponding channel when A/D conversion on each channel has finished. Each 8-bit A/D Data Register retains the last conversion result until they receive the next conversion result transferred, allowing the content to be read out at any time.

11.3 Functional Description of A/D Converter

11.3.1 How to Find Analog Input Voltages

The A/D Converter performs A/D conversion using a 10-bit successive approximation method. The equation shown below is used to calculate the actual analog input voltage from the digital value obtained by executing A/D conversion.

$$\text{Analog input voltage [v]} = \frac{\text{A/D conversion result x VREF input voltage [V]}}{1{,}024}$$

The A/D Converter is a 10-bit converter, providing a resolution of 1,024 discrete voltage levels. Because the reference voltage for the A/D Converter is the voltage applied to the VREF0 pin, make sure that an exact and stable constant-voltage power supply is connected to VREF0. Also make sure the analog circuit power supply and ground (AVCC0, AVSS0) are separated from those of the digital circuit, with sufficient noise prevention measures incorporated.
For details about the conversion accuracy, see Section 11.3.5, “Accuracy of A/D Conversion.”

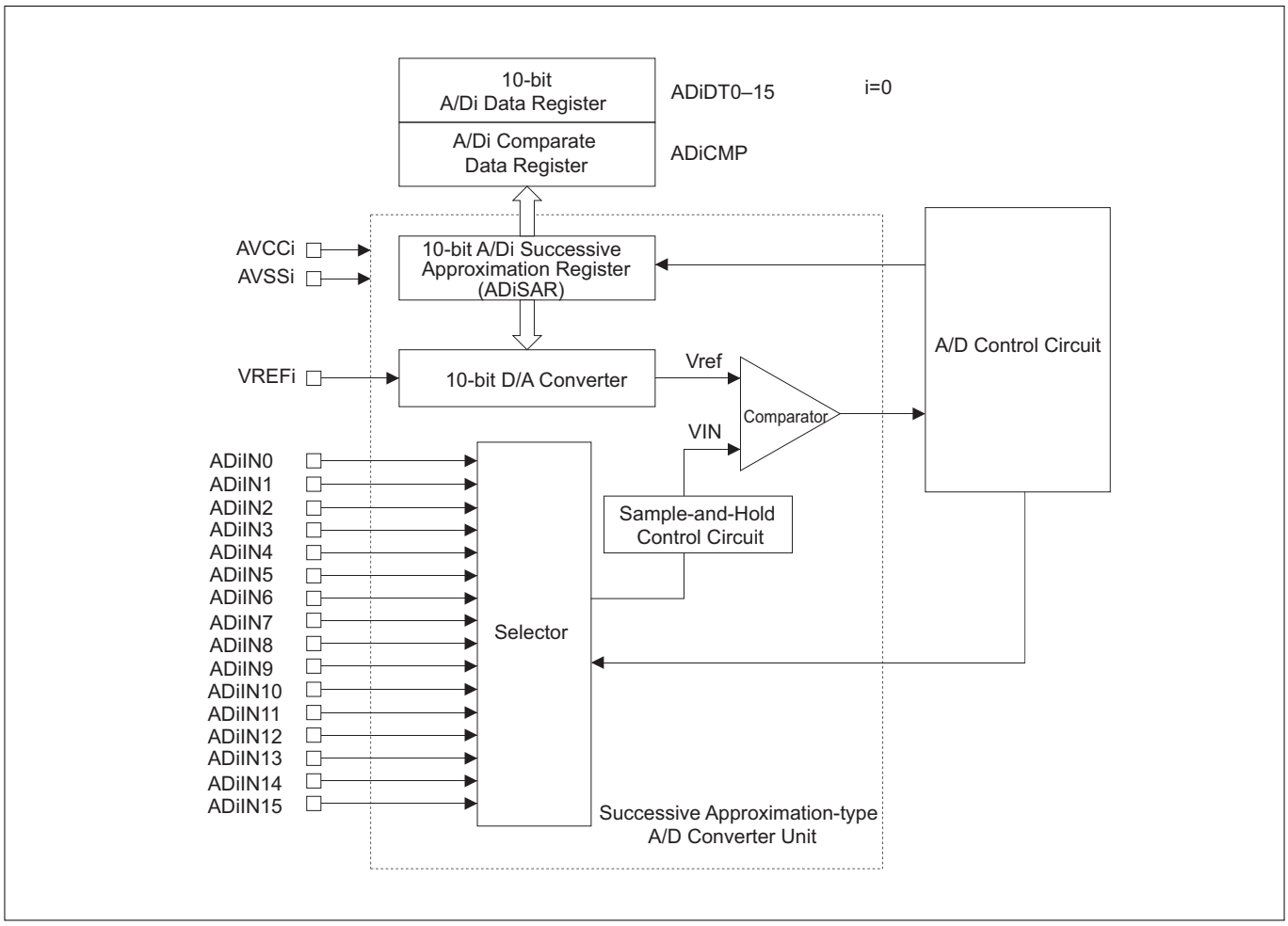


Figure 11.3.1 Outline Block Diagram of the Successive Approximation-type A/D Converter Unit

11.3.2 A/D Conversion by Successive Approximation Method

The A/D Converter use an A/D conversion start trigger (software or hardware) as they start A/D conversion. Once A/D conversion begins, the following operation is automatically performed.

1. During single mode, A/D Single Mode Register 0's A/D conversion/comparate completion bit is cleared to "0". During scan mode, A/D Scan Mode Register 0's A/D conversion completion bit is cleared to "0".
2. The content of the A/D Successive Approximation Register is cleared to H'0000.
3. The A/D Successive Approximation Register's most significant bit (bit 6) is set to "1".
4. The comparison voltage, Vref (Note 1), is fed from the D/A Converter into the comparator.
5. The comparison voltage, Vref, and the analog input voltage, VIN, are compared, and the comparison result will be stored in bit 6.
   If Vref < VIN, then bit 6 = "1"
   If Vref > VIN, then bit 6 = "0"
6. Operations in 3 through 5 above are executed for all other bits from bit 7 to bit 15.
7. The value stored in the A/D Successive Approximation Register by the time comparison for bit 15 has finished is held in it as the A/D conversion result.

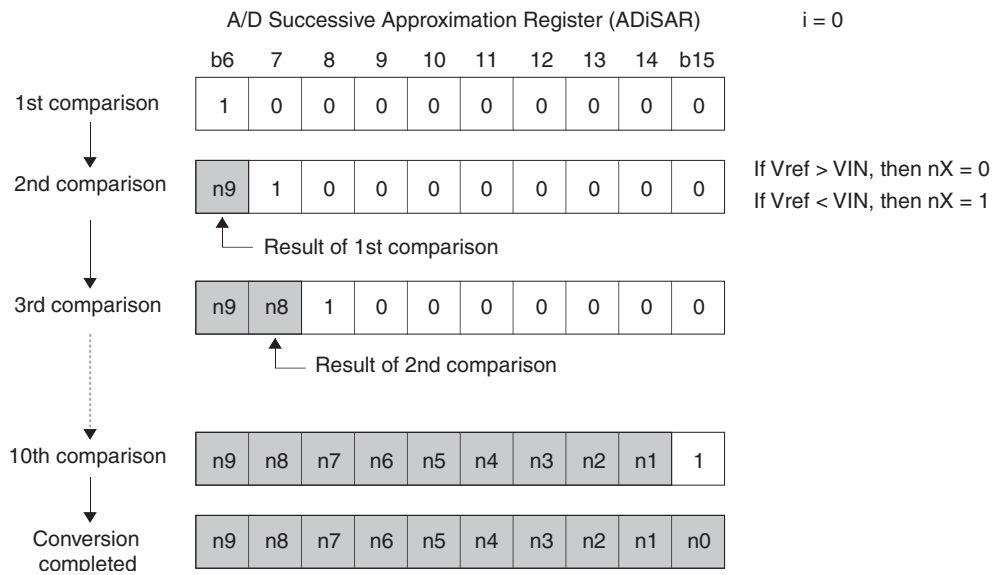


Figure 11.3.2 Changes of the A/D Successive Approximation Register during A/D Convert Operation

Note 1: The comparison voltage, Vref (the voltage fed from the D/A Converter into the comparator), is determined according to changes of the A/D Successive Approximation Register content. Shown below are the equations used to calculate the comparison voltage, Vref.

- If the A/D Successive Approximation Register content = 0
  Vref [V] = 0
- If the A/D Successive Approximation Register content = 1 to 1,023
  Vref [V] = (reference voltage VREF0 / 1,024) × (A/D Successive Approximation Register content – 0.5)

The conversion result is stored in the 10-bit A/D Data Register (AD0DTn) corresponding to each converted channel. There is also an 8-bit A/D Data Register (AD08DTn) for each channel, from which the 8 high-order bits of the 10-bit A/D conversion result can be read out.

The following shows the procedure for A/D conversion by a successive approximation method in each operation mode.

(1) Single mode

The convert operation stops when comparison for the A/D Successive Approximation Register bit 15 is completed. The content (A/D conversion result) of the A/D Successive Approximation Register is transferred to the 10-bit A/D Data Registers 0–15 for the converted channel.

(2) Single-shot scan mode

When comparison for the A/D Successive Approximation Register bit 15 on a specified channel is completed, the content of the A/D Successive Approximation Register is transferred to the corresponding 10-bit A/D Data Registers 0–15, and the convert operations in steps 2 to 7 above are reexecuted for the next channel to be converted. In single-shot scan mode, the convert operation stops when A/D conversion in one specified scan loop is completed.

(3) Continuous scan mode

When comparison for the A/D Successive Approximation Register bit 15 on a specified channel is completed, the content of the A/D Successive Approximation Register is transferred to the corresponding 10-bit A/D Data Registers 0–15, and the convert operations in steps 2 to 7 above are reexecuted for the next channel to be converted.
In continuous scan mode, the convert operation is executed continuously until scan operation is forcibly terminated by setting the A/D conversion stop bit (Scan Mode Register 0 bit 6) to "1".

11.3.3 Comparator Operation

When comparator mode (single mode only) is selected, the A/D Converter functions as a comparator which compares analog input voltages with the comparison voltage that is set by software.

When a comparison value is written to the successive approximation register, the A/D Converter starts “comparating” the analog input voltage selected by the Single Mode Register 1 analog input select bit with the value written into the successive approximation register. Once comparate begins, the following operation is automatically executed.

1. The A/D Single Mode Register 0 A/D conversion/comparate completion bit is cleared to "0".
2. The comparison voltage, Vref (Note 1), is fed from the D/A Converter into the comparator.
3. The comparison voltage, Vref, and the analog input voltage, VIN, are compared, and the comparison result will be stored in the comparate result flag for the corresponding channel.
   If Vref < VIN, then the comparate result flag = 0
   If Vref > VIN, then the comparate result flag = 1
4. The comparate operation is stopped after storing the comparison result.

The comparison result is stored in the A/D Comparate Data Register (AD0CMP)’s corresponding bit.

Note 1: The comparison voltage, Vref (the voltage fed from the D/A Converter into the comparator), is determined according to changes of the A/D Successive Approximation Register content. Shown below are the equations used to calculate the comparison voltage, Vref.
- If the A/D Successive Approximation Register content = 0
  Vref [V] = 0
- If the A/D Successive Approximation Register content = 1 to 1,023
  Vref [V] = (reference voltage VREF0 / 1,024) x (A/D0 Successive Approximation Register content – 0.5)

11.3.4 Calculating the A/D Conversion Time

The A/D conversion time is expressed by the sum of dummy cycle time and actual execution cycle time. The following shows each time factor necessary to calculate the conversion time.

1. Start dummy time

A time from when the CPU executed the A/D conversion start instruction to when the A/D Converter starts A/D conversion

2. A/D conversion execution cycle time

If sample-and-hold is enabled, the sampling time is included in this execution cycle time.

3. Comparate execution cycle time

4. End dummy time

A time from when the A/D Converter has finished A/D conversion to when the CPU can stably read out the conversion result from the A/D data register.

5. Scan to scan dummy time

A time during single-shot or continuous scan mode from when the A/D Converter has finished A/D conversion on a channel to when it starts A/D conversion on the next channel.

The equation to calculate the A/D conversion time is as follows:

A/D conversion time = Start dummy time + Execution cycle time
(+ Scan to scan dummy time + Execution cycle time
+ Scan to scan dummy time + Execution cycle time
+ Scan to scan dummy time .... + Execution cycle time)
+ End dummy time

Note: • Enclosed in ( ) are the conversion time required for the second and subsequent channels to be converted in scan mode.

(1) Calculating the conversion time during A/D conversion mode

The following schematically shows the method for calculating the conversion time during A/D conversion mode.

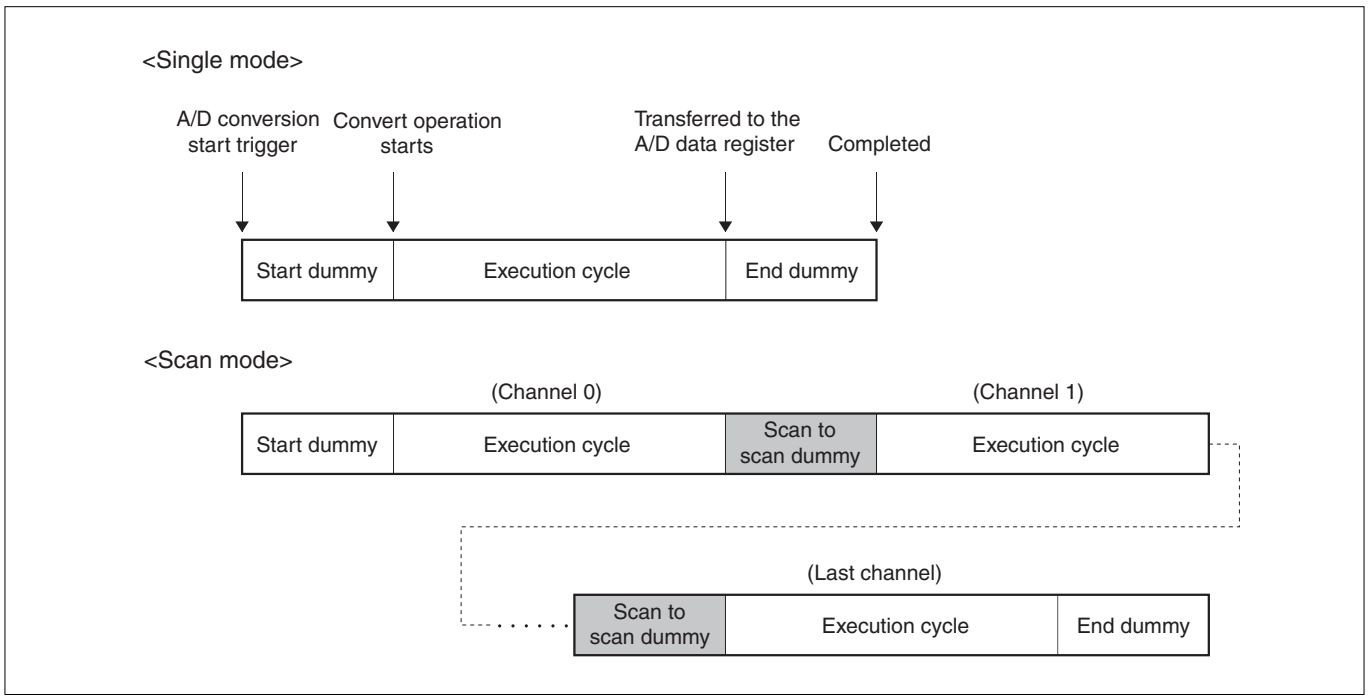


Figure 11.3.3 Conceptual Diagram of A/D Conversion Time

(2) Calculating the conversion time when sample-and-hold is enabled

The following schematically shows the method for calculating the conversion time when the sample-and-hold function is enabled.

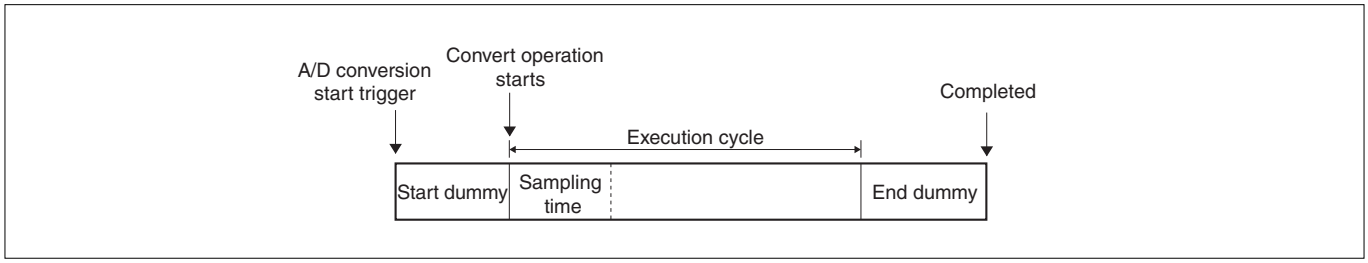


Figure 11.3.4 Conceptual Diagram of A/D Conversion Time when Sample-and-Hold is Enabled

Table 11.3.1 Conversion Clock Periods in A/D conversion Mode when Sample-and-Hold is Disabled or Normal Sample-and-Hold is Enabled (Shortest Period) **Unit: BCLK**

Conversion speed		Start dummy (Note 1)	Execution cycle	End dummy	Scan to scan dummy (Note 2)
Slow mode	Normal speed	4	294	1	4
	Double speed	4	168	1	4
Fast mode	Normal speed	4	126	1	4
	Double speed	4	84	1	4

Note 1: The same applies to both software and hardware triggers.
Note 2: Only during scan mode operation, execution time per channel is added.

Table 11.3.2 Conversion Clock Periods in A/D Conversion Mode when Fast Sample-and-Hold is Enabled (Shortest Period) **Unit: BCLK**

Conversion speed		Start dummy (Note 1)	Execution cycle	End dummy	Scan to scan dummy (Note 2)
Slow mode	Normal speed	4	186	1	4
	Double speed	4	96	1	4
Fast mode	Normal speed	4	90	1	4
	Double speed	4	48	1	4

Note 1: The same applies to both software and hardware triggers.
Note 2: Only during scan mode operation, execution time per channel is added.

(3) Calculating the conversion time during comparator mode

The following schematically shows the method for calculating the conversion time during comparator mode.

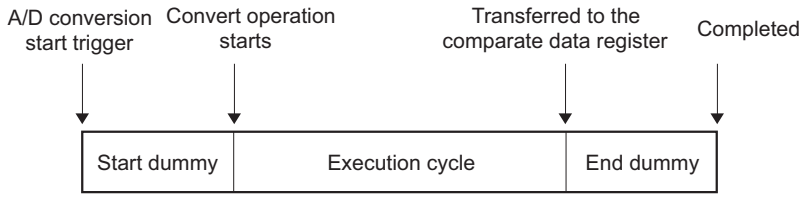


Figure 11.3.5 Conceptual Diagram of A/D Conversion Time during Comparator Mode

Table 11.3.3 Conversion Clock Periods during Comparator Mode (Shortest Period) **Unit: BCLK**

Conversion speed		Start dummy	Execution cycle	End dummy
Slow mode	Normal speed	4	42	1
	Double speed	4	24	1
Fast mode	Normal speed	4	18	1
	Double speed	4	12	1

(4) A/D conversion time

A total A/D conversion time in various modes are shown in the table below.

Table 11.3.4 A/D Conversion Time (Total Time) **Unit: BCLK**

Conversion start method	Conversion speed		Conversion mode (Note 1)	Conversion time	When fast sample-and-hold enabled
Software and hardware triggers (Note 2)	Slow Mode	Normal speed	Single mode	299	191
			n-channel single-shot scan/ continuous scan mode	(298 × n)+1	(190 × n)+1
			Comparator mode	47	47
		Double speed	Single mode	173	101
			n-channel single-shot scan/ continuous scan mode	(172 × n)+1	(100 × n)+1
			Comparator mode	29	29
	Fast Mode	Normal speed	Single mode	131	95
			n-channel single-shot scan/ continuous scan mode	(130 × n)+1	(94 × n)+1
			Comparator mode	23	23
		Double speed	Single mode	89	53
			n-channel single-shot scan/ continuous scan mode	(88 × n)+1	(52 × n)+1
			Comparator mode	17	17

Note 1: For single mode and comparator mode, this indicates an A/D conversion or comparate time per channel. For single-shot and continuous scan modes, this indicates an A/D conversion time per scan loop.

Note 2: This indicates a time from when a register write cycle has finished to when an A/D conversion completion interrupt request is generated, or a time from when an event bus or other MJT event has occurred to when an A/D conversion completion interrupt request is generated.

11.3.5 Accuracy of A/D Conversion

The accuracy of the A/D Converter is indicated by an absolute accuracy. The absolute accuracy refers to a difference expressed by LSB between the output code obtained by A/D converting the analog input voltages and the output code expected for an A/D converter with ideal characteristics. The analog input voltages used during accuracy measurement are the midpoint values of the voltage width in which an A/D converter with ideal characteristics produces the same output code. If VREF = 5.12 V, for example, the width of 1 LSB for a 10-bit A/D converter is 5 mV, so that 0 mV, 5 mV, 10 mV, 15 mV, 20 mV, 25 mV and so on are selected as midpoints of the analog input voltage.

If an A/D converter is said to have the absolute accuracy of ±2 LSB, it means that if the input voltage is 25 mV, for example, the output code expected for an A/D converter with ideal characteristics is H'005, and the actual A/D conversion result is in the range of H'003 to H'007. Note that the absolute accuracy includes zero and full-scale errors.

When actually using the A/D Converter, the analog input voltages are in the range of AVSS0 to VREF0. Note, however, that low VREF0 voltages result in a poor resolution. Note also that output codes for the analog input voltages from VREF0 to AVCC0 are always H'3FF.

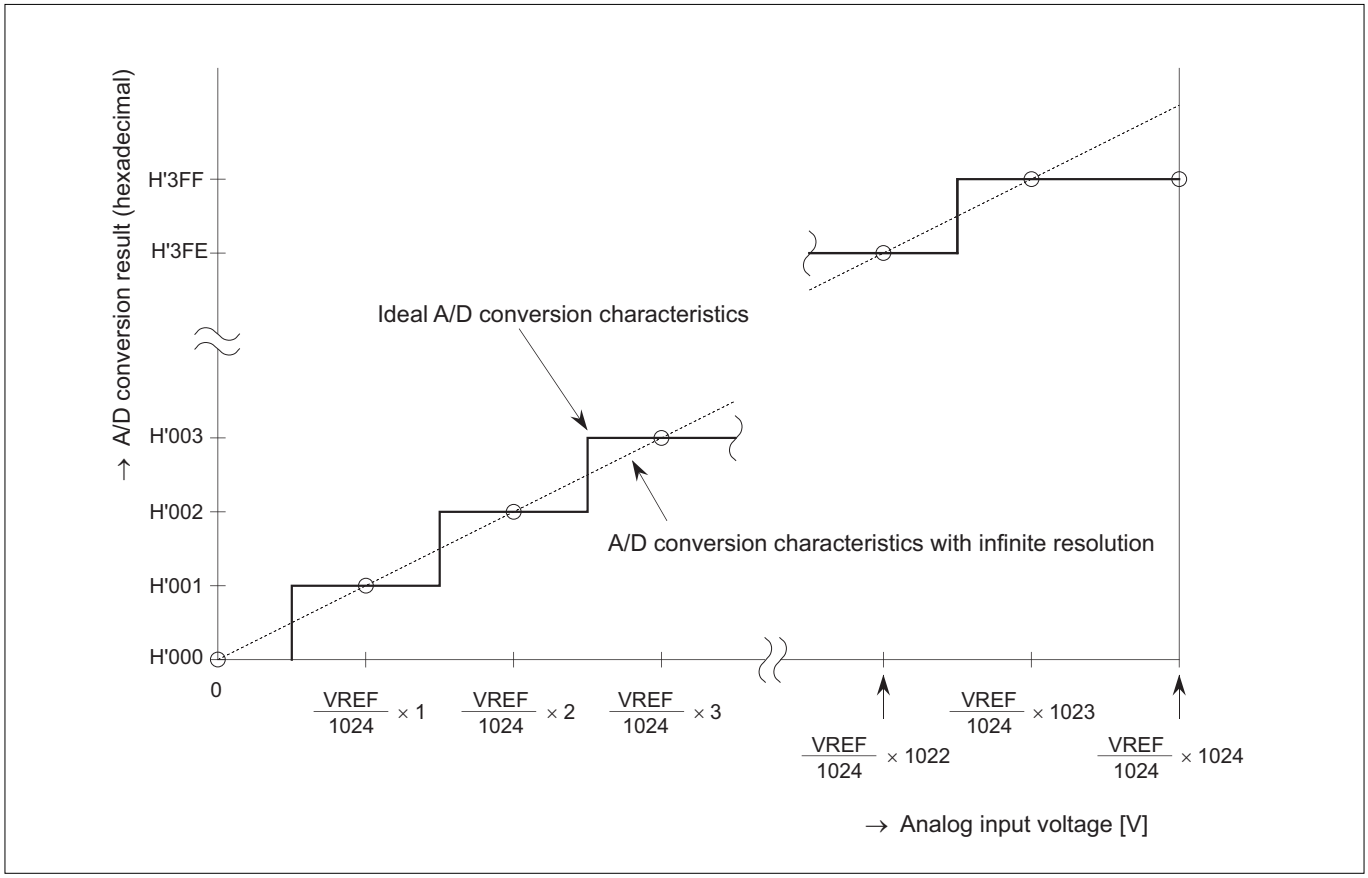


Figure 11.3.6 Ideal A/D Conversion Characteristics Relative to the 10-bit A/D Converter's Analog Input Voltages

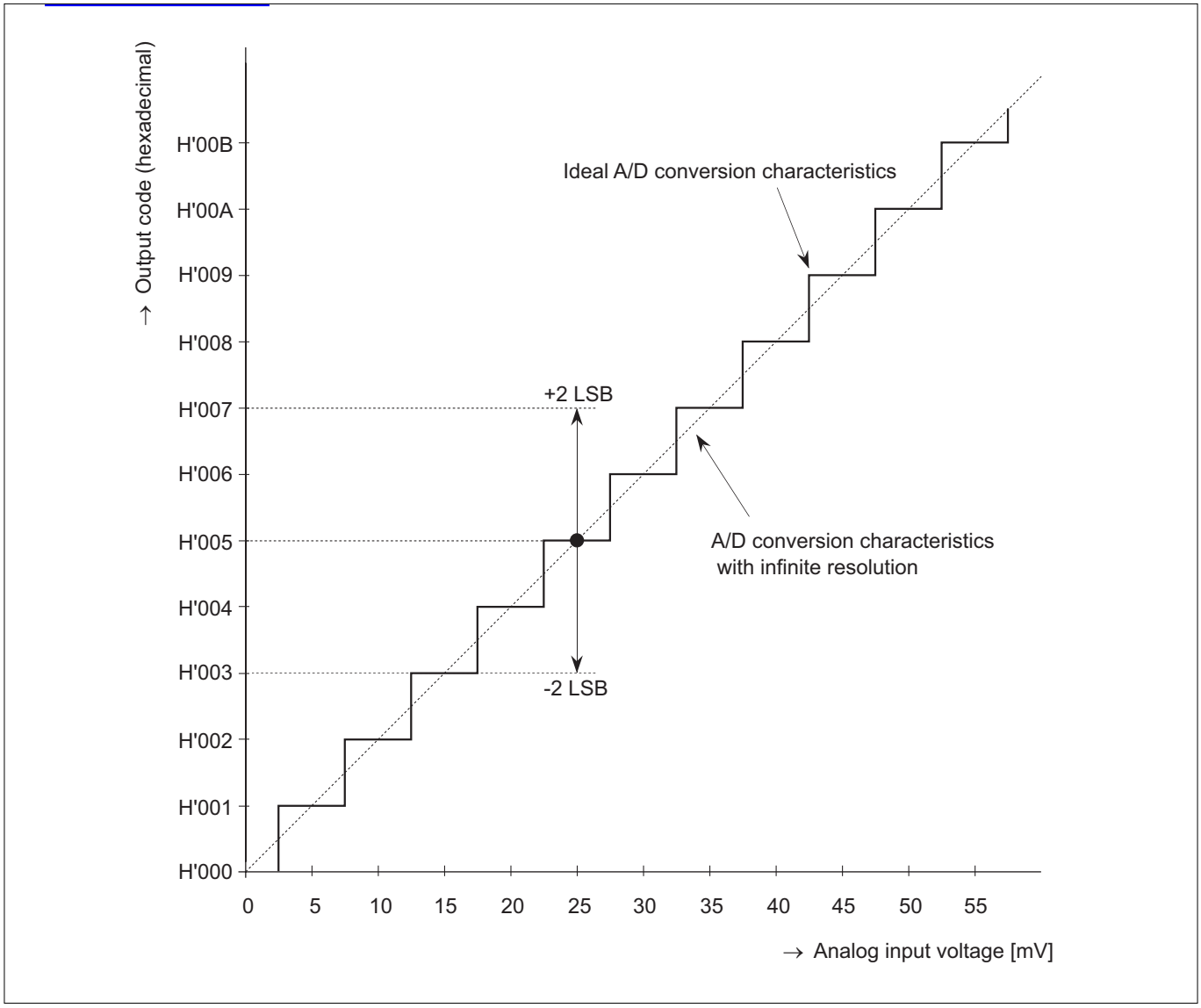


Figure 11.3.7 Absolute Accuracy of A/D Converter

11.4 Inflow Current Bypass Circuit

If when the A/D Converter is A/D converting a selected analog input an overvoltage exceeding the converter's absolute maximum rating is applied to any unselected analog input, the selector for the unselected analog input is inadvertently turned on by that overvoltage. This causes current to leak to the selected analog input, and the accuracy of the A/D conversion result is thereby deteriorated.

The Inflow Current Bypass Circuit fixes the internal signals of unselected analog inputs to the GND level, so that when an overvoltage is applied, this circuit lets the current flow into the GND and prevents it from leaking to the selected analog input. That way, the accuracy of the A/D conversion result is prevented from being deteriorated by overvoltages.

This circuit is always active while the A/D Converter is operating, and does not need to be controlled in software.

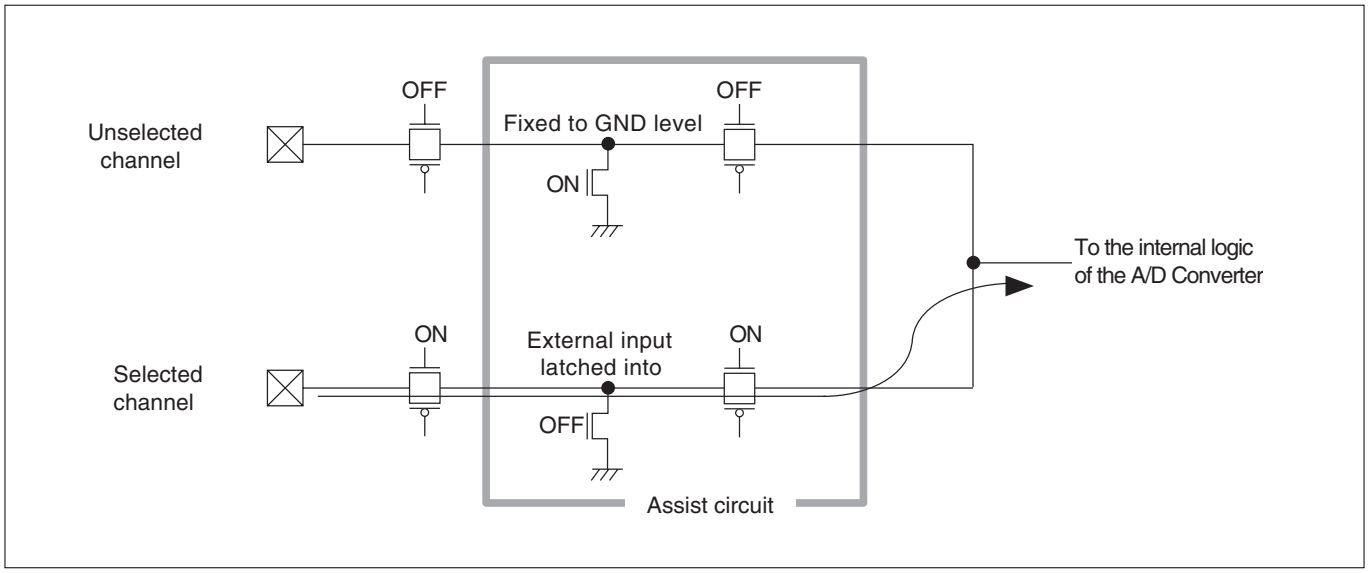


Figure 11.4.1 Configuration of the Inflow Current Bypass Circuit

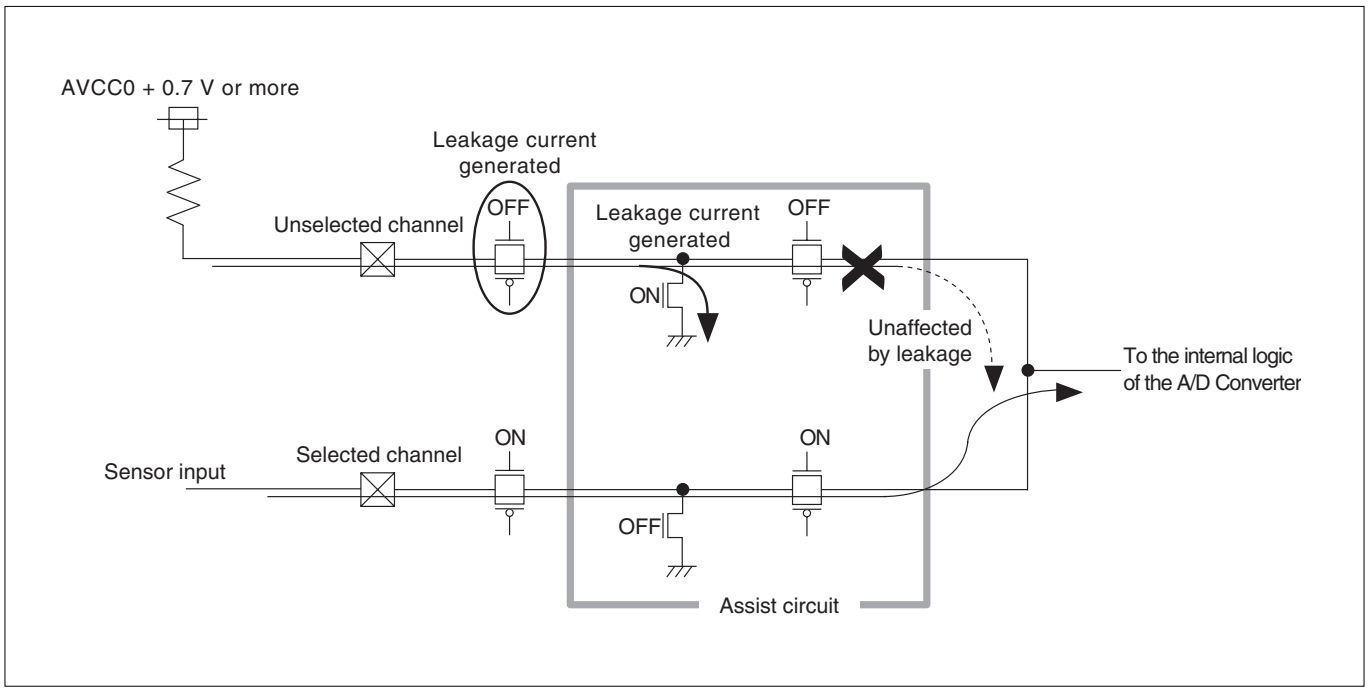


Figure 11.4.2 Example of an Inflow Current Bypass Circuit where AVCC0 + 0.7 V or More is Applied

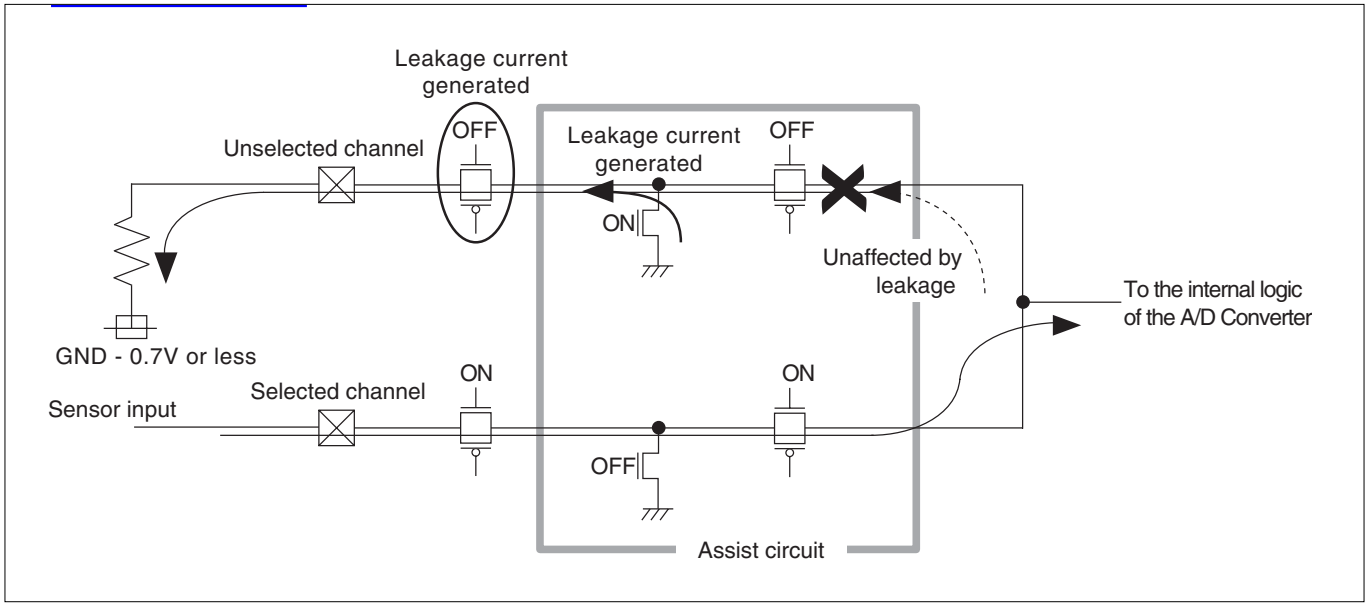


Figure 11.4.3 Example of an Inflow Current Bypass Circuit where GND – 0.7 V or Less is Applied

Table 11.4.1 Accuracy Errors (Actual Performance Values) when Current is Injected into AD0IN0

		Accuracy error on overcurrent injected ports (Unit: LSB)															
Analog input pin		AD0IN0	AD0IN1	AD0IN2	AD0IN3	AD0IN4	AD0IN5	AD0IN6	AD0IN7	AD0IN8	AD0IN9	AD0IN10	AD0IN11	AD0IN12	AD0IN13	AD0IN14	AD0IN15
Injection current (Note 1)	10mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-1mA		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-2mA		-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-3mA		-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-4mA		-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-5mA		-2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
	-6mA		-3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
	-7mA		-3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
	-8mA		-3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
	-9mA		-4	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
	-10mA		-5	-1	0	0	0	0	0	0	0	0	0	0	0	0	0

Note 1: The conversion accuracy is not affected unless the injection current is greater than 1 mA.

11.5 Notes on Using A/D Converter

• Forcible termination during scan operation

If A/D conversion is forcibly terminated by setting the A/D conversion stop bit (AD0CSTP) to "1" during scan mode operation and the A/D data register for the channel that was in the middle of conversion is accessed for read, the read value shows the last conversion result that had been transferred to the data register before the conversion was forcibly terminated.

• Modification of the A/D converter related registers

If the content of any register—A/D Conversion Interrupt Control Register, Single or Scan Mode Registers or A/D Successive Approximation Register, except the A/D conversion stop bit—is modified in the middle of A/D conversion, the conversion result cannot be guaranteed. Therefore, do not modify the contents of these registers while A/D conversion is in progress, or be sure to restart A/D conversion if register contents have been modified.

• Handling of analog input signals

When using the A/D Converter with its sample-and-hold function disabled, make sure the analog input level is fixed during A/D conversion.

• A/D conversion completed bit read timing

To read the A/D conversion completed bit (Single Mode Register 0 bit 5 or Scan Mode Register 0 bit 5) immediately after A/D conversion has started, be sure to adjust the timing 2 BCLK periods by, for example, inserting a NOP instruction before read.

• Regarding the analog input pins

Figure 11.5.1 shows the internal equivalent circuit of the A/D Converter's analog input part. To obtain accurate A/D conversion results, make sure the internal capacitor C2 of the A/D conversion circuit is charged up within a predetermined time (sampling time). To meet this sampling time requirement, it is recommended that a stabilizing capacitor C1 be connected external to the chip.
The method for determining the necessary value of this external stabilizing capacitor with respect to the output impedance of an analog output device is described below. Also, an explanation is made of the case where the output impedance of an analog output device is low and the external stabilizing capacitor C1 is unnecessary.

• Rated value of the absolute accuracy

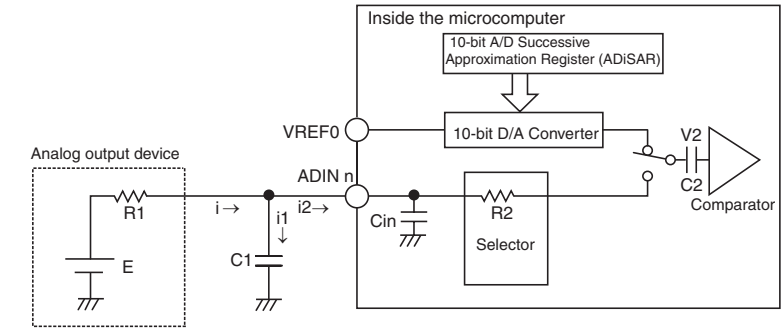
The rated value of the absolute accuracy is the actual performance value of the microcomputer alone, with influences of the power supply wiring and noise on the board not taken into account. When designing the application system, use caution for the board layout by, for example, separating the analog circuit power supply and ground (AVCC0, AVSS0 and VREF0) from those of the digital circuit and incorporating measures to prevent the analog input pins from being affected by noise, etc. from other digital signals.

• Single and scan mode operation under sample-and-hold enabled mode

If either A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIM1) or A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as sample-and-hold enabled, both single and scan mode operate under sample-and-hold enabled mode.

If either A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIMI1) or A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as sample-and-hold enabled, and A/D sample-and-hold conversion speed select (ADSSHSPD) bit in A/D0 Single Mode Register 1 (AD0SM1) or A/D sample-and-hold conversion speed select (ADCSHSPD) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as fast sample-and-hold, both single and scan mode operate in fast sample-and-hold conversion spped.

To use single or scan mode under sample-and-hold enabled mode, make sure that A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIM1), A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1), A/D sample-and-hold conversion spped select (ADSSHSPD)bit in A/D0 Single Mode Register 1 (AD0SIM1) and A/D sample-and-hold conversion speed select (ADCSHSPD) bit in A/D0 Scan Mode Register 1 (AD0SCM1) are set in the same value.



C1 : parasitic capacitance of the board + stabilizing capacitance
R1 : resistance of analog output device
Cin : input pin capacitance (approx. 10 pF)

R2 : parasitic resistance of the selector (1-2 KΩ)
C2 : comparator capacitance (approx. 2.9 pF)

VREF : analog reference voltage
V2 : voltage across C2
E : voltage of analog output device

Figure 11.5.1 Internal Equivalent Circuit of the Analog Input Part

(a) Example for calculating the external stabilizing capacitor C1 (addition of this capacitor is recommended)

Assuming the R1 in Figure 11.5.1 is infinitely large and that the current necessary to charge the internal capacitor C2 is supplied from C1, if the potential fluctuation, Vp, caused by capacitance division of C1 and C2 is to be within 0.1 LSB, then what amount of capacitance C1 should have. For a 10-bit A/D Converter where VREF0 is 5.12 V, 1 LSB determination voltage = 5.12 V / 1,024 = 5 mV. The potential fluctuation of 0.1 LSB means a 0.5 mV fluctuation.

The relationship between the capacitance division of C1 and C2 and the potential fluctuation, Vp, is obtained by the equation below:

$$Vp = \frac{C2}{C1 + C2} \times (E - V2) \quad \text{---------- Eq. A-1}$$

Vp is also obtained by the equation below:

$$Vp = Vp1 \times \sum_{i=0}^{x-1} \frac{1}{2^i} < \frac{VREF0}{10 \times 2^x} \quad \text{---------- Eq. A-2}$$

where Vp1 = potential fluctuation in the first A/D conversion performed and x = 10 for a 10-bit resolution A/D converter

When Eq. A-1 and Eq. A-2 are solved, the following results:

$$C1 = C2 \left\{ \frac{E - V2}{Vp1} - 1 \right\} \quad \text{---------- Eq. A-3}$$

$$\therefore \quad C1 > C2 \{10 \times 2^x \times \sum_{i=0}^{x-1} \frac{1}{2^i} - 1 \} \quad \text{---------- Eq. A-4}$$

Thus, for a 10-bit resolution A/D Converter where C2 = 2.9 pF, C1 is 0.06 µF or more. Use this value for reference when setting up C1.

(b) Maximum value of the output impedance R1 when C1 is not added

If the external capacitor C1 in Figure 11.5.1 is not used, examination must be made to see if the analog output device can fully charge C2 within a predetermined time. First, the equation to find i2 when C1 in Figure 11.5.1 does not exist is shown below.

$$i2 = \frac{C2(E - V2)}{Cin \times R1 + C2(R1 + R2)} \times \exp \left\{ \frac{-t}{Cin \times R1 + C2(R1 + R2)} \right\} \quad \text{---------- Eq. B-1}$$

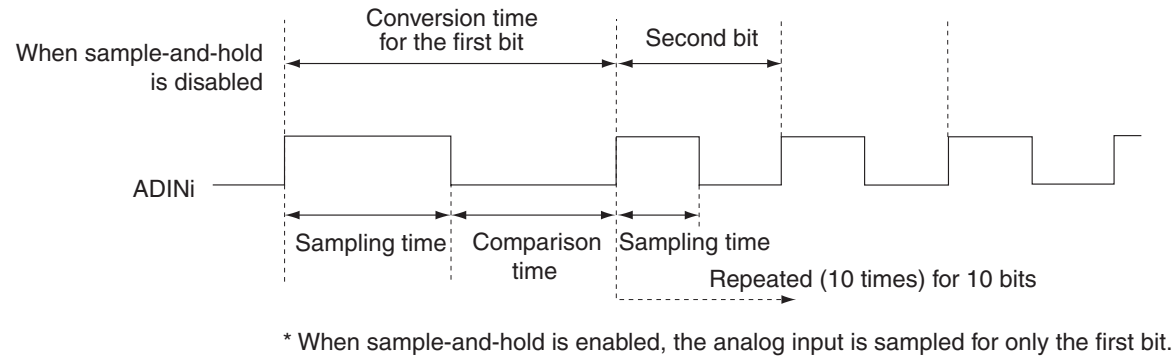


Figure 11.5.2 A/D Conversion Timing Diagram

Figure 11.5.2 shows an A/D conversion timing diagram. C2 must be charged up within the sampling time shown in this diagram. When the sample-and-hold function is disabled, the sampling time for the second and subsequent bits is about half that of the first bit.
The sampling times at the respective conversion speeds are listed in the table 11.5.1. Note that when the sample-and-hold function is enabled, the analog input is sampled for only the first bit.

Table 11.5.1 Sampling Time (in Which C2 Needs to Be Charged)

Conversion start method	Conversion speed		Sampling time for the first bit	Sampling time for the second and subsequent bits
Single mode (when sample-and-hold disabled)	Slow mode	Normal speed	27.5BCLK	13.5BCLK
		Double speed	15.5BCLK	7.5BCLK
	Fast mode	Normal speed	11.5BCLK	5.5BCLK
		Double speed	7.5BCLK	3.5BCLK
Single mode (when sample-and-hold enabled)	Slow mode	Normal speed	27.5BCLK	–
		Double speed	15.5BCLK	–
	Fast mode	Normal speed	11.5BCLK	–
		Double speed	7.5BCLK	–
Comparator mode	Slow mode	Normal speed	27.5BCLK	–
		Double speed	15.5BCLK	–
	Fast mode	Normal speed	11.5BCLK	–
		Double speed	7.5BCLK	–

Therefore, the time in which C2 needs to be charged is found from Eq. B-1, as follows:

Sampling time (in which C2 needs to be charged) > Cin × R1 + C2(R1 + R2) - Eq. B2

Thus, the maximum value of R1 can be obtained as a criterion from the equation below. Note, however, that for single mode (when sample-and-hold is disabled), the sampling time for the second and subsequent bits (C2 charging time) must be applied.

$$R1 < \frac{\text{C2 charging time - C2} \times \text{R2}}{\text{Cin + C2}}$$

CHAPTER 12

SERIAL INTERFACE

12.1 Outline of Serial Interface

The 32176 contains a total of four serial interface channels, SIO0–SIO3. Channels SIO0 and SIO1 can be selected between CSIO mode (clock-synchronous serial interface) and UART mode (clock-asynchronous serial interface). Channels SIO2 and SIO3 are UART mode only.

- **CSIO mode (clock-synchronous serial interface)**

  Communication is performed synchronously with a transfer clock, using the same clock on both transmit and receive sides. The transfer data is 8 bits long (fixed).

- **UART mode (clock-asynchronous serial interface)**

  Communication is performed at any transfer rate in any transfer data format. The transfer data length can be selected from 7, 8 and 9 bits.

  Channels SIO0–SIO3 each have a transmit DMA transfer and a receive DMA transfer request. These serial interfaces, when combined with the internal DMA Controller (DMAC), allow serial communication to be performed at high speed, as well as reduce the data communication load of the CPU.

Serial interface is outlined below.

Table 12.1.1 Outline of Serial Interface

Item	Description
Number of channels	CSIO mode/UART mode : 2 channels (SIO0, SIO1) UART only : 2 channels (SIO2, SIO3)
Clock	During CSIO mode : Internal clock or external clock as selected (Note 1), clock polarity can be selected During UART mode : Internal clock only
Transfer mode	Transmit half-duplex, receive half-duplex, transmit/receive full-duplex
BRG count source (when internal clock selected)	f(BCLK), f(BCLK)/8, f(BCLK)/32, f(BLCK)/256 (Note 2) f(BCLK): Peripheral clock operating frequency
Data format	CSIO mode : Data length = 8 bits (fixed) Order of transfer = LSB first (fixed) UART mode : Start bit = 1 bit Character length = 7, 8 or 9 bits Parity bit = Added (odd, even) or not added Stop bit = 1 or 2 bits Order of transfer = LSB first (fixed)
Baud rate	CSIO mode : 152 bits/sec to 2 Mbits/sec (when f(BCLK) = 20 MHz) UART mode : 19 bits/sec to 1.25 Mbits/sec (when f(BCLK) = 20 MHz)
Error detection	CSIO mode : Overrun error only UART mode : Overrun, parity and framing errors (Occurrence of any of these errors is indicated by an error sum bit)
Fixed period clock output function	When using SIO0 and SIO1 as UART, this function outputs a divided-by-2 BRG clock from the SCLK pin.

Note 1: The maximum input frequency of an external clock during CSIO mode is f(BCLK)/16.
Note 2: If f(BCLK) is selected as the count source, the BRG set value is subject to limitations.

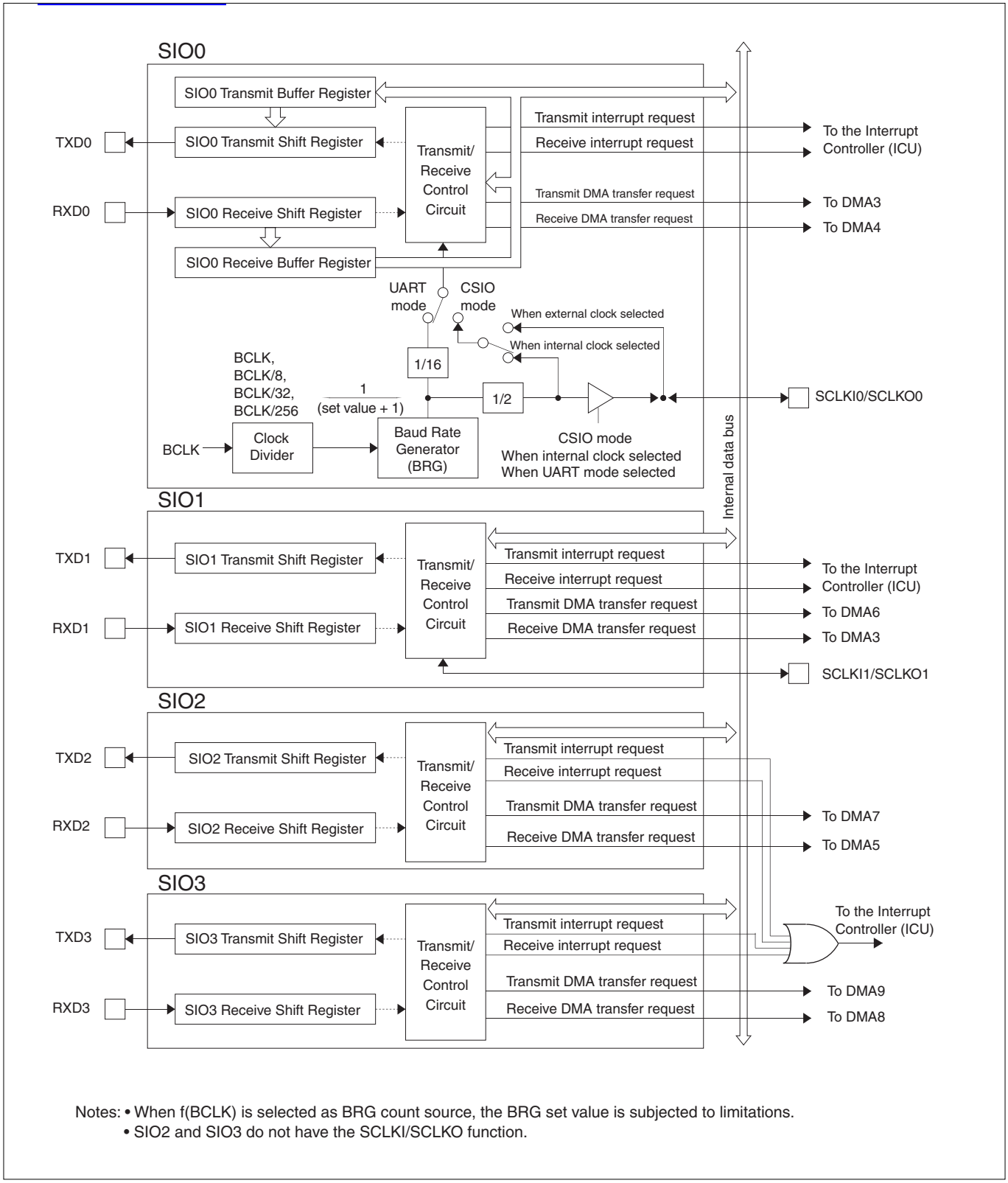
Table 12.1.2 Interrupt Generation Functions of Serial Interface

Serial Interface Interrupt Request Source	ICU Interrupt Sources
SIO0 transmit buffer empty or transmission finished	SIO0 transmit interrupt
SIO0 reception finished or receive error	SIO0 receive interrupt
SIO1 transmit buffer empty or transmission finished	SIO1 transmit interrupt
SIO1 reception finished or receive error	SIO1 receive interrupt
SIO2 transmit buffer empty or transmission finished	SIO2,3 transmit/receive interrupt (group interrupt)
SIO2 reception finished or receive error	SIO2,3 transmit/receive interrupt (group interrupt)
SIO3 transmit buffer empty or transmission finished	SIO2,3 transmit/receive interrupt (group interrupt)
SIO3 reception finished or receive error	SIO2,3 transmit/receive interrupt (group interrupt)

Note: • The transmission-finished interrupt is effective when the internal clock is selected in UART or CSIO mode.

Table 12.1.3 DMA Transfer Request Generation Functions of Serial Interface

Serial Interface DMA Transfer Request	DMAC Input Channels
SIO0 transmit buffer empty	DMA3
SIO0 reception finished	DMA4
SIO1 transmit buffer empty	DMA6
SIO1 reception finished	DMA3
SIO2 transmit buffer empty	DMA7
SIO2 reception finished	DMA5
SIO3 transmit buffer empty	DMA9
SIO3 reception finished	DMA8



Notes: • When f(BCLK) is selected as BRG count source, the BRG set value is subjected to limitations.
• SIO2 and SIO3 do not have the SCLKI/SCLKO function.

Figure 12.1.1 Block Diagram of SIO0–SIO3

12.2 Serial Interface Related Registers

Shown below is a serial interface related register map.

Serial Interface Related Register Map

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 0100	SIO23 Interrupt Request Status Register (SI23STAT)	SIO03 Interrupt Request Mask Register (SI03MASK)	12-9 12-10
H'0080 0102	SIO03 Interrupt Source Select Register (SI03SEL)	(Use inhibited area)	12-11
|	(Use inhibited area)		
H'0080 0110	SIO0 Transmit Control Register (S0TCNT)	SIO0 Transmit/Receive Mode Register (S0MOD)	12-13 12-14
H'0080 0112	SIO0 Transmit Buffer Register (S0TXB)		12-17
H'0080 0114	SIO0 Receive Buffer Register (S0RXB)		12-18
H'0080 0116	SIO0 Receive Control Register (S0RCNT)	SIO0 Baud Rate Register (S0BAUR)	12-19 12-21
H'0080 0118	SIO0 Special Mode Register (S0SMOD)	(Use inhibited area)	12-23
|	(Use inhibited area)		
H'0080 0120	SIO1 Transmit Control Register (S1TCNT)	SIO1 Transmit/Receive Mode Register (S1MOD)	12-13 12-14
H'0080 0122	SIO1 Transmit Buffer Register (S1TXB)		12-17
H'0080 0124	SIO1 Receive Buffer Register (S1RXB)		12-18
H'0080 0126	SIO1 Receive Control Register (S1RCNT)	SIO1 Baud Rate Register (S1BAUR)	12-19 12-21
H'0080 0128	SIO1 Special Mode Register (S1SMOD)	(Use inhibited area)	12-23
|	(Use inhibited area)		
H'0080 0130	SIO2 Transmit Control Register (S2TCNT)	SIO2 Transmit/Receive Mode Register (S2MOD)	12-13 12-14
H'0080 0132	SIO2 Transmit Buffer Register (S2TXB)		12-17
H'0080 0134	SIO2 Receive Buffer Register (S2RXB)		12-18
H'0080 0136	SIO2 Receive Control Register (S2RCNT)	SIO2 Baud Rate Register (S2BAUR)	12-19 12-21
|	(Use inhibited area)		
H'0080 0140	SIO3 Transmit Control Register (S3TCNT)	SIO3 Transmit/Receive Mode Register (S3MOD)	12-13 12-14
H'0080 0142	SIO3 Transmit Buffer Register (S3TXB)		12-17
H'0080 0144	SIO3 Receive Buffer Register (S3RXB)		12-18
H'0080 0146	SIO3 Receive Control Register (S3RCNT)	SIO3 Baud Rate Register (S3BAUR)	12-19 12-21

12.2.1 SIO Interrupt Related Registers

The SIO interrupt related registers are used to control the interrupt request signals output from SIO to the Interrupt Controller (ICU), as well as select the source of each interrupt request.

(1) Interrupt request status bit

This status bit is used to determine whether an interrupt is requested. When an interrupt request occurs, this bit is set in hardware (cannot be set in software). The status bit is cleared by writing "0". Writing "1" has no effect; the bit retains the status it had before the write. Because this bit is unaffected by the interrupt request mask bit, it can also be used to inspect the operating status of peripheral functions.
In interrupt handling, make sure that within the grouped interrupt request status, only the status bit for the interrupt request that has been serviced is cleared. If the status bit for any interrupt request that has not been serviced is cleared, the pending interrupt request is cleared simultaneously with its status bit.

(2) Interrupt request mask bit

This bit is used to disable unnecessary interrupt requests within the grouped interrupt request. Set this bit to "1" to enable interrupt requests or "0" to disable interrupt requests.

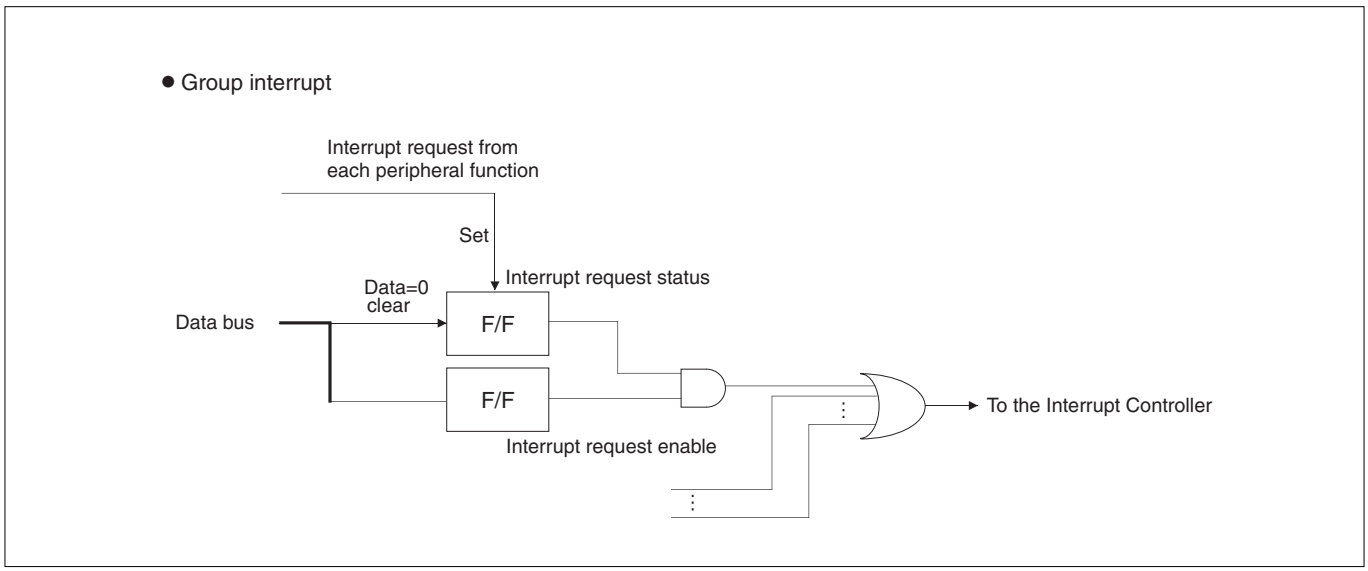
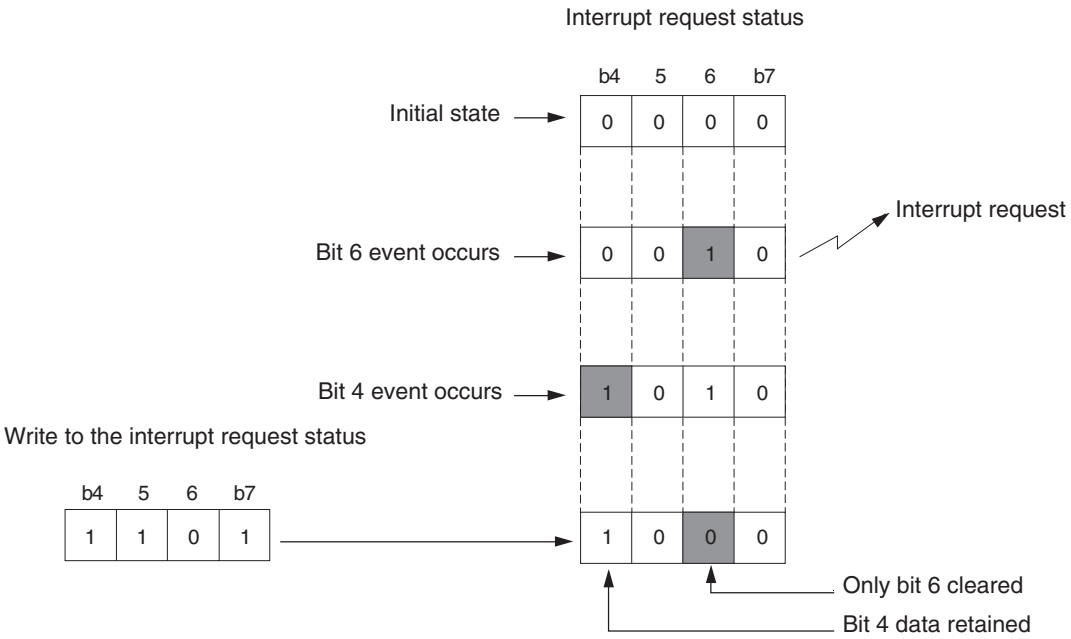


Figure 12.2.1 Interrupt Request Status and Mask Registers

● Example for clearing interrupt request status



● Program example

• To clear the Interrupt Request Status Register 0 (ISTREG) interrupt request status 1: ISTAT1 (0x02 bit)

○ ISTREG = 0xfd; /* Clear ISTAT1 (0x02 bit) only */

To clear an interrupt request status, always be sure to write 1 to all other interrupt request status bits. At this time, avoid using a logic operation like the one shown below. Because it requires three step-ISTREG read, logic operation and write, if another interrupt request occurs between the read and write, status may be inadvertently cleared.

× ISTREG &= 0xfd; /* Clear ISTAT1 (0x02 bit) only */

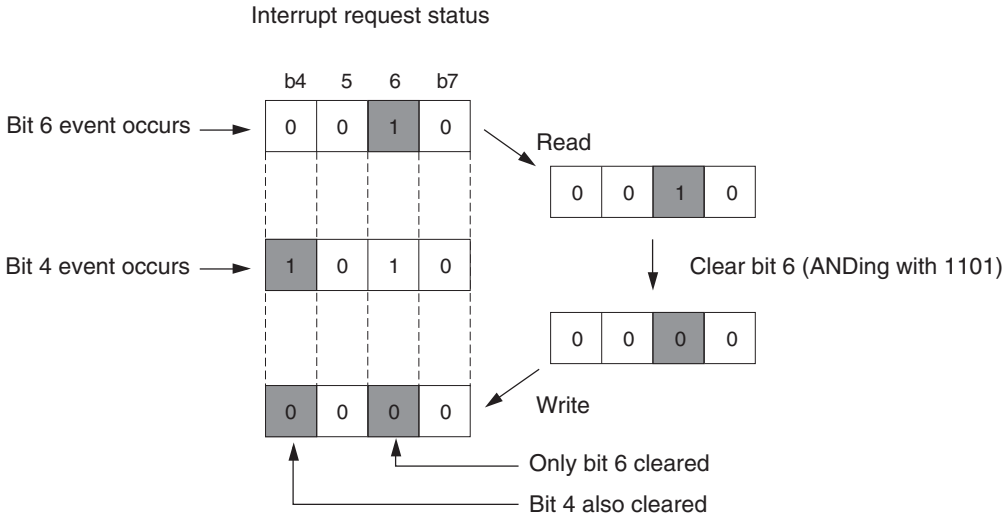


Figure 12.2.2 Example for Clearing Interrupt Request Status

(3) Selecting the source of an interrupt request

The interrupt request signals sent from each SIO to the Interrupt Controller (ICU) are classified into transmit interrupts and receive interrupts. Transmit interrupt requests can be generated when the transmit buffer is empty or transmission is finished, and the receive interrupt requests can be generated when reception is finished or an receive error is detected, as selected by the Interrupt Source Select Register (SI03SEL).

Notes: • No interrupt request signals are generated unless interrupts are generated by the SIO Interrupt Request Mask Register after enabling the TEN (Transmit Enable) bit or REN (Receive Enable) bit for the corresponding SIO.
• SIO2 and SIO3 together comprise one interrupt group.
• The transmission-finished interrupt is effective when the internal clock is selected in UART or CSIO mode.

(4) Notes on using transmit interrupts

When the interupt request is enable in SIO Interrupt Request Mask Register and the transmit buffer empty interrupt is selected in SIO Interrupt Request Source Slect Register, a transmit interrupt request is generated upon enabling the corresponding TEN (Transmit Enable) bit.

(5) About DMA transfer requests from SIO

Each SIO can generate a transmit DMA transfer and a reception-finished DMA transfer request. These DMA transfer requests can be generated by enabling each SIO's corresponding TEN (Transmit Enable) bit or REN (Receive Enable) bit. When using DMA transfers to communicate with external devices, be sure to set the DMA Controller (DMAC) before enabling the TEN or REN bit. No reception-finished DMA transfer requests are generated if a receive error occurs.

• Transmit DMA transfer request

Generated when the transmit buffer is empty and the TEN bit is enabled.

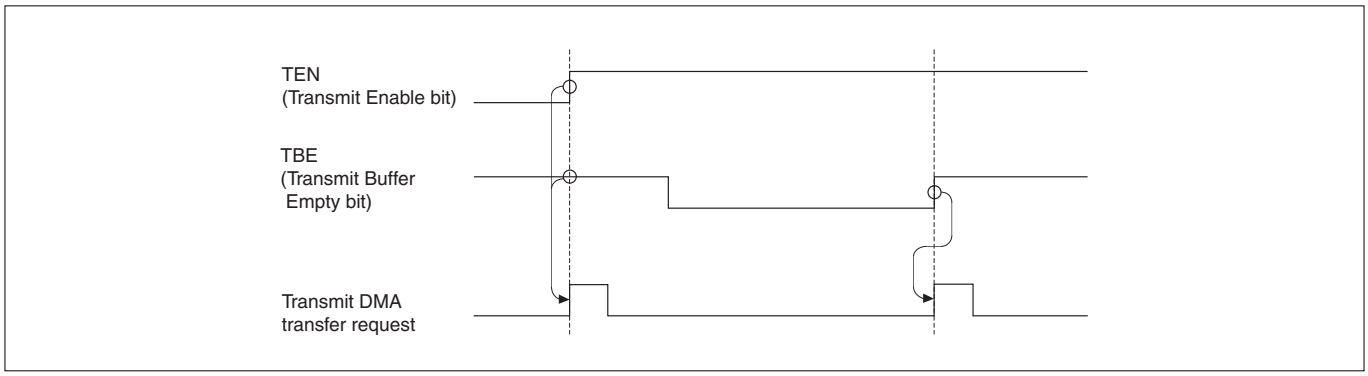


Figure 12.2.3 Transmit DMA Transfer Request

• Reception-finished DMA transfer request

A DMA transfer request is generated when the receive buffer is filled.

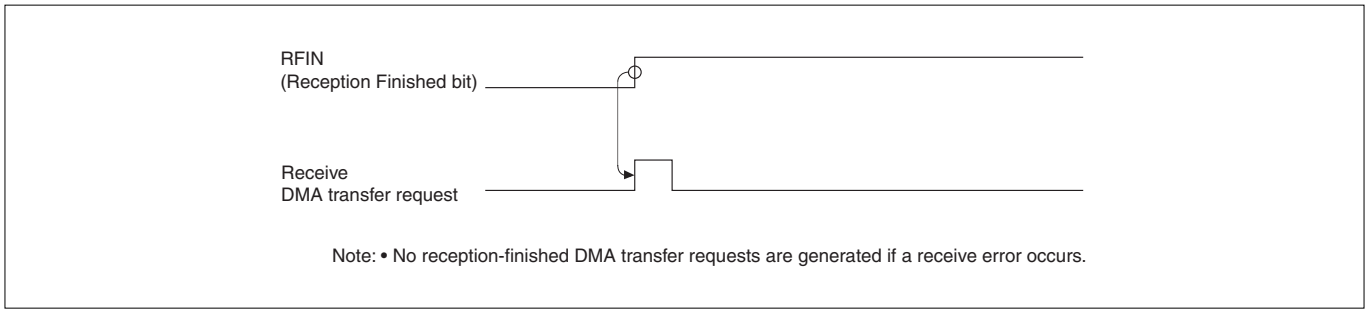


Figure 12.2.4 Reception-finished DMA Transfer Request

SIO23 Interrupt Request Status Register (SI23STAT) <Address: H'0080 0100>

b0	1	2	3	4	5	6	b7
0	0	0	0	IRQT2 0	IRQR2 0	IRQT3 0	IRQR3 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–3	No function assigned. Fix to "0".		0	0
4	IRQT2 SIO2 transmit interrupt request status bit	0: Interrupt not requested 1: Interrupt requested	R	(Note 1)
5	IRQR2 SIO2 receive interrupt request status bit	0: Interrupt not requested 1: Interrupt requested	R	(Note 1)
6	IRQT3 SIO3 transmit interrupt request status bit	0: Interrupt not requested 1: Interrupt requested	R	(Note 1)
7	IRQR3 SIO3 receive interrupt request status bit	0: Interrupt not requested 1: Interrupt requested	R	(Note 1)

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

The register indicates the transmit/receive interrupt requests from SIO2 and SIO3.

[Setting the interrupt request status bit]

This bit can only be set in hardware, and cannot be set in software.

[Clearing the interrupt request status bit]

This bit is cleared by writing "0" in software.

Note: • If the status bit is set in hardware at the same time it is cleared in software, the former has priority and the status bit is set.

When writing to the SIO Interrupt Request Status Register, make sure only the bits to be cleared are set to "0" and all other bits are set to "1". Those bits that have been set to "1" are unaffected by writing in software and retain the value they had before the write.

SIO03 Interrupt Request Mask Register (SI03MASK) <Address: H'0080 0101>

b8	9	10	11	12	13	14	b15
T0MASK	R0MASK	T1MASK	R1MASK	T2MASK	R2MASK	T3MASK	R3MASK
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	T0MASK SIO0 transmit interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W
9	R0MASK SIO0 receive interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W
10	T1MASK SIO1 transmit interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W
11	R1MASK SIO1 receive interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W
12	T2MASK SIO2 transmit interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W
13	R2MASK SIO2 receive interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W
14	T3MASK SIO3 transmit interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W
15	R3MASK SIO3 receive interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W

The register enables or disables the interrupt requests generated by each SIO. Interrupt requests from any SIO are enabled by setting its corresponding interrupt request mask bit to "1".

SIO03 Interrupt Request Source Select Register (SI03SEL) <Address: H'0080 0102>

b0	1	2	3	4	5	6	b7
IST0	IST1	IST2	IST3	ISR0	ISR1	ISR2	ISR3
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	IST0 SIO0 transmit interrupt request source select bit	0: Transmit buffer empty interrupt 1: Transmission finished interrupt	R	W
1	IST1 SIO1 transmit interrupt request source select bit	0: Transmit buffer empty interrupt 1: Transmission finished interrupt	R	W
2	IST2 SIO2 transmit interrupt request source select bit	0: Transmit buffer empty interrupt 1: Transmission finished interrupt	R	W
3	IST3 SIO3 transmit interrupt request source select bit	0: Transmit buffer empty interrupt 1: Transmission finished interrupt	R	W
4	ISR0 SIO0 receive interrupt request source select bit	0: Reception finished interrupt 1: Receive error interrupt	R	W
5	ISR1 SIO1 receive interrupt request source select bit	0: Reception finished interrupt 1: Receive error interrupt	R	W
6	ISR2 SIO2 receive interrupt request source select bit	0: Reception finished interrupt 1: Receive error interrupt	R	W
7	ISR3 SIO3 receive interrupt request source select bit	0: Reception finished interrupt 1: Receive error interrupt	R	W

The register selects the source of interrupt requests generated by each SIO when transmit or receive operation is completed.

(1) SIOn transmit interrupt source select bit

[When set to "0"]

The transmit buffer empty interrupt is selected. A transmit buffer empty interrupt request is generated when data is transferred from the transmit buffer register to the transmit shift register. Also, a transmit buffer empty interrupt request is generated when the TEN (Transmit Enable) bit is set to "1" (interrupt enabled).

[When set to "1"]

The transmission finished (transmit shift buffer empty) interrupt is selected. A transmission finished interrupt request is generated when all of the data in the transmit shift register has been transferred.

Note: • Do not select the transmission finished interrupt when an external clock is selected in CSIO mode.

(2) SIOn receive interrupt request source select bit

[When set to "0"]

The reception finished (receive buffer full) interrupt is selected. A reception finished interrupt request is also generated when a receive error (except overrun error) occurs.

[When set to "1"]

The receive error interrupt is selected. Following types of errors constitute a receive error:

• CSIO mode: Overrun error

• UART mode: Overrun, parity and framing errors

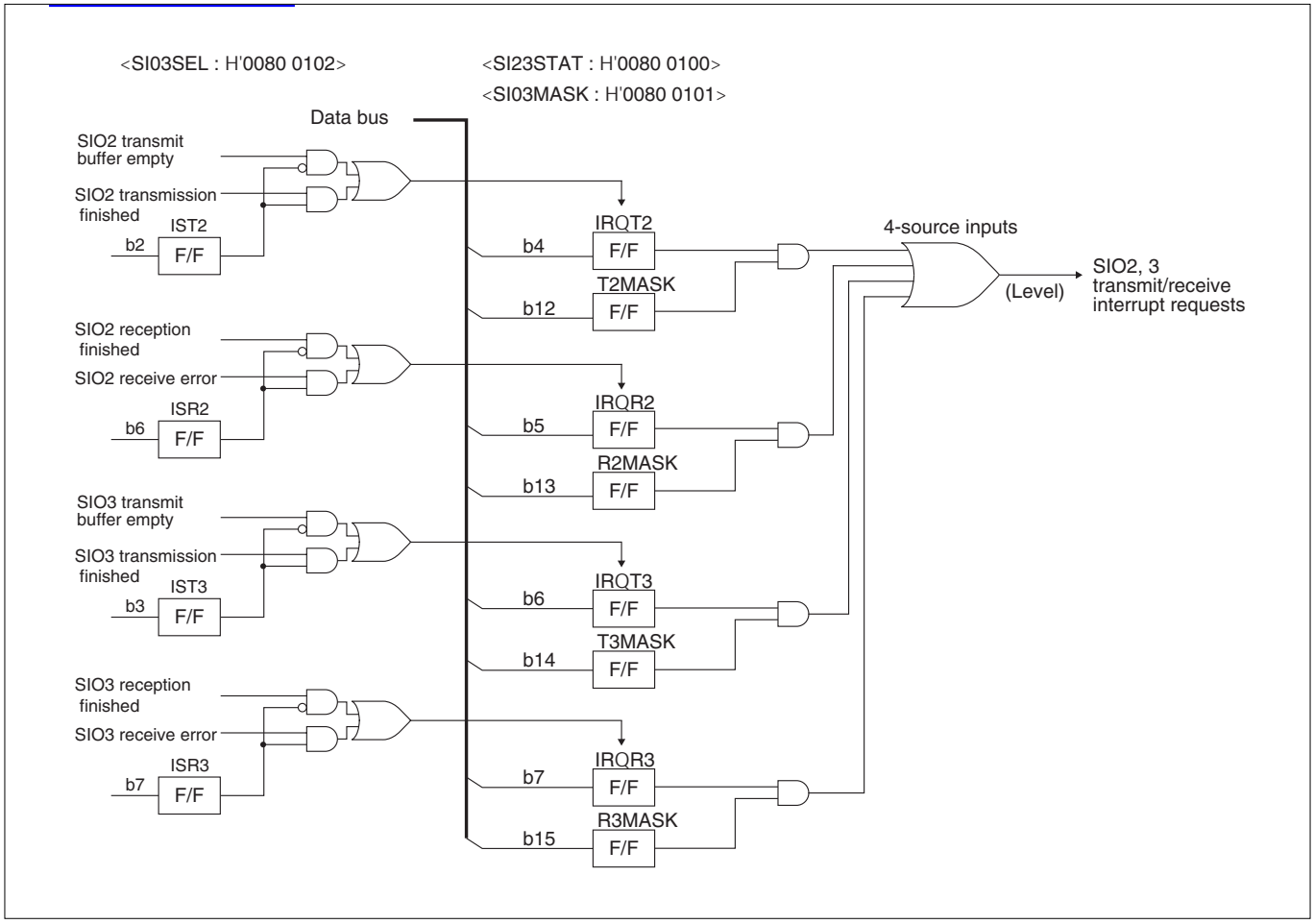


Figure 12.2.5 Block Diagram of SIO2,3 Transmit/Receive Interrupt Requests

12.2.2 SIO Transmit Control Registers

SIO0 Transmit Control Register (S0TCNT) <Address: H'0080 0110>
SIO1 Transmit Control Register (S1TCNT) <Address: H'0080 0120>
SIO2 Transmit Control Register (S2TCNT) <Address: H'0080 0130>
SIO3 Transmit Control Register (S3TCNT) <Address: H'0080 0140>

b0	1	2	3	4	5	6	b7
0	0	CDIV 0	CDIV 1	0	TSTAT 0	TBE 1	TEN 0

<Upon exiting reset: H'12>

b	Bit Name	Function	R	W
0, 1	No function assigned. Fix to "0".		0	0
2, 3	CDIV BRG count source select bit	b2 b3 0 0: Select f(BCLK) 0 1: Select f(BCLK) divided by 8 1 0: Select f(BCLK) divided by 32 1 1: Select f(BCLK) divided by 256	R	W
4	No function assigned. Fix to "0".		0	0
5	TSTAT Transmit status bit	0:Transmission stopped and no data in transmit buffer register 1:Transmitting now or data present in transmit buffer register	R	–
6	TBE Transmit buffer empty bit	0:Data present in transmit buffer register 1: No data in transmit buffer register	R	–
7	TEN Transmit enable bit	0: Disable transmission 1: Enable transmission	R	W

(1) CDIV (baud rate generator count source select) bits (Bits 2–3)

These bits select the count source for the Baud Rate Generator (BRG).

Note: • When using internal clock (the internal clock CSIO mode) and selecting f(BCLK) as a BRG count source, set the BRG value for the transfer rate not to exceed 2Mbits / second.

(2) TSTAT (Transmit Status) bit (Bit 5)

[Set condition]

This bit is set to "1" by a write to the transmit buffer register while transmission is enabled.

[Clear condition]

This bit is cleared to "0" when transmission is idle (no data in the transmit shift register) and no data exists in the transmit buffer register. This bit is also cleared by clearing the transmit enable bit.

(3) TBE (Transmit Buffer Empty) bit (Bit 6)

[Set condition]

This bit is set to "1" when data is transferred from the transmit buffer register to the transmit shift register and the transmit buffer register is thereby emptied. This bit is also set by clearing the transmit enable bit to "0".

[Clear condition]

This bit is cleared to "0" by writing data to the lower byte of the transmit buffer register while transmission is enabled (TEN = "1").

(4) TEN (Transmit Enable) bit (Bit 7)

Transmission is enabled by setting this bit to "1" and disabled by clearing this bit to "0". If this bit is cleared to "0" while transmitting data, the transmit operation stops.

12.2.3 SIO Transmit/Receive Mode Registers

SIO0 Transmit/Receive Mode Register (S0MOD) <Address: H'0080 0111>
SIO1 Transmit/Receive Mode Register (S1MOD) <Address: H'0080 0121>
SIO2 Transmit/Receive Mode Register (S2MOD) <Address: H'0080 0131>
SIO3 Transmit/Receive Mode Register (S3MOD) <Address: H'0080 0141>

b8	9	10	11	12	13	14	b15
SMOD			CKS	STB	PSEL	PEN	SEN
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	SMOD Serial interface mode select bit (Note 1)	b8 b9 b10 0 0 0 : 7-bit UART 0 0 1 : 8-bit UART 0 1 0 : 9-bit UART 0 1 1 : 9-bit UART 1 0 0 : 8-bit clock-synchronous serial interface 1 0 1 : 8-bit clock-synchronous serial interface 1 1 0 : 8-bit clock-synchronous serial interface 1 1 1 : 8-bit clock-synchronous serial interface	R	W
11	CKS Internal/external clock select bit	0: Internal clock 1: External clock (Note 4)	R	W (Note 2)
12	STB Stop bit length select bit, UART mode only	0: One stop bit 1: Two stop bits	R	W (Note 3)
13	PSEL Odd/even parity select bit, UART mode only	0: Odd parity 1: Even parity	R	W (Note 3)
14	PEN Parity enable bit, UART mode only	0: Disable parity 1: Enable parity	R	W (Note 3)
15	SEN Sleep select bit, UART mode only	0: Disable sleep function 1: Enable sleep function	R	W (Note 3)

Note 1: For SIO2 and 3, bit 8 is fixed to "0" in hardware. This bit cannot be set to "1" in software (to select clock-synchronous serial interface).

Note 2: Has no effect when UART mode selected.

Note 3: Bits 12–15 have no effect during clock-synchronous mode.

Note 4:The maximum frequency of the SCLKI pin input clock is f(BCLK)/16 when external clock is selected in CSIO mode.

The SIO Transmit/Receive Mode Registers consist of bits to set the serial interface operation mode, data format and the functions used during communication.

The SIO Transmit/Receive Mode Registers must always be set before the serial interface starts operating. To change register settings after the serial interface starts sending or receiving data, first confirm that transmit and receive operations have finished and then disable transmit/receive operations (by clearing the SIO Transmit Control Register transmit enable bit and SIO Receive Control Register receive enable bit to "0") before making changes.

(1) SMOD (Serial Interface Mode Select) bits (Bits 8–10)

These bits select the operation mode of serial interface.

(2) CKS (Internal/External Clock Select) bit (Bit 11)

This bit is effective when CSIO mode is selected. Setting this bit has no effect when UART mode is selected, in which case the serial interface is clocked by the internal clock.

(3) STB (Stop Bit Length Select) bit (Bit 12)

This bit is effective during UART mode. Use this bit to select the stop bit length that indicates the end of data to transmit. Setting this bit to "0" selects one stop bit, and setting this bit to "1" selects two stop bits.
During clock-synchronous mode, the content of this bit has no effect.

(4) PSEL (Odd/Even Parity Select) bit (Bit 13)

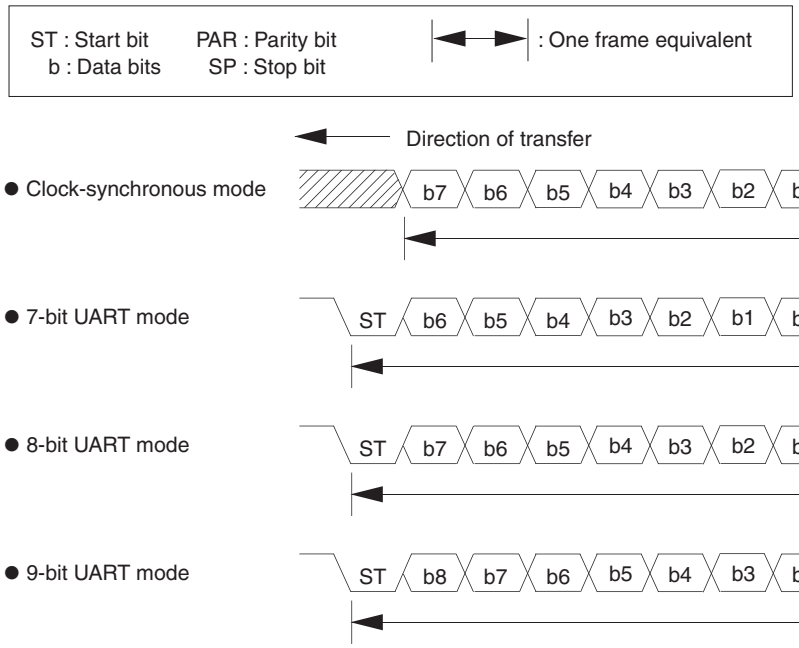
This bit is effective during UART mode. When parity is enabled (bit 14 = "1"), use this bit to select the parity attribute (whether odd or even). Setting this bit to "0" selects an odd parity, and setting this bit to "1" selects an even parity.
When parity is disabled (bit 14 = "0") or during clock-synchronous mode, the content of this bit has no effect.

(5) PEN (Parity Enable) bit (Bit 14)

This bit is effective during UART mode. When this bit is set to "1", a parity bit is added immediately after the data bits of the transmit data, and the received data is checked for parity.
The parity bit added to the transmit data is automatically determined to be "0" or "1" so that the attribute (odd/even) derived by adding the number of 1's in data bits and the content of the parity bit agrees with one that was selected with the odd/even parity select bit (bit 13).
Figure 12.2.6 shows an example of a data format when parity is enabled.

(6) SEN (Sleep Select) bit (Bit 15)

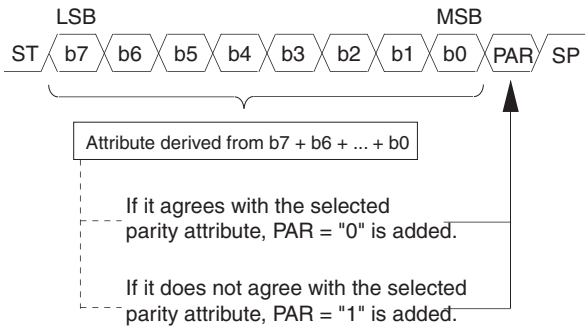
This bit is effective during UART mode. If the sleep function is enabled by setting this bit to "1", data is latched into the UART Receive Buffer Register only when the most significant bit (MSB) of the received data is "1".



Note 1: Whether or not to add a parity bit is selectable.
Note 2: The stop bit can be chosen to be one bit or two bits long.

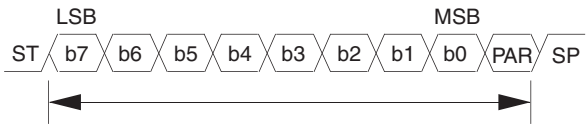
● When transmitting

If the attribute (odd/even) represented by the number of 1's in data bits agrees with the selected parity attribute, a parity bit "0" is added. If the attribute (odd/even) represented by the number of 1's in data bits does not agree with the selected parity attribute, a parity bit "1" is added.



● When receiving

The received data is checked to see if the number of 1's included in its data and parity bits agrees with the parity attribute (known as parity check).



If the result of b7 + b6 + ... + b0 + PAR does not agree with the selected parity attribute, a parity error is assumed

Notes : • Shown above is an example of a data format in 8-bit UART mode.
• The data bit numbers (bn) above indicate bit numbers in a data list, and not the register bit numbers (bn).

Figure 12.2.6 Data Format When Parity is Enabled

12.2.4 SIO Transmit Buffer Registers

SIO0 Transmit Buffer Register (S0TXB) <Address: H'0080 0112>
SIO1 Transmit Buffer Register (S1TXB) <Address: H'0080 0122>
SIO2 Transmit Buffer Register (S2TXB) <Address: H'0080 0132>
SIO3 Transmit Buffer Register (S3TXB) <Address: H'0080 0142>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
							TDATA								
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–6	No function assigned. Fix to "0".		?	0
7–15	TDATA Transmit data	Transmit data is set in these bits.	?	W

The SIO Transmit Buffer Registers are used to set transmit data. These registers are a write-only register, and the contents of these registers cannot be read out. Data must be LSB-aligned when set in these registers. Therefore, write transmit data to bits 9–15 for the 7-bit data format (UART mode only), bits 8–15 for the 8-bit data format, or bits 7–15 for the 9-bit data format (UART mode only).

Before setting transmit data in these registers, enable the Transmit Control Register TEN (Transmit Enable) bit by setting it to "1". Writing data to these registers while the TEN bit is disabled (cleared to "0") has no effect. When data is written to the SIO Transmit Buffer Register while transmission is enabled, the data is transferred from that register to the SIO Transmit Shift Register, upon which the serial interface starts sending data.

Note: For the 7-bit and 8-bit data formats, the register can be accessed bytewise.

12.2.5 SIO Receive Buffer Registers

SIO0 Receive Buffer Register (S0RXB) <Address: H'0080 0114>
SIO1 Receive Buffer Register (S1RXB) <Address: H'0080 0124>
SIO2 Receive Buffer Register (S2RXB) <Address: H'0080 0134>
SIO3 Receive Buffer Register (S3RXB) <Address: H'0080 0144>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
							RDATA								
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–6	No function assigned.		0	–
8–15	RDATA Received data	Received data is stored in these bits.	R	–

The SIO Receive Buffer Registers are used to store the received data. When the serial interface has finished receiving data, the content of the SIO Receive Shift Register is transferred to the SIO Receive Buffer Register. These registers are a read-only register.
For the 7-bit data format (UART mode only), data is set in bits 9–15, with bits 8 and 7 always set to "0". For the 8-bit data format, data is set in bits 8–15, with bit 7 always set to "0".
When reading the content of the SIO Receive Buffer Register after reception is completed, if the serial interface finishes receiving the next data before the previous data is not read out, an overrun error occurs and the subsequent received data are not transferred to the Receive Buffer Register.
To restart normal receive operation, clear the Receive Control Register REN (Receive Enable) bit to "0".

Note: For the 7-bit and 8-bit data formats, the register can be accessed bytewise.

12.2.6 SIO Receive Control Registers

SIO0 Receive Control Register (S0RCNT) <Address: H'0080 0116>
SIO1 Receive Control Register (S1RCNT) <Address: H'0080 0126>
SIO2 Receive Control Register (S2RCNT) <Address: H'0080 0136>
SIO3 Receive Control Register (S3RCNT) <Address: H'0080 0146>

b0	1	2	3	4	5	6	b7
0	RSTAT 0	RFIN 0	REN 0	OVR 0	PTY 0	FLM 0	ERS 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0	No function assigned. Fix to "0".		0	0
1	RSTAT Receive status bit	0: Reception stopped 1: Reception in progress	R	–
2	RFIN Reception finished bit	0: No data in receive buffer register 1: Data present in receive buffer register	R	–
3	REN Receive enable bit	0: Disable reception 1: Enable reception	R	W
4	OVR Overrun error bit	0: No overrun error 1: Overrun error occurred	R	–
5	PTY Parity error bit, UART mode only	0: No parity error 1: Parity error occurred	R	–
6	FLM Framing error bit, UART mode only	0: No framing error 1: Framing error occurred	R	–
7	ERS Error sum bit	0: No error 1: Error occurred	R	–

(1) RSTAT (Receive Status) bit (Bit 1)

[Set condition]

This bit is set to "1" by a start of receive operation. When this bit = "1", the serial interface is receiving data.

[Clear condition]

This bit is cleared to "0" upon completion of receive operation or by clearing the REN (Receive Enable) bit.

(2) RFIN (Reception Finished) bit (Bit 2)

[Set condition]

This bit is set to "1" when all data bits have been received in the Receive Shift Register and whose content is transferred to the Receive Buffer Register.

[Clear condition]

This bit is cleared to "0" by reading out the lower byte of the Receive Buffer Register or by clearing the REN (Receive Enable) bit. However, if an overrun error occurs, this bit cannot be cleared by reading out the lower byte of the Receive Buffer Register. In this case, clear REN (Receive Enable) bit to "0".

(3) REN (Receive Enable) bit (Bit 3)

Reception is enabled by setting this bit to "1", and is disabled by clearing this bit to "0", in which case the receiver unit is initialized. Accordingly, the receive status and reception finished flags, as well as the overrun error, framing error, parity error and error sum flags all are cleared.

The receive operation stops if the Receive Enable bit is cleared to "0" while receiving data.

(4) OVR (Overrun Error) bit (Bit 4)

[Set condition]

This bit is set to "1" when all bits of the next received data have been set in the Receive Shift Register while the Receive Buffer Register still contains the previous received data. In this case, the received data is not stored in the Receive Buffer Register. Although receive operation continues even when the overrun error flag = "1", the received data is not stored in the Receive Buffer Register. This error bit must be cleared before normal reception can be restarted.

[Clear condition]

This bit is cleared to "0" by only clearing the REN (Receive Enable) bit.

(5) PTY (Parity Error) bit (Bit 5)

This bit is effective in only UART mode. It is fixed to "0" during CSIO mode.

[Set condition]

The PTY (Parity Error) bit is set to "1" when the SIO Transmit/Receive Mode Register PEN (Parity Enable/Disable) bit is enabled and the parity (even or odd) of the received data does not agree with one that was set by the said register's PSEL (Parity Select) bit.

[Clear condition]

The PTY bit is cleared to "0" by reading out the lower byte of the SIO Receive Buffer Register or by clearing the SIO Receive Control Register REN (Receive Enable) bit. However, if an overrun error occurs, this bit cannot be cleared by reading out the lower byte of the Receive Buffer Register. In this case, clear the REN (Receive Enable) bit.

(6) FLM (Framing Error) bit (Bit 6)

This bit is effective in only UART mode. It is fixed to "0" during CSIO mode.

[Set condition]

The FLM (Framing Error) bit is set to "1" when the number of received bits does not agree with one that was set by the SIO Transmit/Receive Mode Register.

[Clear condition]

The FLM bit is cleared to "0" by reading out the lower byte of the SIO Receive Buffer Register or by clearing the SIO Receive Control Register REN (Receive Enable) bit.
However, if an overrun error occurs, this bit cannot be cleared by reading out the lower byte of the Receive Buffer Register. In this case, clear the REN (Receive Enable) bit to "0".

(7) ERS (Error Sum) bit (Bit 7)

[Set condition]

This flag is set to "1" when any of overrun, framing or parity errors is detected at completion of reception.

[Clear condition]

If the detected error was an overrun error, this flag is cleared by clearing the REN (Receive Enable) bit to "0". Otherwise, this flag is cleared by reading out the lower byte of the SIO Receive Buffer Register or by clearing the SIO Receive Control Register REN (Receive Enable) bit.

12.2.7 SIO Baud Rate Registers

SIO0 Baud Rate Register (S0BAUR) <Address: H'0080 0117>
SIO1 Baud Rate Register (S1BAUR) <Address: H'0080 0127>
SIO2 Baud Rate Register (S2BAUR) <Address: H'0080 0137>
SIO3 Baud Rate Register (S3BAUR) <Address: H'0080 0147>

b8	9	10	11	12	13	14	b15
BRG							
?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–15	BRG Baud rate divide value	Set a baud rate divide value	R	W

(1) BRG (baud rate divide value) (Bits 8–15)

The SIO Baud Rate Registers are used to set a baud rate divide value, so that the baud rate count source selected by SIO Mode Register is divided by (BRG set value + 1).
Because the BRG value initially is undefined, be sure to set the divide value before the serial interface starts operating. The value written to the BRG during transmit/receive operation takes effect in the next cycle after the BRG counter has finished counting.

When using the internal clock (to output the SCLKO signal) in CSIO mode, the serial interface divides the internal BCLK using a clock divider and then divides the resulting clock by (BRG set value + 1) and further by 2, thereby generating a transmit/receive shift clock.
When using an external clock in CSIO mode, the serial interface does not use the BRG. (Transmit/receive operations are synchronized to the externally supplied clock.)

During UART mode, the serial interface divides the internal BCLK using a clock divider and then divides the resulting clock by (BRG set value + 1) and further by 16, thereby generating a transmit/receive shift clock.
When using SIO0 or SIO1 in UART mode, set the relevant port (P84 or P87) to function as an SCLKO pin, so that a BRG output clock divided by 2 can be output from that SCLKO pin.

When using the internal clock (internally clocked CSIO mode), if f(BCLK) is selected as the BRG count source, make sure the transfer rate does not exceed 2 Mbits/second during CSIO mode.

The baud rate register set value when internal clock CSIO mode is selected can be calcurated by the following equations.

• **CSIO Mode**

$$\text{SIO Baud Rate Register Set Value} = \frac{f(BCLK)}{\text{Baud Rate} \times \text{Clock Divider Divide Value} \times 2} - 1$$

• **UART Mode**

$$\text{SIO Baud Rate Register Set Value} = \frac{f(BCLK)}{\text{Baud Rate} \times \text{Clock Divider Divide Value} \times 16} - 1$$

Clock divider divide value: selected among 1, 8, 32 and 256 by setting the SIO Transmit Control Register BRG count source select bit.

Table 12.2.1 Example Settings of the SIO Baud Rate Register (CSIO Mode)

items / Baud rate [bps]	When f(BCLK) = 16MHz			When f(BCLK) = 20MHz		
	Clock divider divide value [divided-by n]	BRG set value	Actual baud rate [bps]	Clock divider divide value [divided-by n]	BRG set value	Actual baud rate [bps]
250	256	124	250.00	256	155	250.40
500	256	62	496.03	256	77	500.80
1000	32	249	1000.00	256	38	1001.60
2500	32	99	2500.00	32	124	2500.00
5000	8	199	5000.00	8	249	5000.00
10000	8	99	10000.00	8	124	10000.00
25000	8	39	25000.00	8	49	25000.00
50000	1	159	50000.00	1	199	50000.00
100000	1	79	100000.00	1	99	100000.00
250000	1	31	250000.00	1	39	250000.00
500000	1	15	500000.00	1	19	500000.00
1000000	1	7	1000000.00	1	9	1000000.00
2000000	1	3	2000000.00	1	4	2000000.00
2500000	-	-	-	1	3	2500000.00

Notes: • This does not mean that the communication at the above baud rates is guaranteed. Careful consideration and inspection under your environment are required before use.
• Select divide-by value of clock divider in the CDIV bit of SIOn transmit control register (SnTCNT).
• Set BRG set value in the SIOn baud rate register (SnBAUR).

Table 12.2.2 Example Settings of the SIO Baud Rate Register (UART Mode)

items / Baud rate [bps]	When f(BCLK) = 16MHz				When f(BCLK) = 20MHz			
	Clock divider divide value [divided-by n]	BRG set value	A margin of error (%)	Actual baud rate [bps]	Clock divider divide value [divided-by n]	BRG set value	A margin of error (%)	Actual baud rate [bps]
300	32	103	0.16	300.48	32	129	0.16	300.48
600	32	51	0.16	600.96	32	64	0.16	600.96
1200	32	25	0.16	1201.92	32	32	-1.36	1183.71
2400	32	12	0.16	2403.85	32	15	1.73	2441.41
4800	1	207	0.16	4807.69	1	259	0.16	4807.69
9600	1	103	0.16	9615.38	1	129	0.16	9615.38
14400	1	68	0.64	14492.75	1	86	-0.22	14367.82
19200	1	51	0.16	19230.77	1	64	0.16	19230.77
38400	1	25	0.16	38461.54	1	32	-1.36	37878.79
57600	-	-	-	-	1	21	-1.36	56818.18
115200	-	-	-	-	1	10	-1.36	113636.36
128000	-	-	-	-	-	-	-	-
250000	1	3	0.00	250000.00	1	4	0.00	250000.00
500000	1	1	0.00	500000.00	-	-	-	-
625000	-	-	-	-	1	1	0.00	625000.00
1000000	1	0	0.00	1000000.00	-	-	-	-
1250000	-	-	-	-	1	0	0.00	1250000.00

Notes: • This does not mean that the communication at the above baud rates is guaranteed. Careful consideration and inspection under your environment are required before use.
• Select divide-by value of clock divider in the CDIV bit of SIOn transmit control register (SnTCNT).
• Set BRG set value in the SIOn baud rate register (SnBAUR).

12.2.8 SIO Special Mode Registers

SIO0 Special Mode Register (S0SMOD) <Address: H'0080 0118>
SIO1 Special Mode Register (S1SMOD) <Address: H'0080 0128>

b0	1	2	3	4	5	6	b7
0	0	0	0	0	0	0	CKPOL 0

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–6	No function assigned.		0	0
7	CKPOL Transmit/receive clock polarity select bit	0: Transmit data is output at a fall of SCLK receive data is latched in at a rise of SCLK 1: Transmit data is output at a rise of SCLK receive data is latched in at a fall of SCLK	R	W

(1) CKPOL(transmit/receive clock polarity select) bit (Bit 7)

This bit selects the polarity of the transmit/receive clock when in CSIO mode.

When the CKPOL bit is set to "0", data is output from the TXD pin synchronously with a falling edge of SCLK, and data is taken in from the RXD pin synchronously with a rising edge of SCLK.
When the CKPOL bit is set to "1", data is output from the TXD pin synchronously with a rising edge of SCLK, and data is taken in from the RXD pin synchronously with a falling edge of SCLK.

Notes • Do not rewrite the clock polarity select bit when the transmit enable bit or receive enable bit is enabled.

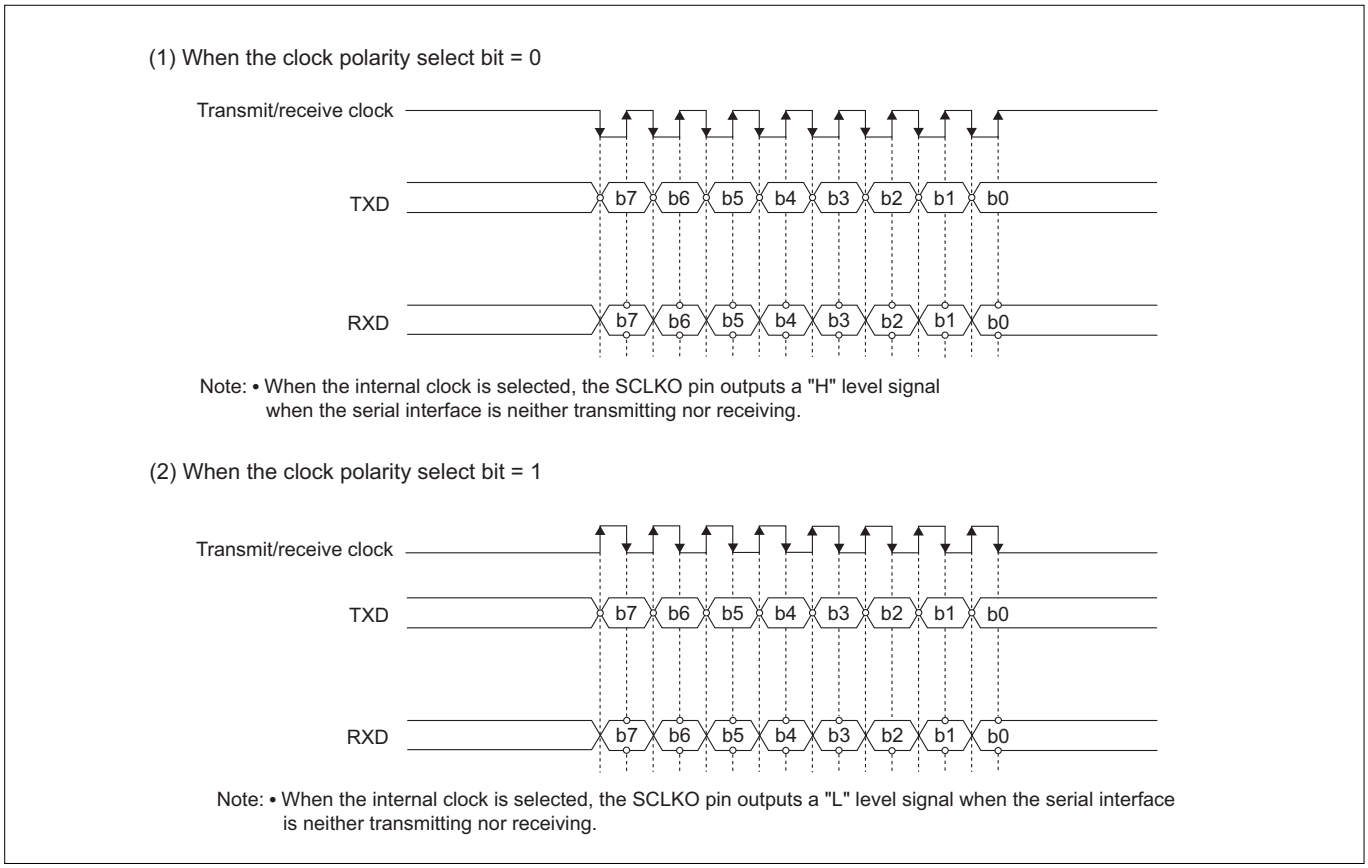


Figure 12.2.7 Selecting the Transmit/receive Clock Polarity

12.3 Transmit Operation in CSIO Mode

12.3.1 Setting the CSIO Baud Rate

The baud rate (data transfer rate) in CSIO mode is determined by a transmit/receive shift clock. The clock source from which a transmit/receive shift clock derives is selected from the internal clock f(BCLK) or external clock. The CKS (Internal/External Clock Select) bit (SIO Transmit/Receive Mode Register bit 11) is used to select the clock source.
The equation used to calculate the transmit/receive baud rate differs depending on whether an internal or external clock is selected.

(1) When internal clock is selected in CSIO mode

When the internal clock is selected, f(BCLK) is divided by a clock divider before being supplied to the Baud Rate Generator (BRG).

The clock divider's divide-by value is selected from 1, 8, 32 or 256 by using the CDIV (baud rate generator count source select) bits (Transmit Control Register bits 2–3).

The Baud Rate Generator divides the clock divider output by (baud rate register set value + 1) and further by 2, thus generating a transmit/receive shift clock.

When the internal clock is selected in CSIO mode, the baud rate is calculated using the equation below.

$$\text{Baud rate [bps]} = \frac{\text{f(BCLK)}}{\text{Clock divider's divide-by value} \times (\text{baud rate register set value} + 1) \times 2}$$

f(BCLK): Peripheral clock operating frequency
Baud rate register set value = H'00 to H'FF (Note 1)
Clock divider's divide-by value = 1, 8, 32 or 256

Note 1: If divide-by-1 (i.e., f(BCLK) itself) is selected as the baud rate generator count source, use caution when setting the baud rate register so that the transfer rate will not exceed 2 Mbps.

(2) When external clock is selected in CSIO mode

In this case, the Baud Rate Generator is not used, and the input clock from the SCLKI pin serves directly as a transmit/receive shift clock for CSIO.
The maximum frequency of the SCLKI pin input clock is f(BCLK)/16.

$$\text{Baud rate [bps]} = \text{SCLKI pin input clock}$$

12.3.2 Initializing CSIO Transmission

To transmit data in CSIO mode, initialize the serial interface following the procedure described below.

(1) Setting SIO Special Mode Register

- Select the clock polarity in CSIO mode.

(2) Setting SIO Transmit/Receive Mode Register

- Set the register to CSIO mode.
- Select the internal or an external clock.

(3) Setting SIO Transmit Control Register

- Select the clock divider's divide-by ratio (when internal clock selected).

(4) Setting SIO Baud Rate Register

When the internal clock is selected, set a baud rate generator value. (See Section 12.3.1, "Setting the CSIO Baud Rate.")

(5) Setting SIO interrupt related registers

- Select the source of transmit interrupt request (transmit buffer empty or transmission finished) (SIO Interrupt Request Source Select Register).
- Enable or disable transmit interrupt requests (SIO Interrupt Request Mask Register).

Note: • Transmission finished interrupt requests are effective only when the internal clock is selected.

(6) Setting the Interrupt Controller (SIO Transmit Interrupt Control Register)

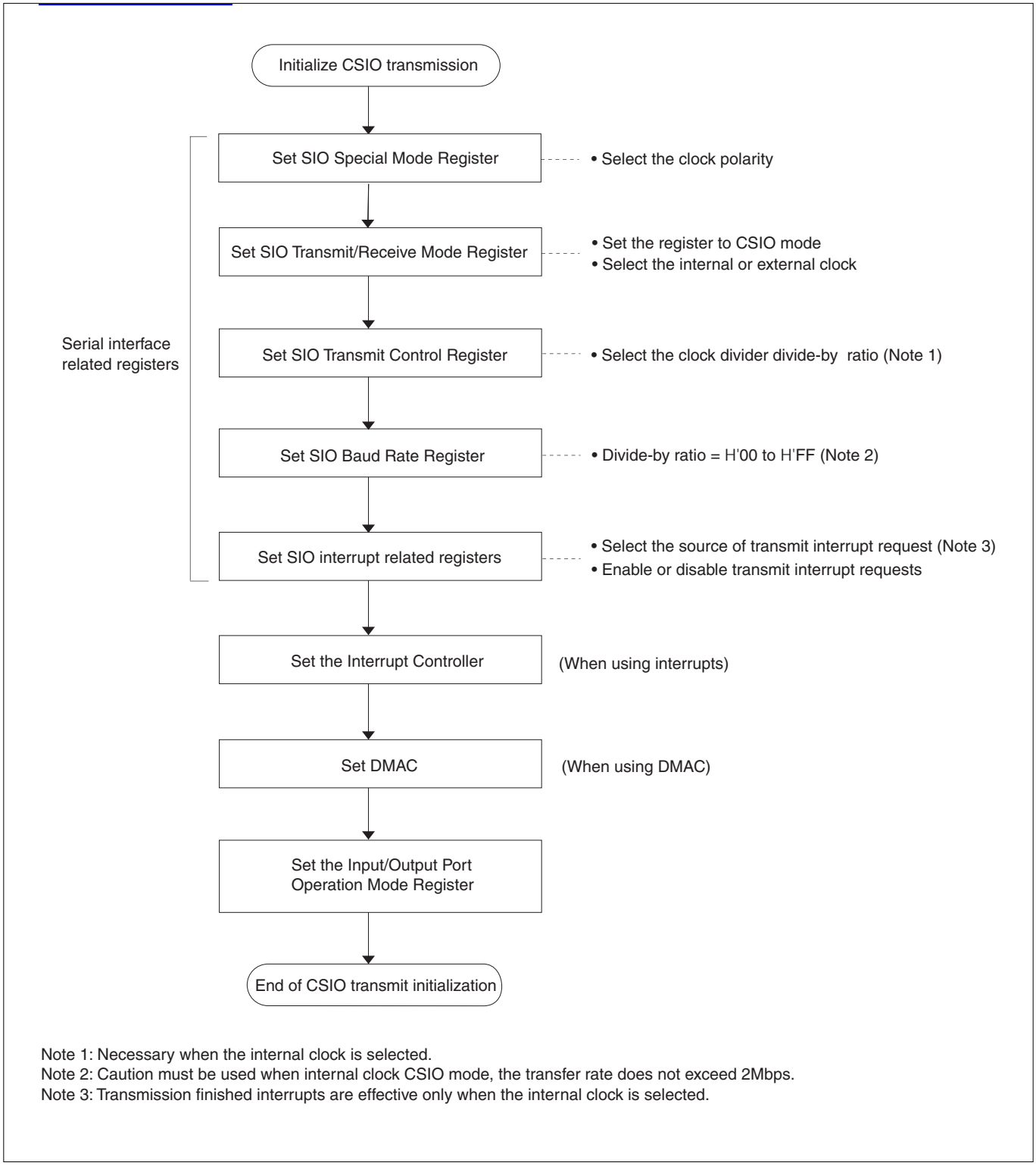
To use transmit interrupts, set their priority levels.

(7) Setting DMAC

To issue DMA transfer requests to the internal DMAC when the transmit buffer is empty, set up the DMAC. (See Chapter 9, "DMAC.")

(8) Selecting pin functions

Because the serial interface related pins serve dual purposes, set the pin functions for use as SIO pins or input/output ports. (See Chapter 8, "Input/Output Ports and Pin Functions.")



Note 1: Necessary when the internal clock is selected.
Note 2: Caution must be used when internal clock CSIO mode, the transfer rate does not exceed 2Mbps.
Note 3: Transmission finished interrupts are effective only when the internal clock is selected.

Figure 12.3.1 Procedure for Initializing CSIO Transmission

12.3.3 Starting CSIO Transmission

The serial interface starts a transmit operation when all of the following conditions are met after being initialized.

(1) Transmit conditions when CSIO mode internal clock is selected

- The SIO Transmit Control Register transmit enable bit is set to "1".
- Transmit data (8 bits) is written to the lower byte of the SIO Transmit Buffer Register (transmit buffer empty bit = "0")

(2) Transmit conditions when CSIO mode external clock is selected

- The SIO Transmit Control Register transmit enable bit is set to "1".
- Transmit data is written to the lower byte of the SIO Transmit Buffer Register (transmit buffer empty bit = "0")
- When the clock polarity select bit = "0", the transmit clock input at the SCLKI pin goes low; when the clock polarity select bit = "1", the transmit clock input at the SCLKI pin goes high.
- A falling edge of transmit clock on the SCLKI pin is detected.
- While the transmit enable bit is cleared to "0", writes to the transmit buffer register are invalid. Always set the transmit enable bit to "1" before writing to the transmit buffer register.
- When the internal clock is selected, a write to the lower byte of the transmit buffer register in above triggers transmission to start.
- The transmit status bit is set to "1" at the time data is set in the lower byte of the SIO Transmit Buffer Register.

When transmission starts, the serial interface sends data following the procedure described below.

- Transfer the content of the SIO Transmit Buffer Register to the SIO Transmit Shift Register.
- Set the transmit buffer empty bit to "1" (Note 1).
- Start sending data synchronously with the shift clock beginning with the LSB.

Note 1: A transmit interrupt request can be generated for reasons that the transmit buffer is empty or transmission has finished. Also, a DMA transfer request can be generated when the transmit buffer is empty. No DMA transfer requests can be generated for reasons that transmission has finished.

12.3.4 Successive CSIO Transmission

Once data has been transferred from the transmit buffer register to the transmit shift register, the next data can be written to the transmit buffer register even when the serial interface has not finished sending the previous data. If the next data is written to the transmit buffer register before transmission has finished, the previous and the next data are transmitted successively.

Check the SIO Transmit Control Register's transmit buffer empty flag to see if data has been transferred from the transmit buffer register to the transmit shift register.

12.3.5 Processing at End of CSIO Transmission

When data transmission finishes, the following operation is automatically performed in hardware.

(1) When not transmitting successively

- The transmit status bit is cleared to "0".

(2) When transmitting successively

- When transmission of the last data in a consecutive data train finishes, the transmit status bit is cleared to "0".

12.3.6 Transmit Interrupts

(1) Transmit buffer empty interrupt

If the transmit buffer empty interrupt was selected using the SIO Interrupt Request Source Select Register, a transmit buffer empty interrupt request is generated when data has been transferred from the transmit buffer register to the transmit shift register. A transmit buffer empty interrupt request is also generated when the TEN (Transmit Enable) bit is set to "1" (disabled → enabled) while the transmit buffer empty interrupt has been enabled.

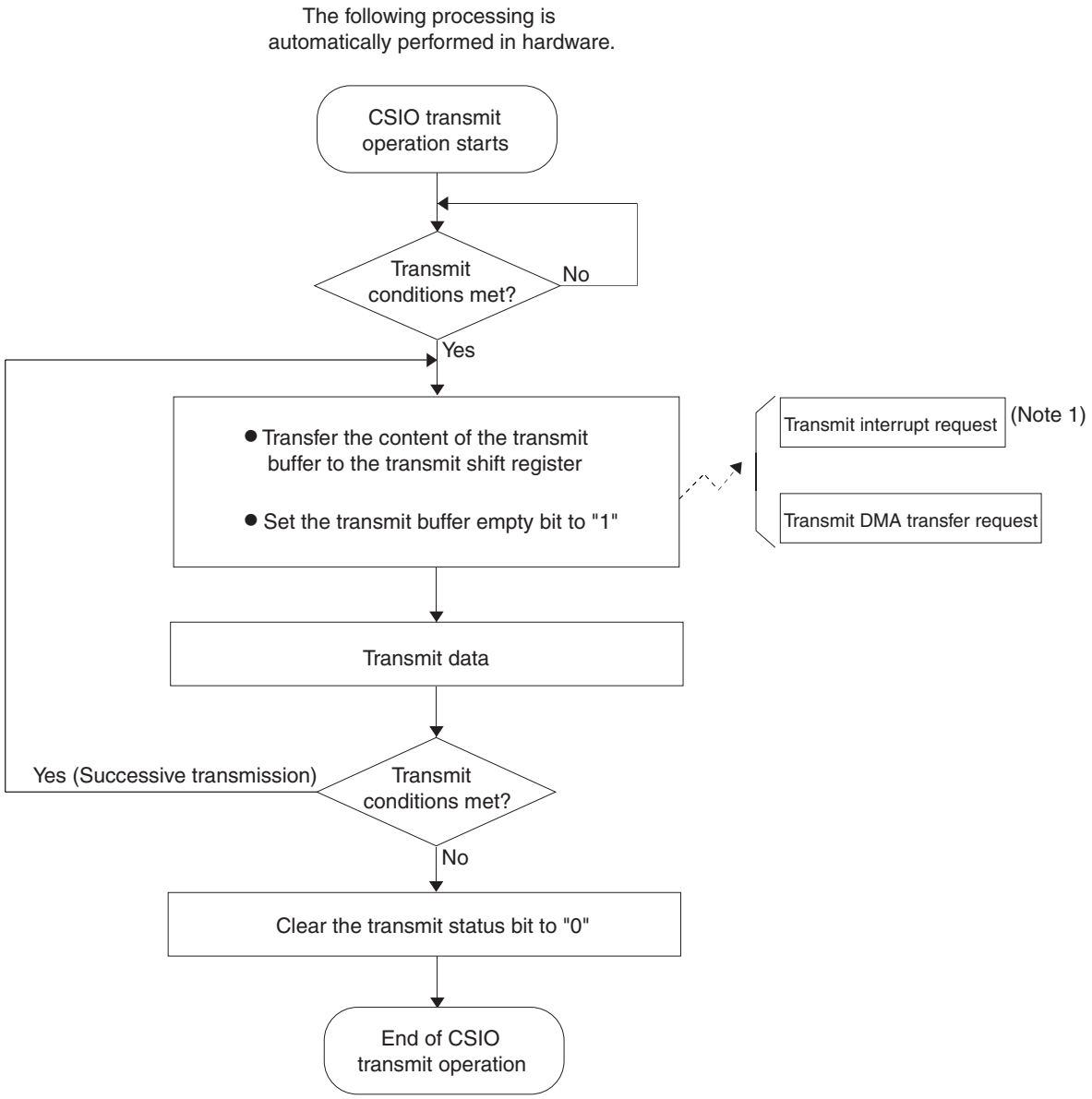
(2) Transmission finished interrupt

If the transmission finished interrupt was selected using the SIO Interrupt Request Source Select Register, a transmission finished interrupt request is generated by a falling edge of the internal transfer clock pulse at which the last bit of data in the transmit shift register has been transmitted.

The SIO Interrupt Request Mask Register and the Interrupt Controller (ICU) must be set before these transmit interrupts can be used.

12.3.7 Transmit DMA Transfer Request

When data has been transferred from the transmit buffer register to the transmit shift register, a transmit DMA transfer request for the corresponding SIO channel is output to the DMAC. A transmit DMA transfer request is also output when the TEN (Transmit Enable) bit is set to "1" (disabled → enabled).
The DMAC must be set before DMA transfers can be used during data transmission.

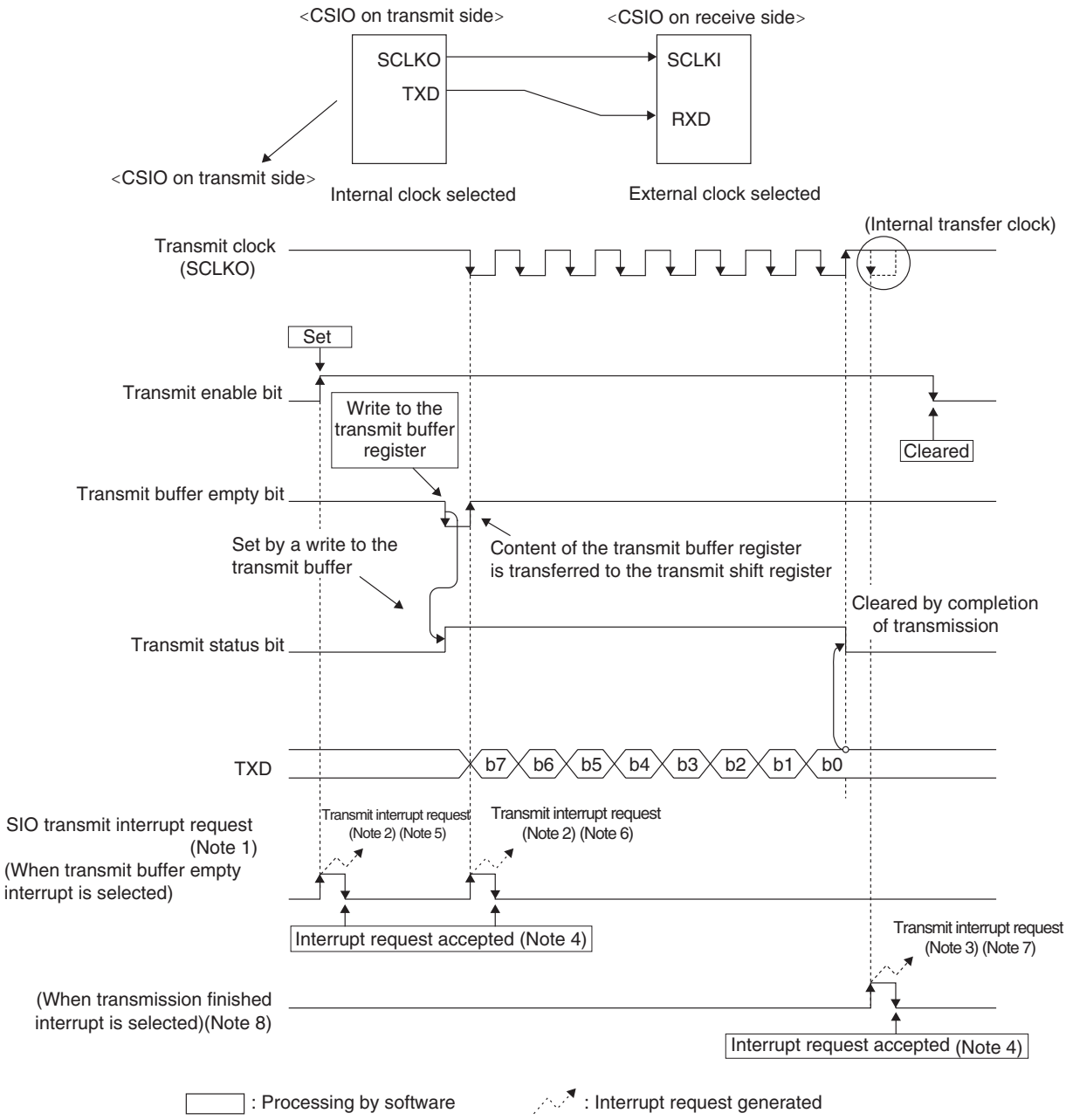


Note 1: This applies when the transmit interrupt request was enabled using the SIO Interrupt Request Enable Register after selecting the transmit buffer empty interrupt with the SIO Interrupt Request Source Select Register.

Figure 12.3.2 Transmit Operation during CSIO Mode (Hardware Processing)

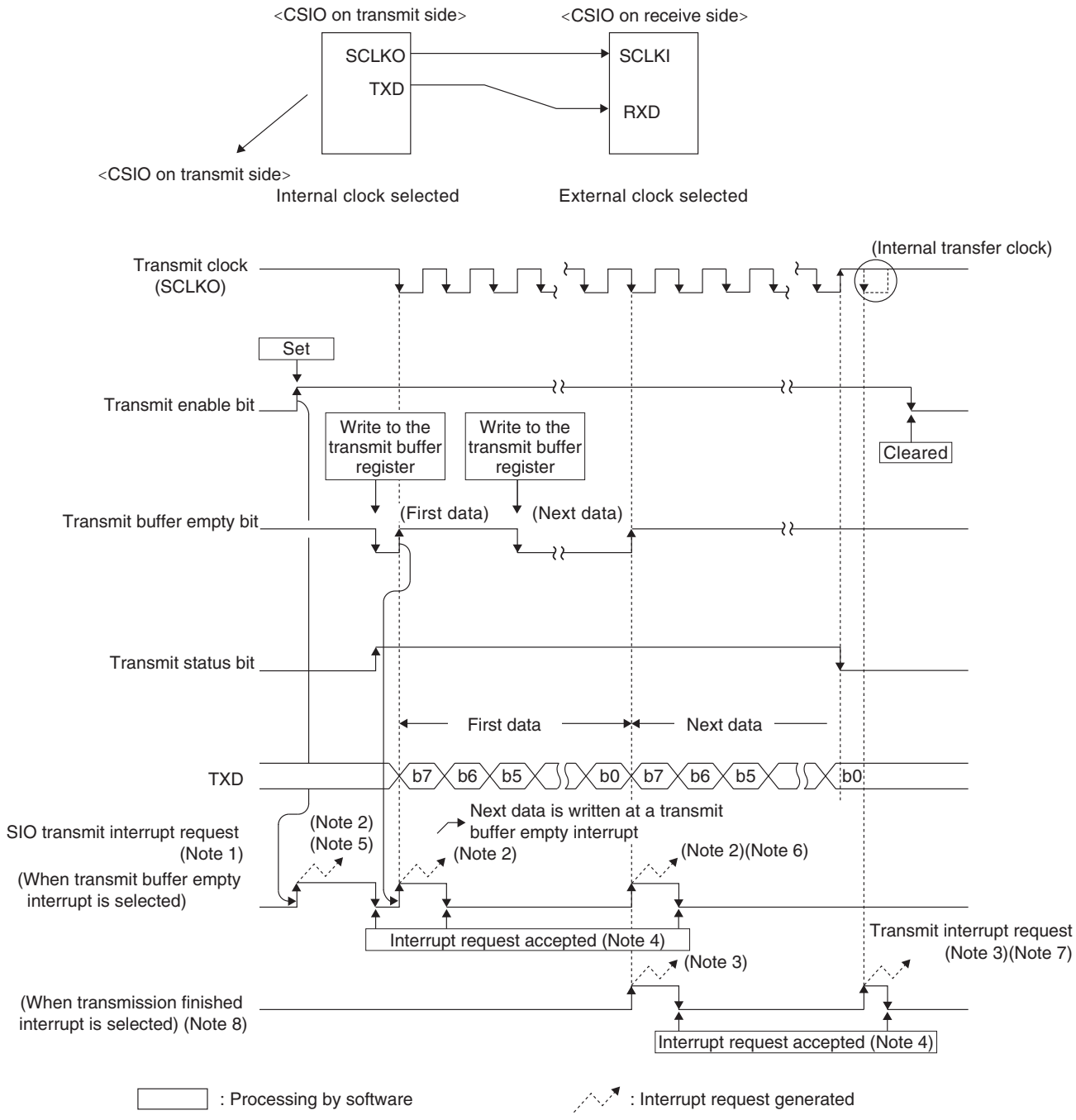
12.3.8 Example of CSIO Transmit Operation

The following shows a typical transmit operation in CSIO mode.



Note 1: Changes of the Interrupt Controller's SIO Transmit Interrupt Control Register interrupt request bit
Note 2: When transmit buffer empty interrupt is enabled (DMA transfer can also be requested at the same time)
Note 3: When transmission finished interrupt is enabled
Note 4: The Interrupt Controller's IVECT register is read or SIO Transmit Interrupt Control Register interrupt request bit is cleared.
Note 5: A transmit interrupt request is generated when transmission is enabled.
Note 6: Be aware that even after transmit data is written to the transmit buffer, a transmit interrupt request is generated when the data is transferred from the transmit buffer to the transmit shift register and the transmit buffer is thereby emptied.
Note 7: A transmission finished interrupt request is generated by a falling edge of the internal transfer clock pulse at which transmission of the transmit shift register data has finished.
Note 8: It is inhibited to select the transmission finished interrupt when an external clock is selected.

Figure 12.3.3 Example of CSIO Transmission (Transmitted Only Once)



Note 1: Changes of the Interrupt Controller's SIO Transmit Interrupt Control Register interrupt request bit
Note 2: When transmit buffer empty interrupt is enabled (DMA transfer can also be requested at the same time)
Note 3: When transmission finished interrupt is enabled
Note 4: The Interrupt Controller's IVECT register is read or SIO Transmit Interrupt Control Register interrupt request bit is cleared.
Note 5: A transmit interrupt request is generated when transmission is enabled.
Note 6: Be aware that even after transmit data is written to the transmit buffer, a transmit interrupt request is generated when the data is transferred from the transmit buffer to the transmit shift register and the transmit buffer is thereby emptied.
Note 7: A transmission finished interrupt request is generated by a falling edge of the internal transfer clock pulse at which transmission of the transmit shift register data has finished.
Note 8: It is inhibited to select the transmission finished interrupt when an external clock is selected.

Figure 12.3.4 Example of CSIO Transmission (Transmitted Successively)

12.4 Receive Operation in CSIO Mode

12.4.1 Initialization for CSIO Reception

To receive data in CSIO mode, initialize the serial interface following the procedure described below. Note, however, that because the receive shift clock is derived by an operation of the transmit circuit, transmit operation must always be executed even when the serial interface is used for only receiving data.

(1) Setting SIO Special Mode Register

- Set the clock polarity in CSIO mode.

(2) Setting SIO Transmit/Receive Mode Register

- Set the register to CSIO mode.
- Select the internal or an external clock.

(3) Setting SIO Transmit Control Register

- Select the clock divider's divide-by ratio (when internal clock selected).

(4) Setting SIO Baud Rate Register

When the internal clock is selected, set a baud rate generator value. (See Section 12.3.1, "Setting the CSIO Baud Rate.")

(5) Setting SIO interrupt related registers

- Select the source of receive interrupt request (reception finished or error) (SIO Interrupt Request Source Select Register).
- Enable or disable receive interrupts (SIO Interrupt Request Mask Register).

(6) Setting SIO Receive Control Register

- Set the receive enable bit.

(7) Setting the Interrupt Controller (SIO Transmit Interrupt Control Register)

To use receive interrupts, set their priority levels.

(8) Setting DMAC

Set up the DMAC when the DMA transfer is requested to the internal DMAC on completion of the transmission. (See Chapter 9, "DMAC.")

(9) Selecting pin functions

Because the serial interface related pins serve dual purposes, set the pin functions for use as SIO pins or input/output ports. (See Chapter 8, "Input/Output Ports and Pin Functions.")

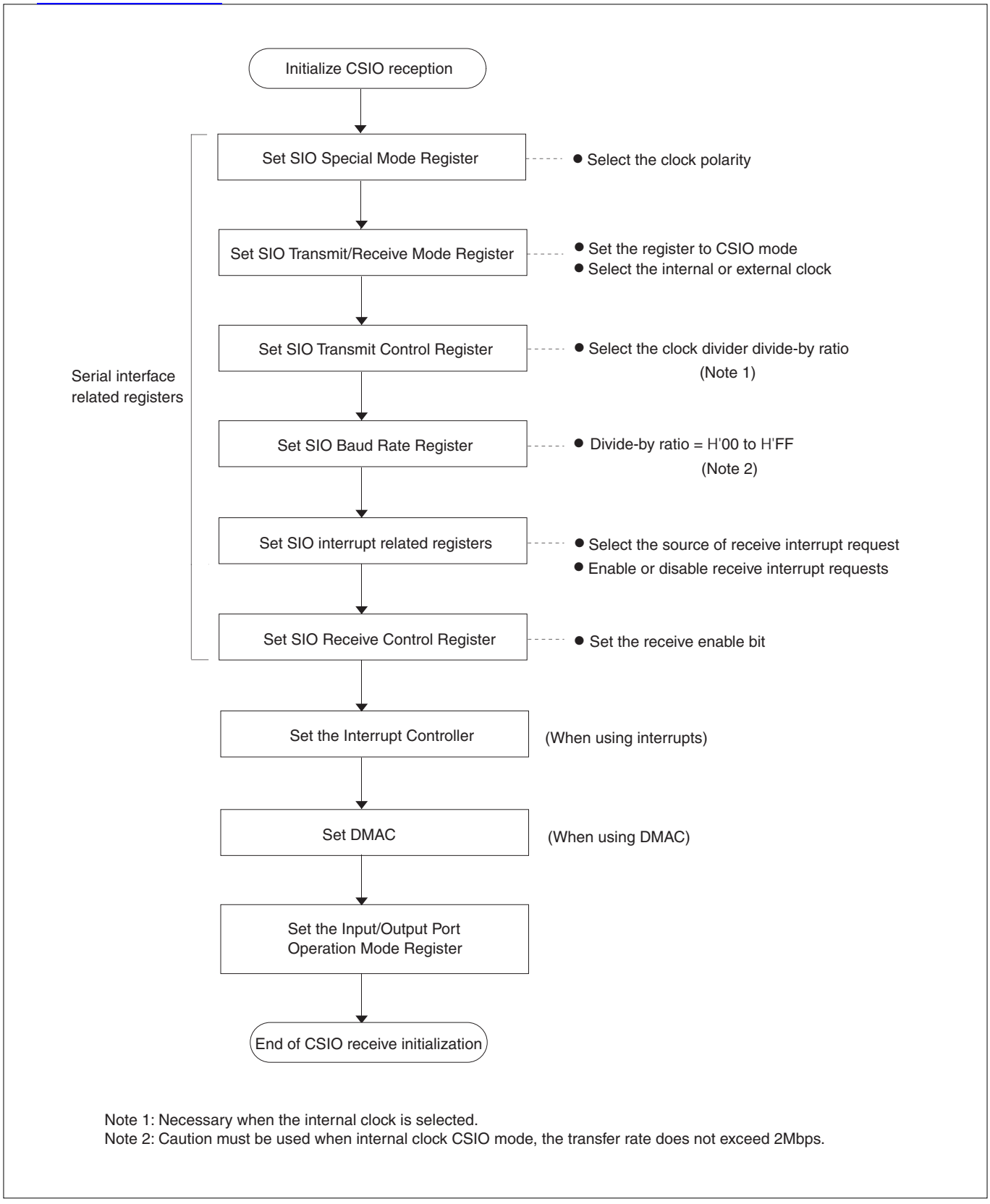


Figure 12.4.1 Procedure for Initializing CSIO Reception

12.4.2 Starting CSIO Reception

The serial interface starts receive operation when all of the following conditions are met after being initialized.

(1) Receive conditions when CSIO mode internal clock is selected

- The SIO Receive Control Register receive enable bit is set to "1".
- Transmit conditions are met. (See Section 12.3.3, “Starting CSIO Transmission.”)

(2) Receive conditions when CSIO mode external clock is selected

- The SIO Receive Control Register receive enable bit is set to "1".
- Transmit conditions are met. (See Section 12.3.3, “Starting CSIO Transmission.”)

Note: • The receive status bit is set to "1" at the time dummy data is set in the lower byte of the SIO Transmit Buffer Register.

When the above conditions are met, the serial interface starts receiving 8-bit serial data (LSB first) synchronously with the receive shift clock.

12.4.3 Processing at End of CSIO Reception

When data reception finishes, the following operation is automatically performed in hardware.

(1) When reception is completed normally

The reception finished (receive buffer full) bit is set to "1".

Notes: • An interrupt request is generated if the reception finished (receive buffer full) interrupt has been enabled.
• A DMA transfer request is generated.

(2) When an error occurred during reception

If an error (only overrun error in CSIO mode) occurred during reception, the overrun error bit and receive error sum bit are set to "1".

Notes: • If the reception finished interrupt has been selected (by SIO Receive Interrupt Request Source Select Register), neither a reception finished interrupt request nor a DMA transfer request is generated.
• If the receive error interrupt has been selected (by SIO Receive Interrupt Request Source Select Register), a receive error interrupt request is generated when interrupt requests are enabled. No DMA transfer requests are generated.

12.4.4 About Successive Reception

If the following conditions are met when data reception has finished, data may be received successively.

- The receive enable bit is set to "1".
- Transmit conditions are met.
- No overrun error has occurred.

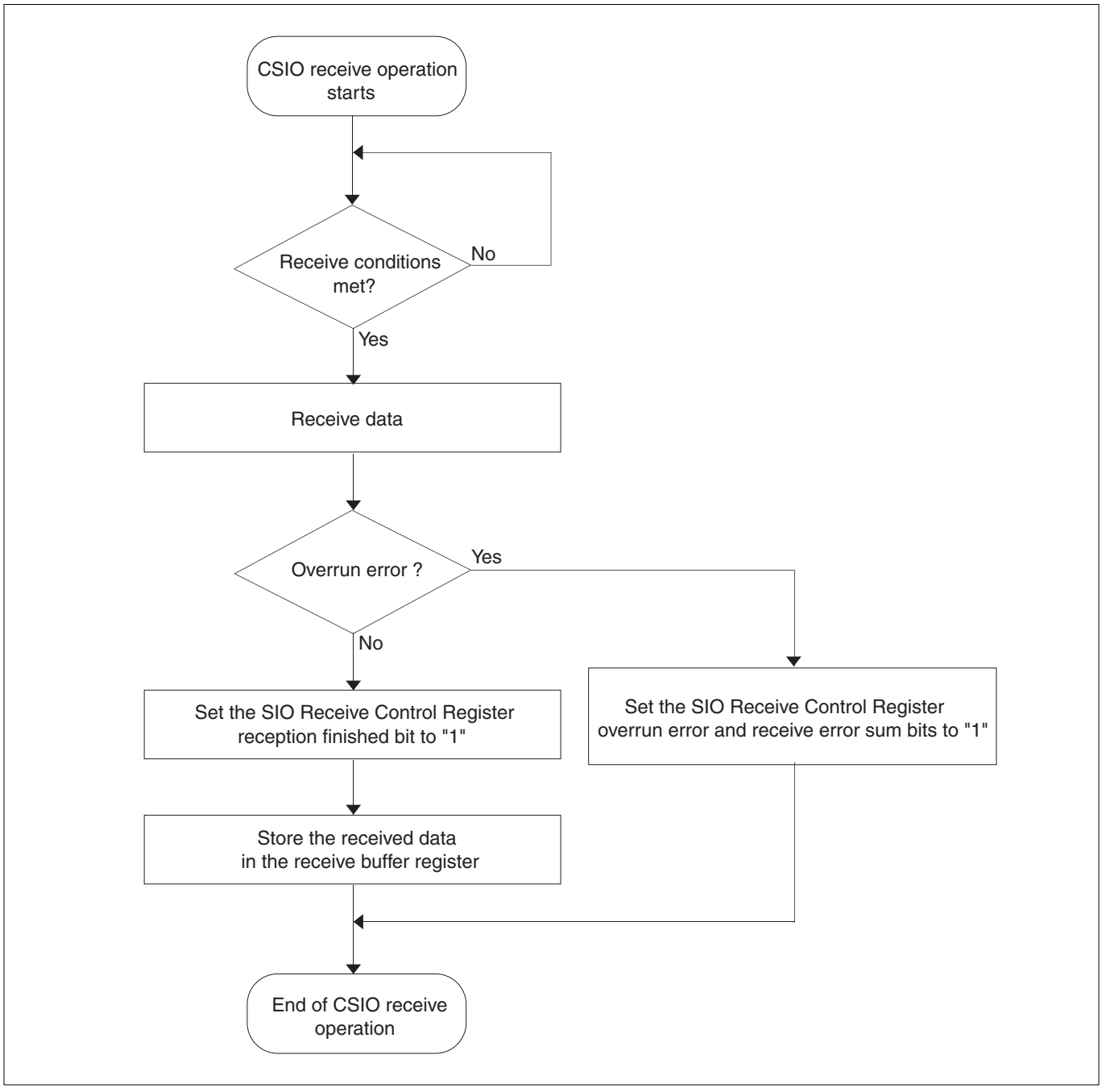


Figure 12.4.2 Receive Operation during CSIO Mode (Hardware Processing)

12.4.5 Flags Showing the Status of CSIO Receive Operation

There are following flags that indicate the status of receive operation during CSIO mode:

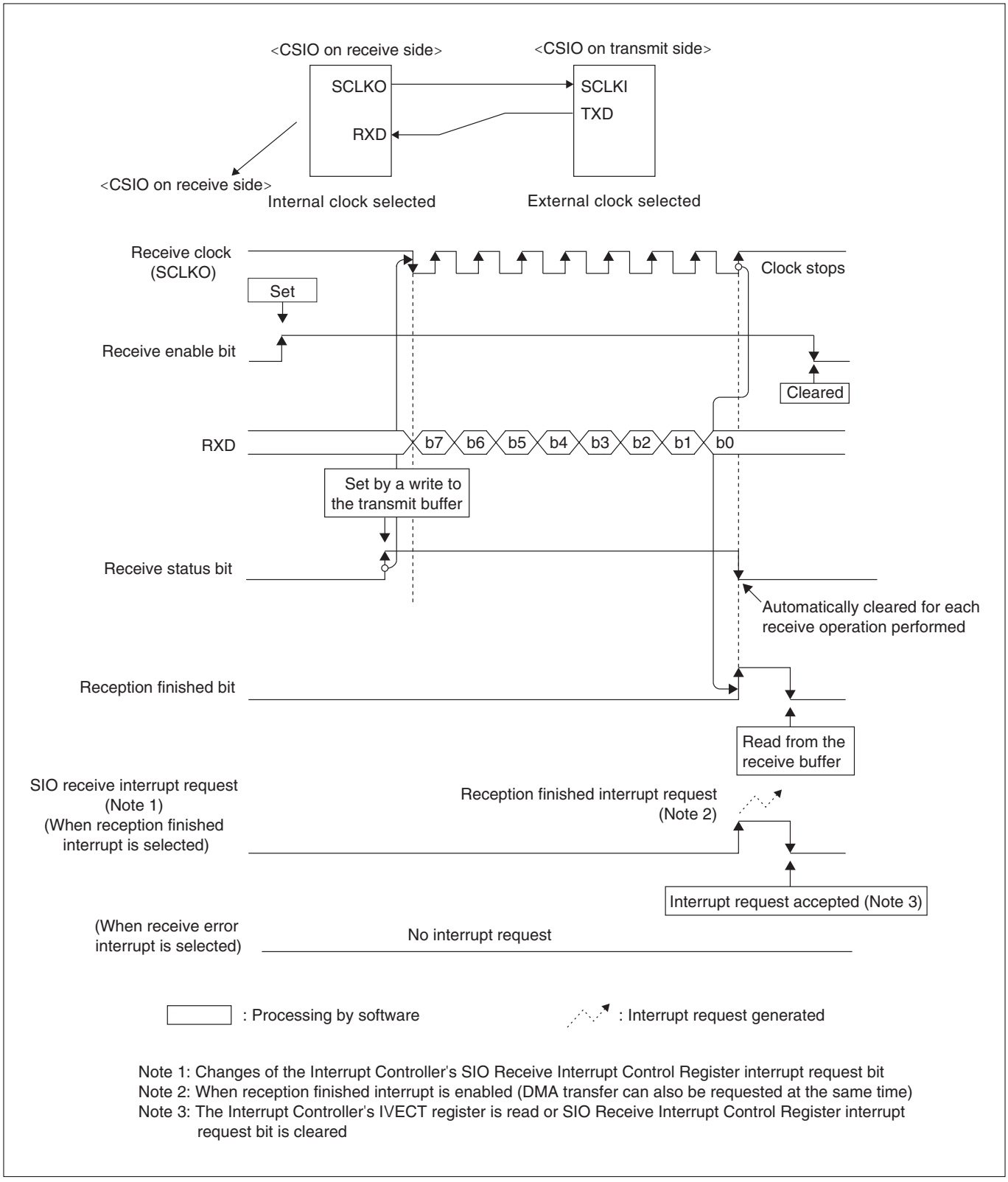
- SIO Receive Control Register receive status bit
- SIO Receive Control Register reception finished bit
- SIO Receive Control Register receive error sum bit
- SIO Receive Control Register overrun error bit

When reading the content of the SIO Receive Buffer Register after reception is completed, if the serial interface finishes receiving the next data before the previous data is not read out, an overrun error occurs and the subsequent received data are not transferred to the receive buffer register.
Before receive operation can be restarted, the receive enable bit must temporarily be cleared to "0" to initialize the receiver control unit.
The above reception finished bit, if no receive errors occurred (Note 1), may be cleared by reading out the lower byte of the SIO Receive Buffer Register or clearing the REN (Receive Enable) bit.
However, if any receive error occurred, the reception finished bit can only be cleared by clearing the REN (Receive Enable) bit, and cannot be cleared by reading out the lower byte of the SIO Receive Buffer Register.

Note 1: Overrun errors are the only error that can be detected during reception in CSIO mode.

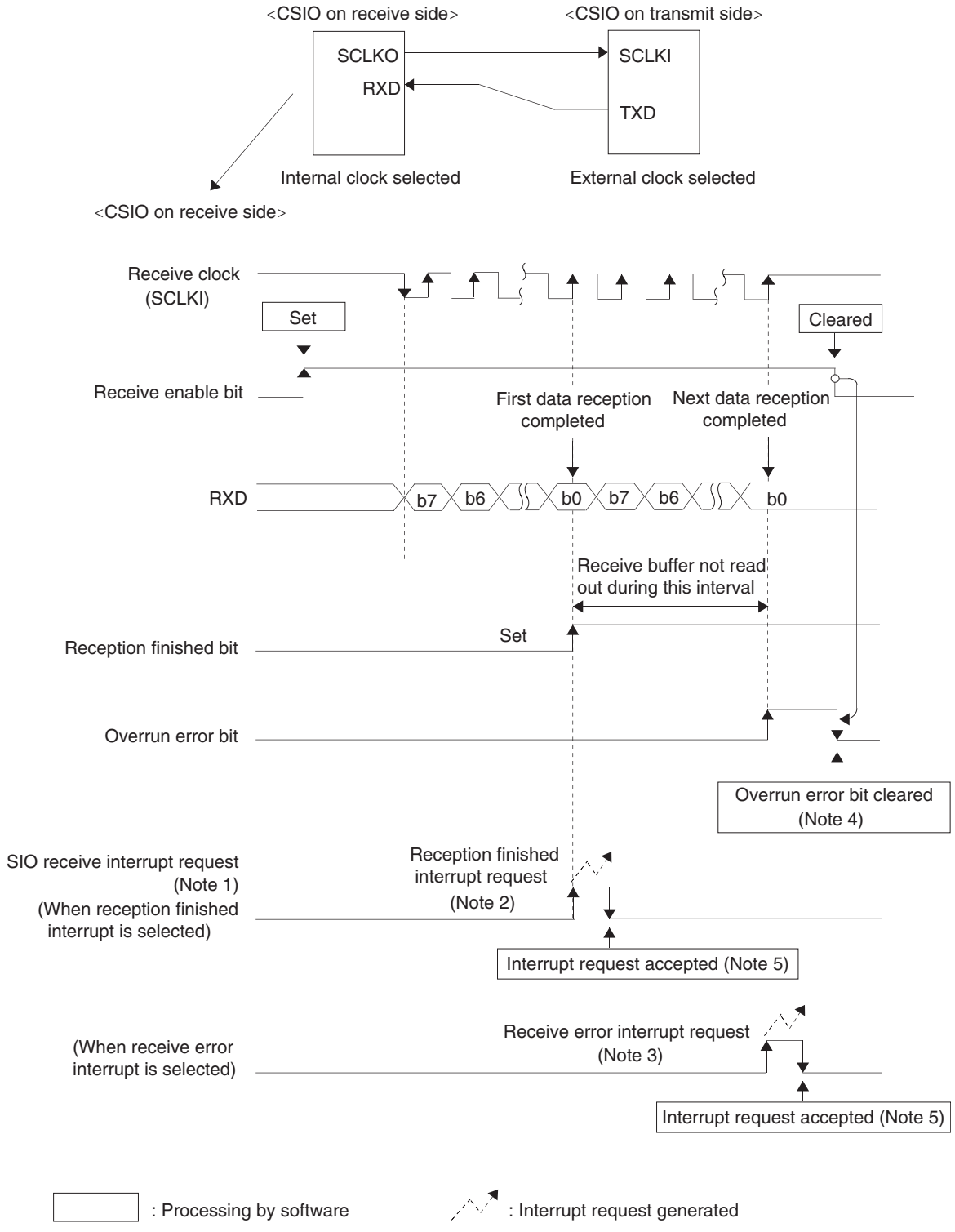
12.4.6 Example of CSIO Receive Operation

The following shows a typical receive operation in CSIO mode.



Note 1: Changes of the Interrupt Controller's SIO Receive Interrupt Control Register interrupt request bit
Note 2: When reception finished interrupt is enabled (DMA transfer can also be requested at the same time)
Note 3: The Interrupt Controller's IVECT register is read or SIO Receive Interrupt Control Register interrupt request bit is cleared

Figure 12.4.3 Example of CSIO Reception (When Received Normally)



Note 1: Changes of the Interrupt Controller's SIO Receive Interrupt Control Register interrupt request bit
Note 2: When reception finished interrupt is enabled
Note 3: When receive error interrupt is enabled
Note 4: The receive enable bit is cleared.
Note 5: The Interrupt Controller's IVECT register is read or SIO Receive Interrupt Control Register interrupt request bit is cleared.

Figure 12.4.4 Example of CSIO Reception (When Overrun Error Occurred)

12.5 Notes on Using CSIO Mode

- **Settings of SIO Transmit/Receive Mode Register and SIO Baud Rate Register**

  The SIO Transmit/Receive Mode Register and SIO Baud Rate Register and the Transmit Control Register's BRG count source select bit must always be set when the serial interface is not operating. If a transmit or receive operation is in progress, wait until the transmit and receive operations are finished and then clear the transmit and receive enable bits before making changes.

- **Settings of BRG (Baud Rate Register)**

  If f(BCLK) is selected with the BRG clock source select bit, use caution when setting the BRG register so that the transfer rate will not exceed 2 Mbps.

- **About successive transmission**

  To transmit data successively, make sure the next transmit data is set in the SIO Transmit Buffer Register before the current data transmission finishes.

- **About reception**

  Because the receive shift clock in CSIO mode is derived by an operation of the transmit circuit, transmit operation must always be executed (by sending dummy data) even when the serial interface is used for only receiving data. In this case, be aware that if the port function is set for the TXD pin (by setting the operation mode register to "1"), dummy data may actually be output from the pin.

- **About successive reception**

  To receive data successively, make sure that data (dummy data) is set in the SIO Transmit Buffer Register before a transmit operation on the transmitter side starts.

- **Transmission/reception using DMA**

  To transmit/receive data in DMA request mode, enable the DMAC to accept transfer requests (by setting the DMA Mode Register) before serial communication starts.

- **About reception finished bit**

  If a receive error (overrun error) occurs, the reception finished bit can only be cleared by clearing the receive enable bit, and cannot be cleared by reading out the receive buffer register.

- **About overrun error**

  If all bits of the next received data have been set in the SIO Receive Shift Register before reading out the SIO Receive Buffer Register (i.e., an overrun error occurred), the received data is not stored in the receive buffer register, with the previous received data retained in it. Although a receive operation continues thereafter, the subsequent received data is not stored in the receive buffer register (receive status bit = "1"). Before normal receive operation can be restarted, the receive enable bit must be temporarily cleared to "0". And this is the only way that the overrun error flag can be cleared.

- **About DMA transfer request generation during SIO transmission**

  If the transmit buffer register becomes empty (transmit buffer empty flag = "1") while the transmit enable bit remains set to "1" (transmission enabled), an SIO transmit buffer empty DMA transfer request is generated.

- **About DMA transfer request generation during SIO reception**

  If the reception finished bit is set to "1" (receive buffer register full), a reception finished DMA transfer request is generated. Be aware, however, that if an overrun error occurred during reception, this DMA transfer request is not generated.

• **Switching from general-purpose to serial interface pin**

When switching general-purpose to serial interface pin, SCLKOn pin outputs "H" level (For the case of selecting internal clock and setting CKPOL bit to "0." When setting CKPOL bit to "1", it outputs "L" level.), and TXDn pin outputs undefined value. However, when switching general-purpose to serial interface pin with setting TEN bit of the SIOn transmit control register to "1" (transmit enable) , TXDn pin outputs the last bit level of the previously output serial data.

12.6 Transmit Operation in UART Mode

12.6.1 Setting the UART Baud Rate

The baud rate (data transfer rate) in UART mode is determined by a transmit/receive shift clock. During UART mode, the source for this transmit/receive shift clock is always the internal clock no matter how the internal/external clock select bit (SIO Transmit/Receive Mode Register bit 11) is set.

(1) Calculating the UART mode baud rate

After being divided by a clock divider, f(BCLK) is supplied to the Baud Rate Generator (BRG), after which it is further divided by 16 to produce a transmit/receive shift clock.
The clock divider's divide-by value is selected from 1, 8, 32 or 256 by using the SIO Transmit Control Register CDIV (baud rate generator count source select) bits (bits 2–3).
The Baud Rate Generator divides the clock divider output by (baud rate register set value + 1) and further by 16, thus generating a transmit/receive shift clock.

When the internal clock is selected in UART mode, the baud rate is calculated using the equation below.

$$\text{Baud rate [bps]} = \frac{f(BCLK)}{\text{Clock divider's divide-by value} \times (\text{baud rate register set value} + 1) \times 16}$$

Baud rate register set value = H'00 to H'FF
Clock divider's divide-by value = 1, 8, 32 or 256

12.6.2 UART Transmit/Receive Data Formats

The transmit/receive data format during UART mode is determined by setting the SIO Transmit/Receive Mode Register. Shown below is the transmit/receive data format that can be used in UART mode.

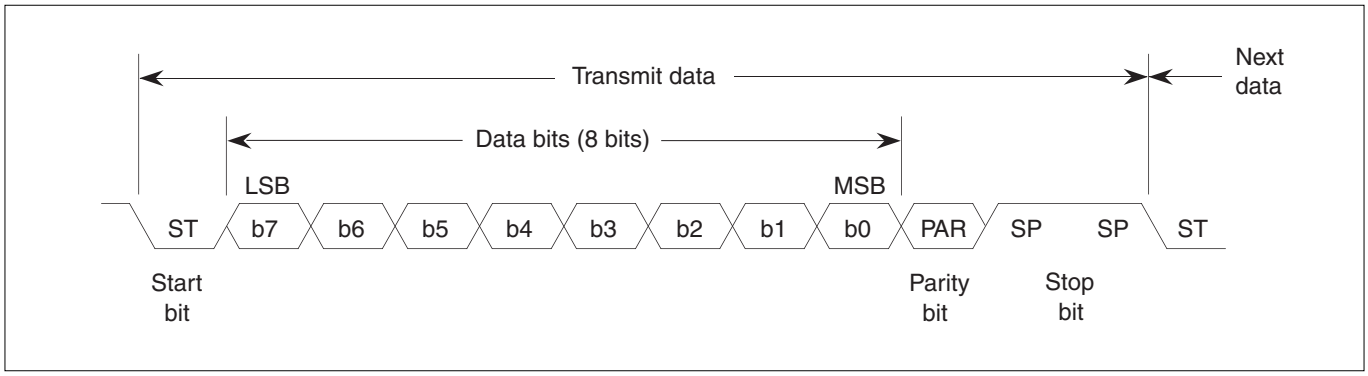


Figure 12.6.1 Example of a Transfer Data Format during UART Mode

Table 12.6.1 Transfer Data in UART Mode

Bit Name	Content
ST (start bit)	Indicates the beginning of data transmission. This is a "L" level signal of a one bit period, which is added immediately preceding the transmit data.
Bits 0–8 (character bits)	Transmit/receive data transferred via serial interface. In UART mode, 7, 8 or 9 bits of data can be transmitted/received.
PAR (parity bit)	Added to the transmit/receive character. When parity is enabled, parity is automatically set in such a way that the number of 1's in the character including the parity bit itself is always even or odd as selected by the even/odd parity select bit.
SP (stop bit)	Indicates the end of data transmission, which is added immediately following the character (or if parity is enabled, immediately following the parity bit). The stop bit can be chosen to be one bit or two bits long.

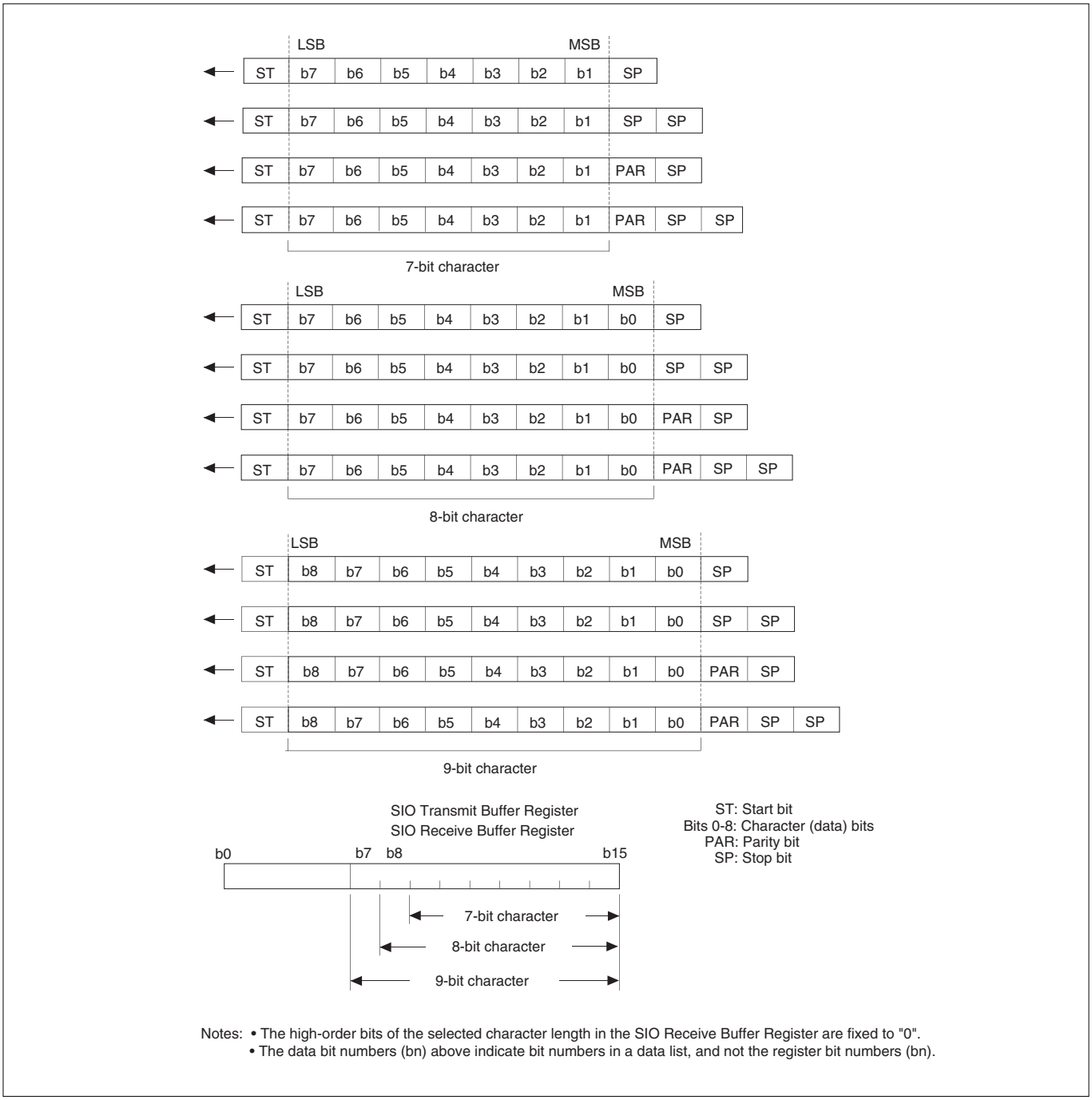


Figure 12.6.2 Selectable Data Formats during UART Mode

12.6.3 Initializing UART Transmission

To transmit data in UART mode, initialize the serial interface following the procedure described below.

(1) Setting SIO Transmit/Receive Mode Register

- Set the register to UART mode.
- Set parity (when enabled, select odd/even).
- Set the stop bit length.
- Set the character length (Note 1).

Note 1: During UART mode, settings of the internal/external clock select bit have no effect (only the internal clock is useful).

(2) Setting SIO Transmit Control Register

- Select the clock divider's divide-by ratio.

(3) Setting SIO Baud Rate Register

Set a baud rate generator value. (See Section 12.6.1, "Setting the UART Baud Rate.")

(4) Setting SIO interrupt related registers

- Select the source of transmit interrupt request (transmit buffer empty or transmission finished) (SIO Interrupt Request Source Select Register).
- Enable or disable SIO transmit interrupt requests (SIO Interrupt Request Mask Register).

(5) Setting the Interrupt Controller (SIO Transmit Interrupt Control Register)

To use transmit interrupts, set their priority levels.

(6) Setting DMAC

To issue DMA transfer requests to the internal DMAC when the transmit buffer is empty, set up the DMAC. (See Chapter 9, "DMAC.")

(7) Selecting pin functions

Because the serial interface related pins serve dual purposes, set the pin functions for use as SIO pins or input/output ports. (See Chapter 8, "Input/Output Ports and Pin Functions.")

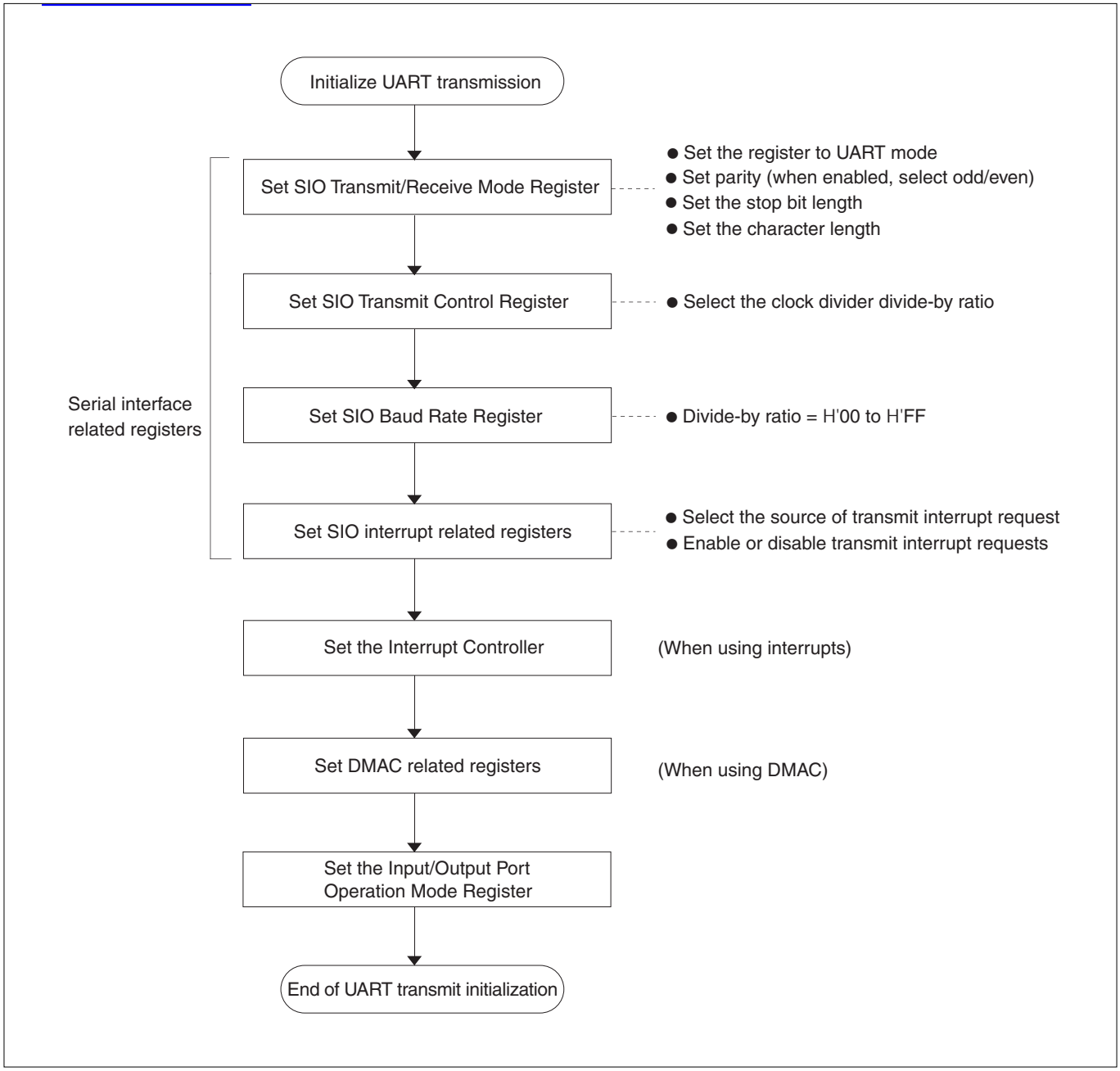


Figure 12.6.3 Procedure for Initializing UART Transmission

12.6.4 Starting UART Transmission

The serial interface starts a transmit operation when all of the following conditions are met after being initialized.

- SIO Transmit Control Register TEN (Transmit Enable) bit is set to "1" (Note 1).
- Transmit data is written to the SIO Transmit Buffer Register (transmit buffer empty bit = "0").

Note 1: While the transmit enable bit is cleared to "0", writes to the transmit buffer are ignored. Always be sure to set the transmit enable bit to "1" before writing to the transmit buffer register.

When transmission starts, the serial interface sends data following the procedure described below.

- Transfer the content of the SIO Transmit Buffer Register to the SIO Transmit Shift Register.
- Set the transmit buffer empty bit to "1" (Note 2).
- Start sending data synchronously with the shift clock beginning with the LSB.

Note 2: A transmit interrupt request can be generated for reasons that the transmit buffer is empty or transmission has finished. Also, a DMA transfer request can be generated when the transmit buffer is empty. No DMA transfer requests can be generated for reasons that transmission has finished.

12.6.5 Successive UART Transmission

Once data has been transferred from the transmit buffer register to the transmit shift register, the next data can be written to the transmit buffer register even when the serial interface has not finished sending the previous data. If the next data is written to the transmit buffer before transmission has finished, the previous and the next data are transmitted successively.

Check the SIO Transmit Control Register's transmit buffer empty flag to see if data has been transferred from the transmit buffer register to the transmit shift register.

12.6.6 Processing at End of UART Transmission

When data transmission finishes, the following operation is automatically performed in hardware.

(1) When not transmitting successively

- The transmit status bit is cleared to "0".

(2) When transmitting successively

- When transmission of the last data in a consecutive data train finishes, the transmit status bit is cleared to "0".

12.6.7 Transmit Interrupts

(1) Transmit buffer empty interrupt

If the transmit buffer empty interrupt was selected using the SIO Interrupt Request Source Select Register, a transmit buffer empty interrupt request is generated when data has been transferred from the transmit buffer register to the transmit shift register. A transmit buffer empty interrupt request is also generated when the TEN (Transmit Enable) bit is set to "1" (reenabled after being disabled) while the transmit buffer empty interrupt has been enabled.

(2) Transmission finished interrupt

If the transmission finished interrupt was selected using the SIO Interrupt Request Source Select Register, a transmission finished interrupt request is generated when data in the transmit shift register has all been transmitted.

The SIO Interrupt Request Mask Register and the Interrupt Controller (ICU) must be set before these transmit interrupts can be used.

12.6.8 Transmit DMA Transfer Request

When data has been transferred from the transmit buffer register to the transmit shift register, a transmit DMA transfer request for the corresponding SIO channel is output to the DMAC. A transmit DMA transfer request is also output when the TEN (Transmit Enable) bit is set to "1" (disabled → enabled).
The DMAC must be set before DMA transfers can be used during data transmission.

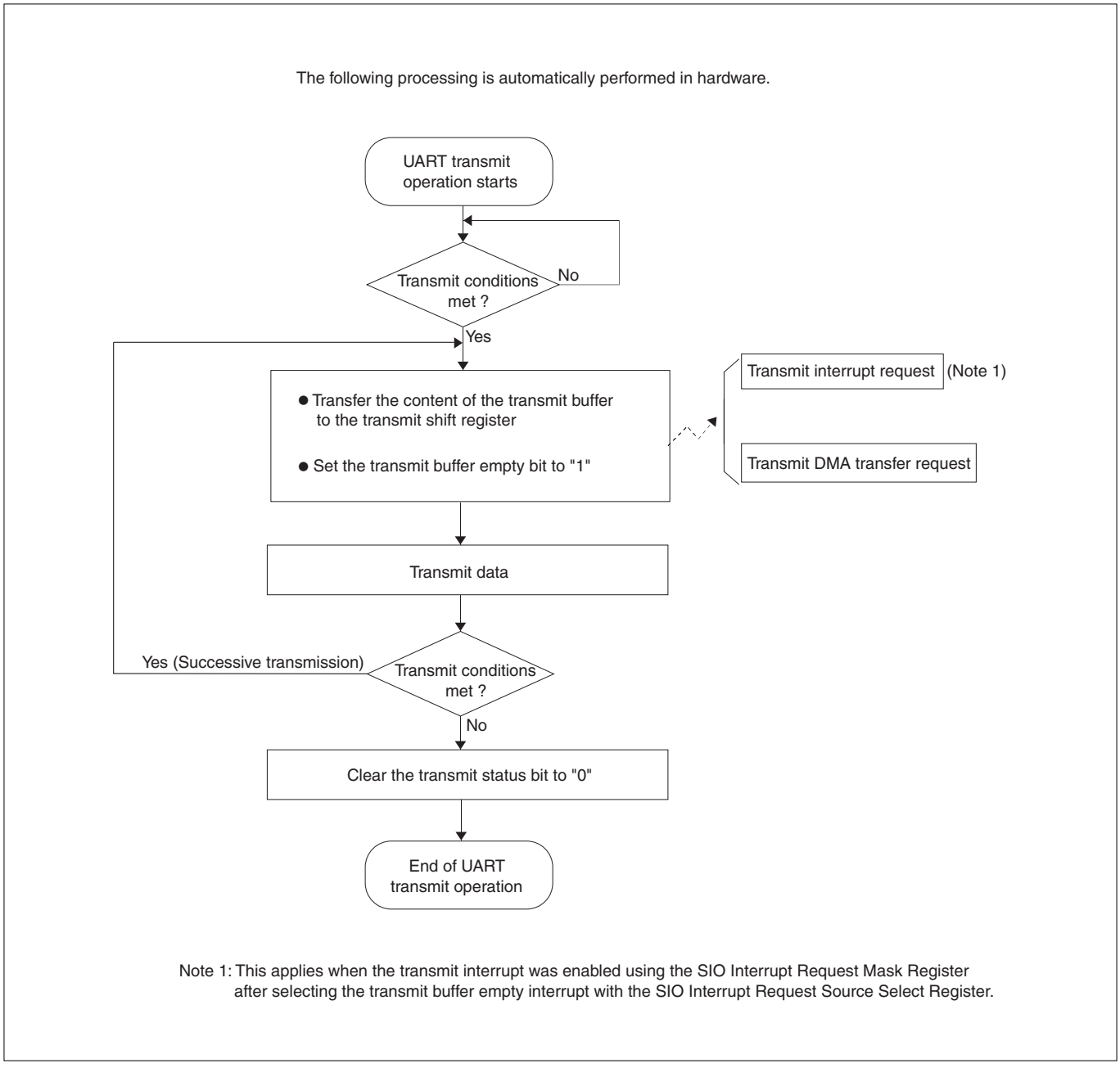
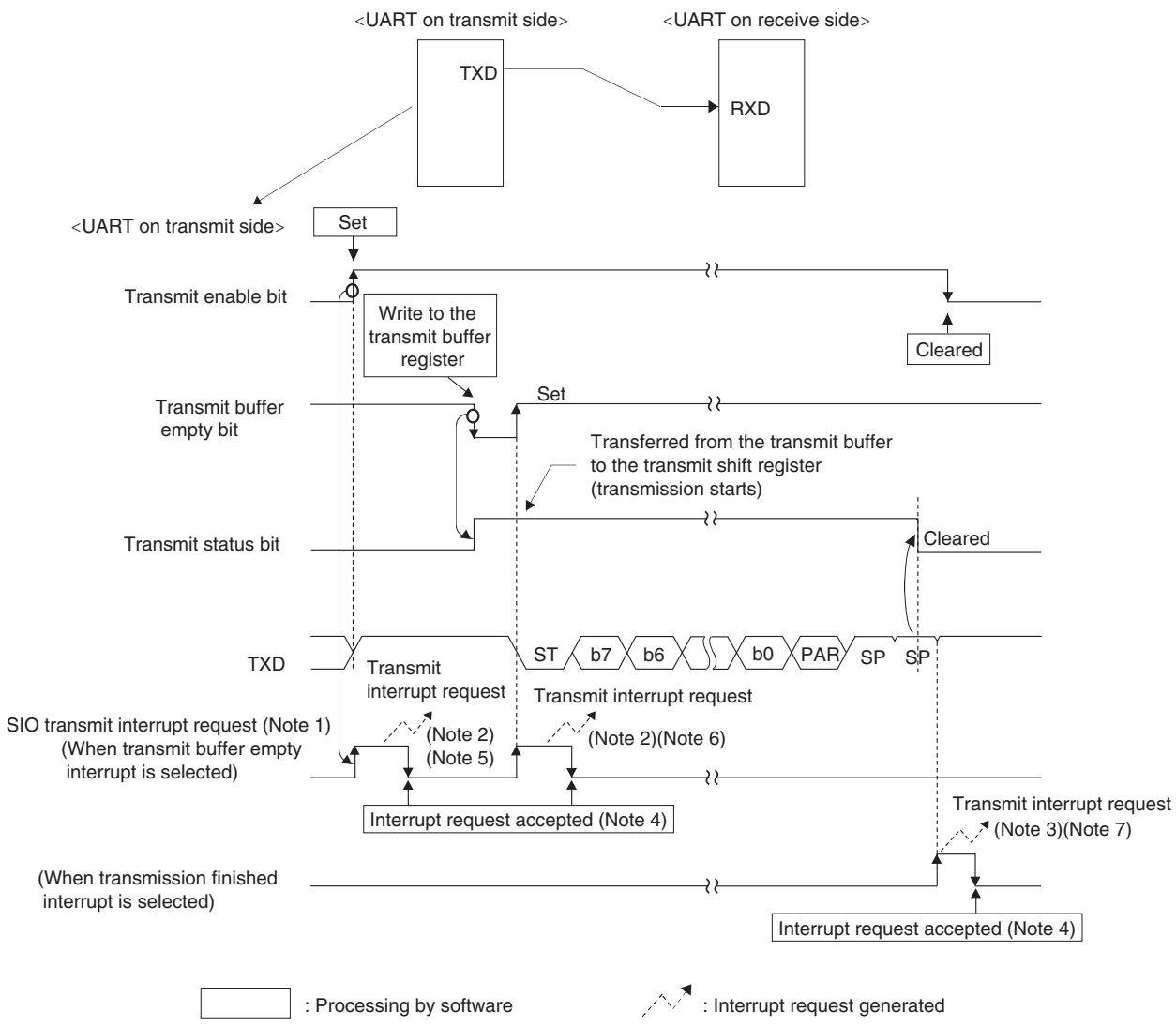


Figure 12.6.4 Transmit Operation during UART Mode (Hardware Processing)

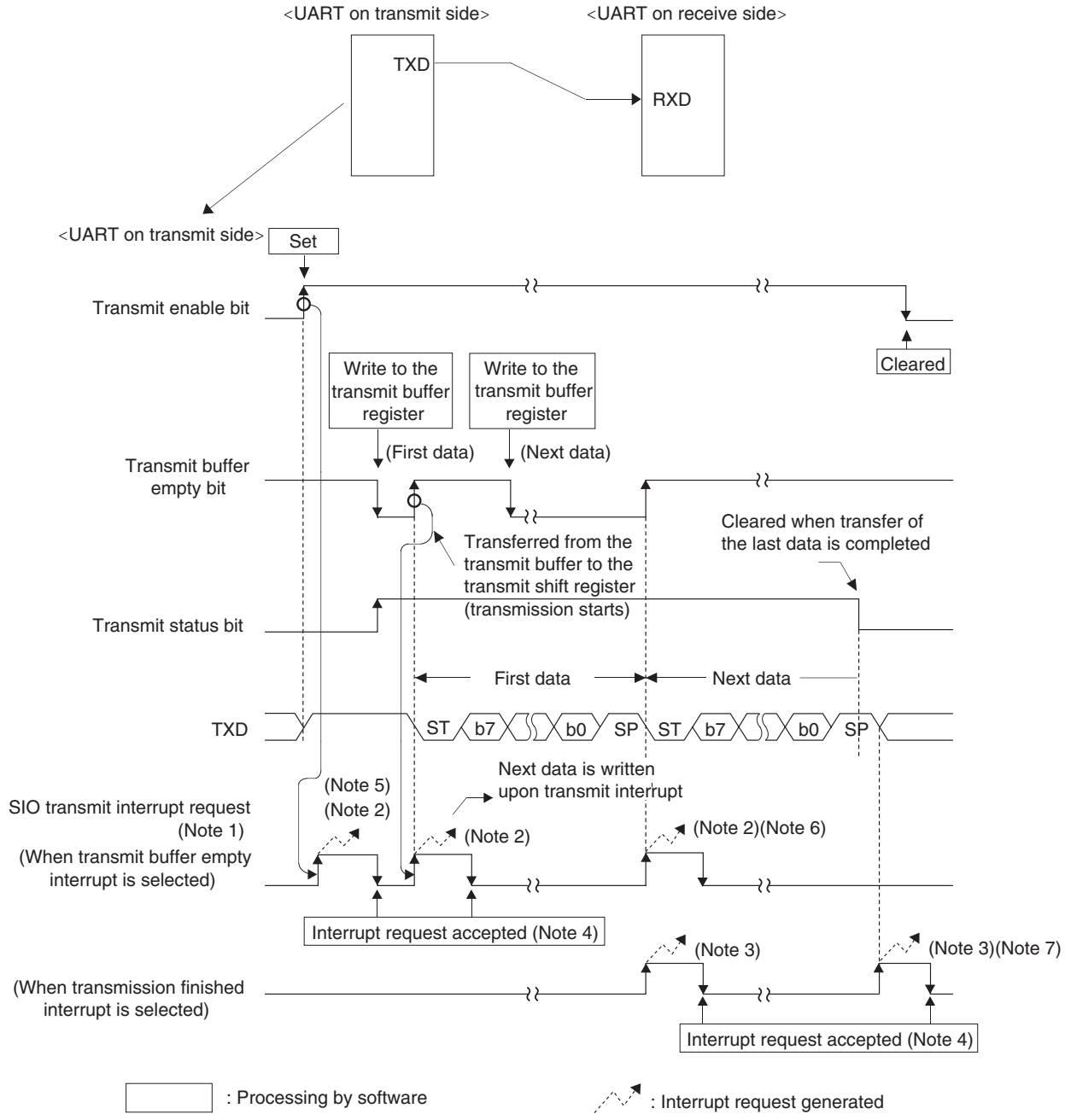
12.6.9 Example of UART Transmit Operation

The following shows a typical transmit operation in UART mode.



Note 1: Changes of the Interrupt Controller's SIO Transmit Interrupt Control Register interrupt request bit
Note 2: When transmit buffer empty interrupt is enabled (DMA transfer can also be requested at the same time)
Note 3: When transmission finished interrupt is enabled
Note 4: The Interrupt Controller's IVECT register is read or SIO Transmit Interrupt Control Register interrupt request bit is cleared.
Note 5: A transmit interrupt request is generated when transmission is enabled.
Note 6: Be aware that even after transmit data is written to the transmit buffer, a transmit interrupt request is generated when the data is transferred from the transmit buffer to the transmit shift register and the transmit buffer is thereby emptied.
Note 7: A transmission finished interrupt request is generated when data in the transmit shift register has all been transmitted.

Figure 12.6.5 Example of UART Transmission (Transmitted Only Once)



Note 1: Changes of the Interrupt Controller's SIO Transmit Interrupt Control Register interrupt request bit
Note 2: When transmit buffer empty interrupt is enabled (DMA transfer can also be requested at the same time)
Note 3: When transmission finished interrupt is enabled
Note 4: The Interrupt Controller's IVECT register is read or SIO Transmit Interrupt Control Register interrupt request bit is cleared.
Note 5: A transmit interrupt request is generated when transmission is enabled.
Note 6: Be aware that even after transmit data is written to the transmit buffer, a transmit interrupt request is generated when the data is transferred from the transmit buffer to the transmit shift register and the transmit buffer is thereby emptied.
Note 7: A transmission finished interrupt request is generated when data in the transmit shift register has all been transmitted.

Figure 12.6.6 Example of UART Transmission (Transmitted Successively)

12.7 Receive Operation in UART Mode

12.7.1 Initialization for UART Reception

To receive data in UART mode, initialize the serial interface following the procedure described below.

(1) Setting SIO Transmit/Receive Mode Register

- Set the register to UART mode.
- Set parity (when enabled, select odd/even).
- Set the stop bit length.
- Set the character length.

Note: • During UART mode, settings of the internal/external clock select bit have no effect (only the internal clock is useful).

(2) Setting SIO Transmit Control Register

- Set the clock divider's divide-by ratio.

(3) Setting SIO Baud Rate Register

Set a baud rate generator value. (See Section 12.6.1, "Setting the UART Baud Rate.")

(4) Setting SIO interrupt related registers

- Select the source of receive interrupt request (reception finished or receive error) (Interrupt Request Source Select Register).
- Enable or disable receive interrupts (Interrupt Request Mask Register).

(5) Setting the Interrupt Controller

To use receive interrupt, set their priority levels.

(6) Setting DMAC

To issue DMA transfer requests to the internal DMAC when reception has finished, set up the DMAC. (See Chapter 9, "DMAC.")

(7) Selecting pin functions

Because the serial interface related pins serve dual purposes, set the pin functions for use as SIO pins or input/output ports. (See Chapter 8, "Input/Output Ports and Pin Functions.")

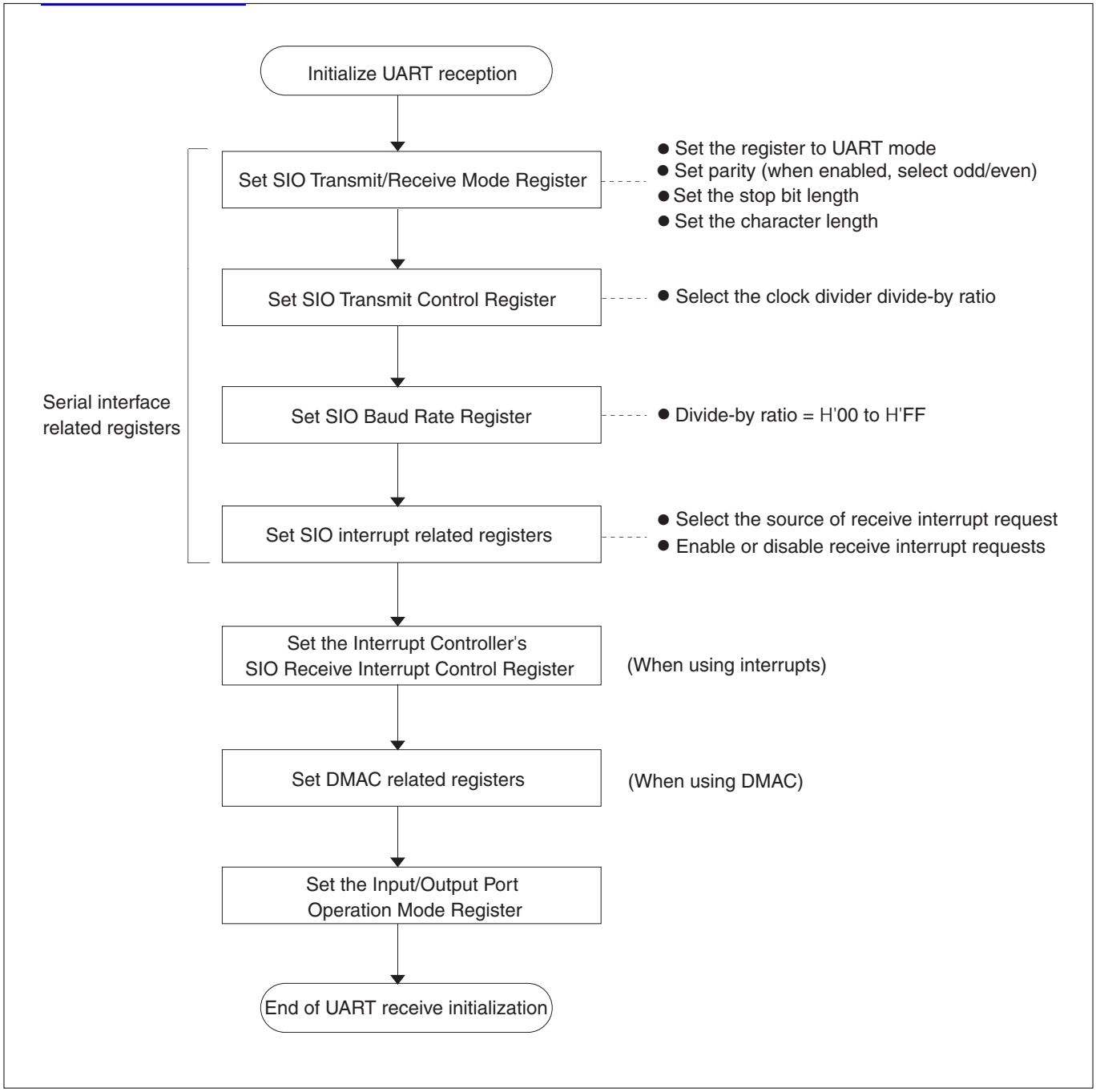


Figure 12.7.1 Procedure for Initializing UART Reception

12.7.2 Starting UART Reception

The serial interface starts receive operation when all of the following conditions are met after being initialized.

- SIO Receive Control Register receive enable bit is set to "1"
- Start bit (falling edge signal) is applied to the RXD pin

When the above conditions are met, the serial interface enters UART receive operation. However, the start bit is checked again at the first rise of the internal receive shift clock and if it is detected "H" for reasons of noise, etc., the serial interface stops receive operation and waits for the start bit again.

12.7.3 Processing at End of UART Reception

When data reception finishes, the following operation is automatically performed in hardware.

(1) When reception is completed normally

The reception finished (receive buffer full) bit is set to "1".

Notes: • An interrupt request is generated if the reception finished (receive buffer full) interrupt has been enabled.
• A DMA transfer request is generated.

(2) When a receive error occurred

If an error occurred, the corresponding error bit (OE, FE or PE) and the receive error sum bit are set to "1".

Notes: • If the reception finished interrupt has been selected (by SIO Receive Interrupt Request Source Select Register), a reception finished interrupt request is generated when interrupt requests are enabled. However, this does not apply when the detected error is an overrun error, in which case no reception finished interrupt requests are generated.
• If the receive error interrupt has been selected (by SIO Receive Interrupt Request Source Select Register), a receive error interrupt request is generated when interrupt requests are enabled.
• No DMA transfer requests are generated.

The following processing is automatically performed in hardware.

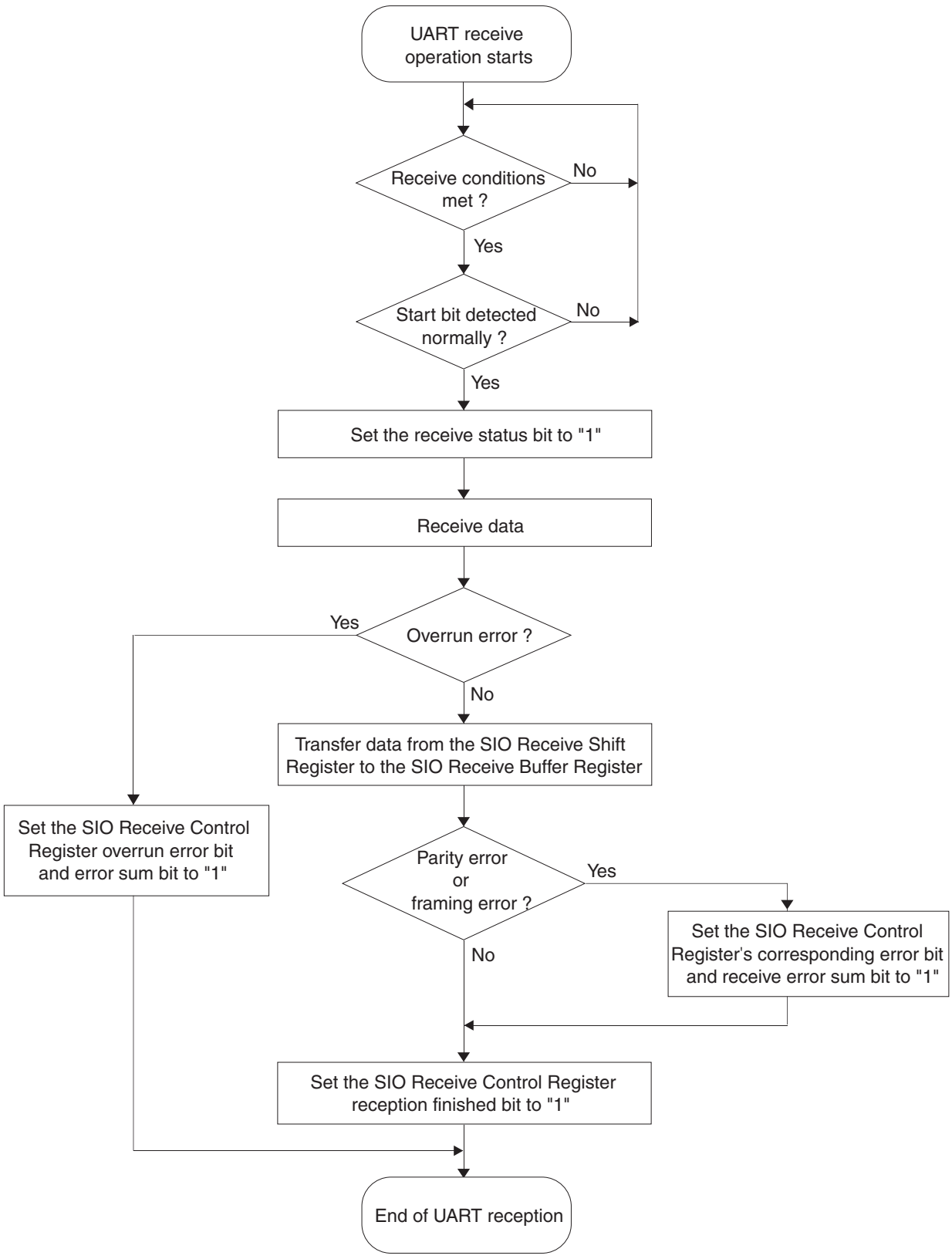
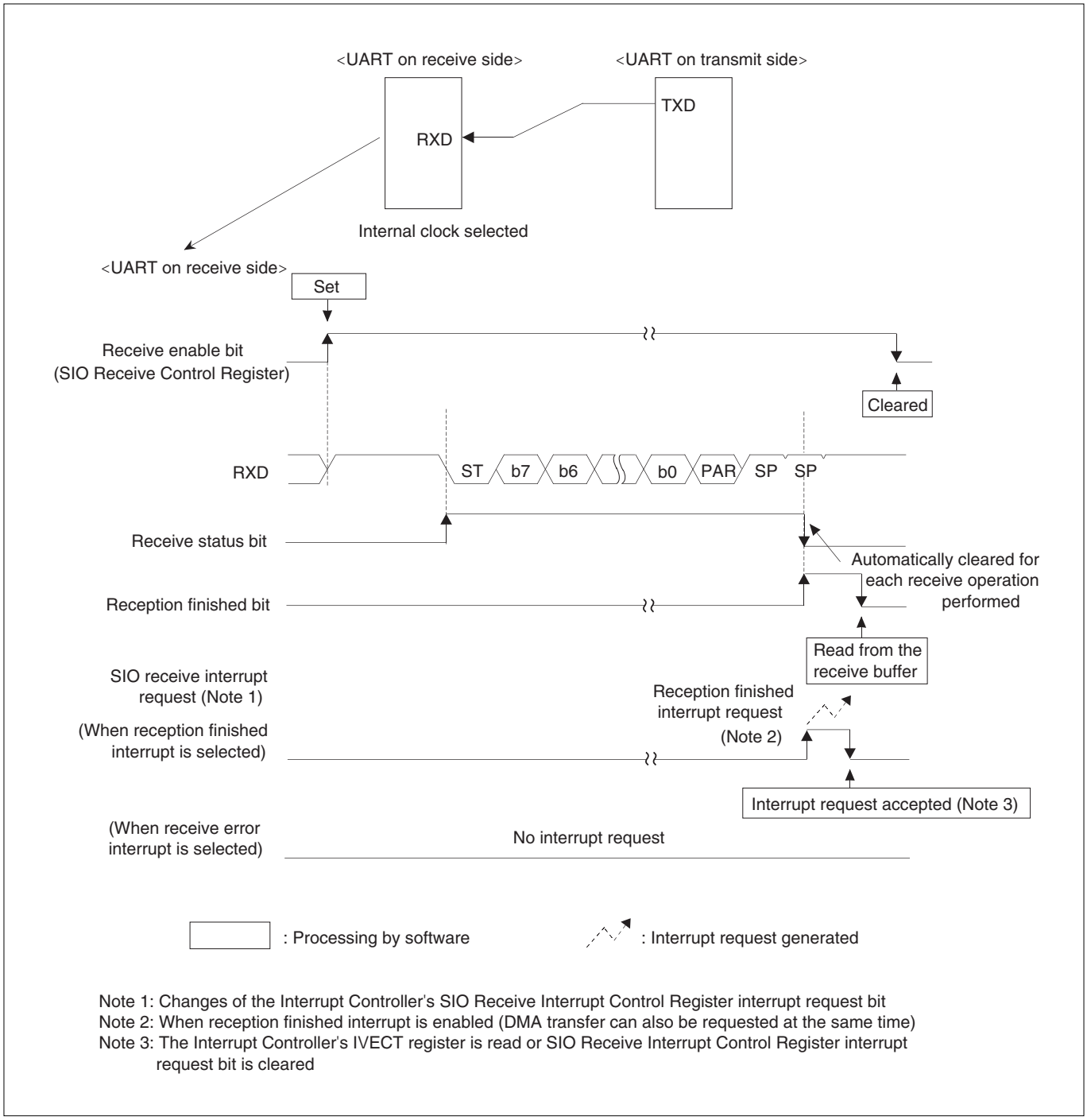


Figure 12.7.2 Receive Operation during UART Mode (Hardware Processing)

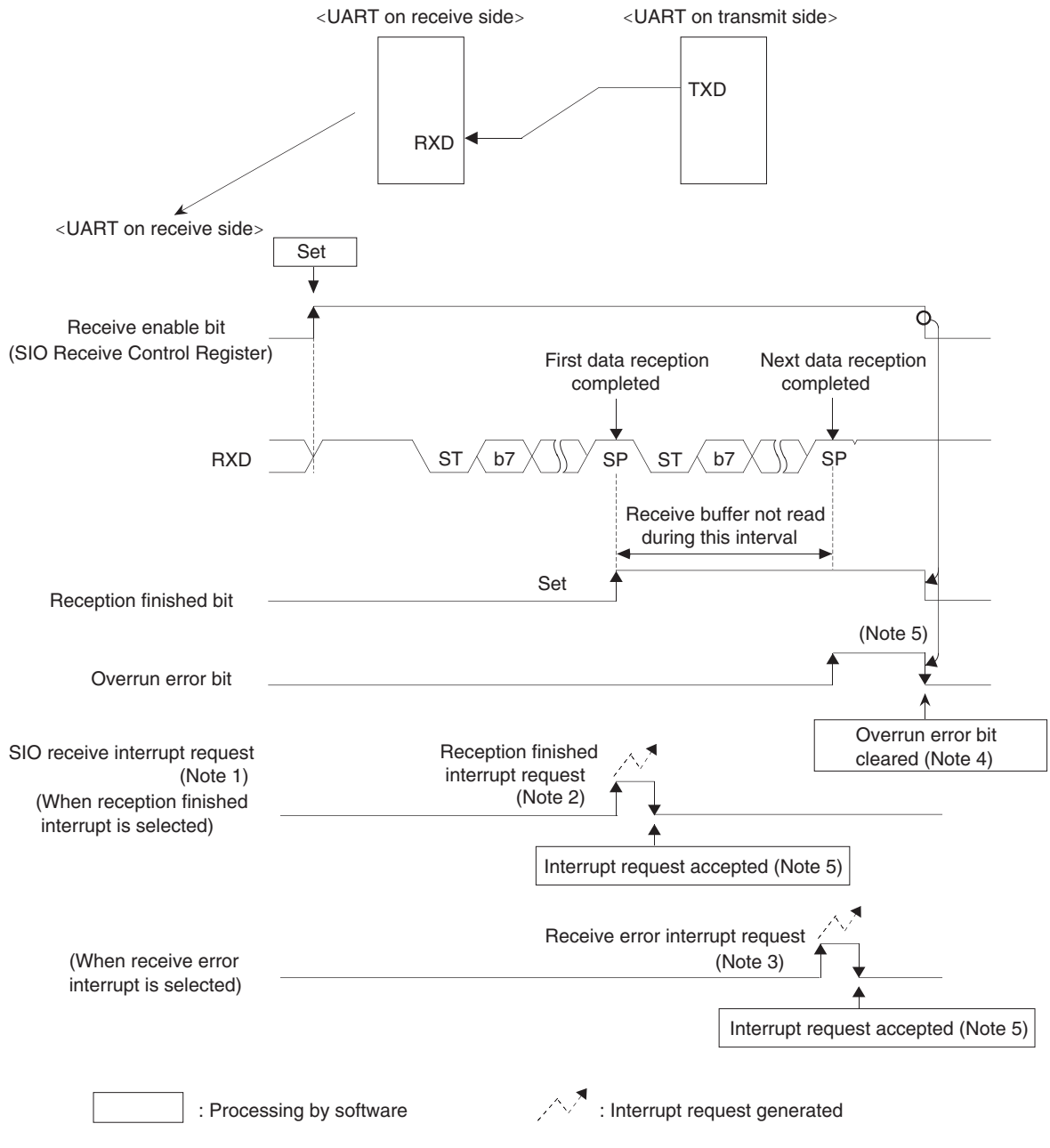
12.7.4 Example of UART Receive Operation

The following shows a typical receive operation in UART mode.



Note 1: Changes of the Interrupt Controller's SIO Receive Interrupt Control Register interrupt request bit
Note 2: When reception finished interrupt is enabled (DMA transfer can also be requested at the same time)
Note 3: The Interrupt Controller's IVECT register is read or SIO Receive Interrupt Control Register interrupt request bit is cleared

Figure 12.7.3 Example of UART Reception (When Received Normally)



Note 1: Changes of the Interrupt Controller's SIO Receive Interrupt Control Register interrupt request bit
Note 2: When reception finished interrupt is enabled
Note 3: When receive error interrupt is enabled
Note 4: This is done by clearing the receive enable bit to "0".
Note 5: The Interrupt Controller's IVECT register is read or SIO Receive Interrupt Control Register interrupt request bit is cleared

Figure 12.7.4 Example of UART Reception (When Overrun Error Occurred)

12.7.5 Start Bit Detection during UART Reception

The start bit is sampled synchronously with the internal BRG output. If the received signal remains "L" for 8 BRG output cycles after the falling edge of the start bit, the CPU recognizes that part of the received signal as the start bit and starts latching the received data another 8 cycles after that, beginning with the LSB (first bit). If some sampled part of the received signal is "H" before being determined to be the start bit, the CPU starts detecting the falling edge of the received signal again. Because the start bit is sampled synchronously with the internal BRG output, there is a delay equivalent to one BRG output cycle at maximum. The subsequent received data is latched into the internal circuit with that delayed timing.

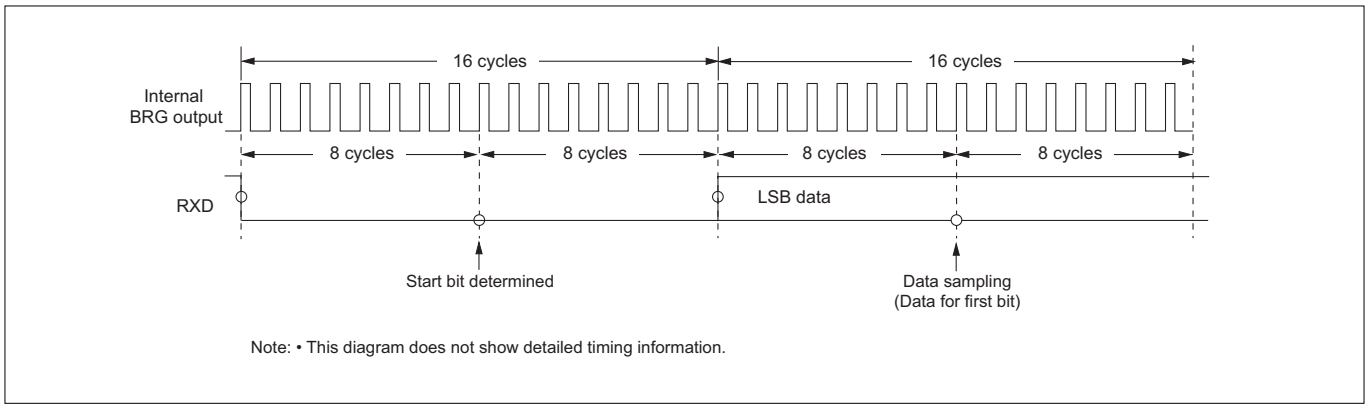


Figure 12.7.5 Start Bit Detection and Data Sampling Timing

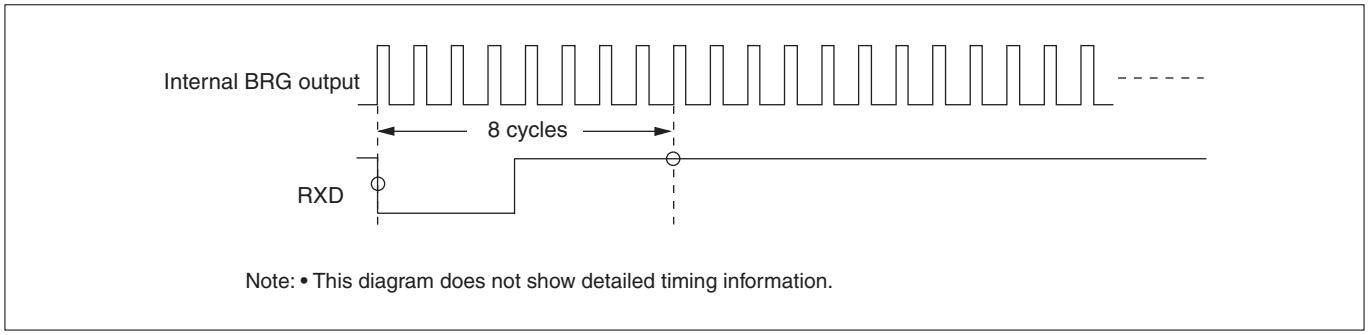


Figure 12.7.6 Example of an Invalid Start Bit (Not Received)

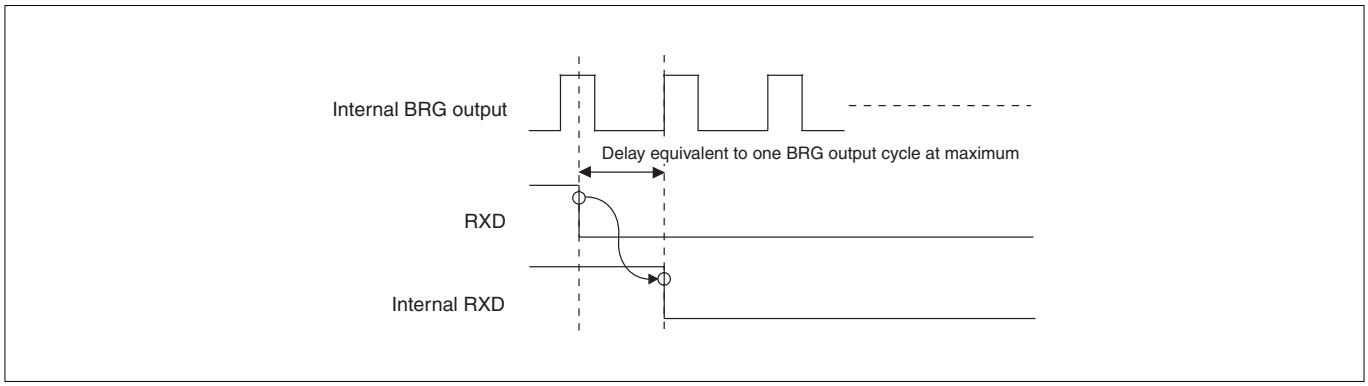


Figure 12.7.7 Delay in Receive Timing

12.8 Fixed Period Clock Output Function

When using SIO0 or SIO1 in UART mode, the relevant port (P84 or P87) can be switched for use as an SCLKO0 or SCLKO1 pin, respectively. That way, a BRG output clock divided by 2 can be output from the SCLKO pin.

Note: • This clock is output not just during data transfer.

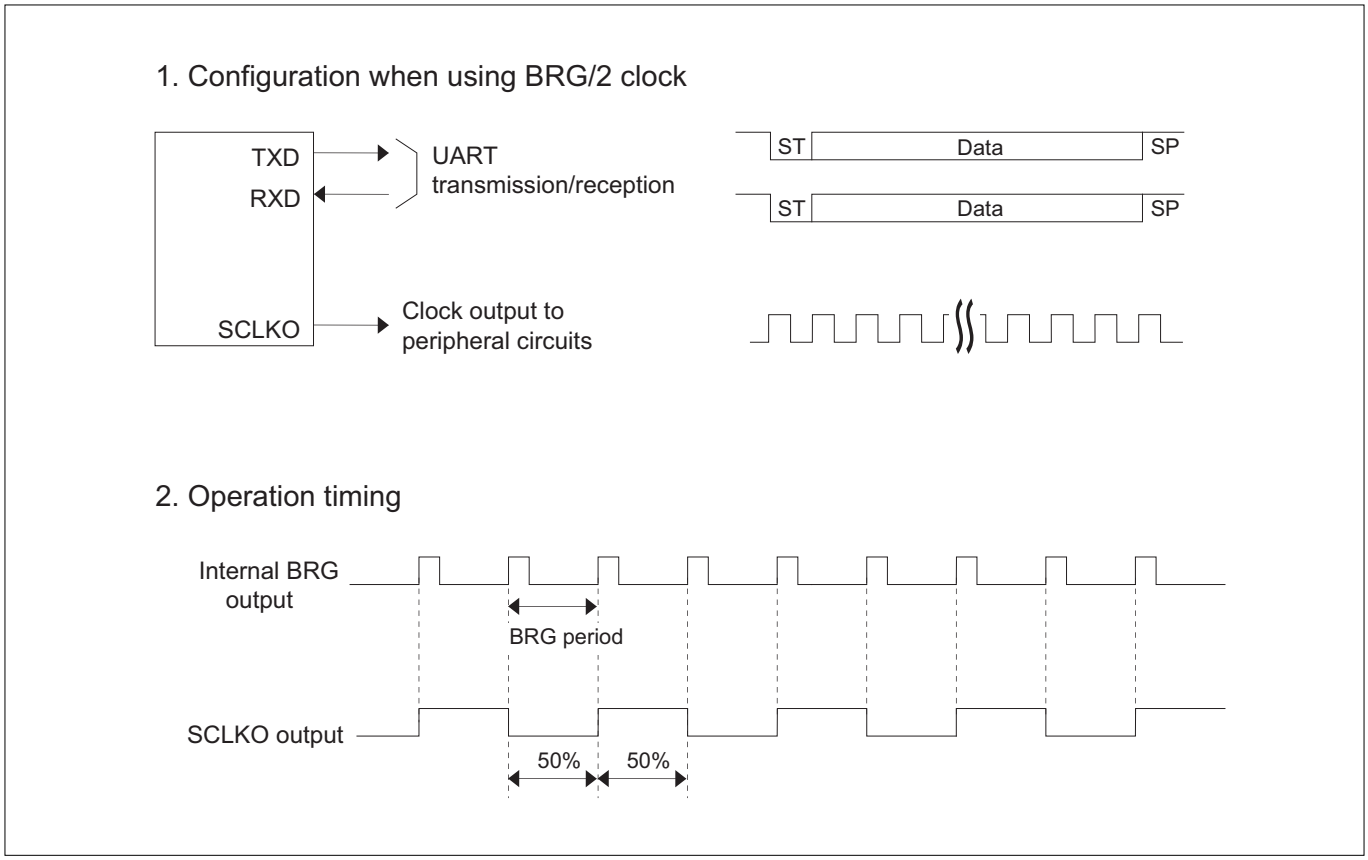


Figure 12.8.1 Example of Fixed Period Clock Output

12.9 Notes on Using UART Mode

• Settings of SIO Transmit/Receive Mode Register and SIO Baud Rate Register

The SIO Transmit/Receive Mode Register and SIO Baud Rate Register and the Transmit Control Register's BRG count source select bit must always be set when the serial interface is not operating. If a transmit or receive operation is in progress, wait until the transmit and receive operations are finished and then clear the transmit and receive enable bits before making changes.

• Settings of BRG (Baud Rate Register)

Writes to the SIO Baud Rate Register take effect in the next cycle after the BRG counter has finished counting. However, if the register is accessed for write while transmission and reception are disabled, the written value takes effect at the same time it is written.

• Transmission/reception using DMA

To transmit/receive data in DMA request mode, enable the DMAC to accept transfer requests (by setting the DMA Mode Register) before serial communication starts.

• About overrun error

If all bits of the next received data have been set in the SIO Receive Shift Register before reading out the SIO Receive Buffer Register (i.e., an overrun error occurred), the received data is not stored in the receive buffer register, with the previous received data retained in it. Once an overrun error occurs, although a receive operation continues, the subsequent received data is not stored in the receive buffer register. Before normal receive operation can be restarted, the receive enable bit must be temporarily cleared. And this is the only way that the overrun error flag can be cleared.

• Flags showing the status of UART receive operation

There are following flags that indicate the status of receive operation during UART mode:

- SIO Receive Control Register receive status bit
- SIO Receive Control Register reception finished bit
- SIO Receive Control Register receive error sum bit
- SIO Receive Control Register overrun error bit
- SIO Receive Control Register parity error bit
- SIO Receive Control Register framing error bit

The manner in which the reception finished bit and various error flags are cleared differs depending on whether an overrun error occurred, as described below.

[When an overrun error did not occur]
Cleared by reading out the lower byte of the receive buffer register or by clearing the receive enable bit.

[When an overrun error occurred]
Cleared by only clearing the receive enable bit.

• Switching from general-purpose to serial interface pin

When switching from general-purpose port to the serial interface pin by the port operation mode register, the terminal TXDn pin outputs "H" level.

This page is blank for reasons of layout.

CHAPTER 13

CAN MODULE

13.1 Outline of the CAN Module

The 32176 contains two-channel Full CAN modules compliant with CAN (Controller Area Network) Specification V2.0 B Active. These CAN modules each have 16 message slots and three mask registers, effective use of which helps to reduce the data processing load of the CPU.
The CAN modules are outlined below.

Table 13.1.1 Outline of the CAN Module

Item	Description
Protocol	CAN Specification V2.0 B Active
Number of message slots	Total 16 slots (14 global slots, two local slots)
Polarity	0: Dominant 1: Recessive
Acceptance filter (Function to receive only a range of IDs specified by receive ID filter)	Global mask: 1 Local mask: 2
Baud rate	1 time quantum (Tq) = (BRP + 1) / CPU clock (BRP: Baud Rate Prescaler set value) Baud rate = $\frac{1}{\text{Tq period} \times \text{number of Tq's for one bit}}$ ····· Max 1 Mbps (Note 1) BRP: 1–255 (0: inhibited) Number of Tq's for one bit = Synchronization Segment + Propagation Segment + Phase Segment 1 + Phase Segment 2 Synchronization Segment : 1Tq Propagation Segment: 1–8Tq Phase Segment 1: 1–8Tq Phase Segment 2: 1–8Tq (IPT = 1)
Remote frame automatic response function	The slot that received a remote frame responds by automatically sending a data frame.
Timestamp function	This function is implemented using a 16-bit counter. The count period is derived from the CAN bus bit period by dividing it by 1, 2, 3 or 4.
BasicCAN mode	Slot 14 and 15 can be alternately received as receive-only.
Transmit abort function	Transmit requests can be canceled.
Loopback function	The CAN module receives the data transmitted by the module itself.
Return bus off function	Error active mode is forcibly entered into after clearing the error counter.
Single shot function	Transmission is not retried even when it failed due to arbitration-lost or a transmit error.
DMA transfer function	DMA transfer request is generated when transmission failed or transmit/receive operation finished.
Self-diagnostic function	Communication module is diagnosed by communicating internally in the CAN module.

Note 1: The maximum allowable error of oscillation depends on the system configuration (e.g., bus length, clock error, CAN bus transceiver, sampling position and bit configuration).

Table 13.1.2 DMA Transfer Requests Generated by CAN

DMA Transfer Request by CAN	DMAC Input Channel
CAN0: Slot 0 transmission failed or slot 15 transmit/receive operation finished	DMA6
CAN0: Slot 1 transmission failed or slot 14 transmit/receive operation finished	DMA7
CAN1: Slot 0 transmission failed or slot 15 transmit/receive operation finished	DMA8
CAN1: Slot 1 transmission failed or slot 14 transmit/receive operation finished	DMA9

Table 13.1.3 Interrupt Requests Generated by CAN Modules

CAN Module Interrupt Request Source	ICU Interrupt Request Source
CAN0 transmission completed	CAN0 transmit/receive & error interrupt
CAN1 transmission completed	CAN1 transmit/receive & error interrupt
CAN0 reception completed	CAN0 transmit/receive & error interrupt
CAN1 reception completed	CAN1 transmit/receive & error interrupt
CAN0 bus error	CAN0 transmit/receive & error interrupt
CAN1 bus error	CAN1 transmit/receive & error interrupt
CAN0 error passive	CAN0 transmit/receive & error interrupt
CAN1 error passive	CAN1 transmit/receive & error interrupt
CAN0 bus off	CAN0 transmit/receive & error interrupt
CAN1 bus off	CAN1 transmit/receive & error interrupt
CAN0 single shot	CAN0 transmit/receive & error interrupt
CAN1 single shot	CAN1 transmit/receive & error interrupt

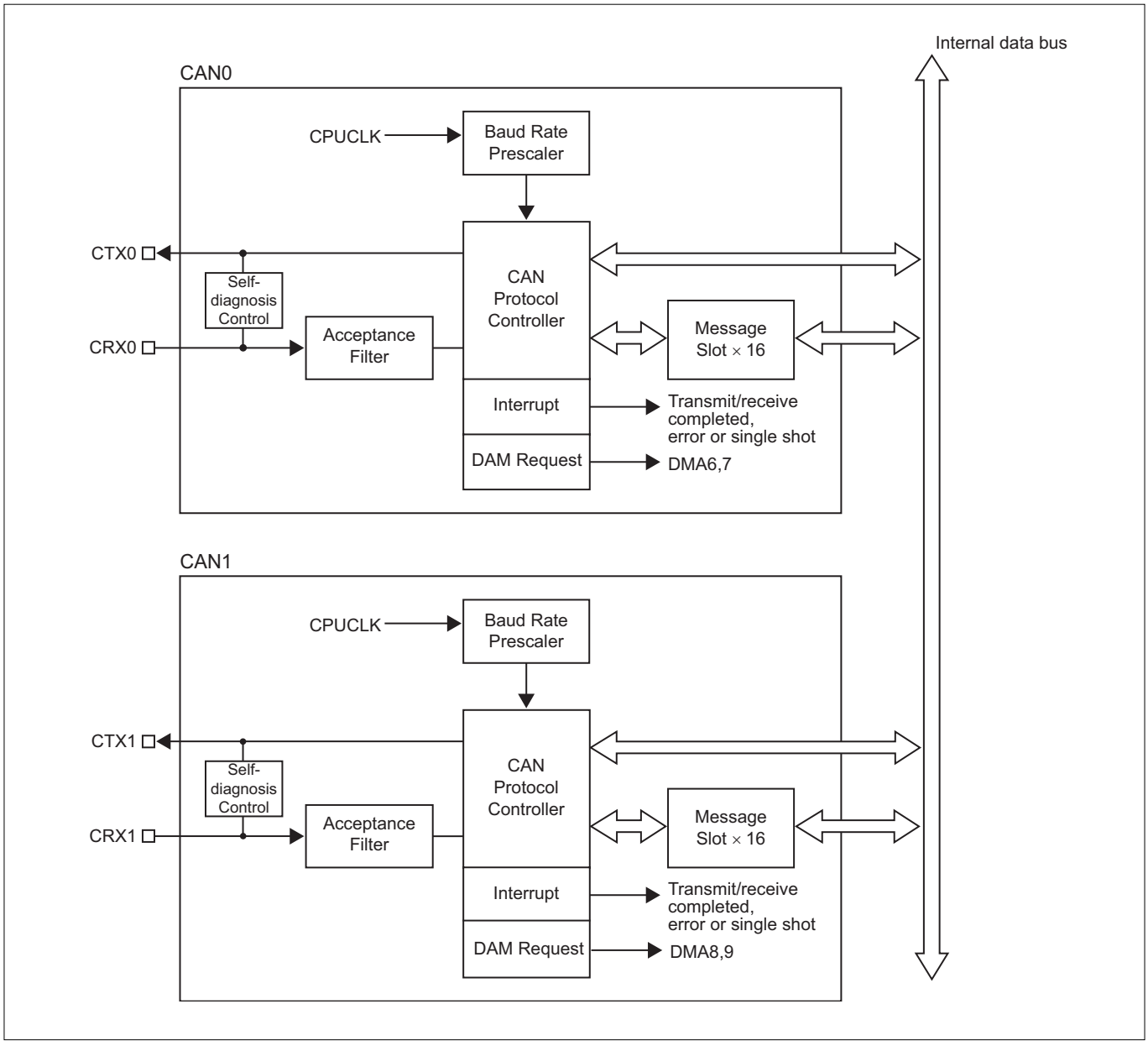


Figure 13.1.1 Block Diagram of the CAN Modules

13.2 CAN Module Related Registers

Shown below is a CAN module related register map.

CAN Module Related Register Map (1/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1000	CAN0 Control Register (CAN0CNT)		13-15
H'0080 1002	CAN0 Status Register (CAN0STAT)		13-18
H'0080 1004	CAN0 Extended ID Register (CAN0EXTID)		13-21
H'0080 1006	CAN0 Configuration Register (CAN0CONF)		13-22
H'0080 1008	CAN0 Timestamp Count Register (CAN0TSTMP)		13-24
H'0080 100A	CAN0 Receive Error Count Register (CAN0REC)	CAN0 Transmit Error Count Register (CAN0TEC)	13-25
H'0080 100C	CAN0 Slot Interrupt Request Status Register (CAN0SLIST)		13-29
H'0080 100E	(Use inhibited area)		
H'0080 1010	CAN0 Slot Interrupt Request Mask Register (CAN0SLIMK)		13-30
H'0080 1012	(Use inhibited area)		
H'0080 1014	CAN0 Error Interrupt Request Status Register (CAN0ERIST)	CAN0 Error Interrupt Request Mask Register (CAN0ERIMK)	13-31 13-32
H'0080 1016	CAN0 Baud Rate Prescaler (CAN0BRP)	CAN0 Cause of Error Register (CAN0EF)	13-26 13-45
H'0080 1018	CAN0 Mode Register (CAN0MOD)	CAN0 DMA Transfer Request Select Register (CAN0DMARQ)	13-47 13-48
⋮	(Use inhibited area)		
H'0080 1028	CAN0 Global Mask Register Standard ID0 (C0GMSKS0)	CAN0 Global Mask Register Standard ID1 (C0GMSKS1)	13-49
H'0080 102A	CAN0 Global Mask Register Extended ID0 (C0GMSKE0)	CAN0 Global Mask Register Extended ID1 (C0GMSKE1)	13-50
H'0080 102C	CAN0 Global Mask Register Extended ID2 (C0GMSKE2)	(Use inhibited area)	13-51
H'0080 102E	(Use inhibited area)		
H'0080 1030	CAN0 Local Mask Register A Standard ID0 (C0LMSKAS0)	CAN0 Local Mask Register A Standard ID1 (C0LMSKAS1)	13-49
H'0080 1032	CAN0 Local Mask Register A Extended ID0 (C0LMSKAE0)	CAN0 Local Mask Register A Extended ID1 (C0LMSKAE1)	13-50
H'0080 1034	CAN0 Local Mask Register A Extended ID2 (C0LMSKAE2)	(Use inhibited area)	13-51
H'0080 1036	(Use inhibited area)		
H'0080 1038	CAN0 Local Mask Register B Standard ID0 (C0LMSKBS0)	CAN0 Local Mask Register B Standard ID1 (C0LMSKBS1)	13-49
H'0080 103A	CAN0 Local Mask Register B Extended ID0 (C0LMSKBE0)	CAN0 Local Mask Register B Extended ID1 (C0LMSKBE1)	13-50
H'0080 103C	CAN0 Local Mask Register B Extended ID2 (C0LMSKBE2)	(Use inhibited area)	13-51
H'0080 103E	(Use inhibited area)		
H'0080 1040	CAN0 Single-Shot Mode Control Register (CAN0SSMODE)		13-53
H'0080 1042	(Use inhibited area)		
H'0080 1044	CAN0 Single-Shot Interrupt Request Status Register (CAN0SSIST)		13-33
H'0080 1046	(Use inhibited area)		

CAN Module Related Register Map (2/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1048	CAN0 Single-Shot Interrupt Request Mask Register (CAN0SSIMK)		13-34
	(Use inhibited area)		
H'0080 1050	CAN0 Message Slot 0 Control Register (C0MSL0CNT)	CAN0 Message Slot 1 Control Register (C0MSL1CNT)	13-54
H'0080 1052	CAN0 Message Slot 2 Control Register (C0MSL2CNT)	CAN0 Message Slot 3 Control Register (C0MSL3CNT)	13-54
H'0080 1054	CAN0 Message Slot 4 Control Register (C0MSL4CNT)	CAN0 Message Slot 5 Control Register (C0MSL5CNT)	13-54
H'0080 1056	CAN0 Message Slot 6 Control Register (C0MSL6CNT)	CAN0 Message Slot 7 Control Register (C0MSL7CNT)	13-54
H'0080 1058	CAN0 Message Slot 8 Control Register (C0MSL8CNT)	CAN0 Message Slot 9 Control Register (C0MSL9CNT)	13-54
H'0080 105A	CAN0 Message Slot 10 Control Register (C0MSL10CNT)	CAN0 Message Slot 11 Control Register (C0MSL11CNT)	13-54
H'0080 105C	CAN0 Message Slot 12 Control Register (C0MSL12CNT)	CAN0 Message Slot 13 Control Register (C0MSL13CNT)	13-54
H'0080 105E	CAN0 Message Slot 14 Control Register (C0MSL14CNT)	CAN0 Message Slot 15 Control Register (C0MSL15CNT)	13-54
	(Use inhibited area)		
H'0080 1100	CAN0 Message Slot 0 Standard ID0 (C0MSL0SID0)	CAN0 Message Slot 0 Standard ID1 (C0MSL0SID1)	13-58 13-59
H'0080 1102	CAN0 Message Slot 0 Extended ID0 (C0MSL0EID0)	CAN0 Message Slot 0 Extended ID1 (C0MSL0EID1)	13-60 13-61
H'0080 1104	CAN0 Message Slot 0 Extended ID2 (C0MSL0EID2)	CAN0 Message Slot 0 Data Length Register (C0MSL0DLC)	13-62 13-63
H'0080 1106	CAN0 Message Slot 0 Data 0 (C0MSL0DT0)	CAN0 Message Slot 0 Data 1 (C0MSL0DT1)	13-64 13-65
H'0080 1108	CAN0 Message Slot 0 Data 2 (C0MSL0DT2)	CAN0 Message Slot 0 Data 3 (C0MSL0DT3)	13-66 13-67
H'0080 110A	CAN0 Message Slot 0 Data 4 (C0MSL0DT4)	CAN0 Message Slot 0 Data 5 (C0MSL0DT5)	13-68 13-69
H'0080 110C	CAN0 Message Slot 0 Data 6 (C0MSL0DT6)	CAN0 Message Slot 0 Data 7 (C0MSL0DT7)	13-70 13-71
H'0080 110E	CAN0 Message Slot 0 Timestamp (C0MSL0TSP)		13-72
H'0080 1110	CAN0 Message Slot 1 Standard ID0 (C0MSL1SID0)	CAN0 Message Slot 1 Standard ID1 (C0MSL1SID1)	13-58 13-59
H'0080 1112	CAN0 Message Slot 1 Extended ID0 (C0MSL1EID0)	CAN0 Message Slot 1 Extended ID1 (C0MSL1EID1)	13-60 13-61
H'0080 1114	CAN0 Message Slot 1 Extended ID2 (C0MSL1EID2)	CAN0 Message Slot 1 Data Length Register (C0MSL1DLC)	13-62 13-63
H'0080 1116	CAN0 Message Slot 1 Data 0 (C0MSL1DT0)	CAN0 Message Slot 1 Data 1 (C0MSL1DT1)	13-64 13-65
H'0080 1118	CAN0 Message Slot 1 Data 2 (C0MSL1DT2)	CAN0 Message Slot 1 Data 3 (C0MSL1DT3)	13-66 13-67
H'0080 111A	CAN0 Message Slot 1 Data 4 (C0MSL1DT4)	CAN0 Message Slot 1 Data 5 (C0MSL1DT5)	13-68 13-69
H'0080 111C	CAN0 Message Slot 1 Data 6 (C0MSL1DT6)	CAN0 Message Slot 1 Data 7 (C0MSL1DT7)	13-70 13-71
H'0080 111E	CAN0 Message Slot 1 Timestamp (C0MSL1TSP)		13-72
H'0080 1120	CAN0 Message Slot 2 Standard ID0 (C0MSL2SID0)	CAN0 Message Slot 2 Standard ID1 (C0MSL2SID1)	13-58 13-59
H'0080 1122	CAN0 Message Slot 2 Extended ID0 (C0MSL2EID0)	CAN0 Message Slot 2 Extended ID1 (C0MSL2EID1)	13-60 13-61
H'0080 1124	CAN0 Message Slot 2 Extended ID2 (C0MSL2EID2)	CAN0 Message Slot 2 Data Length Register (C0MSL2DLC)	13-62 13-63
H'0080 1126	CAN0 Message Slot 2 Data 0 (C0MSL2DT0)	CAN0 Message Slot 2 Data 1 (C0MSL2DT1)	13-64 13-65
H'0080 1128	CAN0 Message Slot 2 Data 2 (C0MSL2DT2)	CAN0 Message Slot 2 Data 3 (C0MSL2DT3)	13-66 13-67
H'0080 112A	CAN0 Message Slot 2 Data 4 (C0MSL2DT4)	CAN0 Message Slot 2 Data 5 (C0MSL2DT5)	13-68 13-69
H'0080 112C	CAN0 Message Slot 2 Data 6 (C0MSL2DT6)	CAN0 Message Slot 2 Data 7 (C0MSL2DT7)	13-70 13-71
H'0080 112E	CAN0 Message Slot 2 Timestamp (C0MSL2TSP)		13-72

CAN Module Related Register Map (3/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1130	CAN0 Message Slot 3 Standard ID0 (C0MSL3SID0)	CAN0 Message Slot 3 Standard ID1 (C0MSL3SID1)	13-58 13-59
H'0080 1132	CAN0 Message Slot 3 Extended ID0 (C0MSL3EID0)	CAN0 Message Slot 3 Extended ID1 (C0MSL3EID1)	13-60 13-61
H'0080 1134	CAN0 Message Slot 3 Extended ID2 (C0MSL3EID2)	CAN0 Message Slot 3 Data Length Register (C0MSL3DLC)	13-62 13-63
H'0080 1136	CAN0 Message Slot 3 Data 0 (C0MSL3DT0)	CAN0 Message Slot 3 Data 1 (C0MSL3DT1)	13-64 13-65
H'0080 1138	CAN0 Message Slot 3 Data 2 (C0MSL3DT2)	CAN0 Message Slot 3 Data 3 (C0MSL3DT3)	13-66 13-67
H'0080 113A	CAN0 Message Slot 3 Data 4 (C0MSL3DT4)	CAN0 Message Slot 3 Data 5 (C0MSL3DT5)	13-68 13-69
H'0080 113C	CAN0 Message Slot 3 Data 6 (C0MSL3DT6)	CAN0 Message Slot 3 Data 7 (C0MSL3DT7)	13-70 13-71
H'0080 113E	CAN0 Message Slot 3 Timestamp (C0MSL3TSP)		13-72
H'0080 1140	CAN0 Message Slot 4 Standard ID0 (C0MSL4SID0)	CAN0 Message Slot 4 Standard ID1 (C0MSL4SID1)	13-58 13-59
H'0080 1142	CAN0 Message Slot 4 Extended ID0 (C0MSL4EID0)	CAN0 Message Slot 4 Extended ID1 (C0MSL4EID1)	13-60 13-61
H'0080 1144	CAN0 Message Slot 4 Extended ID2 (C0MSL4EID2)	CAN0 Message Slot 4 Data Length Register (C0MSL4DLC)	13-62 13-63
H'0080 1146	CAN0 Message Slot 4 Data 0 (C0MSL4DT0)	CAN0 Message Slot 4 Data 1 (C0MSL4DT1)	13-64 13-65
H'0080 1148	CAN0 Message Slot 4 Data 2 (C0MSL4DT2)	CAN0 Message Slot 4 Data 3 (C0MSL4DT3)	13-66 13-67
H'0080 114A	CAN0 Message Slot 4 Data 4 (C0MSL4DT4)	CAN0 Message Slot 4 Data 5 (C0MSL4DT5)	13-68 13-69
H'0080 114C	CAN0 Message Slot 4 Data 6 (C0MSL4DT6)	CAN0 Message Slot 4 Data 7 (C0MSL4DT7)	13-70 13-71
H'0080 114E	CAN0 Message Slot 4 Timestamp (C0MSL4TSP)		13-72
H'0080 1150	CAN0 Message Slot 5 Standard ID0 (C0MSL5SID0)	CAN0 Message Slot 5 Standard ID1 (C0MSL5SID1)	13-58 13-59
H'0080 1152	CAN0 Message Slot 5 Extended ID0 (C0MSL5EID0)	CAN0 Message Slot 5 Extended ID1 (C0MSL5EID1)	13-60 13-61
H'0080 1154	CAN0 Message Slot 5 Extended ID2 (C0MSL5EID2)	CAN0 Message Slot 5 Data Length Register (C0MSL5DLC)	13-62 13-63
H'0080 1156	CAN0 Message Slot 5 Data 0 (C0MSL5DT0)	CAN0 Message Slot 5 Data 1 (C0MSL5DT1)	13-64 13-65
H'0080 1158	CAN0 Message Slot 5 Data 2 (C0MSL5DT2)	CAN0 Message Slot 5 Data 3 (C0MSL5DT3)	13-66 13-67
H'0080 115A	CAN0 Message Slot 5 Data 4 (C0MSL5DT4)	CAN0 Message Slot 5 Data 5 (C0MSL5DT5)	13-68 13-69
H'0080 115C	CAN0 Message Slot 5 Data 6 (C0MSL5DT6)	CAN0 Message Slot 5 Data 7 (C0MSL5DT7)	13-70 13-71
H'0080 115E	CAN0 Message Slot 5 Timestamp (C0MSL5TSP)		13-72
H'0080 1160	CAN0 Message Slot 6 Standard ID0 (C0MSL6SID0)	CAN0 Message Slot 6 Standard ID1 (C0MSL6SID1)	13-58 13-59
H'0080 1162	CAN0 Message Slot 6 Extended ID0 (C0MSL6EID0)	CAN0 Message Slot 6 Extended ID1 (C0MSL6EID1)	13-60 13-61
H'0080 1164	CAN0 Message Slot 6 Extended ID2 (C0MSL6EID2)	CAN0 Message Slot 6 Data Length Register (C0MSL6DLC)	13-62 13-63
H'0080 1166	CAN0 Message Slot 6 Data 0 (C0MSL6DT0)	CAN0 Message Slot 6 Data 1 (C0MSL6DT1)	13-64 13-65
H'0080 1168	CAN0 Message Slot 6 Data 2 (C0MSL6DT2)	CAN0 Message Slot 6 Data 3 (C0MSL6DT3)	13-66 13-67
H'0080 116A	CAN0 Message Slot 6 Data 4 (C0MSL6DT4)	CAN0 Message Slot 6 Data 5 (C0MSL6DT5)	13-68 13-69
H'0080 116C	CAN0 Message Slot 6 Data 6 (C0MSL6DT6)	CAN0 Message Slot 6 Data 7 (C0MSL6DT7)	13-70 13-71
H'0080 116E	CAN0 Message Slot 6 Timestamp (C0MSL6TSP)		13-72

CAN Module Related Register Map (4/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1170	CAN0 Message Slot 7 Standard ID0 (C0MSL7SID0)	CAN0 Message Slot 7 Standard ID1 (C0MSL7SID1)	13-58 13-59
H'0080 1172	CAN0 Message Slot 7 Extended ID0 (C0MSL7EID0)	CAN0 Message Slot 7 Extended ID1 (C0MSL7EID1)	13-60 13-61
H'0080 1174	CAN0 Message Slot 7 Extended ID2 (C0MSL7EID2)	CAN0 Message Slot 7 Data Length Register (C0MSL7DLC)	13-62 13-63
H'0080 1176	CAN0 Message Slot 7 Data 0 (C0MSL7DT0)	CAN0 Message Slot 7 Data 1 (C0MSL7DT1)	13-64 13-65
H'0080 1178	CAN0 Message Slot 7 Data 2 (C0MSL7DT2)	CAN0 Message Slot 7 Data 3 (C0MSL7DT3)	13-66 13-67
H'0080 117A	CAN0 Message Slot 7 Data 4 (C0MSL7DT4)	CAN0 Message Slot 7 Data 5 (C0MSL7DT5)	13-68 13-69
H'0080 117C	CAN0 Message Slot 7 Data 6 (C0MSL7DT6)	CAN0 Message Slot 7 Data 7 (C0MSL7DT7)	13-70 13-71
H'0080 117E	CAN0 Message Slot 7 Timestamp (C0MSL7TSP)		13-72
H'0080 1180	CAN0 Message Slot 8 Standard ID0 (C0MSL8SID0)	CAN0 Message Slot 8 Standard ID1 (C0MSL8SID1)	13-58 13-59
H'0080 1182	CAN0 Message Slot 8 Extended ID0 (C0MSL8EID0)	CAN0 Message Slot 8 Extended ID1 (C0MSL8EID1)	13-60 13-61
H'0080 1184	CAN0 Message Slot 8 Extended ID2 (C0MSL8EID2)	CAN0 Message Slot 8 Data Length Register (C0MSL8DLC)	13-62 13-63
H'0080 1186	CAN0 Message Slot 8 Data 0 (C0MSL8DT0)	CAN0 Message Slot 8 Data 1 (C0MSL8DT1)	13-64 13-65
H'0080 1188	CAN0 Message Slot 8 Data 2 (C0MSL8DT2)	CAN0 Message Slot 8 Data 3 (C0MSL8DT3)	13-66 13-67
H'0080 118A	CAN0 Message Slot 8 Data 4 (C0MSL8DT4)	CAN0 Message Slot 8 Data 5 (C0MSL8DT5)	13-68 13-69
H'0080 118C	CAN0 Message Slot 8 Data 6 (C0MSL8DT6)	CAN0 Message Slot 8 Data 7 (C0MSL8DT7)	13-70 13-71
H'0080 118E	CAN0 Message Slot 8 Timestamp (C0MSL8TSP)		13-72
H'0080 1190	CAN0 Message Slot 9 Standard ID0 (C0MSL9SID0)	CAN0 Message Slot 9 Standard ID1 (C0MSL9SID1)	13-58 13-59
H'0080 1192	CAN0 Message Slot 9 Extended ID0 (C0MSL9EID0)	CAN0 Message Slot 9 Extended ID1 (C0MSL9EID1)	13-60 13-61
H'0080 1194	CAN0 Message Slot 9 Extended ID2 (C0MSL9EID2)	CAN0 Message Slot 9 Data Length Register (C0MSL9DLC)	13-62 13-63
H'0080 1196	CAN0 Message Slot 9 Data 0 (C0MSL9DT0)	CAN0 Message Slot 9 Data 1 (C0MSL9DT1)	13-64 13-65
H'0080 1198	CAN0 Message Slot 9 Data 2 (C0MSL9DT2)	CAN0 Message Slot 9 Data 3 (C0MSL9DT3)	13-66 13-67
H'0080 119A	CAN0 Message Slot 9 Data 4 (C0MSL9DT4)	CAN0 Message Slot 9 Data 5 (C0MSL9DT5)	13-68 13-69
H'0080 119C	CAN0 Message Slot 9 Data 6 (C0MSL9DT6)	CAN0 Message Slot 9 Data 7 (C0MSL9DT7)	13-70 13-71
H'0080 119E	CAN0 Message Slot 9 Timestamp (C0MSL9TSP)		13-72
H'0080 11A0	CAN0 Message Slot 10 Standard ID0 (C0MSL10SID0)	CAN0 Message Slot 10 Standard ID1 (C0MSL10SID1)	13-58 13-59
H'0080 11A2	CAN0 Message Slot 10 Extended ID0 (C0MSL10EID0)	CAN0 Message Slot 10 Extended ID1 (C0MSL10EID1)	13-60 13-61
H'0080 11A4	CAN0 Message Slot 10 Extended ID2 (C0MSL10EID2)	CAN0 Message Slot 10 Data Length Register (C0MSL10DLC)	13-62 13-63
H'0080 11A6	CAN0 Message Slot 10 Data 0 (C0MSL10DT0)	CAN0 Message Slot 10 Data 1 (C0MSL10DT1)	13-64 13-65
H'0080 11A8	CAN0 Message Slot 10 Data 2 (C0MSL10DT2)	CAN0 Message Slot 10 Data 3 (C0MSL10DT3)	13-66 13-67
H'0080 11AA	CAN0 Message Slot 10 Data 4 (C0MSL10DT4)	CAN0 Message Slot 10 Data 5 (C0MSL10DT5)	13-68 13-69
H'0080 11AC	CAN0 Message Slot 10 Data 6 (C0MSL10DT6)	CAN0 Message Slot 10 Data 7 (C0MSL10DT7)	13-70 13-71
H'0080 11AE	CAN0 Message Slot 10 Timestamp (C0MSL10TSP)		13-72

CAN Module Related Register Map (5/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 11B0	CAN0 Message Slot 11 Standard ID0 (C0MSL11SID0)	CAN0 Message Slot 11 Standard ID1 (C0MSL11SID1)	13-58 13-59
H'0080 11B2	CAN0 Message Slot 11 Extended ID0 (C0MSL11EID0)	CAN0 Message Slot 11 Extended ID1 (C0MSL11EID1)	13-60 13-61
H'0080 11B4	CAN0 Message Slot 11 Extended ID2 (C0MSL11EID2)	CAN0 Message Slot 11 Data Length Register (C0MSL11DLC)	13-62 13-63
H'0080 11B6	CAN0 Message Slot 11 Data 0 (C0MSL11DT0)	CAN0 Message Slot 11 Data 1 (C0MSL11DT1)	13-64 13-65
H'0080 11B8	CAN0 Message Slot 11 Data 2 (C0MSL11DT2)	CAN0 Message Slot 11 Data 3 (C0MSL11DT3)	13-66 13-67
H'0080 11BA	CAN0 Message Slot 11 Data 4 (C0MSL11DT4)	CAN0 Message Slot 11 Data 5 (C0MSL11DT5)	13-68 13-69
H'0080 11BC	CAN0 Message Slot 11 Data 6 (C0MSL11DT6)	CAN0 Message Slot 11 Data 7 (C0MSL11DT7)	13-70 13-71
H'0080 11BE	CAN0 Message Slot 11 Timestamp (C0MSL11TSP)		13-72
H'0080 11C0	CAN0 Message Slot 12 Standard ID0 (C0MSL12SID0)	CAN0 Message Slot 12 Standard ID1 (C0MSL12SID1)	13-58 13-59
H'0080 11C2	CAN0 Message Slot 12 Extended ID0 (C0MSL12EID0)	CAN0 Message Slot 12 Extended ID1 (C0MSL12EID1)	13-60 13-61
H'0080 11C4	CAN0 Message Slot 12 Extended ID2 (C0MSL12EID2)	CAN0 Message Slot 12 Data Length Register (C0MSL12DLC)	13-62 13-63
H'0080 11C6	CAN0 Message Slot 12 Data 0 (C0MSL12DT0)	CAN0 Message Slot 12 Data 1 (C0MSL12DT1)	13-64 13-65
H'0080 11C8	CAN0 Message Slot 12 Data 2 (C0MSL12DT2)	CAN0 Message Slot 12 Data 3 (C0MSL12DT3)	13-66 13-67
H'0080 11CA	CAN0 Message Slot 12 Data 4 (C0MSL12DT4)	CAN0 Message Slot 12 Data 5 (C0MSL12DT5)	13-68 13-69
H'0080 11CC	CAN0 Message Slot 12 Data 6 (C0MSL12DT6)	CAN0 Message Slot 12 Data 7 (C0MSL12DT7)	13-70 13-71
H'0080 11CE	CAN0 Message Slot 12 Timestamp (C0MSL12TSP)		13-72
H'0080 11D0	CAN0 Message Slot 13 Standard ID0 (C0MSL13SID0)	CAN0 Message Slot 13 Standard ID1 (C0MSL13SID1)	13-58 13-59
H'0080 11D2	CAN0 Message Slot 13 Extended ID0 (C0MSL13EID0)	CAN0 Message Slot 13 Extended ID1 (C0MSL13EID1)	13-60 13-61
H'0080 11D4	CAN0 Message Slot 13 Extended ID2 (C0MSL13EID2)	CAN0 Message Slot 13 Data Length Register (C0MSL13DLC)	13-62 13-63
H'0080 11D6	CAN0 Message Slot 13 Data 0 (C0MSL13DT0)	CAN0 Message Slot 13 Data 1 (C0MSL13DT1)	13-64 13-65
H'0080 11D8	CAN0 Message Slot 13 Data 2 (C0MSL13DT2)	CAN0 Message Slot 13 Data 3 (C0MSL13DT3)	13-66 13-67
H'0080 11DA	CAN0 Message Slot 13 Data 4 (C0MSL13DT4)	CAN0 Message Slot 13 Data 5 (C0MSL13DT5)	13-68 13-69
H'0080 11DC	CAN0 Message Slot 13 Data 6 (C0MSL13DT6)	CAN0 Message Slot 13 Data 7 (C0MSL13DT7)	13-70 13-71
H'0080 11DE	CAN0 Message Slot 13 Timestamp (C0MSL13TSP)		13-72
H'0080 11E0	CAN0 Message Slot 14 Standard ID0 (C0MSL14SID0)	CAN0 Message Slot 14 Standard ID1 (C0MSL14SID1)	13-58 13-59
H'0080 11E2	CAN0 Message Slot 14 Extended ID0 (C0MSL14EID0)	CAN0 Message Slot 14 Extended ID1 (C0MSL14EID1)	13-60 13-61
H'0080 11E4	CAN0 Message Slot 14 Extended ID2 (C0MSL14EID2)	CAN0 Message Slot 14 Data Length Register (C0MSL14DLC)	13-62 13-63
H'0080 11E6	CAN0 Message Slot 14 Data 0 (C0MSL14DT0)	CAN0 Message Slot 14 Data 1 (C0MSL14DT1)	13-64 13-65
H'0080 11E8	CAN0 Message Slot 14 Data 2 (C0MSL14DT2)	CAN0 Message Slot 14 Data 3 (C0MSL14DT3)	13-66 13-67
H'0080 11EA	CAN0 Message Slot 14 Data 4 (C0MSL14DT4)	CAN0 Message Slot 14 Data 5 (C0MSL14DT5)	13-68 13-69
H'0080 11EC	CAN0 Message Slot 14 Data 6 (C0MSL14DT6)	CAN0 Message Slot 14 Data 7 (C0MSL14DT7)	13-70 13-71
H'0080 11EE	CAN0 Message Slot 14 Timestamp (C0MSL14TSP)		13-72

CAN Module Related Register Map (6/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 11F0	CAN0 Message Slot 15 Standard ID0 (C0MSL15SID0)	CAN0 Message Slot 15 Standard ID1 (C0MSL15SID1)	13-58 13-59
H'0080 11F2	CAN0 Message Slot 15 Extended ID0 (C0MSL15EID0)	CAN0 Message Slot 15 Extended ID1 (C0MSL15EID1)	13-60 13-61
H'0080 11F4	CAN0 Message Slot 15 Extended ID2 (C0MSL15EID2)	CAN0 Message Slot 15 Data Length Register (C0MSL15DLC)	13-62 13-63
H'0080 11F6	CAN0 Message Slot 15 Data 0 (C0MSL15DT0)	CAN0 Message Slot 15 Data 1 (C0MSL15DT1)	13-64 13-65
H'0080 11F8	CAN0 Message Slot 15 Data 2 (C0MSL15DT2)	CAN0 Message Slot 15 Data 3 (C0MSL15DT3)	13-66 13-67
H'0080 11FA	CAN0 Message Slot 15 Data 4 (C0MSL15DT4)	CAN0 Message Slot 15 Data 5 (C0MSL15DT5)	13-68 13-69
H'0080 11FC	CAN0 Message Slot 15 Data 6 (C0MSL15DT6)	CAN0 Message Slot 15 Data 7 (C0MSL15DT7)	13-70 13-71
H'0080 11FE	CAN0 Message Slot 15 Timestamp (C0MSL15TSP)		13-72
	(Use inhibited area)		
H'0080 1400	CAN1 Control Register (CAN1CNT)		13-15
H'0080 1402	CAN1 Status Register (CAN1STAT)		13-18
H'0080 1404	CAN1 Extended ID Register (CAN1EXTID)		13-21
H'0080 1406	CAN1 Configuration Register (CAN1CONF)		13-22
H'0080 1408	CAN1 Timestamp Count Register (CAN1TSTMP)		13-24
H'0080 140A	CAN1 Receive Error Count Register (CAN1REC)	CAN1 Transmit Error Count Register (CAN1TEC)	13-25
H'0080 140C	CAN1 Slot Interrupt Request Status Register (CAN1SLIST)		13-29
H'0080 140E	(Use inhibited area)		
H'0080 1410	CAN1 Slot Interrupt Request Mask Register (CAN1SLIMK)		13-30
H'0080 1412	(Use inhibited area)		
H'0080 1414	CAN1 Error Interrupt Request Status Register (CAN1ERIST)	CAN1 Error Interrupt Request Mask Register (CAN1ERIMK)	13-31 13-32
H'0080 1416	CAN1 Baud Rate Prescaler (CAN1BRP)	CAN1 Cause of Error Register (CAN1EF)	13-26 13-45
H'0080 1418	CAN1 Mode Register (CAN1MOD)	CAN1 DMA Transfer Request Select Register (CAN1DMARQ)	13-47 13-48
	(Use inhibited area)		
H'0080 1428	CAN1 Global Mask Register Standard ID0 (C1GMSKS0)	CAN1 Global Mask Register Standard ID1 (C1GMSKS1)	13-49
H'0080 142A	CAN1 Global Mask Register Extended ID0 (C1GMSKE0)	CAN1 Global Mask Register Extended ID1 (C1GMSKE1)	13-50
H'0080 142C	CAN1 Global Mask Register Extended ID2 (C1GMSKE2)	(Use inhibited area)	13-51
H'0080 142E	(Use inhibited area)		
H'0080 1430	CAN1 Local Mask Register A Standard ID0 (C1LMSKAS0)	CAN1 Local Mask Register A Standard ID1 (C1LMSKAS1)	13-49
H'0080 1432	CAN1 Local Mask Register A Extended ID0 (C1LMSKAE0)	CAN1 Local Mask Register A Extended ID1 (C1LMSKAE1)	13-50
H'0080 1434	CAN1 Local Mask Register A Extended ID2 (C1LMSKAE2)	(Use inhibited area)	13-51
H'0080 1436	(Use inhibited area)		
H'0080 1438	CAN1 Local Mask Register B Standard ID0 (C1LMSKBS0)	CAN1 Local Mask Register B Standard ID1 (C1LMSKBS1)	13-49
H'0080 143A	CAN1 Local Mask Register B Extended ID0 (C1LMSKBE0)	CAN1 Local Mask Register B Extended ID1 (C1LMSKBE1)	13-50
H'0080 143C	CAN1 Local Mask Register B Extended ID2 (C1LMSKBE2)	(Use inhibited area)	13-51
H'0080 143E	(Use inhibited area)		

CAN Module Related Register Map (7/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1440	CAN1 Single-Shot Mode Control Register (CAN1SSMODE)		13-53
H'0080 1442	(Use inhibited area)		
H'0080 1444	CAN1 Single-Shot Interrupt Request Status Register (CAN1SSIST)		13-33
H'0080 1446	(Use inhibited area)		
H'0080 1448	CAN1 Single-Shot Interrupt Request Mask Register (CAN1SSIMK)		13-34
	(Use inhibited area)		
H'0080 1450	CAN1 Message Slot 0 Control Register (C1MSL0CNT)	CAN1 Message Slot 1 Control Register (C1MSL1CNT)	13-54
H'0080 1452	CAN1 Message Slot 2 Control Register (C1MSL2CNT)	CAN1 Message Slot 3 Control Register (C1MSL3CNT)	13-54
H'0080 1454	CAN1 Message Slot 4 Control Register (C1MSL4CNT)	CAN1 Message Slot 5 Control Register (C1MSL5CNT)	13-54
H'0080 1456	CAN1 Message Slot 6 Control Register (C1MSL6CNT)	CAN1 Message Slot 7 Control Register (C1MSL7CNT)	13-54
H'0080 1458	CAN1 Message Slot 8 Control Register (C1MSL8CNT)	CAN1 Message Slot 9 Control Register (C1MSL9CNT)	13-54
H'0080 145A	CAN1 Message Slot 10 Control Register (C1MSL10CNT)	CAN1 Message Slot 11 Control Register (C1MSL11CNT)	13-54
H'0080 145C	CAN1 Message Slot 12 Control Register (C1MSL12CNT)	CAN1 Message Slot 13 Control Register (C1MSL13CNT)	13-54
H'0080 145E	CAN1 Message Slot 14 Control Register (C1MSL14CNT)	CAN1 Message Slot 15 Control Register (C1MSL15CNT)	13-54
	(Use inhibited area)		
H'0080 1500	CAN1 Message Slot 0 Standard ID0 (C1MSL0SID0)	CAN1 Message Slot 0 Standard ID1 (C1MSL0SID1)	13-58 13-59
H'0080 1502	CAN1 Message Slot 0 Extended ID0 (C1MSL0EID0)	CAN1 Message Slot 0 Extended ID1 (C1MSL0EID1)	13-60 13-61
H'0080 1504	CAN1 Message Slot 0 Extended ID2 (C1MSL0EID2)	CAN1 Message Slot 0 Data Length Register (C1MSL0DLC)	13-62 13-63
H'0080 1506	CAN1 Message Slot 0 Data 0 (C1MSL0DT0)	CAN1 Message Slot 0 Data 1 (C1MSL0DT1)	13-64 13-65
H'0080 1508	CAN1 Message Slot 0 Data 2 (C1MSL0DT2)	CAN1 Message Slot 0 Data 3 (C1MSL0DT3)	13-66 13-67
H'0080 150A	CAN1 Message Slot 0 Data 4 (C1MSL0DT4)	CAN1 Message Slot 0 Data 5 (C1MSL0DT5)	13-68 13-69
H'0080 150C	CAN1 Message Slot 0 Data 6 (C1MSL0DT6)	CAN1 Message Slot 0 Data 7 (C1MSL0DT7)	13-70 13-71
H'0080 150E	CAN1 Message Slot 0 Timestamp (C1MSL0TSP)		13-72
H'0080 1510	CAN1 Message Slot 1 Standard ID0 (C1MSL1SID0)	CAN1 Message Slot 1 Standard ID1 (C1MSL1SID1)	13-58 13-59
H'0080 1512	CAN1 Message Slot 1 Extended ID0 (C1MSL1EID0)	CAN1 Message Slot 1 Extended ID1 (C1MSL1EID1)	13-60 13-61
H'0080 1514	CAN1 Message Slot 1 Extended ID2 (C1MSL1EID2)	CAN1 Message Slot 1 Data Length Register (C1MSL1DLC)	13-62 13-63
H'0080 1516	CAN1 Message Slot 1 Data 0 (C1MSL1DT0)	CAN1 Message Slot 1 Data 1 (C1MSL1DT1)	13-64 13-65
H'0080 1518	CAN1 Message Slot 1 Data 2 (C1MSL1DT2)	CAN1 Message Slot 1 Data 3 (C1MSL1DT3)	13-66 13-67
H'0080 151A	CAN1 Message Slot 1 Data 4 (C1MSL1DT4)	CAN1 Message Slot 1 Data 5 (C1MSL1DT5)	13-68 13-69
H'0080 151C	CAN1 Message Slot 1 Data 6 (C1MSL1DT6)	CAN1 Message Slot 1 Data 7 (C1MSL1DT7)	13-70 13-71
H'0080 151E	CAN1 Message Slot 1 Timestamp (C1MSL1TSP)		13-72

CAN Module Related Register Map (8/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1520	CAN1 Message Slot 2 Standard ID0 (C1MSL2SID0)	CAN1 Message Slot 2 Standard ID1 (C1MSL2SID1)	13-58 13-59
H'0080 1522	CAN1 Message Slot 2 Extended ID0 (C1MSL2EID0)	CAN1 Message Slot 2 Extended ID1 (C1MSL2EID1)	13-60 13-61
H'0080 1524	CAN1 Message Slot 2 Extended ID2 (C1MSL2EID2)	CAN1 Message Slot 2 Data Length Register (C1MSL2DLC)	13-62 13-63
H'0080 1526	CAN1 Message Slot 2 Data 0 (C1MSL2DT0)	CAN1 Message Slot 2 Data 1 (C1MSL2DT1)	13-64 13-65
H'0080 1528	CAN1 Message Slot 2 Data 2 (C1MSL2DT2)	CAN1 Message Slot 2 Data 3 (C1MSL2DT3)	13-66 13-67
H'0080 152A	CAN1 Message Slot 2 Data 4 (C1MSL2DT4)	CAN1 Message Slot 2 Data 5 (C1MSL2DT5)	13-68 13-69
H'0080 152C	CAN1 Message Slot 2 Data 6 (C1MSL2DT6)	CAN1 Message Slot 2 Data 7 (C1MSL2DT7)	13-70 13-71
H'0080 152E	CAN1 Message Slot 2 Timestamp (C1MSL2TSP)		13-72
H'0080 1530	CAN1 Message Slot 3 Standard ID0 (C1MSL3SID0)	CAN1 Message Slot 3 Standard ID1 (C1MSL3SID1)	13-58 13-59
H'0080 1532	CAN1 Message Slot 3 Extended ID0 (C1MSL3EID0)	CAN1 Message Slot 3 Extended ID1 (C1MSL3EID1)	13-60 13-61
H'0080 1534	CAN1 Message Slot 3 Extended ID2 (C1MSL3EID2)	CAN1 Message Slot 3 Data Length Register (C1MSL3DLC)	13-62 13-63
H'0080 1536	CAN1 Message Slot 3 Data 0 (C1MSL3DT0)	CAN1 Message Slot 3 Data 1 (C1MSL3DT1)	13-64 13-65
H'0080 1538	CAN1 Message Slot 3 Data 2 (C1MSL3DT2)	CAN1 Message Slot 3 Data 3 (C1MSL3DT3)	13-66 13-67
H'0080 153A	CAN1 Message Slot 3 Data 4 (C1MSL3DT4)	CAN1 Message Slot 3 Data 5 (C1MSL3DT5)	13-68 13-69
H'0080 153C	CAN1 Message Slot 3 Data 6 (C1MSL3DT6)	CAN1 Message Slot 3 Data 7 (C1MSL3DT7)	13-70 13-71
H'0080 153E	CAN1 Message Slot 3 Timestamp (C1MSL3TSP)		13-72
H'0080 1540	CAN1 Message Slot 4 Standard ID0 (C1MSL4SID0)	CAN1 Message Slot 4 Standard ID1 (C1MSL4SID1)	13-58 13-59
H'0080 1542	CAN1 Message Slot 4 Extended ID0 (C1MSL4EID0)	CAN1 Message Slot 4 Extended ID1 (C1MSL4EID1)	13-60 13-61
H'0080 1544	CAN1 Message Slot 4 Extended ID2 (C1MSL4EID2)	CAN1 Message Slot 4 Data Length Register (C1MSL4DLC)	13-62 13-63
H'0080 1546	CAN1 Message Slot 4 Data 0 (C1MSL4DT0)	CAN1 Message Slot 4 Data 1 (C1MSL4DT1)	13-64 13-65
H'0080 1548	CAN1 Message Slot 4 Data 2 (C1MSL4DT2)	CAN1 Message Slot 4 Data 3 (C1MSL4DT3)	13-66 13-67
H'0080 154A	CAN1 Message Slot 4 Data 4 (C1MSL4DT4)	CAN1 Message Slot 4 Data 5 (C1MSL4DT5)	13-68 13-69
H'0080 154C	CAN1 Message Slot 4 Data 6 (C1MSL4DT6)	CAN1 Message Slot 4 Data 7 (C1MSL4DT7)	13-70 13-71
H'0080 154E	CAN1 Message Slot 4 Timestamp (C1MSL4TSP)		13-72
H'0080 1550	CAN1 Message Slot 5 Standard ID0 (C1MSL5SID0)	CAN1 Message Slot 5 Standard ID1 (C1MSL5SID1)	13-58 13-59
H'0080 1552	CAN1 Message Slot 5 Extended ID0 (C1MSL5EID0)	CAN1 Message Slot 5 Extended ID1 (C1MSL5EID1)	13-60 13-61
H'0080 1554	CAN1 Message Slot 5 Extended ID2 (C1MSL5EID2)	CAN1 Message Slot 5 Data Length Register (C1MSL5DLC)	13-62 13-63
H'0080 1556	CAN1 Message Slot 5 Data 0 (C1MSL5DT0)	CAN1 Message Slot 5 Data 1 (C1MSL5DT1)	13-64 13-65
H'0080 1558	CAN1 Message Slot 5 Data 2 (C1MSL5DT2)	CAN1 Message Slot 5 Data 3 (C1MSL5DT3)	13-66 13-67
H'0080 155A	CAN1 Message Slot 5 Data 4 (C1MSL5DT4)	CAN1 Message Slot 5 Data 5 (C1MSL5DT5)	13-68 13-69
H'0080 155C	CAN1 Message Slot 5 Data 6 (C1MSL5DT6)	CAN1 Message Slot 5 Data 7 (C1MSL5DT7)	13-70 13-71
H'0080 155E	CAN1 Message Slot 5 Timestamp (C1MSL5TSP)		13-72

CAN Module Related Register Map (9/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 1560	CAN1 Message Slot 6 Standard ID0 (C1MSL6SID0)	CAN1 Message Slot 6 Standard ID1 (C1MSL6SID1)	13-58 13-59
H'0080 1562	CAN1 Message Slot 6 Extended ID0 (C1MSL6EID0)	CAN1 Message Slot 6 Extended ID1 (C1MSL6EID1)	13-60 13-61
H'0080 1564	CAN1 Message Slot 6 Extended ID2 (C1MSL6EID2)	CAN1 Message Slot 6 Data Length Register (C1MSL6DLC)	13-62 13-63
H'0080 1566	CAN1 Message Slot 6 Data 0 (C1MSL6DT0)	CAN1 Message Slot 6 Data 1 (C1MSL6DT1)	13-64 13-65
H'0080 1568	CAN1 Message Slot 6 Data 2 (C1MSL6DT2)	CAN1 Message Slot 6 Data 3 (C1MSL6DT3)	13-66 13-67
H'0080 156A	CAN1 Message Slot 6 Data 4 (C1MSL6DT4)	CAN1 Message Slot 6 Data 5 (C1MSL6DT5)	13-68 13-69
H'0080 156C	CAN1 Message Slot 6 Data 6 (C1MSL6DT6)	CAN1 Message Slot 6 Data 7 (C1MSL6DT7)	13-70 13-71
H'0080 156E	CAN1 Message Slot 6 Timestamp (C1MSL6TSP)		13-72
H'0080 1570	CAN1 Message Slot 7 Standard ID0 (C1MSL7SID0)	CAN1 Message Slot 7 Standard ID1 (C1MSL7SID1)	13-58 13-59
H'0080 1572	CAN1 Message Slot 7 Extended ID0 (C1MSL7EID0)	CAN1 Message Slot 7 Extended ID1 (C1MSL7EID1)	13-60 13-61
H'0080 1574	CAN1 Message Slot 7 Extended ID2 (C1MSL7EID2)	CAN1 Message Slot 7 Data Length Register (C1MSL7DLC)	13-62 13-63
H'0080 1576	CAN1 Message Slot 7 Data 0 (C1MSL7DT0)	CAN1 Message Slot 7 Data 1 (C1MSL7DT1)	13-64 13-65
H'0080 1578	CAN1 Message Slot 7 Data 2 (C1MSL7DT2)	CAN1 Message Slot 7 Data 3 (C1MSL7DT3)	13-66 13-67
H'0080 157A	CAN1 Message Slot 7 Data 4 (C1MSL7DT4)	CAN1 Message Slot 7 Data 5 (C1MSL7DT5)	13-68 13-69
H'0080 157C	CAN1 Message Slot 7 Data 6 (C1MSL7DT6)	CAN1 Message Slot 7 Data 7 (C1MSL7DT7)	13-70 13-71
H'0080 157E	CAN1 Message Slot 7 Timestamp (C1MSL7TSP)		13-72
H'0080 1580	CAN1 Message Slot 8 Standard ID0 (C1MSL8SID0)	CAN1 Message Slot 8 Standard ID1 (C1MSL8SID1)	13-58 13-59
H'0080 1582	CAN1 Message Slot 8 Extended ID0 (C1MSL8EID0)	CAN1 Message Slot 8 Extended ID1 (C1MSL8EID1)	13-60 13-61
H'0080 1584	CAN1 Message Slot 8 Extended ID2 (C1MSL8EID2)	CAN1 Message Slot 8 Data Length Register (C1MSL8DLC)	13-62 13-63
H'0080 1586	CAN1 Message Slot 8 Data 0 (C1MSL8DT0)	CAN1 Message Slot 8 Data 1 (C1MSL8DT1)	13-64 13-65
H'0080 1588	CAN1 Message Slot 8 Data 2 (C1MSL8DT2)	CAN1 Message Slot 8 Data 3 (C1MSL8DT3)	13-66 13-67
H'0080 158A	CAN1 Message Slot 8 Data 4 (C1MSL8DT4)	CAN1 Message Slot 8 Data 5 (C1MSL8DT5)	13-68 13-69
H'0080 158C	CAN1 Message Slot 8 Data 6 (C1MSL8DT6)	CAN1 Message Slot 8 Data 7 (C1MSL8DT7)	13-70 13-71
H'0080 158E	CAN1 Message Slot 8 Timestamp (C1MSL8TSP)		13-72
H'0080 1590	CAN1 Message Slot 9 Standard ID0 (C1MSL9SID0)	CAN1 Message Slot 9 Standard ID1 (C1MSL9SID1)	13-58 13-59
H'0080 1592	CAN1 Message Slot 9 Extended ID0 (C1MSL9EID0)	CAN1 Message Slot 9 Extended ID1 (C1MSL9EID1)	13-60 13-61
H'0080 1594	CAN1 Message Slot 9 Extended ID2 (C1MSL9EID2)	CAN1 Message Slot 9 Data Length Register (C1MSL9DLC)	13-62 13-63
H'0080 1596	CAN1 Message Slot 9 Data 0 (C1MSL9DT0)	CAN1 Message Slot 9 Data 1 (C1MSL9DT1)	13-64 13-65
H'0080 1598	CAN1 Message Slot 9 Data 2 (C1MSL9DT2)	CAN1 Message Slot 9 Data 3 (C1MSL9DT3)	13-66 13-67
H'0080 159A	CAN1 Message Slot 9 Data 4 (C1MSL9DT4)	CAN1 Message Slot 9 Data 5 (C1MSL9DT5)	13-68 13-69
H'0080 159C	CAN1 Message Slot 9 Data 6 (C1MSL9DT6)	CAN1 Message Slot 9 Data 7 (C1MSL9DT7)	13-70 13-71
H'0080 159E	CAN1 Message Slot 9 Timestamp (C1MSL9TSP)		13-72

CAN Module Related Register Map (10/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 15A0	CAN1 Message Slot 10 Standard ID0 (C1MSL10SID0)	CAN1 Message Slot 10 Standard ID1 (C1MSL10SID1)	13-58 13-59
H'0080 15A2	CAN1 Message Slot 10 Extended ID0 (C1MSL10EID0)	CAN1 Message Slot 10 Extended ID1 (C1MSL10EID1)	13-60 13-61
H'0080 15A4	CAN1 Message Slot 10 Extended ID2 (C1MSL10EID2)	CAN1 Message Slot 10 Data Length Register (C1MSL10DLC)	13-62 13-63
H'0080 15A6	CAN1 Message Slot 10 Data 0 (C1MSL10DT0)	CAN1 Message Slot 10 Data 1 (C1MSL10DT1)	13-64 13-65
H'0080 15A8	CAN1 Message Slot 10 Data 2 (C1MSL10DT2)	CAN1 Message Slot 10 Data 3 (C1MSL10DT3)	13-66 13-67
H'0080 15AA	CAN1 Message Slot 10 Data 4 (C1MSL10DT4)	CAN1 Message Slot 10 Data 5 (C1MSL10DT5)	13-68 13-69
H'0080 15AC	CAN1 Message Slot 10 Data 6 (C1MSL10DT6)	CAN1 Message Slot 10 Data 7 (C1MSL10DT7)	13-70 13-71
H'0080 15AE	CAN1 Message Slot 10 Timestamp (C1MSL10TSP)		13-72
H'0080 15B0	CAN1 Message Slot 11 Standard ID0 (C1MSL11SID0)	CAN1 Message Slot 11 Standard ID1 (C1MSL11SID1)	13-58 13-59
H'0080 15B2	CAN1 Message Slot 11 Extended ID0 (C1MSL11EID0)	CAN1 Message Slot 11 Extended ID1 (C1MSL11EID1)	13-60 13-61
H'0080 15B4	CAN1 Message Slot 11 Extended ID2 (C1MSL11EID2)	CAN1 Message Slot 11 Data Length Register (C1MSL11DLC)	13-62 13-63
H'0080 15B6	CAN1 Message Slot 11 Data 0 (C1MSL11DT0)	CAN1 Message Slot 11 Data 1 (C1MSL11DT1)	13-64 13-65
H'0080 15B8	CAN1 Message Slot 11 Data 2 (C1MSL11DT2)	CAN1 Message Slot 11 Data 3 (C1MSL11DT3)	13-66 13-67
H'0080 15BA	CAN1 Message Slot 11 Data 4 (C1MSL11DT4)	CAN1 Message Slot 11 Data 5 (C1MSL11DT5)	13-68 13-69
H'0080 15BC	CAN1 Message Slot 11 Data 6 (C1MSL11DT6)	CAN1 Message Slot 11 Data 7 (C1MSL11DT7)	13-70 13-71
H'0080 15BE	CAN1 Message Slot 11 Timestamp (C1MSL11TSP)		13-72
H'0080 15C0	CAN1 Message Slot 12 Standard ID0 (C1MSL12SID0)	CAN1 Message Slot 12 Standard ID1 (C1MSL12SID1)	13-58 13-59
H'0080 15C2	CAN1 Message Slot 12 Extended ID0 (C1MSL12EID0)	CAN1 Message Slot 12 Extended ID1 (C1MSL12EID1)	13-60 13-61
H'0080 15C4	CAN1 Message Slot 12 Extended ID2 (C1MSL12EID2)	CAN1 Message Slot 12 Data Length Register (C1MSL12DLC)	13-62 13-63
H'0080 15C6	CAN1 Message Slot 12 Data 0 (C1MSL12DT0)	CAN1 Message Slot 12 Data 1 (C1MSL12DT1)	13-64 13-65
H'0080 15C8	CAN1 Message Slot 12 Data 2 (C1MSL12DT2)	CAN1 Message Slot 12 Data 3 (C1MSL12DT3)	13-66 13-67
H'0080 15CA	CAN1 Message Slot 12 Data 4 (C1MSL12DT4)	CAN1 Message Slot 12 Data 5 (C1MSL12DT5)	13-68 13-69
H'0080 15CC	CAN1 Message Slot 12 Data 6 (C1MSL12DT6)	CAN1 Message Slot 12 Data 7 (C1MSL12DT7)	13-70 13-71
H'0080 15CE	CAN1 Message Slot 12 Timestamp (C1MSL12TSP)		13-72
H'0080 15D0	CAN1 Message Slot 13 Standard ID0 (C1MSL13SID0)	CAN1 Message Slot 13 Standard ID1 (C1MSL13SID1)	13-58 13-59
H'0080 15D2	CAN1 Message Slot 13 Extended ID0 (C1MSL13EID0)	CAN1 Message Slot 13 Extended ID1 (C1MSL13EID1)	13-60 13-61
H'0080 15D4	CAN1 Message Slot 13 Extended ID2 (C1MSL13EID2)	CAN1 Message Slot 13 Data Length Register (C1MSL13DLC)	13-62 13-63
H'0080 15D6	CAN1 Message Slot 13 Data 0 (C1MSL13DT0)	CAN1 Message Slot 13 Data 1 (C1MSL13DT1)	13-64 13-65
H'0080 15D8	CAN1 Message Slot 13 Data 2 (C1MSL13DT2)	CAN1 Message Slot 13 Data 3 (C1MSL13DT3)	13-66 13-67
H'0080 15DA	CAN1 Message Slot 13 Data 4 (C1MSL13DT4)	CAN1 Message Slot 13 Data 5 (C1MSL13DT5)	13-68 13-69
H'0080 15DC	CAN1 Message Slot 13 Data 6 (C1MSL13DT6)	CAN1 Message Slot 13 Data 7 (C1MSL13DT7)	13-70 13-71
H'0080 15DE	CAN1 Message Slot 13 Timestamp (C1MSL13TSP)		13-72

CAN Module Related Register Map (11/11)

Address	+0 address b0 b7	+1 address b8 b15	See pages
H'0080 15E0	CAN1 Message Slot 14 Standard ID0 (C1MSL14SID0)	CAN1 Message Slot 14 Standard ID1 (C1MSL14SID1)	13-58 13-59
H'0080 15E2	CAN1 Message Slot 14 Extended ID0 (C1MSL14EID0)	CAN1 Message Slot 14 Extended ID1 (C1MSL14EID1)	13-60 13-61
H'0080 15E4	CAN1 Message Slot 14 Extended ID2 (C1MSL14EID2)	CAN1 Message Slot 14 Data Length Register (C1MSL14DLC)	13-62 13-63
H'0080 15E6	CAN1 Message Slot 14 Data 0 (C1MSL14DT0)	CAN1 Message Slot 14 Data 1 (C1MSL14DT1)	13-64 13-65
H'0080 15E8	CAN1 Message Slot 14 Data 2 (C1MSL14DT2)	CAN1 Message Slot 14 Data 3 (C1MSL14DT3)	13-66 13-67
H'0080 15EA	CAN1 Message Slot 14 Data 4 (C1MSL14DT4)	CAN1 Message Slot 14 Data 5 (C1MSL14DT5)	13-68 13-69
H'0080 15EC	CAN1 Message Slot 14 Data 6 (C1MSL14DT6)	CAN1 Message Slot 14 Data 7 (C1MSL14DT7)	13-70 13-71
H'0080 15EE	CAN1 Message Slot 14 Timestamp (C1MSL14TSP)		13-72
H'0080 15F0	CAN1 Message Slot 15 Standard ID0 (C1MSL15SID0)	CAN1 Message Slot 15 Standard ID1 (C1MSL15SID1)	13-58 13-59
H'0080 15F2	CAN1 Message Slot 15 Extended ID0 (C1MSL15EID0)	CAN1 Message Slot 15 Extended ID1 (C1MSL15EID1)	13-60 13-61
H'0080 15F4	CAN1 Message Slot 15 Extended ID2 (C1MSL15EID2)	CAN1 Message Slot 15 Data Length Register (C1MSL15DLC)	13-62 13-63
H'0080 15F6	CAN1 Message Slot 15 Data 0 (C1MSL15DT0)	CAN1 Message Slot 15 Data 1 (C1MSL15DT1)	13-64 13-65
H'0080 15F8	CAN1 Message Slot 15 Data 2 (C1MSL15DT2)	CAN1 Message Slot 15 Data 3 (C1MSL15DT3)	13-66 13-67
H'0080 15FA	CAN1 Message Slot 15 Data 4 (C1MSL15DT4)	CAN1 Message Slot 15 Data 5 (C1MSL15DT5)	13-68 13-69
H'0080 15FC	CAN1 Message Slot 15 Data 6 (C1MSL15DT6)	CAN1 Message Slot 15 Data 7 (C1MSL15DT7)	13-70 13-71
H'0080 15FE	CAN1 Message Slot 15 Timestamp (C1MSL15TSP)		13-72
⋮	(Use inhibited area)		
H'0080 3FFE	(Use inhibited area)		

13.2.1 CAN Control Registers

CAN0 Control Register (CAN0CNT) <Address: H'0080 1000>
CAN1 Control Register (CAN1CNT) <Address: H'0080 1400>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
				RBO	TSR	TSP					FRST	BCM		LBM	RST
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

<Upon exiting reset: H'0011>

b	Bit Name	Function	R	W
0–3	No function assigned. Fix to "0".		0	0
4	RBO Return bus off bit	0: Enable normal operation 1: Request clearing of error counter	R(Note 1)	
5	TSR Timestamp counter reset bit	0: Enable count operation 1: Initialize count (to H'0000)	R(Note 1)	
6–7	TSP Timestamp prescaler bit	00: Select CAN bus bit clock 01: Select CAN bus bit clock divided by 2 10: Select CAN bus bit clock divided by 3 11: Select CAN bus bit clock divided by 4	R	W
8-10	No function assigned. Fix to "0".		0	0
11	FRST Forcible reset bit	0: Negate reset 1: Forcibly reset	R	W
12	BCM BasicCAN mode bit	0: Disable BasicCAN mode 1: BasicCAN mode	R	W
13	No function assigned. Fix to "0".		0	0
14	LBM Loopback mode bit	0: Disable loopback function 1: Enable loopback function	R	W
15	RST CAN reset bit	0: Negate reset 1: Request reset	R	W

Note 1: Only writing "1" is effective. Automatically cleared to "0" in hardware.

(1) RBO (Return Bus Off) bit (Bit 4)

Setting this bit to "1" clears the CAN Receive Error Count Register (CANnREC) and CAN Transmit Error Count Register (CANnTEC) to H'00 and forcibly places the CAN module into an error active state. This bit is cleared when the CAN module goes to an error active state.

Note: • Communication becomes possible when 11 consecutive recessive bits are detected on the CAN bus after clearing the error counters.

(2) TSR (Timestamp Counter Reset) bit (Bit 5)

Setting this bit to "1" clears the value of the CAN Timestamp Count Register (CANnTSTMP) to H'0000. This bit is cleared after the value of the CAN Timestamp Count Register (CANnTSTMP) is cleared to H'0000.

(3) TSP (Timestamp Prescaler) bits (Bits 6–7)

These bits select the count clock source for the timestamp counter.

Note: • Do not change settings of the TSP bits while CAN is operating (CAN Status Register CRS bit = "0").

(4) FRST (Forcible Reset) bit (Bit 11)

When the FRST bit is set to "1", the CAN module is separated from the CAN bus and the protocol control unit is reset regardless of whether the CAN module currently is communicating. Up to 5 BCLK periods are required before the protocol control unit is reset after setting the FRST bit.

Notes: • In order for CAN communication to start, the FRST and RST bits must be cleared to "0".
- If the FRST bit is set to "1" during communication, the CTX pin output goes high immediately after that. Therefore, setting the FRST bit to "1" while sending CAN frame may cause a CAN bus error.
- The CAN Message Slot Control Register's transmit/receive requests are not cleared for reasons that the FRST or RST bits are set.
- When the protocol control unit is reset by setting the FRST bit to "1", the CAN Timestamp Count and CAN Transmit/Receive Error Count Registers are initialized to "0".

(5) BCM (BasicCAN Mode) bit (Bit 12)

By setting this bit to "1", local slot 14 and 15 of the CAN module can be operated in BasicCAN mode.

• Operation during BasicCAN mode

During BasicCAN mode, two local slots—slots 14 and 15—are used as dual buffers, and the received frames with matching ID are stored alternately in slots 14 and 15 by acceptance filtering. Used for this acceptance filtering when slot 14 is active (next received frame to be stored in slot 14) are the ID set in slot 14 and local mask A, and those when slot 15 is active are the ID set in slot 15 and local mask B. Two types of frames—data frame and remote frame—can be received in this mode. By setting the same ID and the same mask register value for the two slots, the possibility of loosing messages when, for example, receiving frames which have many IDs may be reduced.

• Procedure for entering BasicCAN mode

Follow the procedure below during initialization:

1) Set the ID for slots 14 and 15 and the local mask registers A and B. (We recommend setting the same value.)
2) Set the frame types to be handled by slots 14 and 15 (standard or extended) in the CAN Extended ID Register. (We recommend setting the same type.)
3) Set the Message Slot Control Registers for slots 14 and 15 for data frame reception.
4) Set the BCM bit to "1".

Notes: • Do not change settings of the BCM bit while CAN is operating (CAN Status Register CRS bit = "0").
• The first slot that is active after clearing the RST bit is slot 14.
• Even during BasicCAN mode, slots 0 to 13 can be used the same way as in normal operation.

(6) LBM (Loopback Mode) bit (Bit 14)

When the LBM bit is set to "1", if a receive slot exists whose ID matches that of the frame sent by the CAN module itself, then the frame can be received.

Notes: • ACK is not returned for the transmit frame.
• Do not change settings of the LBM bit while CAN is operating (CAN Status Register CRS bit = "0").
• After complete sending Frame correctly, TSC bit in CAN status register (CANnSTAT) is "1", but RSC bit is not "1." And it is possible to symbiotic for transmit complete interrupt request and receive complete interrupt request.

(7) RST (CAN Reset) bit (Bit 15)

When the RST bit is cleared to "0", the CAN module is connected to the CAN bus and becomes ready to communicate after detecting 11 consecutive recessive bits. Also, the CAN Timestamp Count Register thereby starts counting.
When the RST bit is set to "1", the bus will enter an idle state (detects 11 consecutive recessive bits) after sending frames from the slots which have transmit requests set by that time, then the protocol control unit is reset and the CAN module is disconnected from the CAN bus. Frames received during this time are processed normally.
When setting RST bit to "1" under bus off state, it exits from bus off state after detecting 11 consecutive recessive bits on CAN bus 128 times, and then protocol control unit enters a reset state. To exit from bus off state forcibly, use either RBO bit or FRST bit.

Notes: • It is inhibited to set a new transmit request until the protocol control unit is reset (until the CAN Status Register CRS bit is set to "1") after setting the RST bit to "1."
• When the protocol control unit is reset by setting the RST bit to "1", the CAN Timestamp Count and CAN Transmit/Receive Error Count Registers are initialized to "0."
• In order for CAN communication to start, the FRST and RST bits must be cleared to "0."

13.2.2 CAN Status Registers

CAN0 Status Register (CAN0STAT) <Address: H'0080 1002>
CAN1 Status Register (CAN1STAT) <Address: H'0080 1402>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
	BOS	EPS	CBS	BCS		LBS	CRS	RSB	TSB	RSC	TSC	MSN			
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0100>

b	Bit Name	Function	R	W
0	No function assigned. Fix to "0".		0	0
1	BOS Bus off status bit	0: Not bus off 1: Bus off state	R	–
2	EPS Error passive status bit	0: Not error passive 1: Error passive state	R	–
3	CBS CAN bus error bit	0: No error occurred 1: Error occurred	R	–
4	BCS BasicCAN status bit	0: Normal mode 1: BasicCAN mode	R	–
5	No function assigned. Fix to "0".		0	0
6	LBS Loopback status bit	0: Normal mode 1: Loopback mode	R	–
7	CRS CAN reset status bit	0: Operating 1: Reset	R	–
8	RSB Receive status bit	0: Not receiving 1: Receiving	R	–
9	TSB Transmit status bit	0: Not sending 1: Sending	R	–
10	RSC Reception completed status bit	0: Reception not completed 1: Reception completed	R	–
11	TSC Transmission completed status bit	0: Transmission not completed 1: Transmission completed	R	–
12–15	MSN Message slot number bit	Number of the message slot which has finished sending or receiving 0000: Slot 0 0001: Slot 1 0010: Slot 2 0011: Slot 3 0100: Slot 4 0101: Slot 5 0110: Slot 6 0111: Slot 7 1000: Slot 8 1001: Slot 9 1010: Slot 10 1011: Slot 11 1100: Slot 12 1101: Slot 13 1110: Slot 14 1111: Slot 15	R	–

(1) BOS (Bus Off Status) bit (Bit 1)

When BOS bit = "1", it means that the CAN module is in a bus off state.

[Set condition]

This bit is set to "1" when the transmit error count register value exceeded 255 and a bus off state is entered.

[Clear condition]

This bit is cleared when restored from the bus off state.

(2) EPS (Error Passive Status) bit (Bit 2)

When EPS bit = "1", it means that the CAN module is in an error passive state.

[Set condition]

This bit is set to "1" when the transmit or receive error count register value exceeded 127 and an error passive state is entered.

[Clear condition]

This bit is cleared when restored from the error passive state.

(3) CBS (CAN Bus Error) bit (Bit 3)

[Set condition]

This bit is set to "1" when an error is detected on the CAN bus.

[Clear condition]

This bit is cleared when the CAN module finished sending or receiving normally.

(4) BCS (BasicCAN Status) bit (Bit 4)

When BCS bit = "1", it means that the CAN module is operating in BasicCAN mode.

[Set condition]

This bit is set to "1" when the CAN module is operating in BasicCAN mode. BasicCAN mode is useful when the following conditions are met:

- CAN Control Register BCM bit = "1"
- Slots 14 and 15 both are set for data frame reception

[Clear condition]

This bit is cleared by clearing the BCM bit to "0".

(5) LBS (Loopback Status) bit (Bit 6)

When LBS bit = "1", it means that the CAN module is operating in loopback mode.

[Set condition]

This bit is set to "1" by setting the CAN Control Register LBM (loopback mode) bit to "1".

[Clear condition]

This bit is cleared by clearing the LBM bit to "0".

(6) CRS (CAN Reset Status) bit (Bit 7)

When CRS bit = "1", it means that the protocol control unit is in a reset state.

[Set condition]

This bit is set to "1" when the CAN protocol control unit is in a reset state.

[Clear condition]

This bit is cleared by clearing the CAN Control Register RST (CAN reset) and FRST bits to "0". However, it requires one bit of set baud rate worth of time to have CRS bit cleared to "0" after RST bit and FRST bit are cleared to "0."

(7) RSB (Receive Status) bit (Bit 8)

[Set condition]

This bit is set to "1" when the CAN module is operating as a receive node.

[Clear condition]

This bit is cleared when the CAN module starts operating as a transmit node or enters a bus idle state.

(8) TSB (Transmit Status) bit (Bit 9)

[Set condition]

This bit is set to "1" when the CAN module is operating as a transmit node.

[Clear condition]

This bit is cleared when the CAN module starts operating as a receive node or enters a bus idle state.

(9) RSC (Reception Completed Status) bit (Bit 10)

[Set condition]

This bit is set to "1" when the CAN module has finished receiving normally (regardless of whether there is any slot that meets receive conditions).

[Clear condition]

This bit is cleared when the CAN module has finished sending normally.

(10) TSC (Transmission Completed Status) bit (Bit 11)

[Set condition]

This bit is set to "1" when the CAN module has finished sending normally.

[Clear condition]

This bit is cleared when the CAN module has finished receiving normally.

(11) MSN (Message Slot Number) bits (Bits 12–15)

These bits indicate the relevant slot number when the CAN module has finished sending or finished storing the received data. These bits cannot be cleared to "0" in software.

Note: • When CAN module receives the frame that is transmitted by the CAN module itself during loopback mode, the MSN bits indicate the transmit slot number.

13.2.3 CAN Extended ID Registers

CAN0 Extended ID Register (CAN0EXTID) <Address: H'0080 1004>
CAN1 Extended ID Register (CAN1EXTID) <Address: H'0080 1404>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
IDE0	IDE1	IDE2	IDE3	IDE4	IDE5	IDE6	IDE7	IDE8	IDE9	IDE10	IDE11	IDE12	IDE13	IDE14	IDE15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	IDE0 (slot 0 extended format bit)	0: Standard ID format	R	W
1	IDE1 (slot 1 extended format bit)	1: Extended ID format		
2	IDE2 (slot 2 extended format bit)			
3	IDE3 (slot 3 extended format bit)			
4	IDE4 (slot 4 extended format bit)			
5	IDE5 (slot 5 extended format bit)			
6	IDE6 (slot 6 extended format bit)			
7	IDE7 (slot 7 extended format bit)			
8	IDE8 (slot 8 extended format bit)			
9	IDE9 (slot 9 extended format bit)			
10	IDE10 (slot 10 extended format bit)			
11	IDE11 (slot 11 extended format bit)			
12	IDE12 (slot 12 extended format bit)			
13	IDE13 (slot 13 extended format bit)			
14	IDE14 (slot 14 extended format bit)			
15	IDE15 (slot 15 extended format bit)			

This register selects the format of frames handled by message slots corresponding to the respective bits in the register. Setting any bit in this register to "0" selects the standard ID format, and setting any bit in this register to "1" selects the extended ID format.

Note: • Settings of any bit in this register can only be changed when the corresponding slot does not have transmit or receive requests set.

13.2.4 CAN Configuration Registers

CAN0 Configuration Register (CAN0CONF) <Address: H'0080 1006>
CAN1 Configuration Register (CAN1CONF) <Address: H'0080 1406>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
SJW		PH2			PH1			PRB			SAM				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0–1	SJW reSynchronization Jump Width setting bit	00: SJW = 1Tq 01: SJW = 2Tq 10: SJW = 3Tq 11: SJW = 4Tq	R	W
2–4	PH2 Phase Segment2 setting bit	000: Phase Segment2 = 1Tq 001: Phase Segment2 = 2Tq 010: Phase Segment2 = 3Tq 011: Phase Segment2 = 4Tq 100: Phase Segment2 = 5Tq 101: Phase Segment2 = 6Tq 110: Phase Segment2 = 7Tq 111: Phase Segment2 = 8Tq	R	W
5–7	PH1 Phase Segment1 setting bit	000: Phase Segment1 = 1Tq 001: Phase Segment1 = 2Tq 010: Phase Segment1 = 3Tq 011: Phase Segment1 = 4Tq 100: Phase Segment1 = 5Tq 101: Phase Segment1 = 6Tq 110: Phase Segment1 = 7Tq 111: Phase Segment1 = 8Tq	R	W
8–10	PRB Propagation Segment setting bit	000: Propagation Segment = 1Tq 001: Propagation Segment = 2Tq 010: Propagation Segment = 3Tq 011: Propagation Segment = 4Tq 100: Propagation Segment = 5Tq 101: Propagation Segment = 6Tq 110: Propagation Segment = 7Tq 111: Propagation Segment = 8Tq	R	W
11	SAM Sampling count select bit	0: Sampled one time 1: Sampled three times	R	W
12–15	No function assigned. Fix to "0".		0	0

Notes: • Do not change settings of the CAN Configuration Register (CAN0CONF or CAN1CONF) during CAN operation (CAN Status Register CRS bit = "0").
• Bit configuration is specified by the CAN protocol specification in such a way that it satisfies the conditions given below:
  • Number of Tq's for one bit: 8–25 Tq's
  • SJW ≤ min (Phase Segment1, Phase Segment2)
  • Phase Segment2 = max (Phase Segment1, IPT) where IPT = 1 for the internal CAN modules of the 32176
  min() is the function that returns the smaller of two values; max() is the function that returns the maximum value.
  * IPT is an abbreviation for Information Processing Time, which is the time immediately after the sampling point.

(1) SJW bits (Bits 0–1)

These bits set the reSynchronization Jump Width.

(2) PH2 bits (Bits 2–4)

These bits set the width of Phase Segment2.

(3) PH1 bits (Bits 5–7)

These bits set the width of Phase Segment1.

(4) PRB bits (Bits 8–10)

These bits set the width of Propagation Segment.

(5) SAM bit (Bit 11)

This bit sets the number of times each bit is sampled. When SAM = "0", the value sampled at the end of Phase Segment1 is assumed to be the value of the bit. When SAM = "1", the value of the bit is determined by a majority circuit from three sampled values, each sampled 2 Tq's before, 1 Tq before, and at the end of Phase Segment1.

Table 13.2.1 Typical Settings of Bit Timing when CPU Clock = 40 MHz

Baud Rate	BRP Set Value	Tq Period (ns)	No. of Tq's in 1 Bit	PROP + PH1	PH2	Sampling Point
1M bps	1	50	20	13	6	70%
	3	100	10	7	2	80%
	3	100	10	6	3	70%
	3	100	10	5	4	60%
	4	125	8	5	2	75%
	4	125	8	4	3	63%
500K bps	4	125	16	13	2	88% (Note 1)
	4	125	16	12	3	81% (Note 1)
	4	125	16	11	4	75%
	7	200	10	7	2	80%
	7	200	10	6	3	70%
	7	200	10	5	4	60%
	9	250	8	5	2	75%
	9	250	8	4	3	63%

Note 1: PH2 = max (PH1, IPT), that is specified in CAN protocol, cannot be met.

Note: ・It does not mean that the communication at the above baud rate settings is guaranteed. Sufficient evaluation and verification are required before use.

Table 13.2.2 Typical Settings of Bit Timing when CPU Clock = 32 MHz

Baud Rate	BRP Set Value	Tq Period (ns)	No. of Tq's in 1 Bit	PROP + PH1	PH2	Sampling Point
1M bps	1	62.5	16	10	5	69%
	3	125	8	5	2	75%
	3	125	8	4	3	63%
500K bps	3	125	16	13	2	88% (Note 1)
	3	125	16	11	4	75%
	7	250	8	5	2	75%
	7	250	8	4	3	63%

Note 1: PH2 = max (PH1, IPT), that is specified in CAN protocol, cannot be met.

Note: ・It does not mean that the communication at the above baud rate settings is guaranteed. Sufficient evaluation and verification are required before use.

13.2.5 CAN Timestamp Count Registers

CAN0 Timestamp Count Register (CAN0TSTMP) <Address: H'0080 1008>
CAN1 Timestamp Count Register (CAN1TSTMP) <Address: H'0080 1408>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
CANTSTMP															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

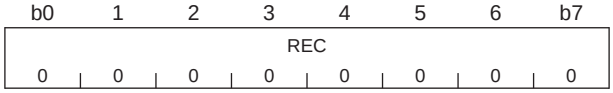
b	Bit Name	Function	R	W
0–15	CANTSTMP	16-bit timestamp count value	R	–

The CAN module contains a 16-bit up-count register. The count period can be selected from the CAN bus bit period divided by 1, 2, 3 or 4 by setting the CAN Control Register (CANnCNT) TSP (Timestamp Prescaler) bits.
When the CAN module finishes sending or receiving, it captures the count register value and stores the value in a message slot. The counter is made to start counting by clearing the CAN Control Register (CANnCNT) RST bit to "0".

Notes: • The CAN protocol control unit can be reset and the counter initialized to H'0000 by setting the CAN Control Register (CANnCNT) RST (CAN Reset) bit to "1". Or the counter can be initialized to H'0000 while the CAN module remains operating by setting the TSR (Timestamp Counter Reset) bit to "1".
• If any slot with the matching ID exists during loopback mode, the CAN module stores the timestamp value in that slot when it finished receiving. (No timestamp values are stored this way when the CAN module finished sending.)
• The count period of the CAN Timestamp Count Register varies with the CAN resynchronization function.

13.2.6 CAN Error Count Registers

CAN0 Receive Error Count Register (CAN0REC) <Address: H'0080 100A>
CAN1 Receive Error Count Register (CAN1REC) <Address: H'0080 140A>

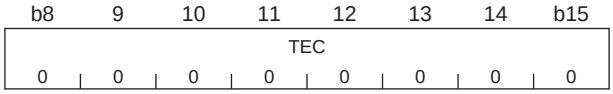


<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–7	REC	Receive error count value	R	–

During an error active/error passive state, a receive error count value is stored in this register.
The count is decremented when frames are received normally or incremented when an error occurred. If the CAN module finished receiving normally when REC ≥ 128 (error passive), REC is set to 127. During a bus off state, an undefined value is stored in this register. The count is reset to H'00 upon returning to an error active state.

CAN0 Transmit Error Count Register (CAN0TEC) <Address: H'0080 100B>
CAN1 Transmit Error Count Register (CAN1TEC) <Address: H'0080 140B>



<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–15	TEC	Transmit error count value	R	–

During an error active/error passive state, a transmit error count value is stored in this register.
The count is decremented when frames are transmitted normally or incremented when an error occurred. During a bus off state, an undefined value is stored in this register. The count is reset to H'00 upon returning to an error active state.

13.2.7 CAN Baud Rate Prescalers

CAN0 Baud Rate Prescaler (CAN0BRP) <Address: H'0080 1016>
CAN1 Baud Rate Prescaler (CAN1BRP) <Address: H'0080 1416>

b0	1	2	3	4	5	6	b7
BRP							
0	0	0	0	0	0	0	1

<Upon exiting reset: H'01>

b	Bit Name	Function	R	W
0–7	BRP	Baud rate prescaler value	R	W

This register sets the Tq period of CAN. The CAN baud rate is determined by (Tq period × number of Tq's in one bit).

Tq period = (BRP + 1) / (CPU clock)

$$\text{CAN transfer baud rate} = \frac{1}{\text{Tq period} \times \text{number of Tq's in one bit}}$$

Number of Tq's in one bit = Synchronization Segment + Propagation Segment
+ Phase Segment 1 + Phase Segment 2

Notes: • Setting H'00 (divide by 1) is inhibited.
• Do not change settings of the CAN Baud Rate Prescaler (CANnBRP) during CAN operation (CAN Status Register CRS bit = "0").

13.2.8 CAN Interrupt Related Registers

The CAN interrupt related registers are used to control the interrupt request signals output to the Interrupt Controller by CAN.

(1) Interrupt request status bit

This status bit is used to determine whether an interrupt is requested. When an interrupt request occurs, this bit is set in hardware (cannot be set in software). The status bit is cleared by writing "0". Writing "1" has no effect; the bit retains the status it had before the write. Because this bit is unaffected by the interrupt request enable bit, it can also be used to inspect the operating status of peripheral functions. In interrupt handling, make sure that within the grouped interrupt request status, only the status bit for the interrupt request that has been serviced is cleared. If the status bit for any interrupt request that has not been serviced is cleared, the pending interrupt request is cleared simultaneously with its status bit.

(2) Interrupt request mask bit

This bit is used to disable unnecessary interrupt requests within the grouped interrupt request. Set this bit to "1" to enable interrupt requests or "0" to disable interrupt requests.

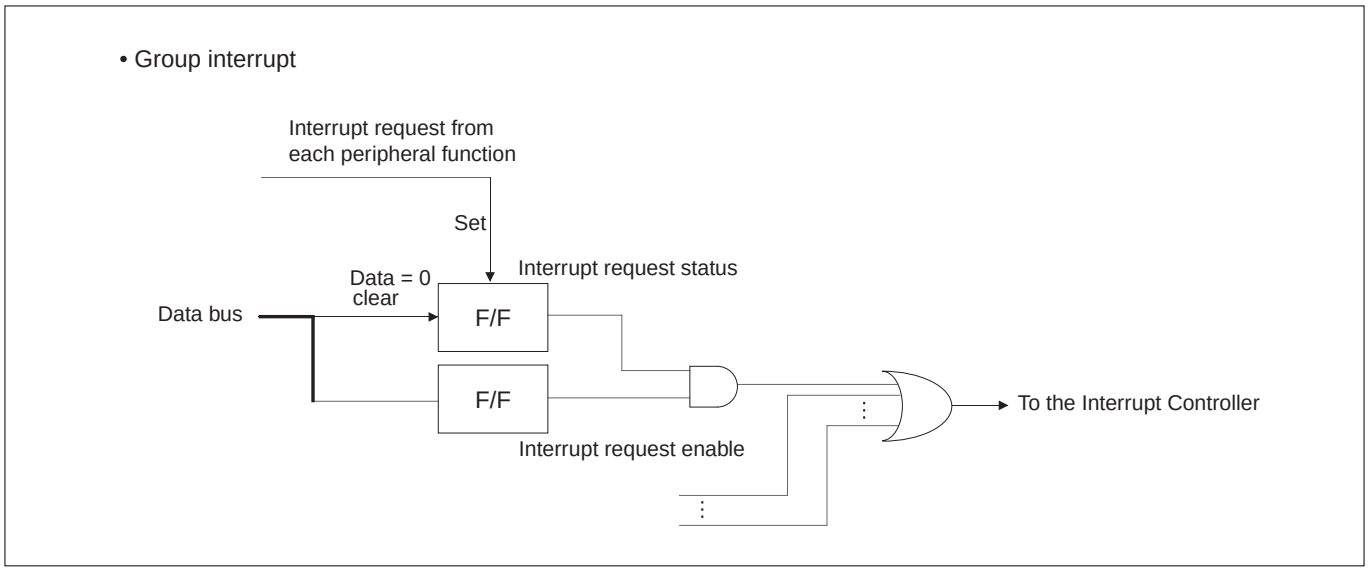
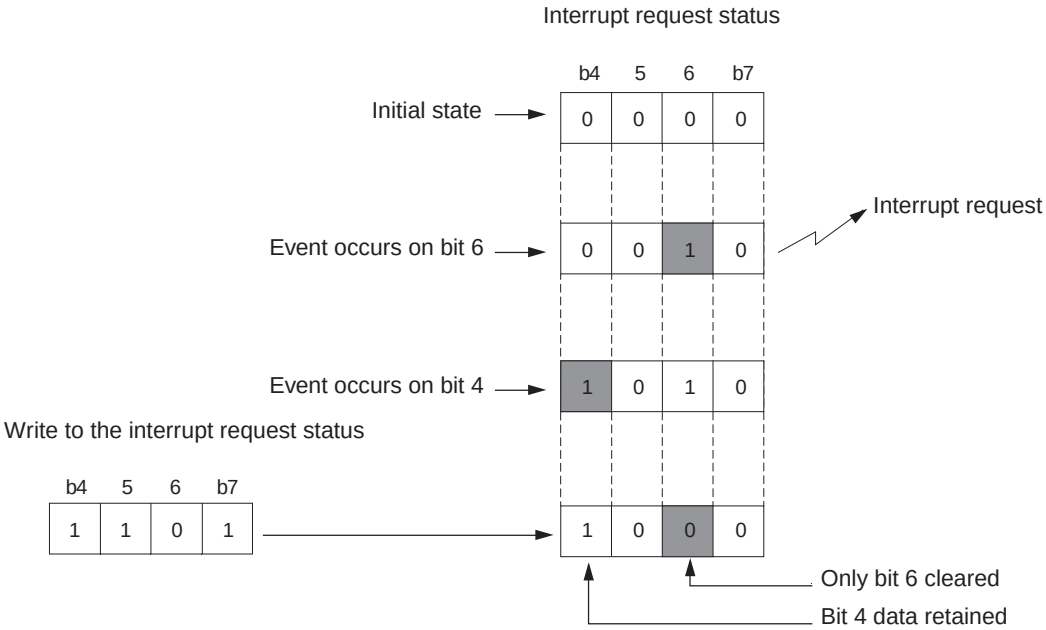


Figure 13.2.1 Interrupt Request Status and Mask Registers

● Example for clearing interrupt request status



● Program example

• To clear the Interrupt Request Status Register 0 (ISTREG) interrupt request status 1, ISTAT1 (0x02 bit)

○ ISTREG = 0xfd; /* Clear ISTAT1 (0x02 bit) only */

To clear an interrupt request status, always be sure to write 1 to all other interrupt request status bits. At this time, avoid using a logic operation like the one shown below. Because it requires three step-ISTREG read, logic operation and write, if another interrupt request occurs between the read and write, status may be inadvertently cleared.

× ISTREG &= 0xfd; /* Clear ISTAT1 (0x02 bit) only */

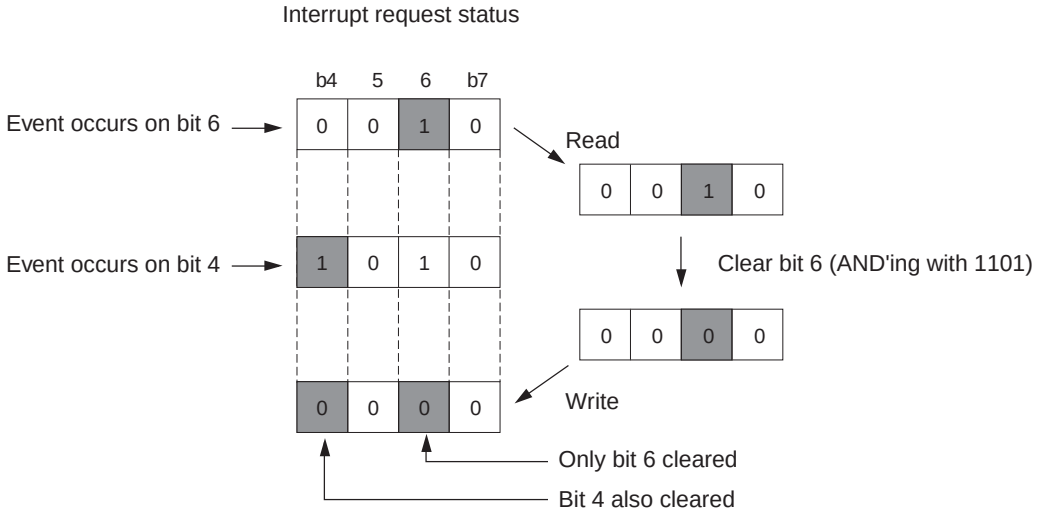


Figure 13.2.2 Example for Clearing Interrupt Request Status

CAN0 Slot Interrupt Request Status Register (CAN0SLIST) <Address: H'0080 100C>
CAN1 Slot Interrupt Request Status Register (CAN1SLIST) <Address: H'0080 140C>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
SSB0	SSB1	SSB2	SSB3	SSB4	SSB5	SSB6	SSB7	SSB8	SSB9	SSB10	SSB11	SSB12	SSB13	SSB14	SSB15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	SSB0 (slot 0 interrupt request status bit)	0: Interrupt not requested	R(Note 1)	
1	SSB1 (slot 1 interrupt request status bit)	1: Interrupt requested		
2	SSB2 (slot 2 interrupt request status bit)			
3	SSB3 (slot 3 interrupt request status bit)			
4	SSB4 (slot 4 interrupt request status bit)			
5	SSB5 (slot 5 interrupt request status bit)			
6	SSB6 (slot 6 interrupt request status bit)			
7	SSB7 (slot 7 interrupt request status bit)			
8	SSB8 (slot 8 interrupt request status bit)			
9	SSB9 (slot 9 interrupt request status bit)			
10	SSB10 (slot 10 interrupt request status bit)			
11	SSB11 (slot 11 interrupt request status bit)			
12	SSB12 (slot 12 interrupt request status bit)			
13	SSB13 (slot 13 interrupt request status bit)			
14	SSB14 (slot 14 interrupt request status bit)			
15	SSB15 (slot 15 interrupt request status bit)			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the status it had before the write.

When using CAN interrupts, this register helps to know which slot requested an interrupt.

• Slots set for transmission

The corresponding bit is set to "1" when the CAN module finished sending. This bit is cleared by writing "0" in software.

• Slots set for reception

The corresponding bit is set to "1" when the CAN module finished receiving and finished storing the received message in the message slot. This bit is cleared by writing "0" in software.

When writing to the CAN slot interrupt request status, make sure only the bits to be cleared are set to "0" and all other bits are set to "1". Those bits that have been set to "1" are unaffected by writing in software and retain the value they had before the write.

Notes: • If the automatic response function is enabled for remote frame receive slots, the request status is set after the CAN module finished receiving a remote frame and after it finished sending a data frame.
• For remote frame transmit slots, the request status is set after the CAN module finished sending a remote frame and after it finished receiving a data frame.
• If the request status is set by an interrupt request at the same time it is cleared in software, the former has priority so that the request status is set.

CAN0 Slot Interrupt Request Mask Register (CAN0SLIMK) <Address: H'0080 1010>
CAN1 Slot Interrupt Request Mask Register (CAN1SLIMK) <Address: H'0080 1410>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
IRB0	IRB1	IRB2	IRB3	IRB4	IRB5	IRB6	IRB7	IRB8	IRB9	IRB10	IRB11	IRB12	IRB13	IRB14	IRB15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	IRB0 (slot 0 interrupt request mask bit)	0: Mask (disable) interrupt request	R	W
1	IRB1 (slot 1 interrupt request mask bit)	1: Enable interrupt request		
2	IRB2 (slot 2 interrupt request mask bit)			
3	IRB3 (slot 3 interrupt request mask bit)			
4	IRB4 (slot 4 interrupt request mask bit)			
5	IRB5 (slot 5 interrupt request mask bit)			
6	IRB6 (slot 6 interrupt request mask bit)			
7	IRB7 (slot 7 interrupt request mask bit)			
8	IRB8 (slot 8 interrupt request mask bit)			
9	IRB9 (slot 9 interrupt request mask bit)			
10	IRB10 (slot 10 interrupt request mask bit)			
11	IRB11 (slot 11 interrupt request mask bit)			
12	IRB12 (slot 12 interrupt request mask bit)			
13	IRB13 (slot 13 interrupt request mask bit)			
14	IRB14 (slot 14 interrupt request mask bit)			
15	IRB15 (slot 15 interrupt request mask bit)			

This register is used to enable or disable the interrupt requests that will be generated when data transmission or reception in each corresponding slot is completed. Setting IRBn (n = 0–15) to "1" enables the interrupt request to be generated when data transmission or reception in the corresponding slot is completed. The CAN Slot Interrupt Request Status Register (CANnSLIST) helps to know which slot requested the interrupt.

CAN0 Error Interrupt Request Status Register (CAN0ERIST) <Address: H'0080 1014>
CAN1 Error Interrupt Request Status Register (CAN1ERIST) <Address: H'0080 1414>

b0	1	2	3	4	5	6	b7
					EIS	PIS	OIS
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–4	No function assigned. Fix to "0".		0	0
5	EIS CAN bus error interrupt request status bit	0: Interrupt not requested 1: Interrupt requested	R(Note 1)	
6	PIS Error passive interrupt request status bit			
7	OIS Bus off interrupt request status bit			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the status it had before the write.

When using CAN interrupts, if the interrupt request sources are associated with errors, this register helps to know which source generated the interrupt.

(1) EIS (CAN Bus Error Interrupt Request Status) bit (Bit 5)

The EIS bit is set to "1" when a communication error is detected. This bit is cleared by writing "0" in software.

(2) PIS (Error Passive Interrupt Request Status) bit (Bit 6)

The PIS bit is set to "1" when the CAN module goes to an error passive state. This bit is cleared by writing "0" in software.

(3) OIS (Bus Off Interrupt Request Status) bit (Bit 7)

The OIS bit is set to "1" when the CAN module goes to a bus off passive state. This bit is cleared by writing "0" in software.

When writing to the CAN error interrupt request status, make sure only the bits to be cleared are set to "0" and all other bits are set to "1". Those bits that have been set to "1" are unaffected by writing in software and retain the value they had before the write.

CAN0 Error Interrupt Request Mask Register (CAN0ERIMK) <Address: H'0080 1015>
CAN1 Error Interrupt Request Mask Register (CAN1ERIMK) <Address: H'0080 1415>

b8	9	10	11	12	13	14	b15
0	0	0	0	0	EIM 0	PIM 0	OIM 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–12	No function assigned. Fix to "0".		0	0
13	EIM CAN bus error interrupt request mask bit	0: Mask (disable) interrupt request 1: Enable interrupt request	R	W
14	PIM Error passive interrupt request mask bit			
15	OIM Bus off interrupt request mask bit			

(1) EIM (CAN Bus Error Interrupt Request Mask) bit (Bit 13)

The EIM bit enables or disables the interrupt requests to be generated when CAN bus errors occurred. CAN bus error interrupt requests are enabled by setting this bit to "1".

(2) PIM (Error Passive Interrupt Request Mask) bit (Bit 14)

The PIM bit enables or disables the interrupt requests to be generated when the CAN module entered an error passive state. Error passive interrupt requests are enabled by setting this bit to "1".

(3) OIM (Bus Off Interrupt Request Mask) bit (Bit 15)

The OIM bit enables or disables the interrupt requests to be generated when the CAN module entered a bus off state. Bus off interrupt requests are enabled by setting this bit to "1".

CAN0 Single-Shot Interrupt Request Status Register (CAN0SSIST) <address: H'0080 1044>
CAN1 Single-Shot Interrupt Request Status Register (CAN1SSIST) <Address: H'0080 1444>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
SSIST0	SSIST1	SSIST2	SSIST3	SSIST4	SSIST5	SSIST6	SSIST7	SSIST8	SSIST9	SSIST10	SSIST11	SSIST12	SSIST13	SSIST14	SSIST15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	SSIST0 Slot 0 single-shot interrupt request status bit	0: No arbitration-lost or transmit error 1: Arbitration-lost or transmit error occurred	R(Note 1)	
1	SSIST1 Slot 1 single-shot interrupt request status bit			
2	SSIST2 Slot 2 single-shot interrupt request status bit			
3	SSIST3 Slot 3 single-shot interrupt request status bit			
4	SSIST4 Slot 4 single-shot interrupt request status bit			
5	SSIST5 Slot 5 single-shot interrupt request status bit			
6	SSIST6 Slot 6 single-shot interrupt request status bit			
7	SSIST7 Slot 7 single-shot interrupt request status bit			
8	SSIST8 Slot 8 single-shot interrupt request status bit			
9	SSIST9 Slot 9 single-shot interrupt request status bit			
10	SSIST10 Slot 10 single-shot interrupt request status bit			
11	SSIST11 Slot 11 single-shot interrupt request status bit			
12	SSIST12 Slot 12 single-shot interrupt request status bit			
13	SSIST13 Slot 13 single-shot interrupt request status bit			
14	SSIST14 Slot 14 single-shot interrupt request status bit			
15	SSIST15 Slot 15 single-shot interrupt request status bit			

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the status it had before the write.

If transmission in any slot failed for reasons of a detection of arbitration-lost or a transmit error while operating Single-shot mode, the corresponding bit in this register is set to "1". The bit is cleared by writing "0" in software.

Furthermore, if the corresponding bit in the CAN single-shot interrupt request mask register has been set to "1", an interrupt request can be generated when transmission failed.

When writing to the CAN single-shot interrupt request status, make sure only the bits to be cleared are set to "0" and all other bits are set to "1". Those bits that have been set to "1" are unaffected by writing in software and retain the value they had before the write.

CAN0 Single-Shot Interrupt Request Mask Register (CAN0SSIMK) <Address: H'0080 1048>
CAN1 Single-Shot Interrupt Request Mask Register (CAN1SSIMK) <Address: H'0080 1448>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
SSIMK0	SSIMK1	SSIMK2	SSIMK3	SSIMK4	SSIMK5	SSIMK6	SSIMK7	SSIMK8	SSIMK9	SSIMK10	SSIMK11	SSIMK12	SSIMK13	SSIMK14	SSIMK15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	SSIMK0 Slot 0 single-shot interrupt request mask bit	0: Disable interrupt request 1: Enable interrupt request	R	W
1	SSIMK1 Slot 1 single-shot interrupt request mask bit			
2	SSIMK2 Slot 2 single-shot interrupt request mask bit			
3	SSIMK3 Slot 3 single-shot interrupt request mask bit			
4	SSIMK4 Slot 4 single-shot interrupt request mask bit			
5	SSIMK5 Slot 5 single-shot interrupt request mask bit			
6	SSIMK6 Slot 6 single-shot interrupt request mask bit			
7	SSIMK7 Slot 7 single-shot interrupt request mask bit			
8	SSIMK8 Slot 8 single-shot interrupt request mask bit			
9	SSIMK9 Slot 9 single-shot interrupt request mask bit			
10	SSIMK10 Slot 10 single-shot interrupt request mask bit			
11	SSIMK11 Slot 11 single-shot interrupt request mask bit			
12	SSIMK12 Slot 12 single-shot interrupt request mask bit			
13	SSIMK13 Slot 13 single-shot interrupt request mask bit			
14	SSIMK14 Slot 14 single-shot interrupt request mask bit			
15	SSIMK15 Slot 15 single-shot interrupt request mask bit			

This register is used to enable or disable the interrupt requests that will be generated when transmission in each corresponding slot has failed. Setting any bit in this register to "1" enables the interrupt request to be generated when transmission in the corresponding slot (in single-shot mode only) has failed. The CAN Single-Shot Interrupt Request Status Register helps to know which slot requested the interrupt.

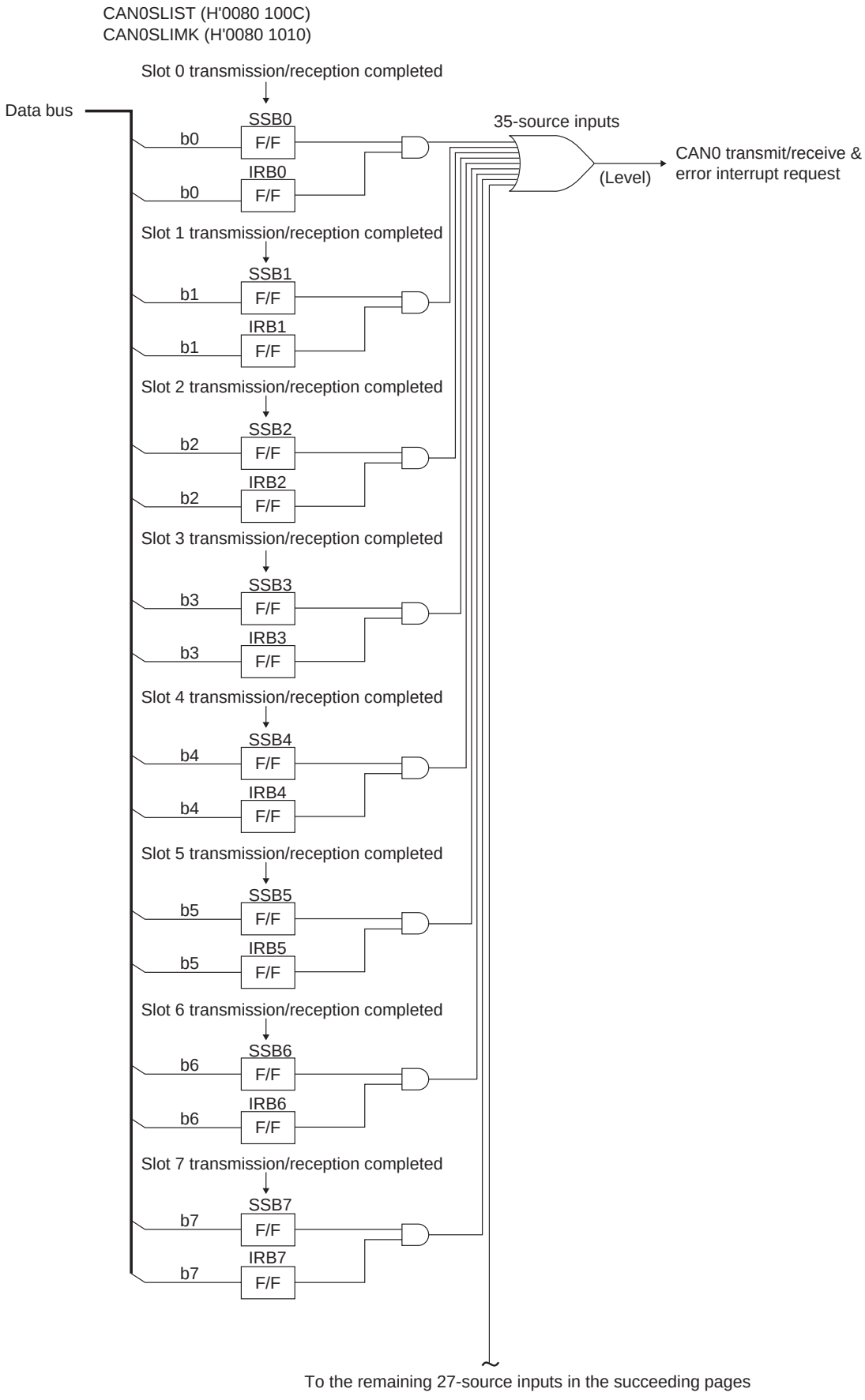


Figure 13.2.3 Block Diagram of CAN0 Transmit/Receive & Error Interrupt Requests (1/5)

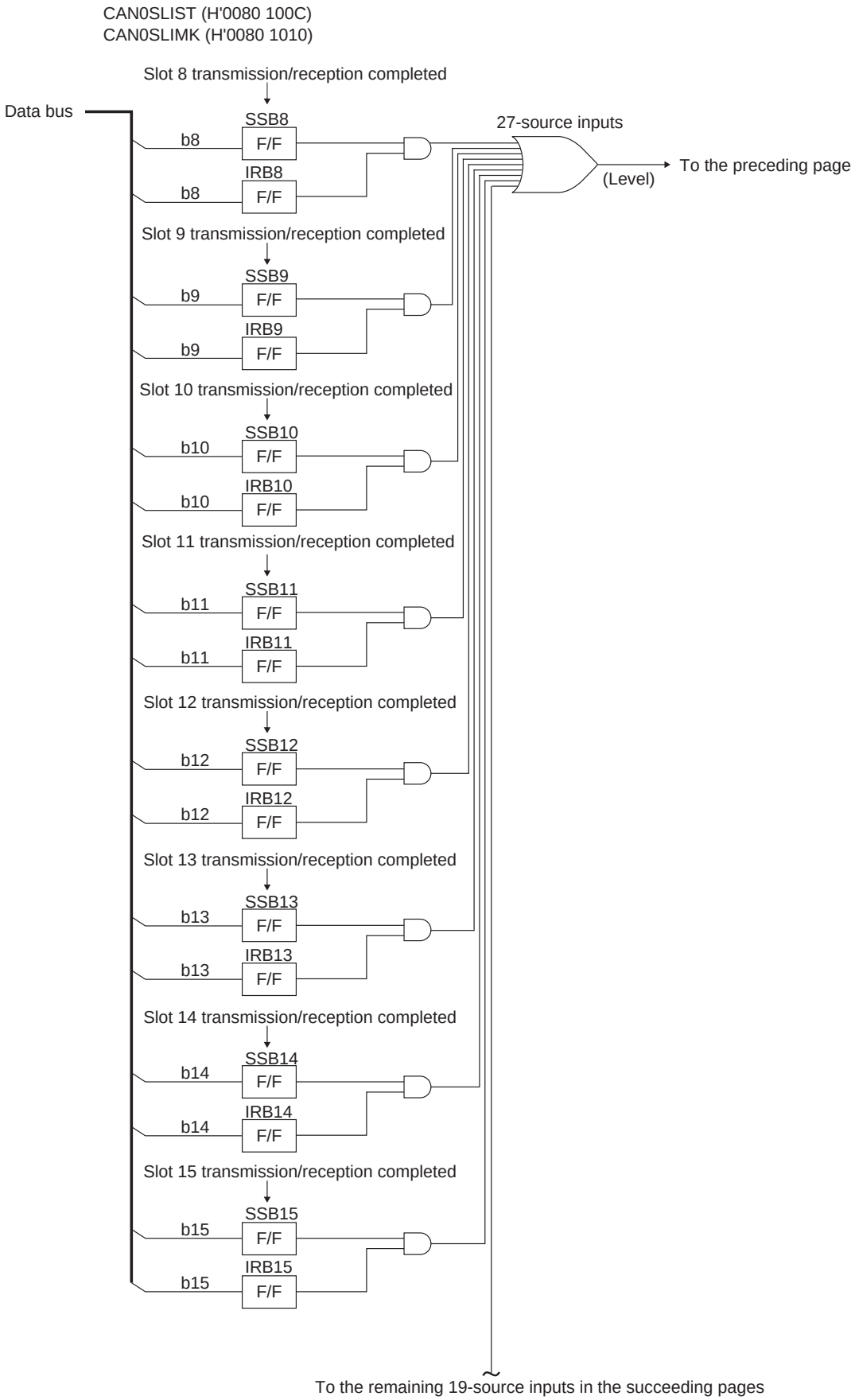


Figure 13.2.4 Block Diagram of CAN0 Transmit/Receive & Error Interrupt Requests (2/5)

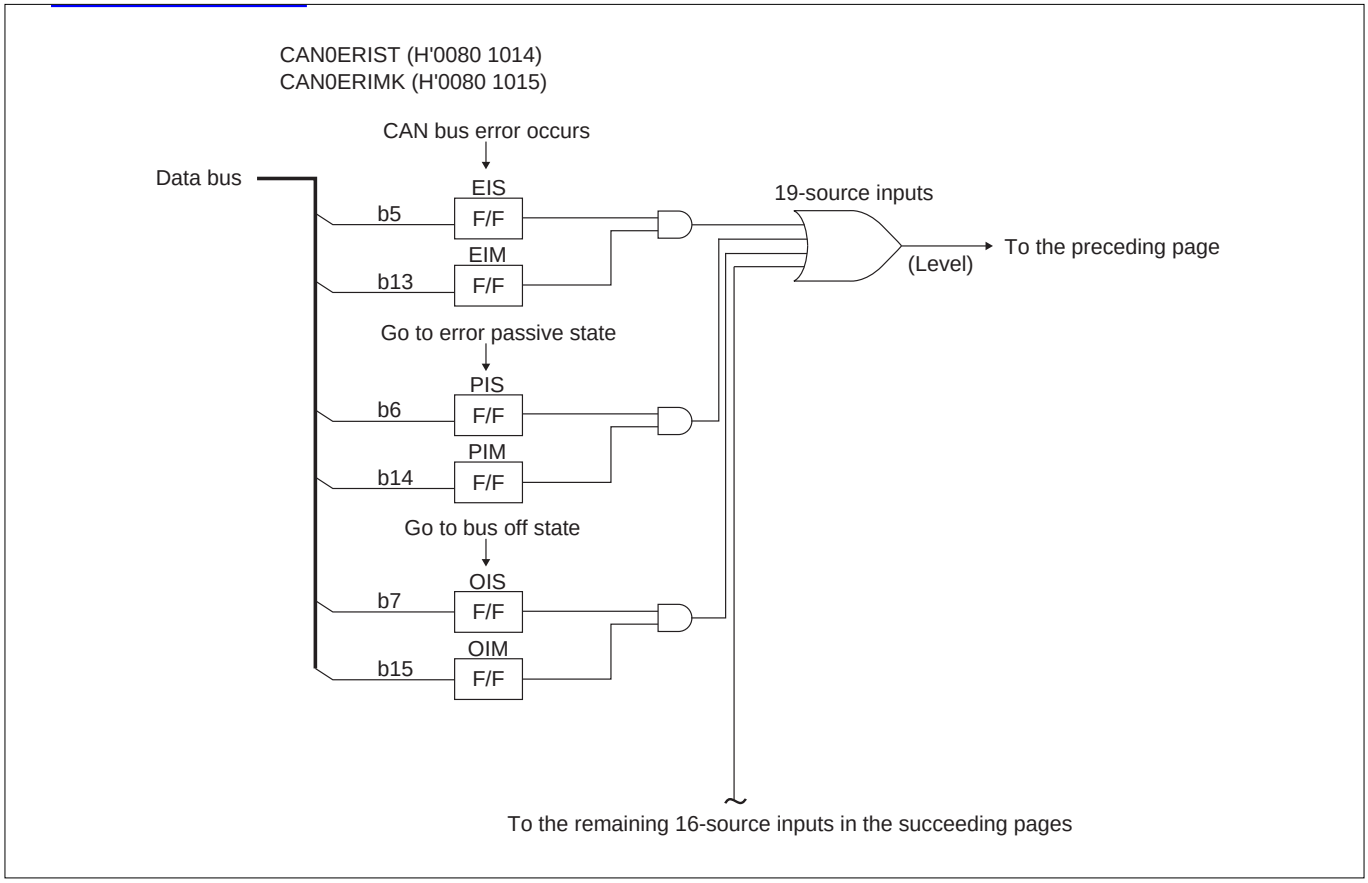


Figure 13.2.5 Block Diagram of CAN0 Transmit/Receive & Error Interrupt Requests (3/5)

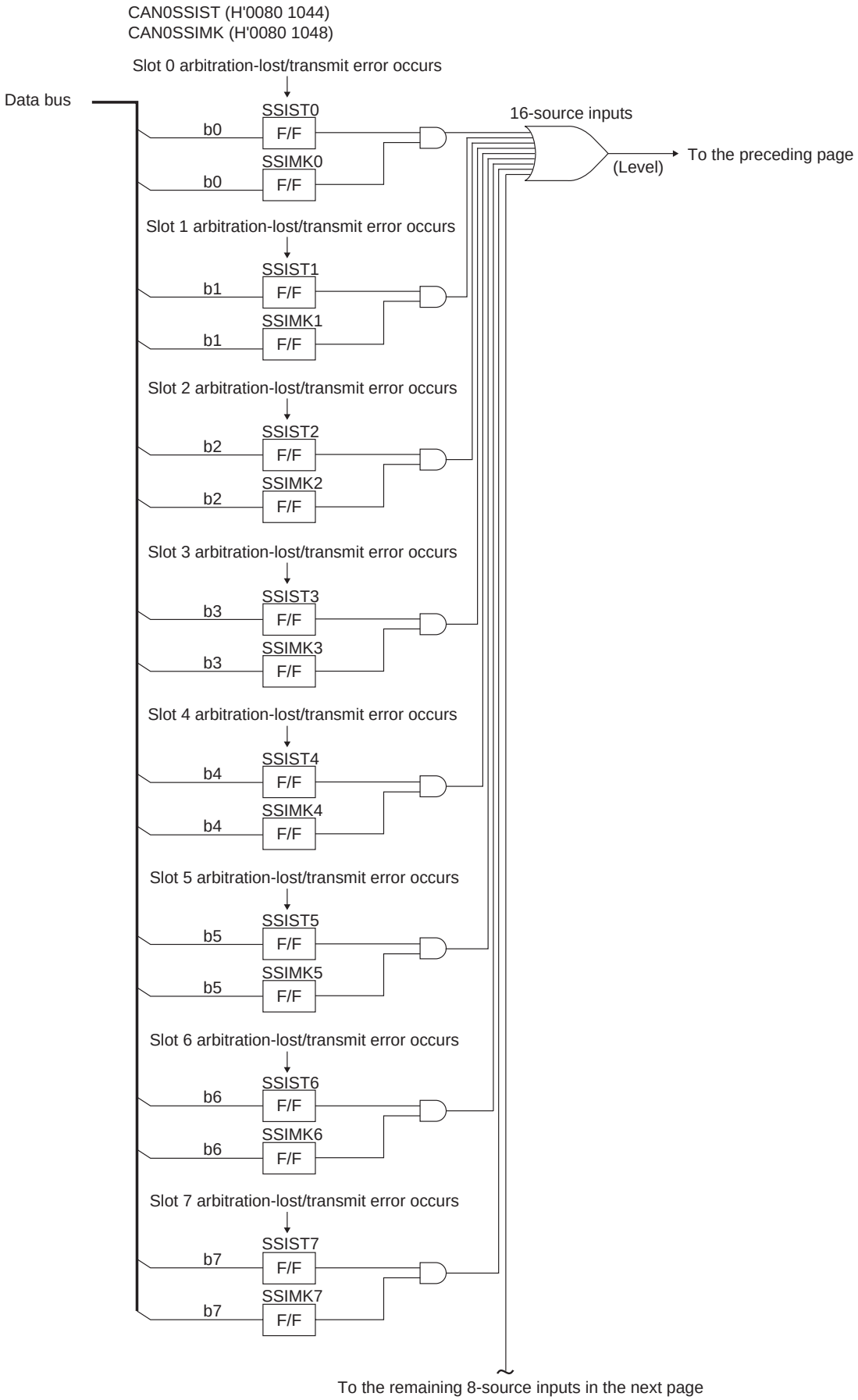


Figure 13.2.6 Block Diagram of CAN0 Transmit/Receive & Error Interrupt Requests (4/5)

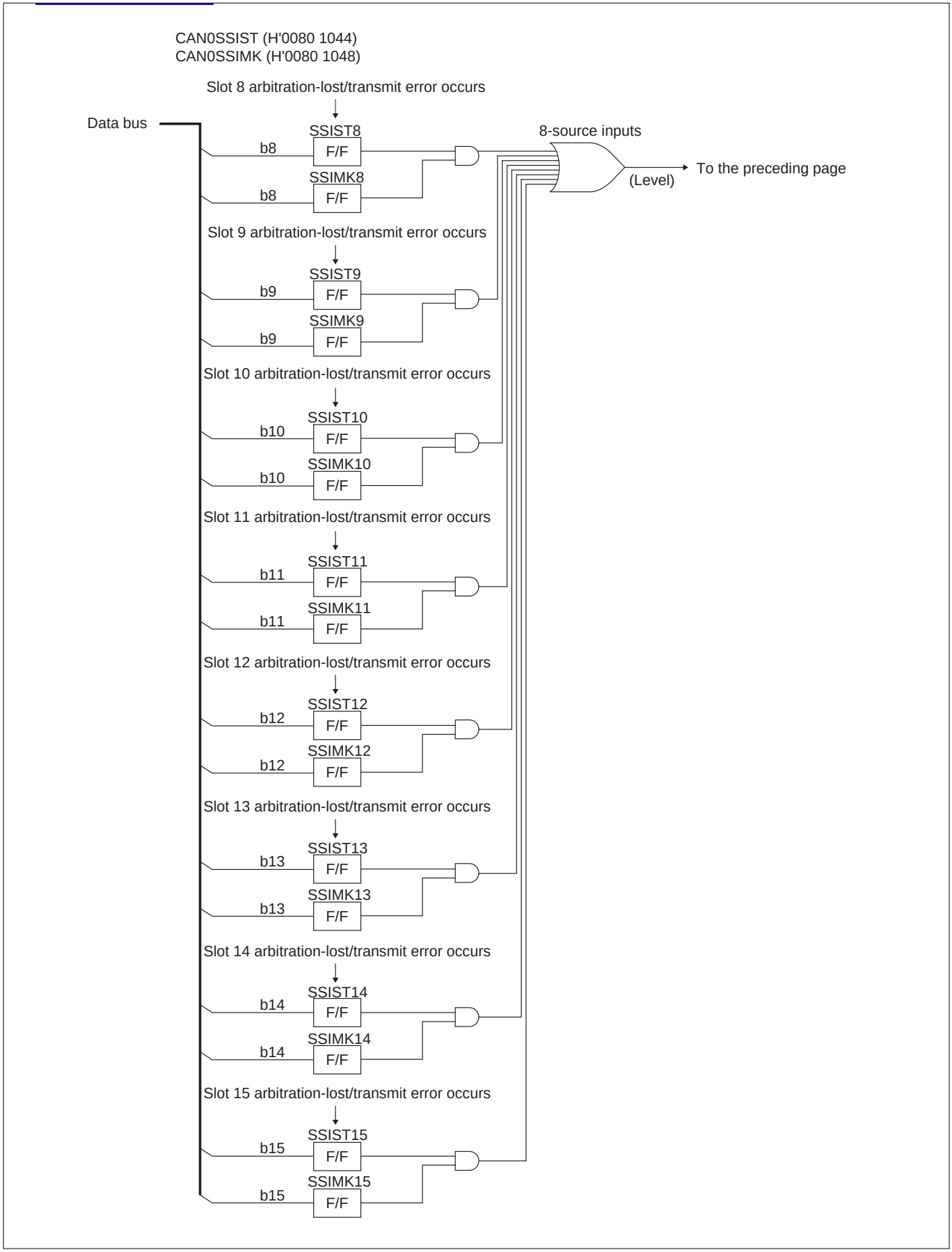


Figure 13.2.7 Block Diagram of CAN0 Transmit/Receive & Error Interrupt Requests (5/5)

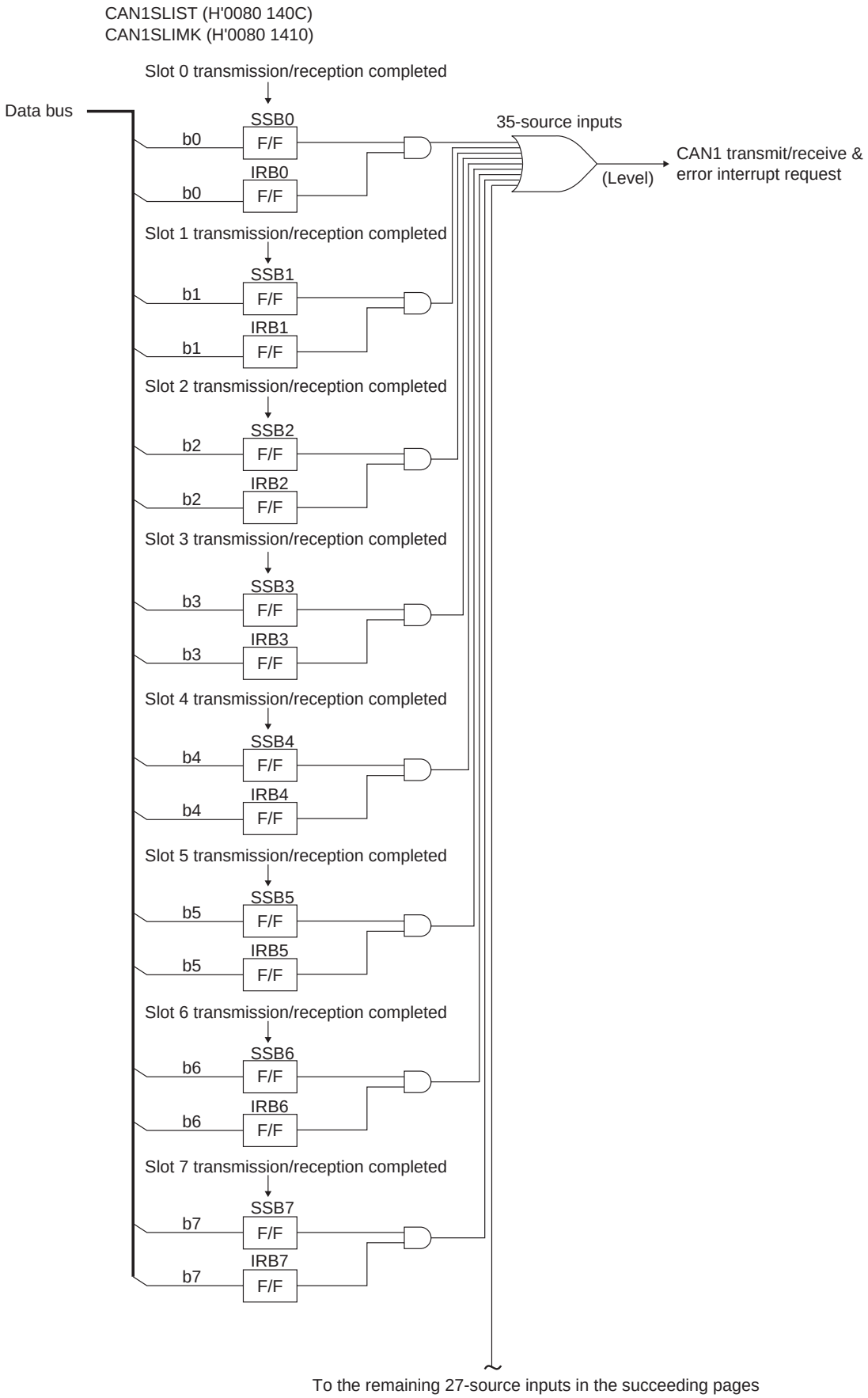


Figure 13.2.8 Block Diagram of CAN1 Transmit/Receive & Error Interrupt Requests (1/5)

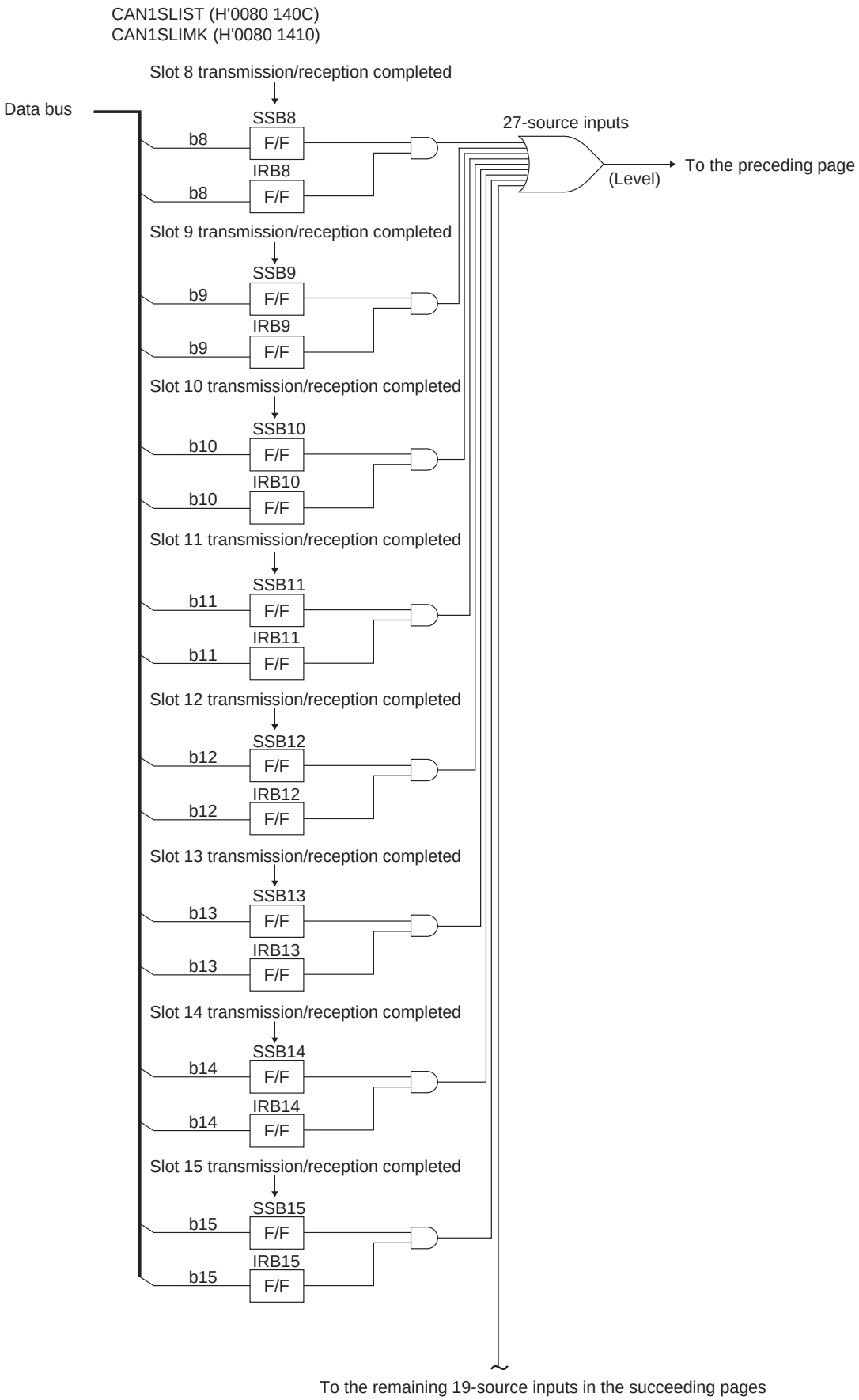


Figure 13.2.9 Block Diagram of CAN1 Transmit/Receive & Error Interrupt Requests (2/5)

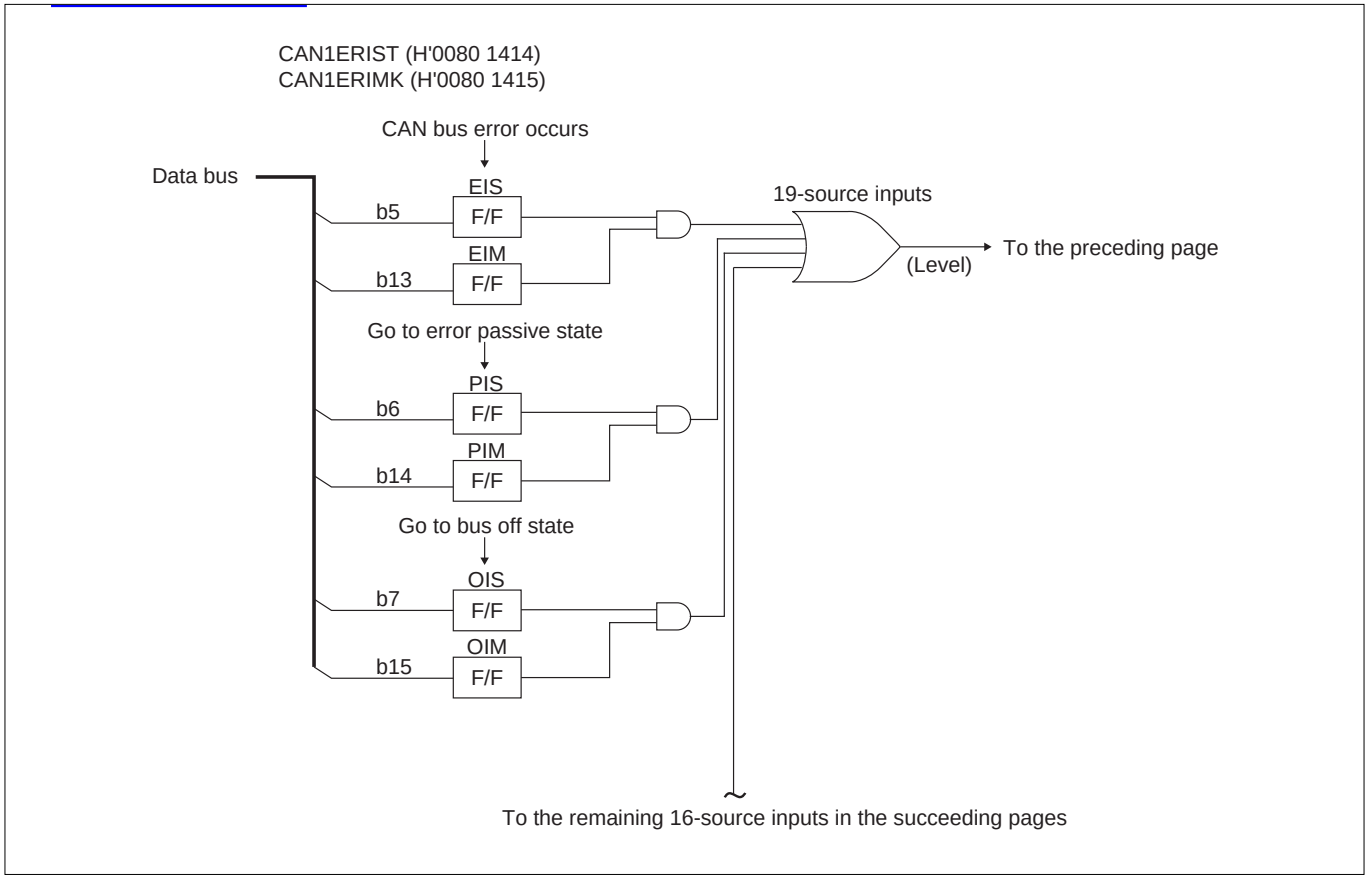


Figure 13.2.10 Block Diagram of CAN1 Transmit/Receive & Error Interrupt Requests (3/5)

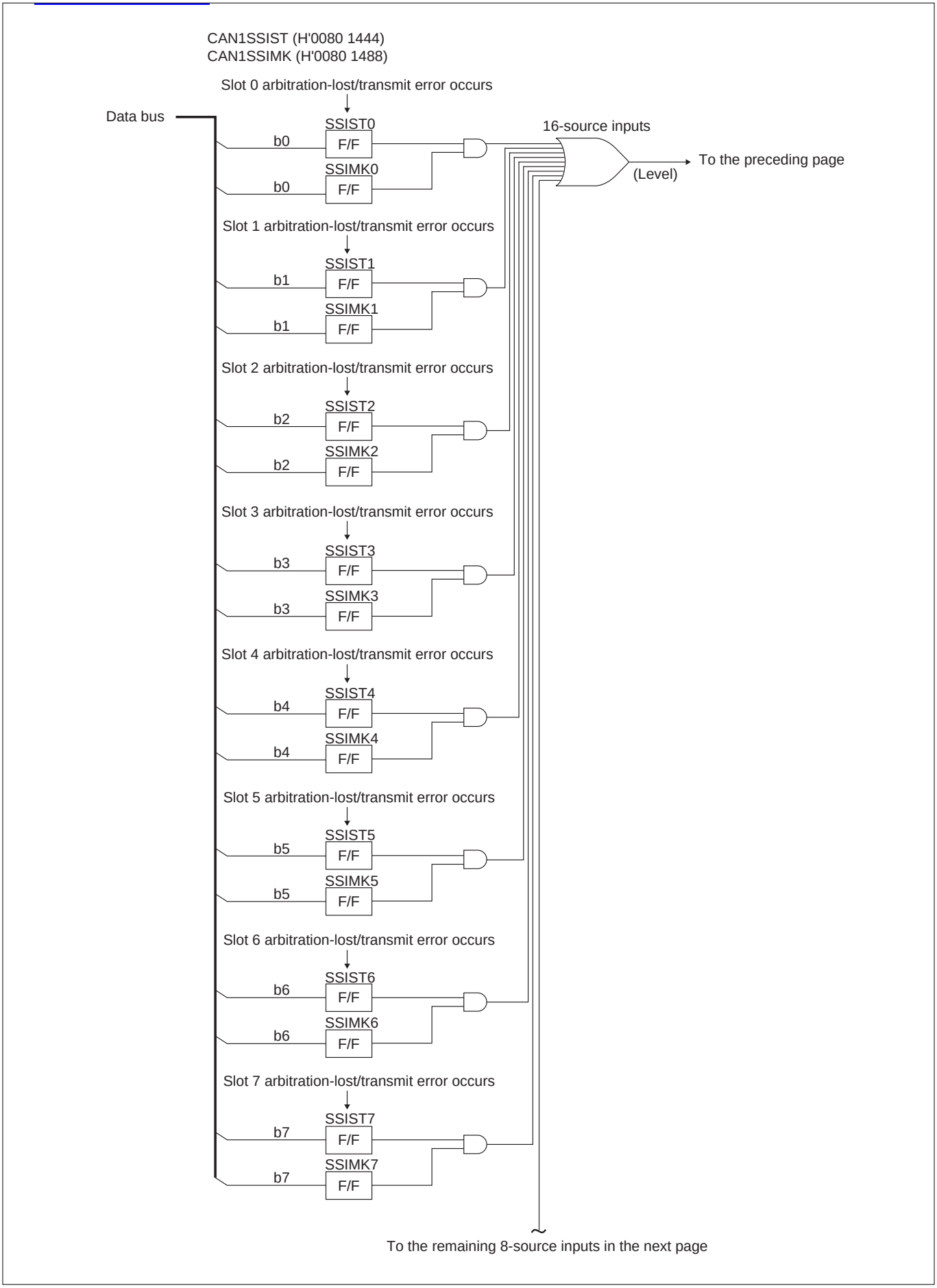


Figure 13.2.11 Block Diagram of CAN1 Transmit/Receive & Error Interrupt Requests (4/5)

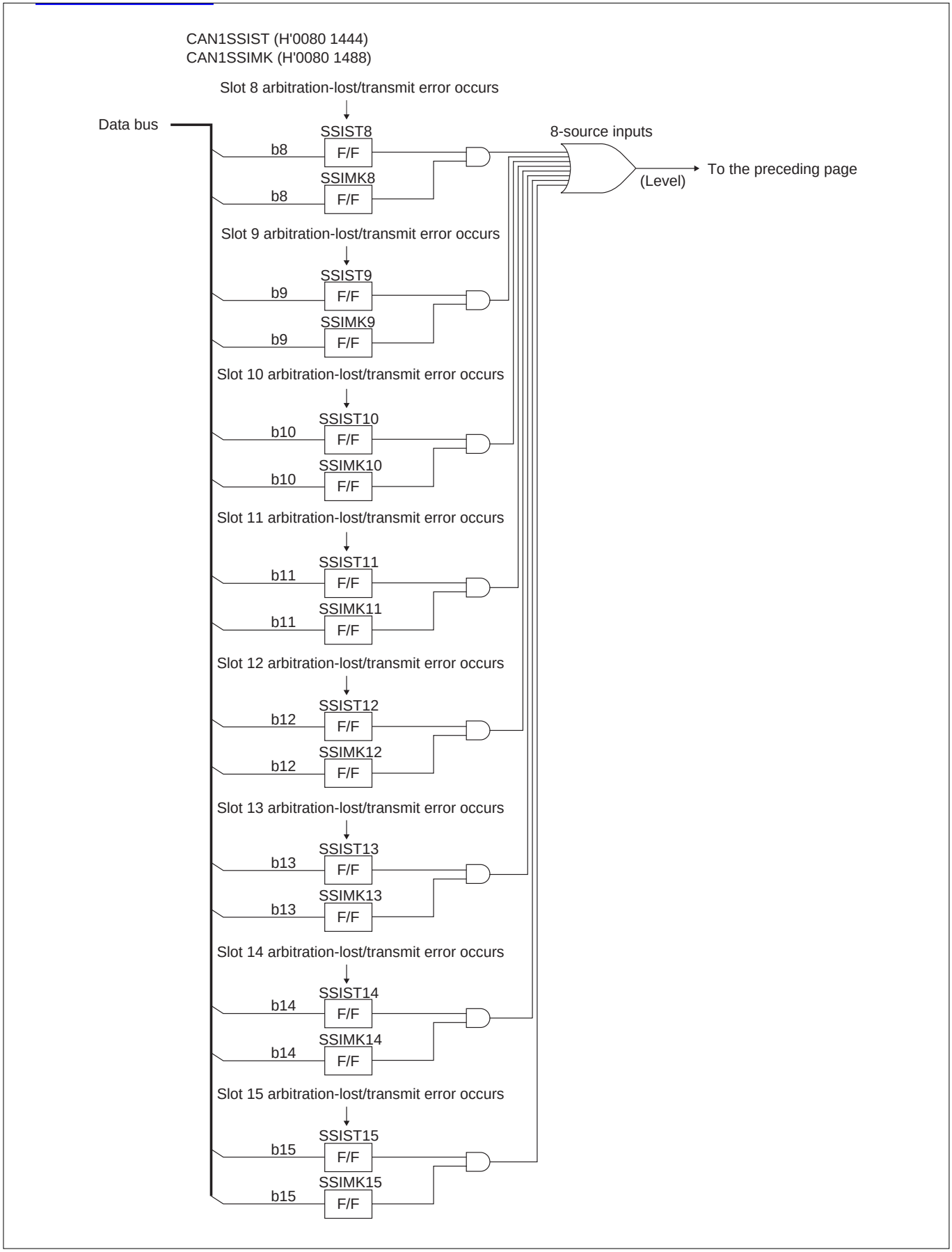


Figure 13.2.12 Block Diagram of CAN1 Transmit/Receive & Error Interrupt Requests (5/5)

13.2.9 CAN Cause of Error Registers

CAN0 Cause of Error Register (CAN0EF) <Address: H'0080 1017>
CAN1 Cause of Error Register (CAN1EF) <Address: H'0080 1417>

b8	9	10	11	12	13	14	b15
TRE	RCVE	BITE0	BITE1	STFE	FORME	CRCE	ACKE
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R W
8	TRE Transmit error detection bit	0: Error not detected 1: Transmit error detected	R (Note 1)
9	RCVE Receive error detection bit	0: Error not detected 1: Receive error detected	R (Note 1)
10	BITE0 "0" sending bit error detection bit	0: No bit error is detected 1: Bit error is detected when sending a "0"	R (Note 1)
11	BITE1 "1" sending bit error detection bit	0: No bit error is detected 1: Bit error is detected when sending a "1"	R (Note 1)
12	STFE Stuff error detection bit	0: Error not detected 1: Stuff error detected	R (Note 1)
13	FORME Form error detection bit	0: Error not detected 1: Form error detected	R (Note 1)
14	CRCE CRC error detection bit	0: Error not detected 1: CRC error detected	R (Note 1)
15	ACKE ACK error detection bit	0: Error not detected 1: ACK error detected	R (Note 1)

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the status it had before the write.

This register indicates error information when a communication error occurred.
Each bit in this register is set every time a communication error is detected, and is not cleared unless a program writes a "0" to the relevant bit.

(1) TRE (Transmit Error Detection) bit (Bit 8)

This bit is set to "1" when a communication error is detected while operating as a transmit node. The bit is cleared by writing a "0" in software.

(2) RCVE (Receive Error Detection) bit (Bit 9)

This bit is set to "1" when a communication error is detected while operating as a receive node. The bit is cleared by writing a "0" in software.

(3) BITE0 ("0" Sending Bit Error Detection) bit (Bit 10)

This bit is set to "1" when a bit error is detected while sending a "0" from CTX. The bit is cleared by writing a "0" in software.

(4) BITE1 ("1" Sending Bit Error Detection) bit (Bit 11)

This bit is set to "1" when a bit error is detected while sending a "1" from CTX. The bit is cleared by writing a "0" in software.

(5) STFE (Stuff Error Detection) bit (Bit 12)

This bit is set to "1" when a stuff error was detected. The bit is cleared by writing a "0" in software.

(6) FORME (Form Error Detection) bit (Bit 13)

This bit is set to "1" when a form error was detected. The bit is cleared by writing a "0" in software.

(7) CRCE (CRC Error Detection) bit (Bit 14)

This bit is set to "1" when a CRC error was detected. The bit is cleared by writing a "0" in software.

(8) ACKE (ACK Error Detection) bit (Bit 15)

This bit is set to "1" when an ACK error was detected. The bit is cleared by writing a "0" in software

Note: • Depending on the error status, two or more bits may be set at the same time.

13.2.10 CAN Mode Registers

CAN0 Mode Register (CAN0MOD) <Address: H'0080 1018>
CAN1 Mode Register (CAN1MOD) <Address: H'0080 1418>

b0	1	2	3	4	5	6	b7
						CMOD	
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–5	No function assigned. Fix to "0".		0	0
6–7	CMOD CAN operation mode select bit	00: Normal mode 01: Bus monitor mode 10: Self-diagnostic mode 11: Settings inhibited	R	W

(1) CMOD (CAN Operation Mode Select) bits (Bit 6, Bit 7)

These bits select the CAN operation mode.

- **Normal operation mode**
  Normal transmit/receive operations can be performed.
- **Bus monitor mode**
  Only receive operation is performed. During bus monitor mode, the CTX output is fixed high and neither ACK nor an error frame can be returned.

  Note: • During bus monitor mode, issuing transmit requests is inhibited. The ACK bit is handled as "Don't care" during bus monitor mode. Therefore, if all bits of data including the CRC delimiter are received normally, it is assumed that data has been received normally no matter whether the ACK bit is high.
- **Self-diagnostic mode**
  CTX and CRX are connected together internally in the CAN module. When combined with loopback mode, this mode allows communication to be performed within the CAN module alone. During self-diagnostic mode, the CTX pin output is fixed high even when transmitting.

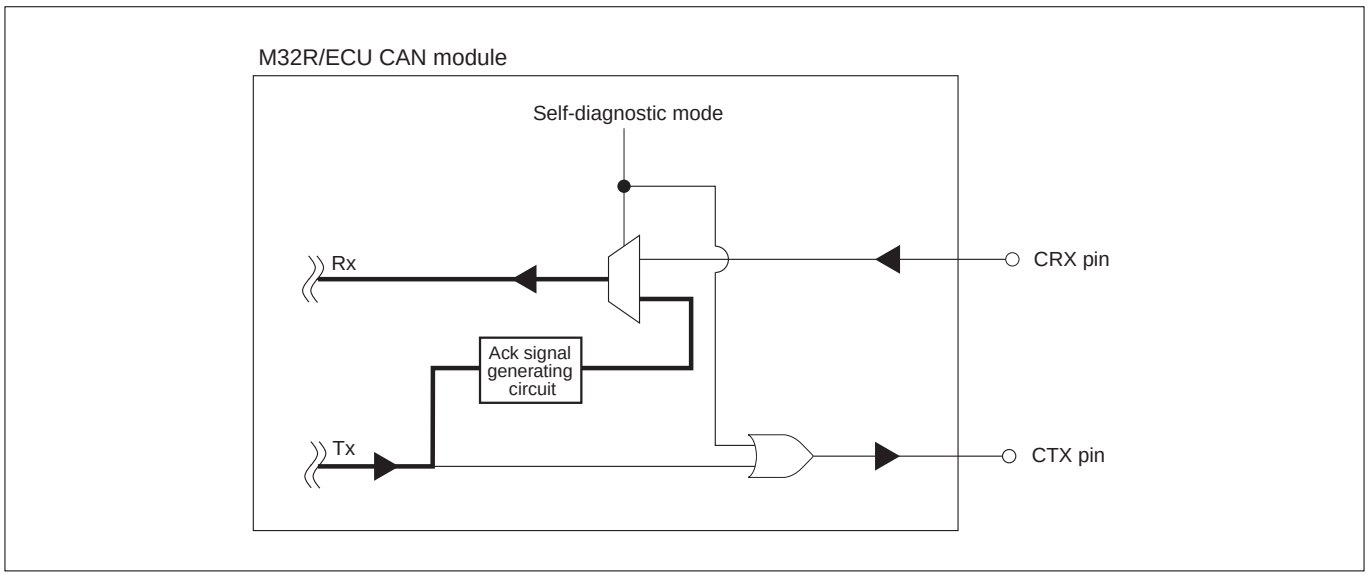


Figure 13.2.13 Conceptual Diagram of Self-Diagnostic Mode

13.2.11 CAN DMA Transfer Request Select Registers

CAN0 DMA Transfer Request Select Register (CAN0DMARQ) <Address: H'0080 1019>
CAN1 DMA Transfer Request Select Register (CAN1DMARQ) <Address: H'0080 1419>

b8	9	10	11	12	13	14	b15
0	0	0	0	0	0	CDMSEL1 0	CDMSEL0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–13	No function assigned. Fix to "0".		0	0
14	CDMSEL1 CAN DMA1 transfer request source select bit	0: Slot 1 transmission failed 1: Slot 14 transmission/reception completed	R	W
15	CDMSEL0 CAN DMA0 transfer request source select bit	0: Slot 0 transmission failed 1: Slot 15 transmission/reception completed	R	W

CAN0 and 1 can generate DMA transfer requests. This register is used to select the cause or source of that request.

(1) CDMSEL1 (CAN DMA1 Transfer Request Source Select) bit (Bit 14)

This bit selects one of the following two as the cause or source of a transfer request to DMA7 and DMA9.

- **Slot 1 transmission failed**
  If the CDMSEL1 bit is set to "0", a transfer request is generated when transmission in slot 1 has failed for reasons of arbitration-lost or transmit error.
- **Slot 14 transmission/reception completed**
  If the CDMSEL1 bit is set to "1", a transfer request is generated when transmission/reception in slot 14 is completed.

  Notes: • If slot 14 has been set for remote frame transmission, a DMA transfer request is generated when remote frame transmission is completed as well as when data frame reception is completed.
  • If slot 14 has been set for remote frame reception (automatic response), a DMA transfer request is generated when remote frame reception is completed as well as when data frame transmission is completed.

(2) CDMSEL0 (CAN DMA0 Transfer Request Source Select) bit (Bit 15)

This bit selects one of the following two as the cause or source of a transfer request to DMA6 and DMA8.

- **Slot 0 transmission failed**
  If the CDMSEL0 bit is set to "0", a transfer request is generated when transmission in slot 0 has failed for reasons of arbitration-lost or transmit error.
- **Slot 15 transmission/reception completed**
  If the CDMSEL0 bit is set to "1", a transfer request is generated when transmission/reception in slot 15 is completed.

  Notes: • If slot 15 has been set for remote frame transmission, a DMA transfer request is generated when remote frame transmission is completed as well as when data frame reception is completed.
  • If slot 15 has been set for remote frame reception (automatic response), a DMA transfer request is generated when remote frame reception is completed as well as when data frame transmission is completed.

13.2.12 CAN Mask Registers

CAN0 Global Mask Register Standard ID0 (C0GMSKS0) <Address: H'0080 1028>
CAN0 Local Mask Register A Standard ID0 (C0LMSKAS0) <Address: H'0080 1030>
CAN0 Local Mask Register B Standard ID0 (C0LMSKBS0) <Address: H'0080 1038>

CAN1 Global Mask Register Standard ID0 (C1GMSKS0) <Address: H'0080 1428>
CAN1 Local Mask Register A Standard ID0 (C1LMSKAS0) <Address: H'0080 1430>
CAN1 Local Mask Register B Standard ID0 (C1LMSKBS0) <Address: H'0080 1438>

b0	1	2	3	4	5	6	b7
0	0	0	SID0M 0	SID1M 0	SID2M 0	SID3M 0	SID4M 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–2	No function assigned. Fix to "0".		0	0
3–7	SID0M–SID4M (Standard mask ID0–standard mask ID4)	0: ID not checked 1: ID checked	R	W

CAN0 Global Mask Register Standard ID1 (C0GMSKS1) <Address: H'0080 1029>
CAN0 Local Mask Register A Standard ID1 (C0LMSKAS1) <Address: H'0080 1031>
CAN0 Local Mask Register B Standard ID1 (C0LMSKBS1) <Address: H'0080 1039>

CAN1 Global Mask Register Standard ID1 (C1GMSKS1) <Address: H'0080 1429>
CAN1 Local Mask Register A Standard ID1 (C1LMSKAS1) <Address: H'0080 1431>
CAN1 Local Mask Register B Standard ID1 (C1LMSKBS1) <Address: H'0080 1439>

b8	9	10	11	12	13	14	b15
0	0	SID5M 0	SID6M 0	SID7M 0	SID8M 0	SID9M 0	SID10M 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–9	No function assigned. Fix to "0".		0	0
10–15	SID5M–SID10M (Standard mask ID5–standard mask ID10)	0: ID not checked 1: ID checked	R	W

Three mask registers are used in acceptance filtering: global mask register, local mask register A and local mask register B. The global mask register is used for message slots 0-13, while local mask registers A and B are used for message slots 14 and 15, respectively.

- If any bit in this register is set to "0", the corresponding ID bit is masked (assumed to have matched) during acceptance filtering.
- If any bit in this register is set to "1", the corresponding ID bit is compared with the receive ID during acceptance filtering and when it matches the ID set in the message slot, the received data is stored in it.

Notes: • SID0M corresponds to the MSB of the standard ID.
• The global mask register can only be modified when none of slots 0-13 have receive requests set.
• The local mask register A can only be modified when slot 14 does not have a receive request set.
• The local mask register B can only be modified when slot 15 does not have a receive request set.

CAN0 Global Mask Register Extended ID0 (C0GMSKE0) <Address: H'0080 102A>
CAN0 Local Mask Register A Extended ID0 (C0LMSKAE0) <Address: H'0080 1032>
CAN0 Local Mask Register B Extended ID0 (C0LMSKBE0) <Address: H'0080 103A>

CAN1 Global Mask Register Extended ID0 (C1GMSKE0) <Address: H'0080 142A>
CAN1 Local Mask Register A Extended ID0 (C1LMSKAE0) <Address: H'0080 1432>
CAN1 Local Mask Register B Extended ID0 (C1LMSKBE0) <Address: H'0080 143A>

b0	1	2	3	4	5	6	b7
0	0	0	0	EID0M 0	EID1M 0	EID2M 0	EID3M 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0–3	No function assigned. Fix to "0".		0	0
4–7	EID0M–EID3M (Extended mask ID0–extended mask ID3)	0: ID not checked 1: ID checked	R	W

CAN0 Global Mask Register Extended ID1 (C0GMSKE1) <Address: H'0080 102B>
CAN0 Local Mask Register A Extended ID1 (C0LMSKAE1) <Address: H'0080 1033>
CAN0 Local Mask Register B Extended ID1 (C0LMSKBE1) <Address: H'0080 103B>

CAN1 Global Mask Register Extended ID1 (C1GMSKE1) <Address: H'0080 142B>
CAN1 Local Mask Register A Extended ID1 (C1LMSKAE1) <Address: H'0080 1433>
CAN1 Local Mask Register B Extended ID1 (C1LMSKBE1) <Address: H'0080 143B>

b8	9	10	11	12	13	14	b15
EID4M 0	EID5M 0	EID6M 0	EID7M 0	EID8M 0	EID9M 0	EID10M 0	EID11M 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–15	EID4M–EID11M (Extended mask ID4–extended mask ID11)	0: ID not checked 1: ID checked	R	W

CAN0 Global Mask Register Extended ID2 (C0GMSKE2) <Address: H'0080 102C>
CAN0 Local Mask Register A Extended ID2 (C0LMSKAE2) <Address: H'0080 1034>
CAN0 Local Mask Register B Extended ID2 (C0LMSKBE2) <Address: H'0080 103C>

CAN1 Global Mask Register Extended ID2 (C1GMSKE2) <Address: H'0080 142C>
CAN1 Local Mask Register A Extended ID2 (C1LMSKAE2) <Address: H'0080 1434>
CAN1 Local Mask Register B Extended ID2 (C1LMSKBE2) <Address: H'0080 143C>

b0	1	2	3	4	5	6	B7
0	0	EID12M 0	EID13M 0	EID14M 0	EID15M 0	EID16M 0	EID17M 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0,1	No function assigned. Fix to "0".		0	0
2–7	EID12M–EID17M (Extended mask ID12–extended mask ID17)	0: ID not checked 1: ID checked	R	W

Three mask registers are used in acceptance filtering: global mask register, local mask register A and local mask register B. The global mask register is used for message slots 0-13, while local mask registers A and B are used for message slots 14 and 15, respectively.

- If any bit in this register is set to "0", the corresponding ID bit is masked (assumed to have matched) during acceptance filtering.
- If any bit in this register is set to "1", the corresponding ID bit is compared with the receive ID during acceptance filtering and when it matches the ID set in the message slot, the received data is stored in it.

Notes: • EID0M corresponds to the MSB of the extended ID.
• The global mask register can only be modified when none of slots 0-13 have receive requests set.
• The local mask register A can only be modified when slot 14 does not have a receive request set.
• The local mask register B can only be modified when slot 15 does not have a receive request set.

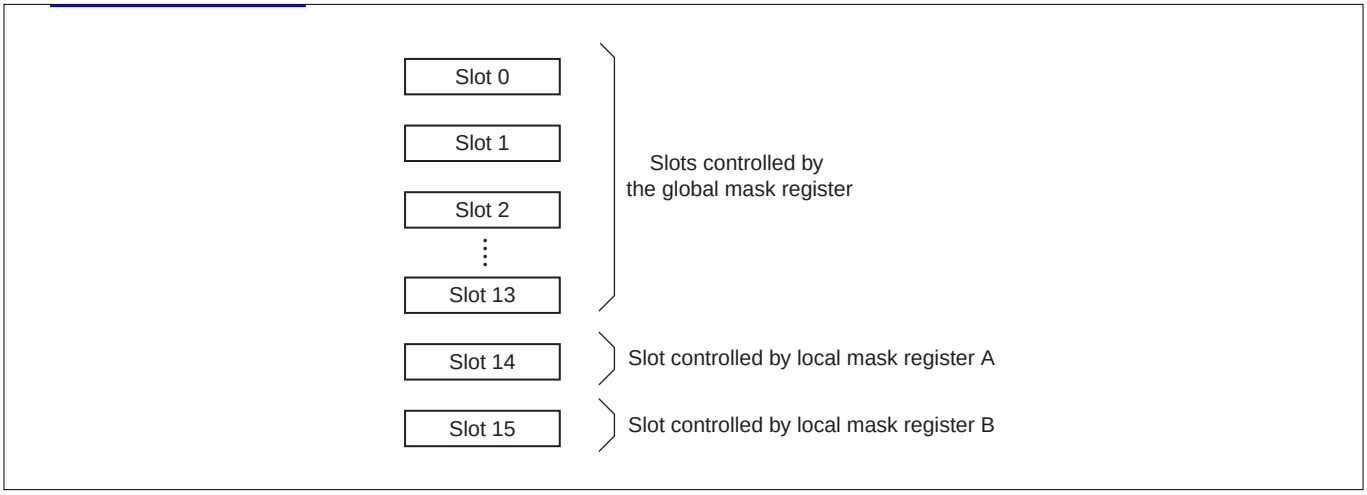


Figure 13.2.14 Relationship between the Mask Registers and the Controlled Slots

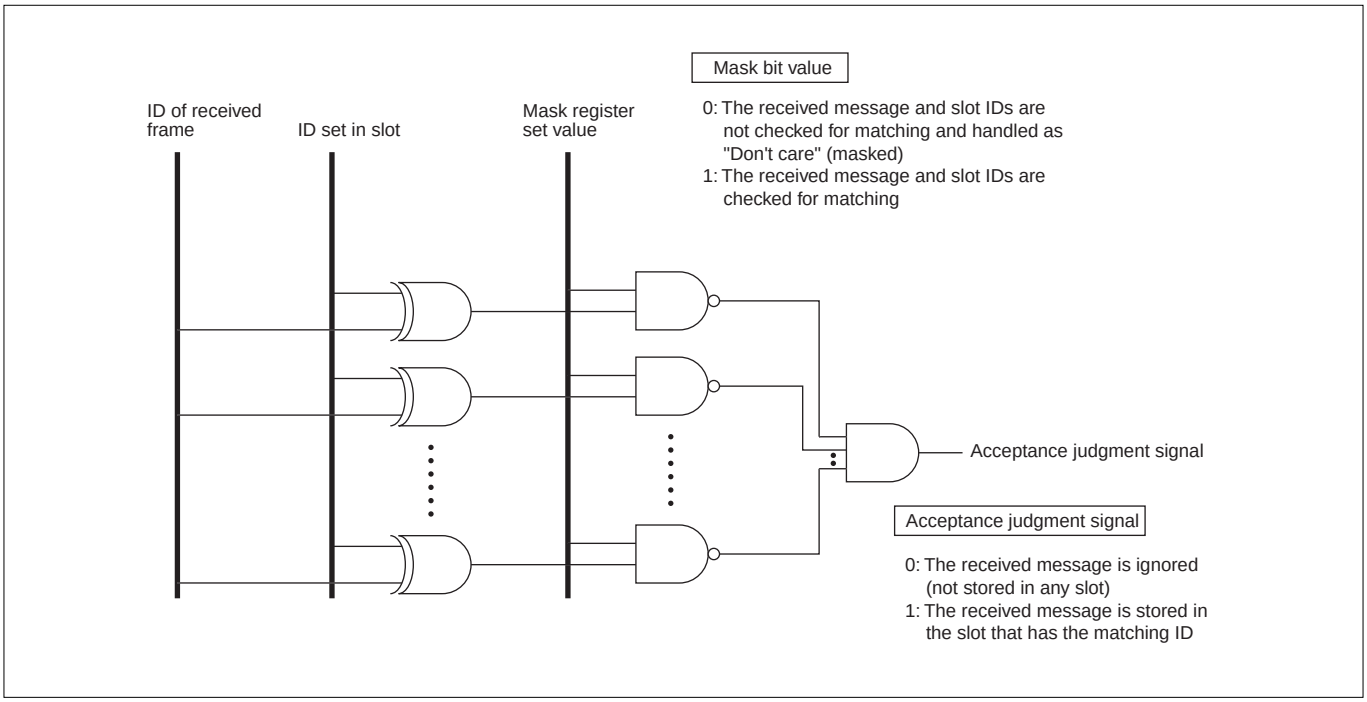


Figure 13.2.15 Concept of Acceptance Filtering

13.2.13 CAN Single-Shot Mode Control Registers

CAN0 Single-Shot Mode Control Register (CAN0SSMODE) <Address: H'0080 1040>
CAN1 Single-Shot Mode Control Register (CAN1SSMODE) <Address: H'0080 1440>

b0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b15
SSCNT0	SSCNT1	SSCNT2	SSCNT3	SSCNT4	SSCNT5	SSCNT6	SSCNT7	SSCNT8	SSCNT9	SSCNT10	SSCNT11	SSCNT12	SSCNT13	SSCNT14	SSCNT15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<Upon exiting reset: H'0000>

b	Bit Name	Function	R	W
0	SSCNT0 (Slot 0 single-shot mode bit)	0: Normal mode	R	W
1	SSCNT1 (Slot 1 single-shot mode bit)	1: Single-shot mode		
2	SSCNT2 (Slot 2 single-shot mode bit)			
3	SSCNT3 (Slot 3 single-shot mode bit)			
4	SSCNT4 (Slot 4 single-shot mode bit)			
5	SSCNT5 (Slot 5 single-shot mode bit)			
6	SSCNT6 (Slot 6 single-shot mode bit)			
7	SSCNT7 (Slot 7 single-shot mode bit)			
8	SSCNT8 (Slot 8 single-shot mode bit)			
9	SSCNT9 (Slot 9 single-shot mode bit)			
10	SSCNT10 (Slot 10 single-shot mode bit)			
11	SSCNT11 (Slot 11 single-shot mode bit)			
12	SSCNT12 (Slot 12 single-shot mode bit)			
13	SSCNT13 (Slot 13 single-shot mode bit)			
14	SSCNT14 (Slot 14 single-shot mode bit)			
15	SSCNT15 (Slot 15 single-shot mode bit)			

Normally in CAN, if transmission has failed for reasons of arbitration-lost or transmit error, the transmit operation is continued until successfully transmitted. This register is used to specify for each slot whether or not to retry a transmit operation in such a case.
In single-shot mode, if transmission fails for reasons of arbitration-lost or transmit error, the transmit operation is not retried. If any bit in this register is set to "1", the corresponding slot operates in single-shot mode.

Note: • Settings of this register can only be changed when the message slot control register for the slot whose corresponding bit is to be modified is in the H'00 state.

13.2.14 CAN Message Slot Control Registers

CAN0 Message Slot 0 Control Register (C0MSL0CNT)	<Address: H'0080 1050>
CAN0 Message Slot 1 Control Register (C0MSL1CNT)	<Address: H'0080 1051>
CAN0 Message Slot 2 Control Register (C0MSL2CNT)	<Address: H'0080 1052>
CAN0 Message Slot 3 Control Register (C0MSL3CNT)	<Address: H'0080 1053>
CAN0 Message Slot 4 Control Register (C0MSL4CNT)	<Address: H'0080 1054>
CAN0 Message Slot 5 Control Register (C0MSL5CNT)	<Address: H'0080 1055>
CAN0 Message Slot 6 Control Register (C0MSL6CNT)	<Address: H'0080 1056>
CAN0 Message Slot 7 Control Register (C0MSL7CNT)	<Address: H'0080 1057>
CAN0 Message Slot 8 Control Register (C0MSL8CNT)	<Address: H'0080 1058>
CAN0 Message Slot 9 Control Register (C0MSL9CNT)	<Address: H'0080 1059>
CAN0 Message Slot 10 Control Register (C0MSL10CNT)	<Address: H'0080 105A>
CAN0 Message Slot 11 Control Register (C0MSL11CNT)	<Address: H'0080 105B>
CAN0 Message Slot12 Control Register (C0MSL12CNT)	<Address: H'0080 105C>
CAN0 Message Slot 13 Control Register (C0MSL13CNT)	<Address: H'0080 105D>
CAN0 Message Slot 14 Control Register (C0MSL14CNT)	<Address: H'0080 105E>
CAN0 Message Slot 15 Control Register (C0MSL15CNT)	<Address: H'0080 105F>
CAN1 Message Slot 0 Control Register (C1MSL0CNT)	<Address: H'0080 1450>
CAN1 Message Slot 1 Control Register (C1MSL1CNT)	<Address: H'0080 1451>
CAN1 Message Slot 2 Control Register (C1MSL2CNT)	<Address: H'0080 1452>
CAN1 Message Slot 3 Control Register (C1MSL3CNT)	<Address: H'0080 1453>
CAN1 Message Slot 4 Control Register (C1MSL4CNT)	<Address: H'0080 1454>
CAN1 Message Slot 5 Control Register (C1MSL5CNT)	<Address: H'0080 1455>
CAN1 Message Slot 6 Control Register (C1MSL6CNT)	<Address: H'0080 1456>
CAN1 Message Slot 7 Control Register (C1MSL7CNT)	<Address: H'0080 1457>
CAN1 Message Slot 8 Control Register (C1MSL8CNT)	<Address: H'0080 1458>
CAN1 Message Slot 9 Control Register (C1MSL9CNT)	<Address: H'0080 1459>
CAN1 Message Slot 10 Control Register (C1MSL10CNT)	<Address: H'0080 145A>
CAN1 Message Slot 11 Control Register (C1MSL11CNT)	<Address: H'0080 145B>
CAN1 Message Slot 12 Control Register (C1MSL12CNT)	<Address: H'0080 145C>
CAN1 Message Slot 13 Control Register (C1MSL13CNT)	<Address: H'0080 145D>
CAN1 Message Slot 14 Control Register (C1MSL14CNT)	<Address: H'0080 145E>
CAN1 Message Slot 15 Control Register (C1MSL15CNT)	<Address: H'0080 145F>

b0(b8)	1	2	3	4	5	6	b7(b15)
TR	RR	RM	RL	RA	ML	TRSTAT	TRFIN
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
0 (8)	TR Transmit request bit	0: Do not use the message slot as transmit slot 1: Use the message slot as transmit slot	R	W
1 (9)	RR Receive request bit	0: Do not use the message slot as receive slot 1: Use the message slot as receive slot	R	W
2 (10)	RM Remote bit	0: Transmit/receive data frame 1: Transmit/receive remote frame	R	W
3 (11)	RL Automatic response inhibit bit	0: Enable automatic response for remote frame 1: Disable automatic response for remote frame	R	W
4 (12)	RA Remote active bit	During BasicCAN mode 0: Receive data frame (status) 1: Receive remote frame (status) During normal mode 0: Data frame 1: Remote frame	R	–
5 (13)	ML Message lost bit	0: No message was lost 1: Message was lost	R(Note 1)	
6 (14)	TRSTAT Transmit/receive status bit	During a transmit slot 0: Transmission idle 1: Transmit request accepted During a receive slot 0: Reception idle 1: Storing received data	R	–
7 (15)	TRFIN Transmission/reception finished bit	During a transmit slot 0: Not transmitted yet 1: Finished transmitting During a receive slot 0: Not received yet 1: Finished receiving	R(Note 1)	

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the status it had before the write.

Notes: • If a transmit request is written to this register while the CAN module is reset (CANnCNT FRST or RST bit = "1"), it starts sending upon detecting 11 consecutive recessive bits on the CAN bus after exiting the reset state.
• If data/remote frame transmit requests are issued for two or more slots, the slot with the smallest slot number sends a frame. If data/remote frame receive requests are issued for two or more slots, the slot with the smallest slot number among the slots satisfying the receive condition receives a frame.
• If transmission failed when single-shot mode is selected, this register is cleared to H'00.

(1) TR (Transmit Request) bit (Bit 0, 8)

To use the message slot as a transmit slot, set this bit to "1". To use the message slot as a data frame or remote frame receive slot, set this bit to "0".

(2) RR (Receive Request) bit (Bit 1, 9)

To use the message slot as a receive slot, set this bit to "1". To use the message slot as a data frame or remote frame transmit slot, set this bit to "0".
If TR (Transmit Request) bit and RR (Receive Request) bit both are set to "1", device operation is undefined.

(3) RM (Remote) bit (Bit 2, 10)

To handle remote frames in the message slot, set this bit to "1". There are following two methods of settings to handle remote frames:

- **Set for remote frame transmission**
  The data set in the message slot is transmitted as a remote frame. When the CAN module finished sending, the slot automatically changes to a data frame receive slot. However, if a data frame is received before the CAN module finished sending a remote frame, the received data is stored in the message slot and the remote frame is not transmitted.
- **Set for remote frame reception**
  Remote frames are received. The processing to be performed after receiving a remote frame is selected by RL (automatic response inhibit) bit.

(4) RL (Automatic Response Inhibit) bit (Bit 3, 11)

This bit is effective when the message slot has been set as a remote frame receive slot. It selects the processing to be performed after receiving a remote frame. If this bit is set to "0", the message slot automatically changes to a transmit slot after receiving a remote frame and transmits the data set in it as a data frame. If this bit is set to "1", the message slot stops operating after receiving a remote frame.

Note: • Always set this bit to "0" unless the message slot is set for remote frame reception.

(5) RA (Remote Active) bit (Bit 4, 12)

This bit functions differently for slots 0-13 and slots 14 and 15.

- **Slots 0–13**
  This bit is set to "1" when the message slot is set for remote frame transmission (reception). Then, when remote frame transmission (reception) is completed, the bit is cleared to "0".
- **Slots 14 and 15**
  The function of this bit differs depending on how the CAN Control Register BCM (BasicCAN Mode) bit is set. If BCM = "0" (normal operation), this bit is set to "1" when the message slot is set for remote frame transmission (reception). If BCM = "1" (BasicCAN), this bit indicates which type of frame is received. During BasicCAN mode, the received data is stored in slots 14 and 15 for both data and remote frames. If RA = "0", it means that the frame stored in the slot is a data frame. If RA = "1", it means that the frame stored in the slot is a remote frame.

(6) ML (Message Lost) bit (Bit 5, 13)

This bit is effective for receive slots. It is set to "1" when unread received data contained in the message slot is overwritten by reception. This bit is cleared by writing "0" in software.

(7) TRSTAT (Transmit/Receive Status) bit (Bit 6, 14)

This bit indicates that the CAN module is sending or receiving and is accessing the message slot. This bit is set to "1" when the CAN module is accessing, and set to "0" when not accessing.

- **During a transmit slot**
  This bit is set to "1" when a transmit request for the message slot is accepted. It is cleared to "0" when the CAN module lost in bus arbitration, when a CAN bus error occurs, or when transmission is completed.
- **During a receive slot**
  This bit is set to "1" while the CAN module is receiving data, with the received data being stored in the message slot. Note that the value read from the message slot while the TRSTAT bit remains set is undefined.

(8) TRFIN (Transmit/Receive Finished) bit (Bit 7, 15)

This bit indicates that the CAN module finished sending or receiving.

- **When set for a transmit slot**
  This bit is set to "1" when the CAN module finished sending the data stored in the message slot.
  This bit is cleared by writing "0" in software. However, it cannot be cleared when the TRSTAT (Transmit/Receive Status) bit = "1".
- **When set for a receive slot**
  This bit is set to "1" when the CAN module finished receiving normally the data to be stored in the message slot. This bit is cleared by writing "0" in software. However, it cannot be cleared when the TRSTAT (Transmit/Receive Status) bit = "1".

Notes:
- Before reading the received data out of the message slot, be sure to clear the TRFIN (Transmit/Receive Finished) bit to "0". If the TRFIN (Transmit/Receive Finished) bit happens to be set to "1" after a read, it means that new received data was stored while reading and the read data contains an undefined value. In that case, discard the read data, clear the TRFIN bit to "0" and read out data again.
- When sending/receiving remote frames, the TRFIN bit is automatically cleared to "0" by hardware. Therefore, the TRFIN bit cannot be used as a transmission/reception-finished flag.

13.2.15 CAN Message Slots

CAN0 Message Slot 0 Standard ID0 (C0MSL0SID0)	<Address: H'0080 1100>
CAN0 Message Slot 1 Standard ID0 (C0MSL1SID0)	<Address: H'0080 1110>
CAN0 Message Slot 2 Standard ID0 (C0MSL2SID0)	<Address: H'0080 1120>
CAN0 Message Slot 3 Standard ID0 (C0MSL3SID0)	<Address: H'0080 1130>
CAN0 Message Slot 4 Standard ID0 (C0MSL4SID0)	<Address: H'0080 1140>
CAN0 Message Slot 5 Standard ID0 (C0MSL5SID0)	<Address: H'0080 1150>
CAN0 Message Slot 6 Standard ID0 (C0MSL6SID0)	<Address: H'0080 1160>
CAN0 Message Slot 7 Standard ID0 (C0MSL7SID0)	<Address: H'0080 1170>
CAN0 Message Slot 8 Standard ID0 (C0MSL8SID0)	<Address: H'0080 1180>
CAN0 Message Slot 9 Standard ID0 (C0MSL9SID0)	<Address: H'0080 1190>
CAN0 Message Slot 10 Standard ID0 (C0MSL10SID0)	<Address: H'0080 11A0>
CAN0 Message Slot 11 Standard ID0 (C0MSL11SID0)	<Address: H'0080 11B0>
CAN0 Message Slot 12 Standard ID0 (C0MSL12SID0)	<Address: H'0080 11C0>
CAN0 Message Slot 13 Standard ID0 (C0MSL13SID0)	<Address: H'0080 11D0>
CAN0 Message Slot 14 Standard ID0 (C0MSL14SID0)	<Address: H'0080 11E0>
CAN0 Message Slot 15 Standard ID0 (C0MSL15SID0)	<Address: H'0080 11F0>

CAN1 Message Slot 0 Standard ID0 (C1MSL0SID0)	<Address: H'0080 1500>
CAN1 Message Slot 1 Standard ID0 (C1MSL1SID0)	<Address: H'0080 1510>
CAN1 Message Slot 2 Standard ID0 (C1MSL2SID0)	<Address: H'0080 1520>
CAN1 Message Slot 3 Standard ID0 (C1MSL3SID0)	<Address: H'0080 1530>
CAN1 Message Slot 4 Standard ID0 (C1MSL4SID0)	<Address: H'0080 1540>
CAN1 Message Slot 5 Standard ID0 (C1MSL5SID0)	<Address: H'0080 1550>
CAN1 Message Slot 6 Standard ID0 (C1MSL6SID0)	<Address: H'0080 1560>
CAN1 Message Slot 7 Standard ID0 (C1MSL7SID0)	<Address: H'0080 1570>
CAN1 Message Slot 8 Standard ID0 (C1MSL8SID0)	<Address: H'0080 1580>
CAN1 Message Slot 9 Standard ID0 (C1MSL9SID0)	<Address: H'0080 1590>
CAN1 Message Slot 10 Standard ID0 (C1MSL10SID0)	<Address: H'0080 15A0>
CAN1 Message Slot 11 Standard ID0 (C1MSL11SID0)	<Address: H'0080 15B0>
CAN1 Message Slot 12 Standard ID0 (C1MSL12SID0)	<Address: H'0080 15C0>
CAN1 Message Slot 13 Standard ID0 (C1MSL13SID0)	<Address: H'0080 15D0>
CAN1 Message Slot 14 Standard ID0 (C1MSL14SID0)	<Address: H'0080 15E0>
CAN1 Message Slot 15 Standard ID0 (C1MSL15SID0)	<Address: H'0080 15F0>

b0	1	2	3	4	5	6	b7
?	?	?	SID0 ?	SID1 ?	SID2 ?	SID3 ?	SID4 ?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–2	No function assigned. Fix to "0".		0	0
3–7	SID0–SID4 (Standard ID0–standard ID4)	Standard ID0–standard ID4	R	W

These registers are the memory space for transmit and receive frames.

CAN0 Message Slot 0 Standard ID1 (C0MSL0SID1)	<Address: H'0080 1101>
CAN0 Message Slot 1 Standard ID1 (C0MSL1SID1)	<Address: H'0080 1111>
CAN0 Message Slot 2 Standard ID1 (C0MSL2SID1)	<Address: H'0080 1121>
CAN0 Message Slot 3 Standard ID1 (C0MSL3SID1)	<Address: H'0080 1131>
CAN0 Message Slot 4 Standard ID1 (C0MSL4SID1)	<Address: H'0080 1141>
CAN0 Message Slot 5 Standard ID1 (C0MSL5SID1)	<Address: H'0080 1151>
CAN0 Message Slot 6 Standard ID1 (C0MSL6SID1)	<Address: H'0080 1161>
CAN0 Message Slot 7 Standard ID1 (C0MSL7SID1)	<Address: H'0080 1171>
CAN0 Message Slot 8 Standard ID1 (C0MSL8SID1)	<Address: H'0080 1181>
CAN0 Message Slot 9 Standard ID1 (C0MSL9SID1)	<Address: H'0080 1191>
CAN0 Message Slot 10 Standard ID1 (C0MSL10SID1)	<Address: H'0080 11A1>
CAN0 Message Slot 11 Standard ID1 (C0MSL11SID1)	<Address: H'0080 11B1>
CAN0 Message Slot 12 Standard ID1 (C0MSL12SID1)	<Address: H'0080 11C1>
CAN0 Message Slot 13 Standard ID1 (C0MSL13SID1)	<Address: H'0080 11D1>
CAN0 Message Slot 14 Standard ID1 (C0MSL14SID1)	<Address: H'0080 11E1>
CAN0 Message Slot 15 Standard ID1 (C0MSL15SID1)	<Address: H'0080 11F1>

CAN1 Message Slot 0 Standard ID1 (C1MSL0SID1)	<Address: H'0080 1501>
CAN1 Message Slot 1 Standard ID1 (C1MSL1SID1)	<Address: H'0080 1511>
CAN1 Message Slot 2 Standard ID1 (C1MSL2SID1)	<Address: H'0080 1521>
CAN1 Message Slot 3 Standard ID1 (C1MSL3SID1)	<Address: H'0080 1531>
CAN1 Message Slot 4 Standard ID1 (C1MSL4SID1)	<Address: H'0080 1541>
CAN1 Message Slot 5 Standard ID1 (C1MSL5SID1)	<Address: H'0080 1551>
CAN1 Message Slot 6 Standard ID1 (C1MSL6SID1)	<Address: H'0080 1561>
CAN1 Message Slot 7 Standard ID1 (C1MSL7SID1)	<Address: H'0080 1571>
CAN1 Message Slot 8 Standard ID1 (C1MSL8SID1)	<Address: H'0080 1581>
CAN1 Message Slot 9 Standard ID1 (C1MSL9SID1)	<Address: H'0080 1591>
CAN1 Message Slot 10 Standard ID1 (C1MSL10SID1)	<Address: H'0080 15A1>
CAN1 Message Slot 11 Standard ID1 (C1MSL11SID1)	<Address: H'0080 15B1>
CAN1 Message Slot 12 Standard ID1 (C1MSL12SID1)	<Address: H'0080 15C1>
CAN1 Message Slot 13 Standard ID1 (C1MSL13SID1)	<Address: H'0080 15D1>
CAN1 Message Slot 14 Standard ID1 (C1MSL14SID1)	<Address: H'0080 15E1>
CAN1 Message Slot 15 Standard ID1 (C1MSL15SID1)	<Address: H'0080 15F1>

b8	9	10	11	12	13	14	b15
?	?	SID5 ?	SID6 ?	SID7 ?	SID8 ?	SID9 ?	SID10 ?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8, 9	No function assigned. Fix to "0".		0	0
10–15	SID5–SID10 (Standard ID5–standard ID10)	Standard ID5–standard ID10	R	W

These registers are the memory space for transmit and receive frames.

CAN0 Message Slot 0 Extended ID0 (C0MSL0EID0)	<Address: H'0080 1102>
CAN0 Message Slot 1 Extended ID0 (C0MSL1EID0)	<Address: H'0080 1112>
CAN0 Message Slot 2 Extended ID0 (C0MSL2EID0)	<Address: H'0080 1122>
CAN0 Message Slot 3 Extended ID0 (C0MSL3EID0)	<Address: H'0080 1132>
CAN0 Message Slot 4 Extended ID0 (C0MSL4EID0)	<Address: H'0080 1142>
CAN0 Message Slot 5 Extended ID0 (C0MSL5EID0)	<Address: H'0080 1152>
CAN0 Message Slot 6 Extended ID0 (C0MSL6EID0)	<Address: H'0080 1162>
CAN0 Message Slot 7 Extended ID0 (C0MSL7EID0)	<Address: H'0080 1172>
CAN0 Message Slot 8 Extended ID0 (C0MSL8EID0)	<Address: H'0080 1182>
CAN0 Message Slot 9 Extended ID0 (C0MSL9EID0)	<Address: H'0080 1192>
CAN0 Message Slot 10 Extended ID0 (C0MSL10EID0)	<Address: H'0080 11A2>
CAN0 Message Slot 11 Extended ID0 (C0MSL11EID0)	<Address: H'0080 11B2>
CAN0 Message Slot 12 Extended ID0 (C0MSL12EID0)	<Address: H'0080 11C2>
CAN0 Message Slot 13 Extended ID0 (C0MSL13EID0)	<Address: H'0080 11D2>
CAN0 Message Slot 14 Extended ID0 (C0MSL14EID0)	<Address: H'0080 11E2>
CAN0 Message Slot 15 Extended ID0 (C0MSL15EID0)	<Address: H'0080 11F2>

CAN1 Message Slot 0 Extended ID0 (C1MSL0EID0)	<Address: H'0080 1502>
CAN1 Message Slot 1 Extended ID0 (C1MSL1EID0)	<Address: H'0080 1512>
CAN1 Message Slot 2 Extended ID0 (C1MSL2EID0)	<Address: H'0080 1522>
CAN1 Message Slot 3 Extended ID0 (C1MSL3EID0)	<Address: H'0080 1532>
CAN1 Message Slot 4 Extended ID0 (C1MSL4EID0)	<Address: H'0080 1542>
CAN1 Message Slot 5 Extended ID0 (C1MSL5EID0)	<Address: H'0080 1552>
CAN1 Message Slot 6 Extended ID0 (C1MSL6EID0)	<Address: H'0080 1562>
CAN1 Message Slot 7 Extended ID0 (C1MSL7EID0)	<Address: H'0080 1572>
CAN1 Message Slot 8 Extended ID0 (C1MSL8EID0)	<Address: H'0080 1582>
CAN1 Message Slot 9 Extended ID0 (C1MSL9EID0)	<Address: H'0080 1592>
CAN1 Message Slot 10 Extended ID0 (C1MSL10EID0)	<Address: H'0080 15A2>
CAN1 Message Slot 11 Extended ID0 (C1MSL11EID0)	<Address: H'0080 15B2>
CAN1 Message Slot 12 Extended ID0 (C1MSL12EID0)	<Address: H'0080 15C2>
CAN1 Message Slot 13 Extended ID0 (C1MSL13EID0)	<Address: H'0080 15D2>
CAN1 Message Slot 14 Extended ID0 (C1MSL14EID0)	<Address: H'0080 15E2>
CAN1 Message Slot 15 Extended ID0 (C1MSL15EID0)	<Address: H'0080 15F2>

b0	1	2	3	4	5	6	b7
?	?	?	?	EID0 ?	EID1 ?	EID2 ?	EID3 ?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–3	No function assigned. Fix to "0".		0	0
4–7	EID0–EID3 (Extended ID0–extended ID3)	Extended ID0–extended ID3	R	W

These registers are the memory space for transmit and receive frames.

Note: • If the message slot is set for the receive slot standard ID format, an undefined value is written to the EID bits when storing received data.

CAN0 Message Slot 0 Extended ID1 (C0MSL0EID1)	<Address: H'0080 1103>
CAN0 Message Slot 1 Extended ID1 (C0MSL1EID1)	<Address: H'0080 1113>
CAN0 Message Slot 2 Extended ID1 (C0MSL2EID1)	<Address: H'0080 1123>
CAN0 Message Slot 3 Extended ID1 (C0MSL3EID1)	<Address: H'0080 1133>
CAN0 Message Slot 4 Extended ID1 (C0MSL4EID1)	<Address: H'0080 1143>
CAN0 Message Slot 5 Extended ID1 (C0MSL5EID1)	<Address: H'0080 1153>
CAN0 Message Slot 6 Extended ID1 (C0MSL6EID1)	<Address: H'0080 1163>
CAN0 Message Slot 7 Extended ID1 (C0MSL7EID1)	<Address: H'0080 1173>
CAN0 Message Slot 8 Extended ID1 (C0MSL8EID1)	<Address: H'0080 1183>
CAN0 Message Slot 9 Extended ID1 (C0MSL9EID1)	<Address: H'0080 1193>
CAN0 Message Slot 10 Extended ID1 (C0MSL10EID1)	<Address: H'0080 11A3>
CAN0 Message Slot 11 Extended ID1 (C0MSL11EID1)	<Address: H'0080 11B3>
CAN0 Message Slot 12 Extended ID1 (C0MSL12EID1)	<Address: H'0080 11C3>
CAN0 Message Slot 13 Extended ID1 (C0MSL13EID1)	<Address: H'0080 11D3>
CAN0 Message Slot 14 Extended ID1 (C0MSL14EID1)	<Address: H'0080 11E3>
CAN0 Message Slot 15 Extended ID1 (C0MSL15EID1)	<Address: H'0080 11F3>

CAN1 Message Slot 0 Extended ID1 (C1MSL0EID1)	<Address: H'0080 1503>
CAN1 Message Slot 1 Extended ID1 (C1MSL1EID1)	<Address: H'0080 1513>
CAN1 Message Slot 2 Extended ID1 (C1MSL2EID1)	<Address: H'0080 1523>
CAN1 Message Slot 3 Extended ID1 (C1MSL3EID1)	<Address: H'0080 1533>
CAN1 Message Slot 4 Extended ID1 (C1MSL4EID1)	<Address: H'0080 1543>
CAN1 Message Slot 5 Extended ID1 (C1MSL5EID1)	<Address: H'0080 1553>
CAN1 Message Slot 6 Extended ID1 (C1MSL6EID1)	<Address: H'0080 1563>
CAN1 Message Slot 7 Extended ID1 (C1MSL7EID1)	<Address: H'0080 1573>
CAN1 Message Slot 8 Extended ID1 (C1MSL8EID1)	<Address: H'0080 1583>
CAN1 Message Slot 9 Extended ID1 (C1MSL9EID1)	<Address: H'0080 1593>
CAN1 Message Slot 10 Extended ID1 (C1MSL10EID1)	<Address: H'0080 15A3>
CAN1 Message Slot 11 Extended ID1 (C1MSL11EID1)	<Address: H'0080 15B3>
CAN1 Message Slot 12 Extended ID1 (C1MSL12EID1)	<Address: H'0080 15C3>
CAN1 Message Slot 13 Extended ID1 (C1MSL13EID1)	<Address: H'0080 15D3>
CAN1 Message Slot 14 Extended ID1 (C1MSL14EID1)	<Address: H'0080 15E3>
CAN1 Message Slot 15 Extended ID1 (C1MSL15EID1)	<Address: H'0080 15F3>

b8	9	10	11	12	13	14	b15
EID4	EID5	EID6	EID7	EID8	EID9	EID10	EID11
?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–15	EID4–EID11 (Extended ID4–extended ID11)	Extended ID4–extended ID11	R	W

These registers are the memory space for transmit and receive frames.

Note: • If the message slot is set for the receive slot standard ID format, an undefined value is written to the EID bits when storing received data.

CAN0 Message Slot 0 Extended ID2 (C0MSL0EID2)	<Address: H'0080 1104>
CAN0 Message Slot 1 Extended ID2 (C0MSL1EID2)	<Address: H'0080 1114>
CAN0 Message Slot 2 Extended ID2 (C0MSL2EID2)	<Address: H'0080 1124>
CAN0 Message Slot 3 Extended ID2 (C0MSL3EID2)	<Address: H'0080 1134>
CAN0 Message Slot 4 Extended ID2 (C0MSL4EID2)	<Address: H'0080 1144>
CAN0 Message Slot 5 Extended ID2 (C0MSL5EID2)	<Address: H'0080 1154>
CAN0 Message Slot 6 Extended ID2 (C0MSL6EID2)	<Address: H'0080 1164>
CAN0 Message Slot 7 Extended ID2 (C0MSL7EID2)	<Address: H'0080 1174>
CAN0 Message Slot 8 Extended ID2 (C0MSL8EID2)	<Address: H'0080 1184>
CAN0 Message Slot 9 Extended ID2 (C0MSL9EID2)	<Address: H'0080 1194>
CAN0 Message Slot 10 Extended ID2 (C0MSL10EID2)	<Address: H'0080 11A4>
CAN0 Message Slot 11 Extended ID2 (C0MSL11EID2)	<Address: H'0080 11B4>
CAN0 Message Slot 12 Extended ID2 (C0MSL12EID2)	<Address: H'0080 11C4>
CAN0 Message Slot 13 Extended ID2 (C0MSL13EID2)	<Address: H'0080 11D4>
CAN0 Message Slot 14 Extended ID2 (C0MSL14EID2)	<Address: H'0080 11E4>
CAN0 Message Slot 15 Extended ID2 (C0MSL15EID2)	<Address: H'0080 11F4>

CAN1 Message Slot 0 Extended ID2 (C1MSL0EID2)	<Address: H'0080 1504>
CAN1 Message Slot 1 Extended ID2 (C1MSL1EID2)	<Address: H'0080 1514>
CAN1 Message Slot 2 Extended ID2 (C1MSL2EID2)	<Address: H'0080 1524>
CAN1 Message Slot 3 Extended ID2 (C1MSL3EID2)	<Address: H'0080 1534>
CAN1 Message Slot 4 Extended ID2 (C1MSL4EID2)	<Address: H'0080 1544>
CAN1 Message Slot 5 Extended ID2 (C1MSL5EID2)	<Address: H'0080 1554>
CAN1 Message Slot 6 Extended ID2 (C1MSL6EID2)	<Address: H'0080 1564>
CAN1 Message Slot 7 Extended ID2 (C1MSL7EID2)	<Address: H'0080 1574>
CAN1 Message Slot 8 Extended ID2 (C1MSL8EID2)	<Address: H'0080 1584>
CAN1 Message Slot 9 Extended ID2 (C1MSL9EID2)	<Address: H'0080 1594>
CAN1 Message Slot 10 Extended ID2 (C1MSL10EID2)	<Address: H'0080 15A4>
CAN1 Message Slot 11 Extended ID2 (C1MSL11EID2)	<Address: H'0080 15B4>
CAN1 Message Slot 12 Extended ID2 (C1MSL12EID2)	<Address: H'0080 15C4>
CAN1 Message Slot 13 Extended ID2 (C1MSL13EID2)	<Address: H'0080 15D4>
CAN1 Message Slot 14 Extended ID2 (C1MSL14EID2)	<Address: H'0080 15E4>
CAN1 Message Slot 15 Extended ID2 (C1MSL15EID2)	<Address: H'0080 15F4>

b0	1	2	3	4	5	6	b7
?	?	EID12 ?	EID13 ?	EID14 ?	EID15 ?	EID16 ?	EID17 ?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0, 1	No function assigned. Fix to "0".		0	0
2–7	EID12–EID17 (Extended ID12–extended ID17)	Extended ID12–extended ID17	R	W

These registers are the memory space for transmit and receive frames.

Note: • If the message slot is set for the receive slot standard ID format, an undefined value is written to the EID bits when storing received data.

CAN0 Message Slot 0 Data Length Register (C0MSL0DLC)	<Address: H'0080 1105>
CAN0 Message Slot 1 Data Length Register (C0MSL1DLC)	<Address: H'0080 1115>
CAN0 Message Slot 2 Data Length Register (C0MSL2DLC)	<Address: H'0080 1125>
CAN0 Message Slot 3 Data Length Register (C0MSL3DLC)	<Address: H'0080 1135>
CAN0 Message Slot 4 Data Length Register (C0MSL4DLC)	<Address: H'0080 1145>
CAN0 Message Slot 5 Data Length Register (C0MSL5DLC)	<Address: H'0080 1155>
CAN0 Message Slot 6 Data Length Register (C0MSL6DLC)	<Address: H'0080 1165>
CAN0 Message Slot 7 Data Length Register (C0MSL7DLC)	<Address: H'0080 1175>
CAN0 Message Slot 8 Data Length Register (C0MSL8DLC)	<Address: H'0080 1185>
CAN0 Message Slot 9 Data Length Register (C0MSL9DLC)	<Address: H'0080 1195>
CAN0 Message Slot 10 Data Length Register (C0MSL10DLC)	<Address: H'0080 11A5>
CAN0 Message Slot 11 Data Length Register (C0MSL11DLC)	<Address: H'0080 11B5>
CAN0 Message Slot 12 Data Length Register (C0MSL12DLC)	<Address: H'0080 11C5>
CAN0 Message Slot 13 Data Length Register (C0MSL13DLC)	<Address: H'0080 11D5>
CAN0 Message Slot 14 Data Length Register (C0MSL14DLC)	<Address: H'0080 11E5>
CAN0 Message Slot 15 Data Length Register (C0MSL15DLC)	<Address: H'0080 11F5>

CAN1 Message Slot 0 Data Length Register (C1MSL0DLC)	<Address: H'0080 1505>
CAN1 Message Slot 1 Data Length Register (C1MSL1DLC)	<Address: H'0080 1515>
CAN1 Message Slot 2 Data Length Register (C1MSL2DLC)	<Address: H'0080 1525>
CAN1 Message Slot 3 Data Length Register (C1MSL3DLC)	<Address: H'0080 1535>
CAN1 Message Slot 4 Data Length Register (C1MSL4DLC)	<Address: H'0080 1545>
CAN1 Message Slot 5 Data Length Register (C1MSL5DLC)	<Address: H'0080 1555>
CAN1 Message Slot 6 Data Length Register (C1MSL6DLC)	<Address: H'0080 1565>
CAN1 Message Slot 7 Data Length Register (C1MSL7DLC)	<Address: H'0080 1575>
CAN1 Message Slot 8 Data Length Register (C1MSL8DLC)	<Address: H'0080 1585>
CAN1 Message Slot 9 Data Length Register (C1MSL9DLC)	<Address: H'0080 1595>
CAN1 Message Slot 10 Data Length Register (C1MSL10DLC)	<Address: H'0080 15A5>
CAN1 Message Slot 11 Data Length Register (C1MSL11DLC)	<Address: H'0080 15B5>
CAN1 Message Slot 12 Data Length Register (C1MSL12DLC)	<Address: H'0080 15C5>
CAN1 Message Slot 13 Data Length Register (C1MSL13DLC)	<Address: H'0080 15D5>
CAN1 Message Slot 14 Data Length Register (C1MSL14DLC)	<Address: H'0080 15E5>
CAN1 Message Slot 15 Data Length Register (C1MSL15DLC)	<Address: H'0080 15F5>

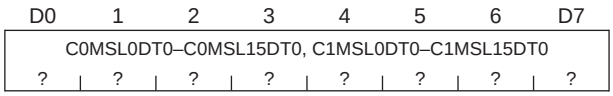
b8	9	10	11	12	13	14	b15
?	?	?	?	DLC0 ?	DLC1 ?	DLC2 ?	DLC3 ?

<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–11	No function assigned. Fix to "0".		0	0
12–15	DLC0–DLC3 Data length setting bit	0000: 0 bytes 0001: 1 bytes 0010: 2 bytes 0011: 3 bytes 0100: 4 bytes 0101: 5 bytes 0110: 6 bytes 0111: 7 bytes 1000: 8 bytes | | 1111: 8 bytes	R	W

These registers are the memory space for transmit and receive frames. When sending, the register is used to set the transmit data length. When receiving, the register is used to store the receive frame DLC.

CAN0 Message Slot 0 Data 0 (C0MSL0DT0)	<Address: H'0080 1106>
CAN0 Message Slot 1 Data 0 (C0MSL1DT0)	<Address: H'0080 1116>
CAN0 Message Slot 2 Data 0 (C0MSL2DT0)	<Address: H'0080 1126>
CAN0 Message Slot 3 Data 0 (C0MSL3DT0)	<Address: H'0080 1136>
CAN0 Message Slot 4 Data 0 (C0MSL4DT0)	<Address: H'0080 1146>
CAN0 Message Slot 5 Data 0 (C0MSL5DT0)	<Address: H'0080 1156>
CAN0 Message Slot 6 Data 0 (C0MSL6DT0)	<Address: H'0080 1166>
CAN0 Message Slot 7 Data 0 (C0MSL7DT0)	<Address: H'0080 1176>
CAN0 Message Slot 8 Data 0 (C0MSL8DT0)	<Address: H'0080 1186>
CAN0 Message Slot 9 Data 0 (C0MSL9DT0)	<Address: H'0080 1196>
CAN0 Message Slot 10 Data 0 (C0MSL10DT0)	<Address: H'0080 11A6>
CAN0 Message Slot 11 Data 0 (C0MSL11DT0)	<Address: H'0080 11B6>
CAN0 Message Slot 12 Data 0 (C0MSL12DT0)	<Address: H'0080 11C6>
CAN0 Message Slot 13 Data 0 (C0MSL13DT0)	<Address: H'0080 11D6>
CAN0 Message Slot 14 Data 0 (C0MSL14DT0)	<Address: H'0080 11E6>
CAN0 Message Slot 15 Data 0 (C0MSL15DT0)	<Address: H'0080 11F6>
CAN1 Message Slot 0 Data 0 (C1MSL0DT0)	<Address: H'0080 1506>
CAN1 Message Slot 1 Data 0 (C1MSL1DT0)	<Address: H'0080 1516>
CAN1 Message Slot 2 Data 0 (C1MSL2DT0)	<Address: H'0080 1526>
CAN1 Message Slot 3 Data 0 (C1MSL3DT0)	<Address: H'0080 1536>
CAN1 Message Slot 4 Data 0 (C1MSL4DT0)	<Address: H'0080 1546>
CAN1 Message Slot 5 Data 0 (C1MSL5DT0)	<Address: H'0080 1556>
CAN1 Message Slot 6 Data 0 (C1MSL6DT0)	<Address: H'0080 1566>
CAN1 Message Slot 7 Data 0 (C1MSL7DT0)	<Address: H'0080 1576>
CAN1 Message Slot 8 Data 0 (C1MSL8DT0)	<Address: H'0080 1586>
CAN1 Message Slot 9 Data 0 (C1MSL9DT0)	<Address: H'0080 1596>
CAN1 Message Slot 10 Data 0 (C1MSL10DT0)	<Address: H'0080 15A6>
CAN1 Message Slot 11 Data 0 (C1MSL11DT0)	<Address: H'0080 15B6>
CAN1 Message Slot 12 Data 0 (C1MSL12DT0)	<Address: H'0080 15C6>
CAN1 Message Slot 13 Data 0 (C1MSL13DT0)	<Address: H'0080 15D6>
CAN1 Message Slot 14 Data 0 (C1MSL14DT0)	<Address: H'0080 15E6>
CAN1 Message Slot 15 Data 0 (C1MSL15DT0)	<Address: H'0080 15F6>



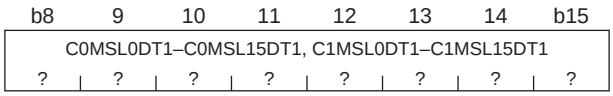
<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–7	C0MSL0DT0–C0MSL15DT0, C1MSL0DT0–C1MSL15DT0	Message slot data 0	R	W

These registers are the memory space for transmit and receive frames.

Notes: • During a receive slot, an undefined value is written to the register if the data length of the data frame being stored (DLC value) = "0".
• The first byte of the CAN frame data field corresponds to message slot n data 0. Data is transmitted or received beginning with the MSB side of the register.

CAN0 Message Slot 0 Data 1 (C0MSL0DT1)	<Address: H'0080 1107>
CAN0 Message Slot 1 Data 1 (C0MSL1DT1)	<Address: H'0080 1117>
CAN0 Message Slot 2 Data 1 (C0MSL2DT1)	<Address: H'0080 1127>
CAN0 Message Slot 3 Data 1 (C0MSL3DT1)	<Address: H'0080 1137>
CAN0 Message Slot 4 Data 1 (C0MSL4DT1)	<Address: H'0080 1147>
CAN0 Message Slot 5 Data 1 (C0MSL5DT1)	<Address: H'0080 1157>
CAN0 Message Slot 6 Data 1 (C0MSL6DT1)	<Address: H'0080 1167>
CAN0 Message Slot 7 Data 1 (C0MSL7DT1)	<Address: H'0080 1177>
CAN0 Message Slot 8 Data 1 (C0MSL8DT1)	<Address: H'0080 1187>
CAN0 Message Slot 9 Data 1 (C0MSL9DT1)	<Address: H'0080 1197>
CAN0 Message Slot 10 Data 1 (C0MSL10DT1)	<Address: H'0080 11A7>
CAN0 Message Slot 11 Data 1 (C0MSL11DT1)	<Address: H'0080 11B7>
CAN0 Message Slot 12 Data 1 (C0MSL12DT1)	<Address: H'0080 11C7>
CAN0 Message Slot 13 Data 1 (C0MSL13DT1)	<Address: H'0080 11D7>
CAN0 Message Slot 14 Data 1 (C0MSL14DT1)	<Address: H'0080 11E7>
CAN0 Message Slot 15 Data 1 (C0MSL15DT1)	<Address: H'0080 11F7>
CAN1 Message Slot 0 Data 1 (C1MSL0DT1)	<Address: H'0080 1507>
CAN1 Message Slot 1 Data 1 (C1MSL1DT1)	<Address: H'0080 1517>
CAN1 Message Slot 2 Data 1 (C1MSL2DT1)	<Address: H'0080 1527>
CAN1 Message Slot 3 Data 1 (C1MSL3DT1)	<Address: H'0080 1537>
CAN1 Message Slot 4 Data 1 (C1MSL4DT1)	<Address: H'0080 1547>
CAN1 Message Slot 5 Data 1 (C1MSL5DT1)	<Address: H'0080 1557>
CAN1 Message Slot 6 Data 1 (C1MSL6DT1)	<Address: H'0080 1567>
CAN1 Message Slot 7 Data 1 (C1MSL7DT1)	<Address: H'0080 1577>
CAN1 Message Slot 8 Data 1 (C1MSL8DT1)	<Address: H'0080 1587>
CAN1 Message Slot 9 Data 1 (C1MSL9DT1)	<Address: H'0080 1597>
CAN1 Message Slot 10 Data 1 (C1MSL10DT1)	<Address: H'0080 15A7>
CAN1 Message Slot 11 Data 1 (C1MSL11DT1)	<Address: H'0080 15B7>
CAN1 Message Slot 12 Data 1 (C1MSL12DT1)	<Address: H'0080 15C7>
CAN1 Message Slot 13 Data 1 (C1MSL13DT1)	<Address: H'0080 15D7>
CAN1 Message Slot 14 Data 1 (C1MSL14DT1)	<Address: H'0080 15E7>
CAN1 Message Slot 15 Data 1 (C1MSL15DT1)	<Address: H'0080 15F7>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–15	C0MSL0DT1–C0MSL15DT1, C1MSL0DT1–C1MSL15DT1	Message slot data 1	R	W

These registers are the memory space for transmit and receive frames.

Note: • During a receive slot, an undefined value is written to the register if the data length of the data frame being stored (DLC value) is equal to or less than 1.

CAN0 Message Slot 0 Data 2 (C0MSL0DT2)	<Address: H'0080 1108>
CAN0 Message Slot 1 Data 2 (C0MSL1DT2)	<Address: H'0080 1118>
CAN0 Message Slot 2 Data 2 (C0MSL2DT2)	<Address: H'0080 1128>
CAN0 Message Slot 3 Data 2 (C0MSL3DT2)	<Address: H'0080 1138>
CAN0 Message Slot 4 Data 2 (C0MSL4DT2)	<Address: H'0080 1148>
CAN0 Message Slot 5 Data 2 (C0MSL5DT2)	<Address: H'0080 1158>
CAN0 Message Slot 6 Data 2 (C0MSL6DT2)	<Address: H'0080 1168>
CAN0 Message Slot 7 Data 2 (C0MSL7DT2)	<Address: H'0080 1178>
CAN0 Message Slot 8 Data 2 (C0MSL8DT2)	<Address: H'0080 1188>
CAN0 Message Slot 9 Data 2 (C0MSL9DT2)	<Address: H'0080 1198>
CAN0 Message Slot 10 Data 2 (C0MSL10DT2)	<Address: H'0080 11A8>
CAN0 Message Slot 11 Data 2 (C0MSL11DT2)	<Address: H'0080 11B8>
CAN0 Message Slot 12 Data 2 (C0MSL12DT2)	<Address: H'0080 11C8>
CAN0 Message Slot 13 Data 2 (C0MSL13DT2)	<Address: H'0080 11D8>
CAN0 Message Slot 14 Data 2 (C0MSL14DT2)	<Address: H'0080 11E8>
CAN0 Message Slot 15 Data 2 (C0MSL15DT2)	<Address: H'0080 11F8>
CAN1 Message Slot 0 Data 2 (C1MSL0DT2)	<Address: H'0080 1508>
CAN1 Message Slot 1 Data 2 (C1MSL1DT2)	<Address: H'0080 1518>
CAN1 Message Slot 2 Data 2 (C1MSL2DT2)	<Address: H'0080 1528>
CAN1 Message Slot 3 Data 2 (C1MSL3DT2)	<Address: H'0080 1538>
CAN1 Message Slot 4 Data 2 (C1MSL4DT2)	<Address: H'0080 1548>
CAN1 Message Slot 5 Data 2 (C1MSL5DT2)	<Address: H'0080 1558>
CAN1 Message Slot 6 Data 2 (C1MSL6DT2)	<Address: H'0080 1568>
CAN1 Message Slot 7 Data 2 (C1MSL7DT2)	<Address: H'0080 1578>
CAN1 Message Slot 8 Data 2 (C1MSL8DT2)	<Address: H'0080 1588>
CAN1 Message Slot 9 Data 2 (C1MSL9DT2)	<Address: H'0080 1598>
CAN1 Message Slot 10 Data 2 (C1MSL10DT2)	<Address: H'0080 15A8>
CAN1 Message Slot 11 Data 2 (C1MSL11DT2)	<Address: H'0080 15B8>
CAN1 Message Slot 12 Data 2 (C1MSL12DT2)	<Address: H'0080 15C8>
CAN1 Message Slot 13 Data 2 (C1MSL13DT2)	<Address: H'0080 15D8>
CAN1 Message Slot 14 Data 2 (C1MSL14DT2)	<Address: H'0080 15E8>
CAN1 Message Slot 15 Data 2 (C1MSL15DT2)	<Address: H'0080 15F8>

b0	1	2	3	4	5	6	b7
C0MSL0DT2–C0MSL15DT2, C1MSL0DT2–C1MSL15DT2							
?	?	?	?	?	?	?	?

<Upon exiting reset: Undefined>

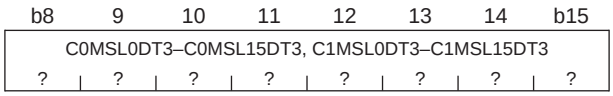
b	Bit Name	Function	R	W
0–7	C0MSL0DT2–C0MSL15DT2, C1MSL0DT2–C1MSL15DT2	Message slot data 2	R	W

These registers are the memory space for transmit and receive frames.

Note: • During a receive slot, an undefined value is written to the register if the data length of the data frame being stored (DLC value) is equal to or less than 2.

CAN0 Message Slot 0 Data 3 (C0MSL0DT3)	<Address: H'0080 1109>
CAN0 Message Slot 1 Data 3 (C0MSL1DT3)	<Address: H'0080 1119>
CAN0 Message Slot 2 Data 3 (C0MSL2DT3)	<Address: H'0080 1129>
CAN0 Message Slot 3 Data 3 (C0MSL3DT3)	<Address: H'0080 1139>
CAN0 Message Slot 4 Data 3 (C0MSL4DT3)	<Address: H'0080 1149>
CAN0 Message Slot 5 Data 3 (C0MSL5DT3)	<Address: H'0080 1159>
CAN0 Message Slot 6 Data 3 (C0MSL6DT3)	<Address: H'0080 1169>
CAN0 Message Slot 7 Data 3 (C0MSL7DT3)	<Address: H'0080 1179>
CAN0 Message Slot 8 Data 3 (C0MSL8DT3)	<Address: H'0080 1189>
CAN0 Message Slot 9 Data 3 (C0MSL9DT3)	<Address: H'0080 1199>
CAN0 Message Slot 10 Data 3 (C0MSL10DT3)	<Address: H'0080 11A9>
CAN0 Message Slot 11 Data 3 (C0MSL11DT3)	<Address: H'0080 11B9>
CAN0 Message Slot 12 Data 3 (C0MSL12DT3)	<Address: H'0080 11C9>
CAN0 Message Slot 13 Data 3 (C0MSL13DT3)	<Address: H'0080 11D9>
CAN0 Message Slot 14 Data 3 (C0MSL14DT3)	<Address: H'0080 11E9>
CAN0 Message Slot 15 Data 3 (C0MSL15DT3)	<Address: H'0080 11F9>

CAN1 Message Slot 0 Data 3 (C1MSL0DT3)	<Address: H'0080 1509>
CAN1 Message Slot 1 Data 3 (C1MSL1DT3)	<Address: H'0080 1519>
CAN1 Message Slot 2 Data 3 (C1MSL2DT3)	<Address: H'0080 1529>
CAN1 Message Slot 3 Data 3 (C1MSL3DT3)	<Address: H'0080 1539>
CAN1 Message Slot 4 Data 3 (C1MSL4DT3)	<Address: H'0080 1549>
CAN1 Message Slot 5 Data 3 (C1MSL5DT3)	<Address: H'0080 1559>
CAN1 Message Slot 6 Data 3 (C1MSL6DT3)	<Address: H'0080 1569>
CAN1 Message Slot 7 Data 3 (C1MSL7DT3)	<Address: H'0080 1579>
CAN1 Message Slot 8 Data 3 (C1MSL8DT3)	<Address: H'0080 1589>
CAN1 Message Slot 9 Data 3 (C1MSL9DT3)	<Address: H'0080 1599>
CAN1 Message Slot 10 Data 3 (C1MSL10DT3)	<Address: H'0080 15A9>
CAN1 Message Slot 11 Data 3 (C1MSL11DT3)	<Address: H'0080 15B9>
CAN1 Message Slot 12 Data 3 (C1MSL12DT3)	<Address: H'0080 15C9>
CAN1 Message Slot 13 Data 3 (C1MSL13DT3)	<Address: H'0080 15D9>
CAN1 Message Slot 14 Data 3 (C1MSL14DT3)	<Address: H'0080 15E9>
CAN1 Message Slot 15 Data 3 (C1MSL15DT3)	<Address: H'0080 15F9>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–15	C0MSL0DT3–C0MSL15DT3, C1MSL0DT3–C1MSL15DT3	Message slot data 3	R	W

These registers are the memory space for transmit and receive frames.

Note: • During a receive slot, an undefined value is written to the register if the data length of the data frame being stored (DLC value) is equal to or less than 3.

CAN0 Message Slot 0 Data 4 (C0MSL0DT4)	<Address: H'0080 110A>
CAN0 Message Slot 1 Data 4 (C0MSL1DT4)	<Address: H'0080 111A>
CAN0 Message Slot 2 Data 4 (C0MSL2DT4)	<Address: H'0080 112A>
CAN0 Message Slot 3 Data 4 (C0MSL3DT4)	<Address: H'0080 113A>
CAN0 Message Slot 4 Data 4 (C0MSL4DT4)	<Address: H'0080 114A>
CAN0 Message Slot 5 Data 4 (C0MSL5DT4)	<Address: H'0080 115A>
CAN0 Message Slot 6 Data 4 (C0MSL6DT4)	<Address: H'0080 116A>
CAN0 Message Slot 7 Data 4 (C0MSL7DT4)	<Address: H'0080 117A>
CAN0 Message Slot 8 Data 4 (C0MSL8DT4)	<Address: H'0080 118A>
CAN0 Message Slot 9 Data 4 (C0MSL9DT4)	<Address: H'0080 119A>
CAN0 Message Slot 10 Data 4 (C0MSL10DT4)	<Address: H'0080 11AA>
CAN0 Message Slot 11 Data 4 (C0MSL11DT4)	<Address: H'0080 11BA>
CAN0 Message Slot 12 Data 4 (C0MSL12DT4)	<Address: H'0080 11CA>
CAN0 Message Slot 13 Data 4 (C0MSL13DT4)	<Address: H'0080 11DA>
CAN0 Message Slot 14 Data 4 (C0MSL14DT4)	<Address: H'0080 11EA>
CAN0 Message Slot 15 Data 4 (C0MSL15DT4)	<Address: H'0080 11FA>
CAN1 Message Slot 0 Data 4 (C1MSL0DT4)	<Address: H'0080 150A>
CAN1 Message Slot 1 Data 4 (C1MSL1DT4)	<Address: H'0080 151A>
CAN1 Message Slot 2 Data 4 (C1MSL2DT4)	<Address: H'0080 152A>
CAN1 Message Slot 3 Data 4 (C1MSL3DT4)	<Address: H'0080 153A>
CAN1 Message Slot 4 Data 4 (C1MSL4DT4)	<Address: H'0080 154A>
CAN1 Message Slot 5 Data 4 (C1MSL5DT4)	<Address: H'0080 155A>
CAN1 Message Slot 6 Data 4 (C1MSL6DT4)	<Address: H'0080 156A>
CAN1 Message Slot 7 Data 4 (C1MSL7DT4)	<Address: H'0080 157A>
CAN1 Message Slot 8 Data 4 (C1MSL8DT4)	<Address: H'0080 158A>
CAN1 Message Slot 9 Data 4 (C1MSL9DT4)	<Address: H'0080 159A>
CAN1 Message Slot 10 Data 4 (C1MSL10DT4)	<Address: H'0080 15AA>
CAN1 Message Slot 11 Data 4 (C1MSL11DT4)	<Address: H'0080 15BA>
CAN1 Message Slot 12 Data 4 (C1MSL12DT4)	<Address: H'0080 15CA>
CAN1 Message Slot 13 Data 4 (C1MSL13DT4)	<Address: H'0080 15DA>
CAN1 Message Slot 14 Data 4 (C1MSL14DT4)	<Address: H'0080 15EA>
CAN1 Message Slot 15 Data 4 (C1MSL15DT4)	<Address: H'0080 15FA>

b0	1	2	3	4	5	6	b7
C0MSL0DT4–C0MSL15DT4, C1MSL0DT4–C1MSL15DT4							
?	?	?	?	?	?	?	?

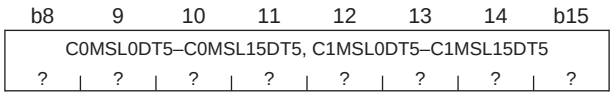
<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–7	C0MSL0DT4–C0MSL15DT4, C1MSL0DT4–C1MSL15DT4	Message slot data 4	R	W

These registers are the memory space for transmit and receive frames.

Note: • During a receive slot, an undefined value is written to the register if the data length of the data frame being stored (DLC value) is equal to or less than 4.

CAN0 Message Slot 0 Data 5 (C0MSL0DT5)	<Address: H'0080 110B>
CAN0 Message Slot 1 Data 5 (C0MSL1DT5)	<Address: H'0080 111B>
CAN0 Message Slot 2 Data 5 (C0MSL2DT5)	<Address: H'0080 112B>
CAN0 Message Slot 3 Data 5 (C0MSL3DT5)	<Address: H'0080 113B>
CAN0 Message Slot 4 Data 5 (C0MSL4DT5)	<Address: H'0080 114B>
CAN0 Message Slot 5 Data 5 (C0MSL5DT5)	<Address: H'0080 115B>
CAN0 Message Slot 6 Data 5 (C0MSL6DT5)	<Address: H'0080 116B>
CAN0 Message Slot 7 Data 5 (C0MSL7DT5)	<Address: H'0080 117B>
CAN0 Message Slot 8 Data 5 (C0MSL8DT5)	<Address: H'0080 118B>
CAN0 Message Slot 9 Data 5 (C0MSL9DT5)	<Address: H'0080 119B>
CAN0 Message Slot 10 Data 5 (C0MSL10DT5)	<Address: H'0080 11AB>
CAN0 Message Slot 11 Data 5 (C0MSL11DT5)	<Address: H'0080 11BB>
CAN0 Message Slot 12 Data 5 (C0MSL12DT5)	<Address: H'0080 11CB>
CAN0 Message Slot 13 Data 5 (C0MSL13DT5)	<Address: H'0080 11DB>
CAN0 Message Slot 14 Data 5 (C0MSL14DT5)	<Address: H'0080 11EB>
CAN0 Message Slot 15 Data 5 (C0MSL15DT5)	<Address: H'0080 11FB>
CAN1 Message Slot 0 Data 5 (C1MSL0DT5)	<Address: H'0080 150B>
CAN1 Message Slot 1 Data 5 (C1MSL1DT5)	<Address: H'0080 151B>
CAN1 Message Slot 2 Data 5 (C1MSL2DT5)	<Address: H'0080 152B>
CAN1 Message Slot 3 Data 5 (C1MSL3DT5)	<Address: H'0080 153B>
CAN1 Message Slot 4 Data 5 (C1MSL4DT5)	<Address: H'0080 154B>
CAN1 Message Slot 5 Data 5 (C1MSL5DT5)	<Address: H'0080 155B>
CAN1 Message Slot 6 Data 5 (C1MSL6DT5)	<Address: H'0080 156B>
CAN1 Message Slot 7 Data 5 (C1MSL7DT5)	<Address: H'0080 157B>
CAN1 Message Slot 8 Data 5 (C1MSL8DT5)	<Address: H'0080 158B>
CAN1 Message Slot 9 Data 5 (C1MSL9DT5)	<Address: H'0080 159B>
CAN1 Message Slot 10 Data 5 (C1MSL10DT5)	<Address: H'0080 15AB>
CAN1 Message Slot 11 Data 5 (C1MSL11DT5)	<Address: H'0080 15BB>
CAN1 Message Slot 12 Data 5 (C1MSL12DT5)	<Address: H'0080 15CB>
CAN1 Message Slot 13 Data 5 (C1MSL13DT5)	<Address: H'0080 15DB>
CAN1 Message Slot 14 Data 5 (C1MSL14DT5)	<Address: H'0080 15EB>
CAN1 Message Slot 15 Data 5 (C1MSL15DT5)	<Address: H'0080 15FB>



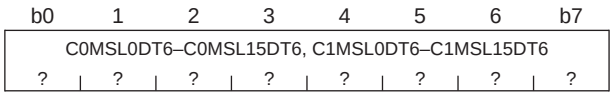
<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–15	C0MSL0DT5–C0MSL15DT5, C1MSL0DT5–C1MSL15DT5	Message slot data 5	R	W

These registers are the memory space for transmit and receive frames.

Note: • During a receive slot, an undefined value is written to the register if the data length of the data frame being stored (DLC value) is equal to or less than 5.

CAN0 Message Slot 0 Data 6 (C0MSL0DT6)	<Address: H'0080 110C>
CAN0 Message Slot 1 Data 6 (C0MSL1DT6)	<Address: H'0080 111C>
CAN0 Message Slot 2 Data 6 (C0MSL2DT6)	<Address: H'0080 112C>
CAN0 Message Slot 3 Data 6 (C0MSL3DT6)	<Address: H'0080 113C>
CAN0 Message Slot 4 Data 6 (C0MSL4DT6)	<Address: H'0080 114C>
CAN0 Message Slot 5 Data 6 (C0MSL5DT6)	<Address: H'0080 115C>
CAN0 Message Slot 6 Data 6 (C0MSL6DT6)	<Address: H'0080 116C>
CAN0 Message Slot 7 Data 6 (C0MSL7DT6)	<Address: H'0080 117C>
CAN0 Message Slot 8 Data 6 (C0MSL8DT6)	<Address: H'0080 118C>
CAN0 Message Slot 9 Data 6 (C0MSL9DT6)	<Address: H'0080 119C>
CAN0 Message Slot 10 Data 6 (C0MSL10DT6)	<Address: H'0080 11AC>
CAN0 Message Slot 11 Data 6 (C0MSL11DT6)	<Address: H'0080 11BC>
CAN0 Message Slot 12 Data 6 (C0MSL12DT6)	<Address: H'0080 11CC>
CAN0 Message Slot 13 Data 6 (C0MSL13DT6)	<Address: H'0080 11DC>
CAN0 Message Slot 14 Data 6 (C0MSL14DT6)	<Address: H'0080 11EC>
CAN0 Message Slot 15 Data 6 (C0MSL15DT6)	<Address: H'0080 11FC>
CAN1 Message Slot 0 Data 6 (C1MSL0DT6)	<Address: H'0080 150C>
CAN1 Message Slot 1 Data 6 (C1MSL1DT6)	<Address: H'0080 151C>
CAN1 Message Slot 2 Data 6 (C1MSL2DT6)	<Address: H'0080 152C>
CAN1 Message Slot 3 Data 6 (C1MSL3DT6)	<Address: H'0080 153C>
CAN1 Message Slot 4 Data 6 (C1MSL4DT6)	<Address: H'0080 154C>
CAN1 Message Slot 5 Data 6 (C1MSL5DT6)	<Address: H'0080 155C>
CAN1 Message Slot 6 Data 6 (C1MSL6DT6)	<Address: H'0080 156C>
CAN1 Message Slot 7 Data 6 (C1MSL7DT6)	<Address: H'0080 157C>
CAN1 Message Slot 8 Data 6 (C1MSL8DT6)	<Address: H'0080 158C>
CAN1 Message Slot 9 Data 6 (C1MSL9DT6)	<Address: H'0080 159C>
CAN1 Message Slot 10 Data 6 (C1MSL10DT6)	<Address: H'0080 15AC>
CAN1 Message Slot 11 Data 6 (C1MSL11DT6)	<Address: H'0080 15BC>
CAN1 Message Slot 12 Data 6 (C1MSL12DT6)	<Address: H'0080 15CC>
CAN1 Message Slot 13 Data 6 (C1MSL13DT6)	<Address: H'0080 15DC>
CAN1 Message Slot 14 Data 6 (C1MSL14DT6)	<Address: H'0080 15EC>
CAN1 Message Slot 15 Data 6 (C1MSL15DT6)	<Address: H'0080 15FC>



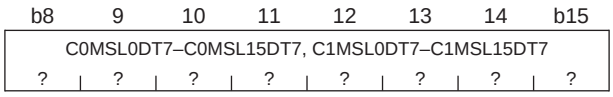
<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–7	C0MSL0DT6–C0MSL15DT6, C1MSL0DT6–C1MSL15DT6	Message slot data 6	R	W

These registers are the memory space for transmit and receive frames.

Note: • During a receive slot, an undefined value is written to the register if the data length of the data frame being stored (DLC value) is equal to or less than 6.

CAN0 Message Slot 0 Data 7 (C0MSL0DT7)	<Address: H'0080 110D>
CAN0 Message Slot 1 Data 7 (C0MSL1DT7)	<Address: H'0080 111D>
CAN0 Message Slot 2 Data 7 (C0MSL2DT7)	<Address: H'0080 112D>
CAN0 Message Slot 3 Data 7 (C0MSL3DT7)	<Address: H'0080 113D>
CAN0 Message Slot 4 Data 7 (C0MSL4DT7)	<Address: H'0080 114D>
CAN0 Message Slot 5 Data 7 (C0MSL5DT7)	<Address: H'0080 115D>
CAN0 Message Slot 6 Data 7 (C0MSL6DT7)	<Address: H'0080 116D>
CAN0 Message Slot 7 Data 7 (C0MSL7DT7)	<Address: H'0080 117D>
CAN0 Message Slot 8 Data 7 (C0MSL8DT7)	<Address: H'0080 118D>
CAN0 Message Slot 9 Data 7 (C0MSL9DT7)	<Address: H'0080 119D>
CAN0 Message Slot 10 Data 7 (C0MSL10DT7)	<Address: H'0080 11AD>
CAN0 Message Slot 11 Data 7 (C0MSL11DT7)	<Address: H'0080 11BD>
CAN0 Message Slot 12 Data 7 (C0MSL12DT7)	<Address: H'0080 11CD>
CAN0 Message Slot 13 Data 7 (C0MSL13DT7)	<Address: H'0080 11DD>
CAN0 Message Slot 14 Data 7 (C0MSL14DT7)	<Address: H'0080 11ED>
CAN0 Message Slot 15 Data 7 (C0MSL15DT7)	<Address: H'0080 11FD>
CAN1 Message Slot 0 Data 7 (C1MSL0DT7)	<Address: H'0080 150D>
CAN1 Message Slot 1 Data 7 (C1MSL1DT7)	<Address: H'0080 151D>
CAN1 Message Slot 2 Data 7 (C1MSL2DT7)	<Address: H'0080 152D>
CAN1 Message Slot 3 Data 7 (C1MSL3DT7)	<Address: H'0080 153D>
CAN1 Message Slot 4 Data 7 (C1MSL4DT7)	<Address: H'0080 154D>
CAN1 Message Slot 5 Data 7 (C1MSL5DT7)	<Address: H'0080 155D>
CAN1 Message Slot 6 Data 7 (C1MSL6DT7)	<Address: H'0080 156D>
CAN1 Message Slot 7 Data 7 (C1MSL7DT7)	<Address: H'0080 157D>
CAN1 Message Slot 8 Data 7 (C1MSL8DT7)	<Address: H'0080 158D>
CAN1 Message Slot 9 Data 7 (C1MSL9DT7)	<Address: H'0080 159D>
CAN1 Message Slot 10 Data 7 (C1MSL10DT7)	<Address: H'0080 15AD>
CAN1 Message Slot 11 Data 7 (C1MSL11DT7)	<Address: H'0080 15BD>
CAN1 Message Slot 12 Data 7 (C1MSL12DT7)	<Address: H'0080 15CD>
CAN1 Message Slot 13 Data 7 (C1MSL13DT7)	<Address: H'0080 15DD>
CAN1 Message Slot 14 Data 7 (C1MSL14DT7)	<Address: H'0080 15ED>
CAN1 Message Slot 15 Data 7 (C1MSL15DT7)	<Address: H'0080 15FD>



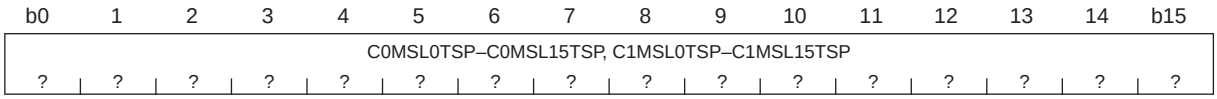
<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
8–15	C0MSL0DT7–C0MSL15DT7, C1MSL0DT7–C1MSL15DT7	Message slot data 7	R	W

These registers are the memory space for transmit and receive frames.

Note: • During a receive slot, an undefined value is written to the register if the data length of the data frame being stored (DLC value) is equal to or less than 7.

CAN0 Message Slot 0 Timestamp (C0MSL0TSP)	<Address: H'0080 110E>
CAN0 Message Slot 1 Timestamp (C0MSL1TSP)	<Address: H'0080 111E>
CAN0 Message Slot 2 Timestamp (C0MSL2TSP)	<Address: H'0080 112E>
CAN0 Message Slot 3 Timestamp (C0MSL3TSP)	<Address: H'0080 113E>
CAN0 Message Slot 4 Timestamp (C0MSL4TSP)	<Address: H'0080 114E>
CAN0 Message Slot 5 Timestamp (C0MSL5TSP)	<Address: H'0080 115E>
CAN0 Message Slot 6 Timestamp (C0MSL6TSP)	<Address: H'0080 116E>
CAN0 Message Slot 7 Timestamp (C0MSL7TSP)	<Address: H'0080 117E>
CAN0 Message Slot 8 Timestamp (C0MSL8TSP)	<Address: H'0080 118E>
CAN0 Message Slot 9 Timestamp (C0MSL9TSP)	<Address: H'0080 119E>
CAN0 Message Slot 10 Timestamp (C0MSL10TSP)	<Address: H'0080 11AE>
CAN0 Message Slot 11 Timestamp (C0MSL11TSP)	<Address: H'0080 11BE>
CAN0 Message Slot 12 Timestamp (C0MSL12TSP)	<Address: H'0080 11CE>
CAN0 Message Slot 13 Timestamp (C0MSL13TSP)	<Address: H'0080 11DE>
CAN0 Message Slot 14 Timestamp (C0MSL14TSP)	<Address: H'0080 11EE>
CAN0 Message Slot 15 Timestamp (C0MSL15TSP)	<Address: H'0080 11FE>
CAN1 Message Slot 0 Timestamp (C1MSL0TSP)	<Address: H'0080 150E>
CAN1 Message Slot 1 Timestamp (C1MSL1TSP)	<Address: H'0080 151E>
CAN1 Message Slot 2 Timestamp (C1MSL2TSP)	<Address: H'0080 152E>
CAN1 Message Slot 3 Timestamp (C1MSL3TSP)	<Address: H'0080 153E>
CAN1 Message Slot 4 Timestamp (C1MSL4TSP)	<Address: H'0080 154E>
CAN1 Message Slot 5 Timestamp (C1MSL5TSP)	<Address: H'0080 155E>
CAN1 Message Slot 6 Timestamp (C1MSL6TSP)	<Address: H'0080 156E>
CAN1 Message Slot 7 Timestamp (C1MSL7TSP)	<Address: H'0080 157E>
CAN1 Message Slot 8 Timestamp (C1MSL8TSP)	<Address: H'0080 158E>
CAN1 Message Slot 9 Timestamp (C1MSL9TSP)	<Address: H'0080 159E>
CAN1 Message Slot 10 Timestamp (C1MSL10TSP)	<Address: H'0080 15AE>
CAN1 Message Slot 11 Timestamp (C1MSL11TSP)	<Address: H'0080 15BE>
CAN1 Message Slot 12 Timestamp (C1MSL12TSP)	<Address: H'0080 15CE>
CAN1 Message Slot 13 Timestamp (C1MSL13TSP)	<Address: H'0080 15DE>
CAN1 Message Slot 14 Timestamp (C1MSL14TSP)	<Address: H'0080 15EE>
CAN1 Message Slot 15 Timestamp (C1MSL15TSP)	<Address: H'0080 15FE>



<Upon exiting reset: Undefined>

b	Bit Name	Function	R	W
0–15	C0MSL0TSP–C0MSL15TSP, C1MSL0TSP–C1MSL15TSP	Message slot timestamp	R	W

These registers are the memory space for transmit and receive frames. When transmission/reception has finished, the CAN timestamp count register value is written to the register.

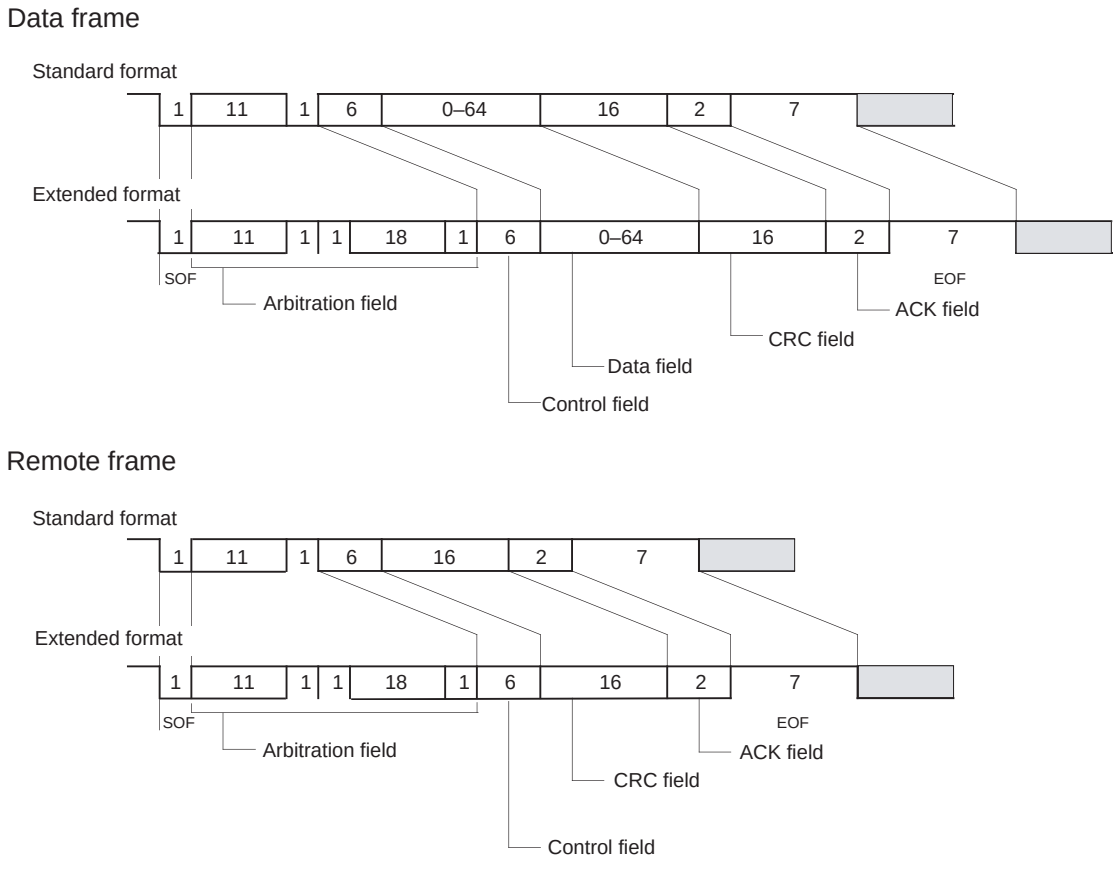
13.3 CAN Protocol

13.3.1 CAN Protocol Frames

There are four types of frames that are handled by CAN protocol:

(1) Data frame
(2) Remote frame
(3) Error frame
(4) Overload frame

Frames are separated from each other by an interframe space.



Note: • The number in each field denotes the number of bits.

Figure 13.3.1 CAN Protocol Frames (1)

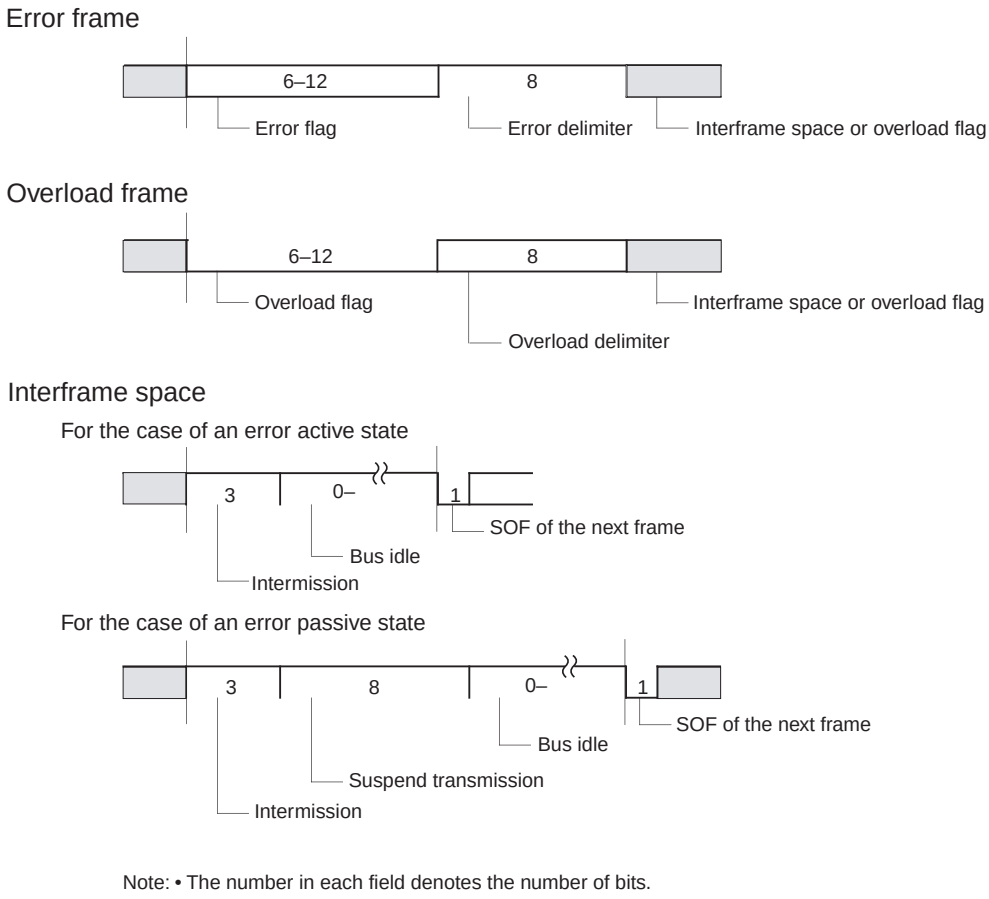


Figure 13.3.2 CAN Protocol Frames (2)

13.3.2 Data Formats during CAN Transmission/Reception

Figure 13.3.3 shows an example of the transmit/receive transfer data format that can be used in CAN. Data is transmitted/received sequentially beginning with the MSB side of the CAN message slot (C0MSLnSID0-C0MSLnDT7 and C1MSLnSID0-C1MSLnDT7).

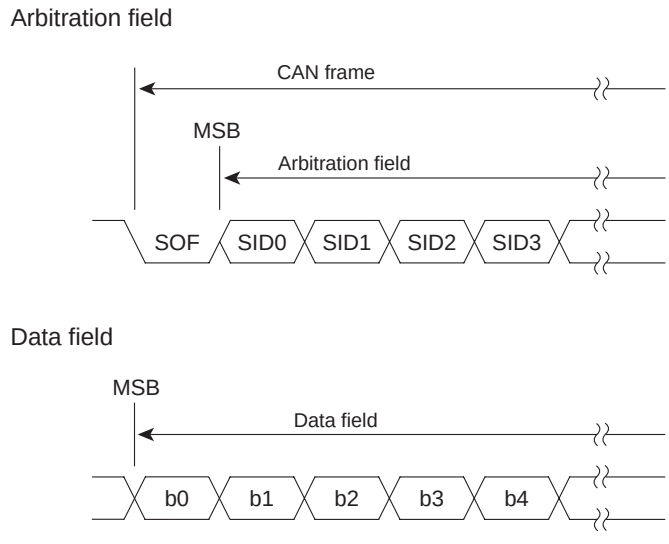


Figure 13.3.3 Example of CAN Transmit/Receive Transfer Data Format

13.3.3 CAN Controller Error States

The CAN controller assumes one of the following three error states depending on the transmit error and receive error counter values.

(1) Error active state

- This is a state where almost no errors have occurred.
- When an error is detected, an active error flag is transmitted.
- The CAN controller is in the state immediately after being initialized.

(2) Error passive state

- This is a state where many errors have occurred.
- When an error is detected, a passive error flag is transmitted.

(3) Bus off state

- This is a state where a very large number of errors have occurred.
- CAN communication with other nodes cannot be performed until the CAN module returns to an error active state.

Error Status of the Unit	Transmit Error Counter		Receive Error Counter
Error active state	0–127	AND	0–127
Error passive state	128–255	OR	128 and over
Bus off state	256 and over		–

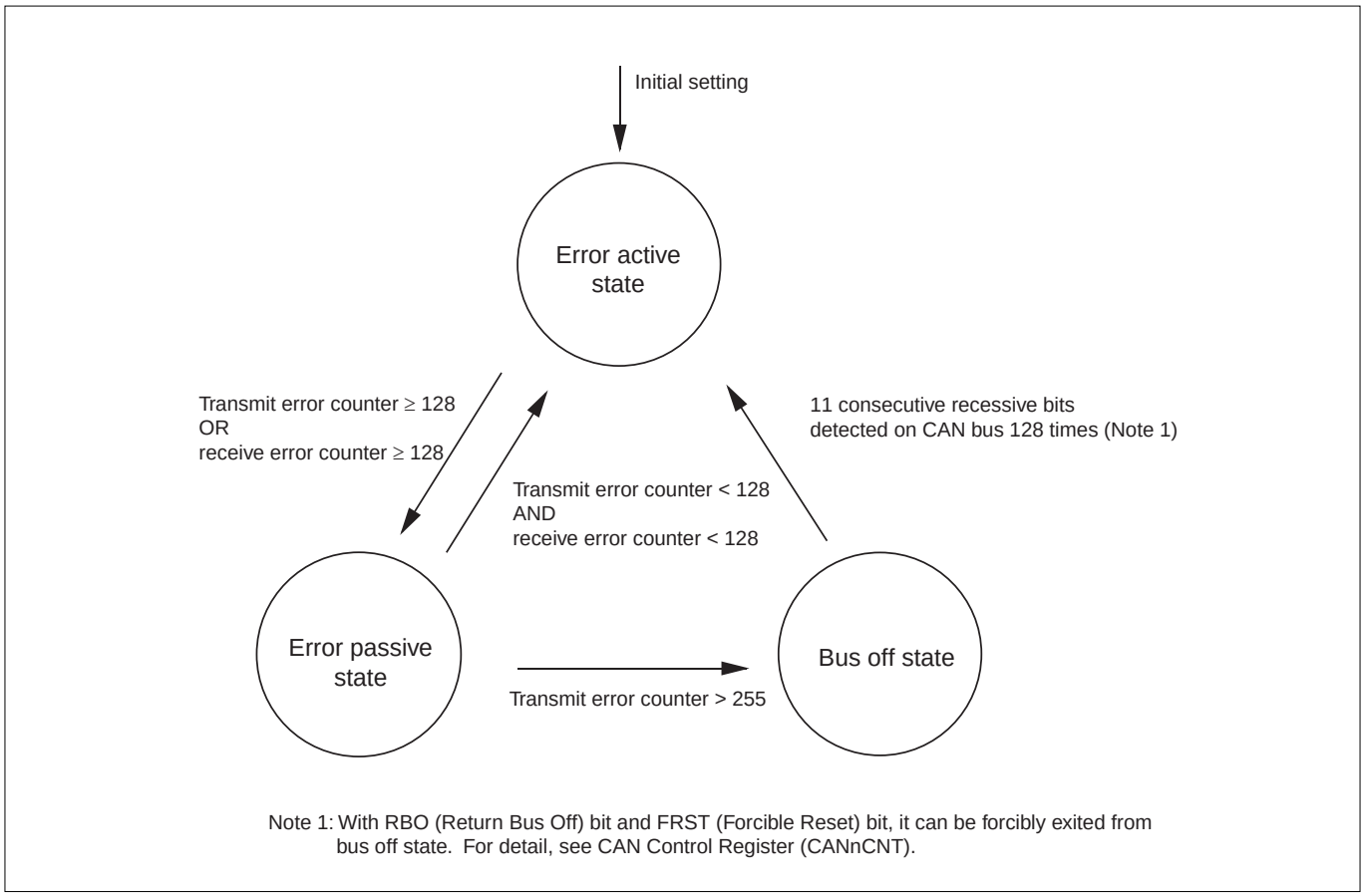


Figure 13.3.4 CAN Controller Error States

13.4 Initializing the CAN Module

13.4.1 Initializing the CAN Module

Before performing communication, set up the CAN module as described below.

(1) Selecting pin functions

The CAN transmit data output pin (CTX) and CAN receive data input pin (CRX) are shared with input/output ports. Be sure to select the functions of these pins. (See Chapter 8, “Input/Output Ports and Pin Functions.”

(2) Setting the Interrupt Controller (ICU)

To use CAN module interrupts, set their interrupt priority levels.

(3) Setting CAN Error, CAN Single-Shot and CAN Slot Interrupt Request Mask Registers

To use CAN bus error, CAN error passive, CAN error bus off, CAN single-shot or CAN slot interrupts, set each corresponding bit to "1" to enable the interrupt request.

(4) Setting DMAC

To use DMA transfers by CAN, be sure to set the DMAC.

(5) Setting CAN DMA transfer request select register

To use DMA transfers by CAN, set the CAN DMA transfer request select register to choose the cause of transfer request.

(6) Setting the bit timing and the number of times sampled

Using the CAN Configuration Register and CAN Baud Rate Prescaler, set the bit timing and the number of times the CAN bus is sampled.

1) Setting the bit timing

Determine the period Tq that is the base of bit timing, the configuration of Propagation Segment, Phase Segment1 and Phase Segment2, and reSynchronization Jump Width. The equation to calculate Tq is given below.

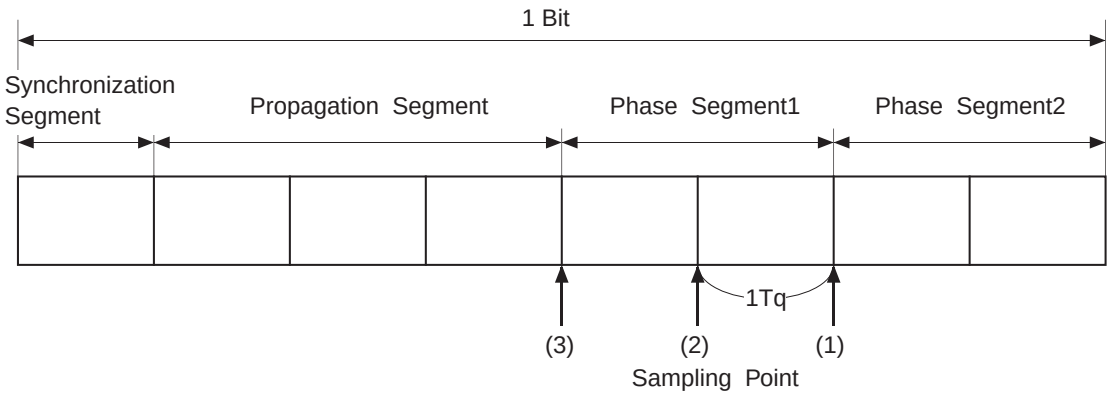
$$Tq = (BRP + 1) / (CPU\ clock)$$

The baud rate is determined by the number of Tq’s that comprise one bit. The equation to calculate the baud rate is given below.

$$\text{Baud rate (bps)} = \frac{1}{\text{Tq period} \times \text{number of Tq's in one bit}}$$

Number of Tq’s in one bit = Synchronization Segment + Propagation Segment + Phase Segment 1 + Phase Segment 2

Note: • The maximum baud rate for communication depends on the system configuration (e.g., bus length, clock error, CAN bus transceiver, sampling position and bit configuration). Consider the system configuration when setting the baud rate and number of Tq's.



- This diagram shows the bit timing when one bit consists of 8 Tq's.
- If one-time sampling is selected, the value sampled at Sampling Point (1) is assumed to be the value of the bit.
- If three-time sampling is selected, the value of the bit is determined by majority from CAN bus values sampled at Sampling Points (1), (2) and (3).

Figure 13.4.1 Example of Bit Timing

2) Setting the number of times sampled

Select the number of times the CAN bus is sampled from “one time” and “three times.”

- If one-time sampling is selected, the value sampled at only the end of Phase Segment1 is assumed to be the value of the bit.
- If three-time sampling is selected, the value of the bit is determined by majority from three sampled values, one sampled at the end of Phase Segment1 and the other sampled 1 Tq before and 2 Tq’s before that.

(7) Setting the ID mask registers

Set the values of ID mask registers (Global Mask Register, Local Mask Register A and Local Mask Register B) that are used in acceptance filtering of received messages.

(8) Settings for use in BasicCAN mode

- Set the CAN Extended ID Register IDE14 and IDE15 bits. (We recommend setting the same value in these bits.)
- Set IDs in message slots 14 and 15.
- Set the Message Control Registers 14 and 15 for data frame reception (H’40).

(9) Settings for use in single-shot mode

Using the CAN Mode Register (CANnMODE) and CAN Control Register (CANnCNT), select CAN module operation mode (BasicCAN, loopback mode) and the clock source for the timestamp counter.

(10) Setting CAN module operation mode

In the CAN Single-Shot Mode Control Register, set the slot that is to be operated in single-shot mode.

(11) Releasing CAN module from reset

When settings (1) through (10) above are finished, clear the CAN Control Register (CANnCNT)’s forcible reset (FRST) and reset (RST) bits to "0". Then, after detecting 11 consecutive recessive bits on the CAN bus, the CAN module becomes ready to communicate.

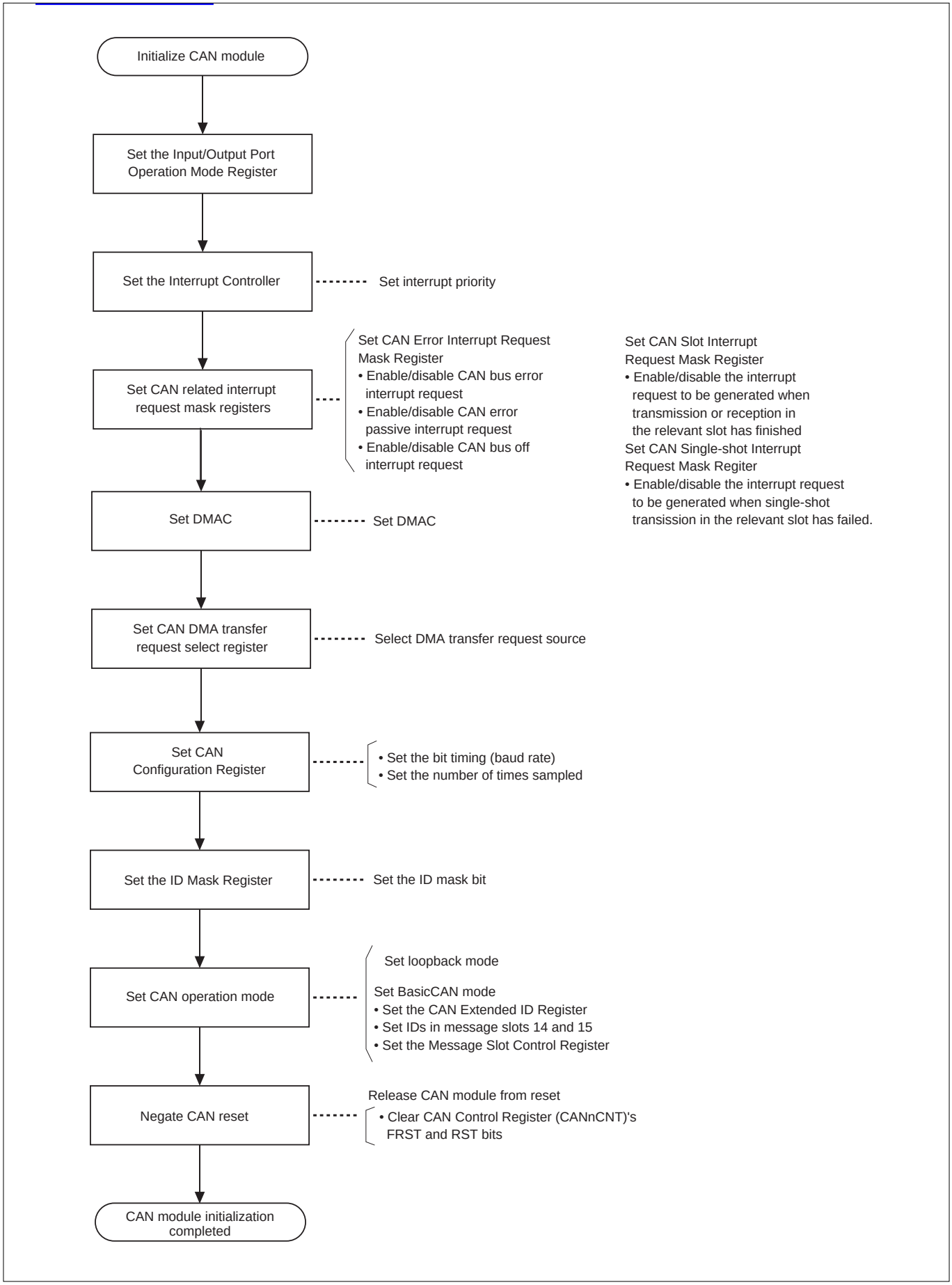


Figure 13.4.2 Initializing CAN Module

13.5 Transmitting Data Frames

13.5.1 Data Frame Transmit Procedure

The following describes the procedure for transmitting data frames.

(1) Initializing CAN Message Slot Control Register

Initialize the CAN Message Slot Control Register for the slot to be transmitted by writing H'00 to the register.

(2) Confirming that transmission is idle

Read the CAN Message Slot Control Register that has been initialized and check the TRSTAT (Transmit/ Receive Status) bit to see that transmission/reception has stopped and remains idle. If this bit = "1", it means that the CAN module is accessing the message slot. Therefore, wait until the bit is cleared to "0".

(3) Setting transmit data

Set the transmit ID and transmit data in the message slot.

(4) Setting the Extended ID Register

Set the corresponding bit in the Extended ID Register to "0" if the data is to be transmitted as a standard frame, or "1" if the data is to be transmitted as an extended frame.

(5) Setting CAN Message Slot Control Register

Write H'80 (Note 1) to the CAN Message Slot Control Register to set the TR (Transmit Request) bit to "1".

Note 1: Always be sure to write H'80 when transmitting data frames.

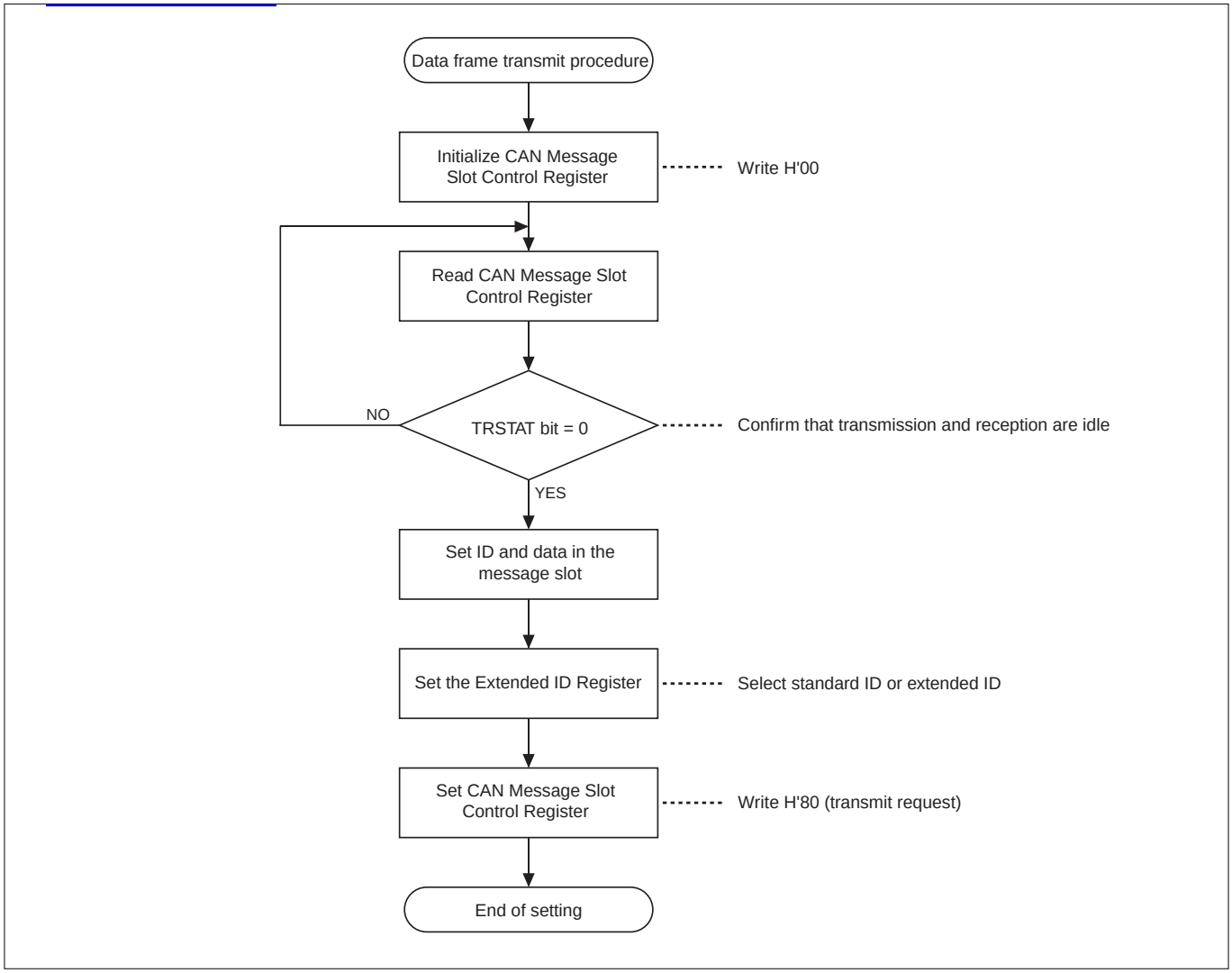


Figure 13.5.1 Data Frame Transmit Procedure

13.5.2 Data Frame Transmit Operation

The following describes data frame transmit operation. The operations described below are automatically performed in hardware.

(1) Selecting a transmit frame

The CAN module checks slots which have transmit requests (including remote frame transmit slots) every intermission to determine the frame to transmit. If two or more transmit slots exist, frames are transmitted in order of slot numbers beginning with the smallest.

(2) Transmitting a data frame

After determining the transmit slot, the CAN module sets the corresponding CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit to "1" and starts transmitting.

(3) If lost in CAN bus arbitration or a CAN bus error occurs

If the CAN module lost in CAN bus arbitration or a CAN bus error occurs in the middle of transmission, the CAN module clears the CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit to "0". If the CAN module requested a transmit abort, the transmit abort is accepted and the message slot is enabled for write.

(4) Completion of data frame transmission

When data frame transmission has finished, the CAN Message Slot Control Register's TRFIN (Transmit/Receive Finished) bit and the CAN Slot Interrupt Request Status Register are set to "1". Also, a timestamp count value at which transmission has finished is written to the CAN Message Slot Timestamp (C0MSLnTSP, C1MSLnTSP), and the transmit operation is thereby completed.
If the CAN slot interrupt request has been enabled, an interrupt request is generated at completion of transmit operation. The slot which has had transmission completed goes to an inactive state and remains inactive (neither transmit nor receive) until it is newly set in software.

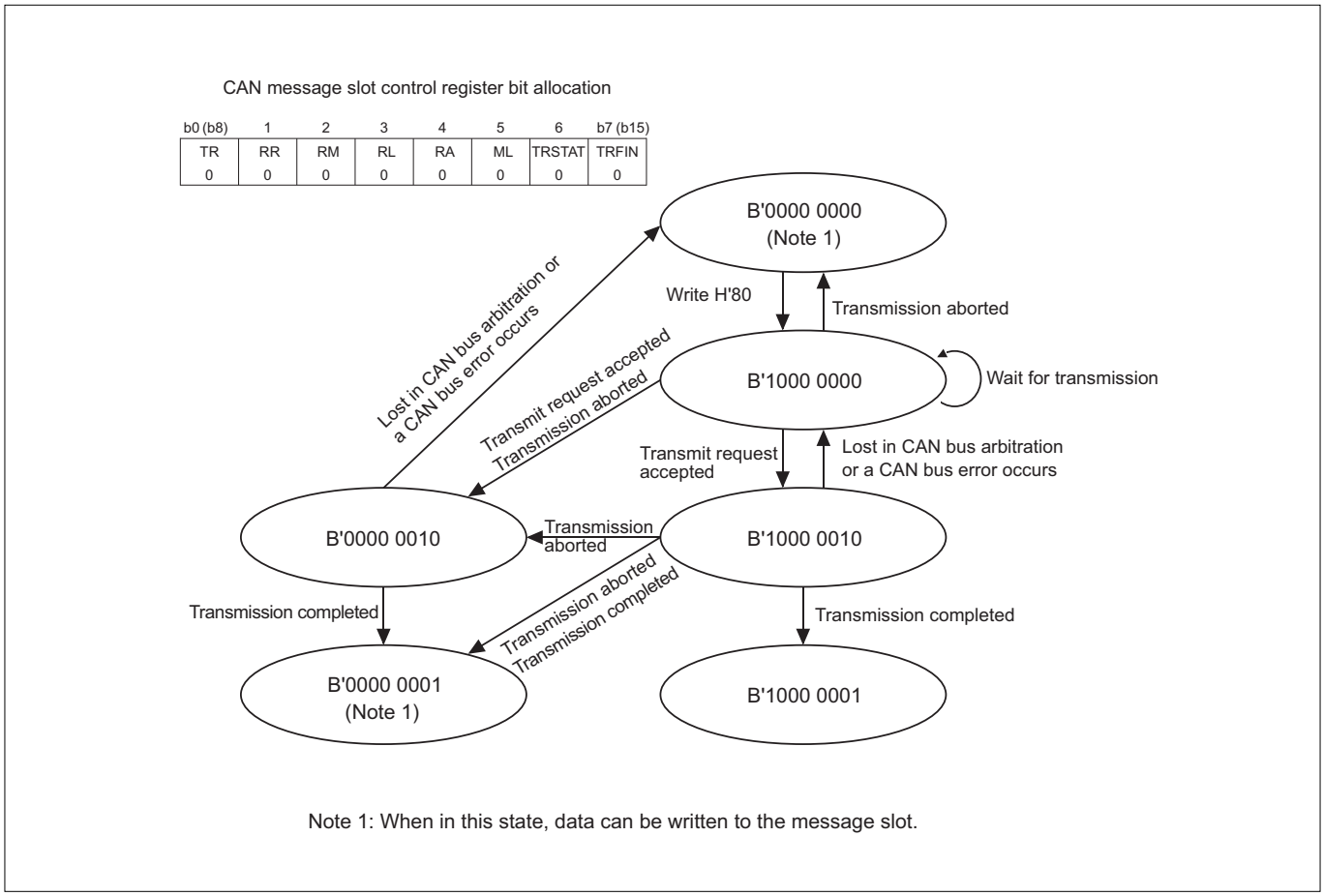


Figure 13.5.2 Operation of CAN Message Slot Control Register during Data Frame Transmission

13.5.3 Transmit Abort Function

The transmit abort function is used to cancel a transmit request that has once been set. This is accomplished by writing H'0F to the CAN Message Slot Control Register for the slot to be canceled. When transmit abort is accepted, the CAN module clears the CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit to "0", allowing for data to be written to the message slot. The following shows the conditions under which transmit abort is accepted.

[Conditions]

- When the target message is waiting for transmission
- When a CAN bus error occurs during transmission
- When lost in CAN bus arbitration

13.6 Receiving Data Frames

13.6.1 Data Frame Receive Procedure

The following describes the procedure for receiving data frames.

(1) Initializing CAN Message Slot Control Register

Initialize the CAN Message Slot Control Register for the slot to be received by writing H’00 to the register.

(2) Confirming that reception is idle

Read the CAN Message Slot Control Register that has been initialized and check the TRSTAT (Transmit/ Receive Status) bit to see that transmission and reception have stopped and remains idle. If this bit = "1", it means that the CAN module is accessing the message slot. Therefore, wait until the bit is cleared to "0".

(3) Setting the receive ID

Set the desired receive ID in the message slot.

(4) Setting the Extended ID Register

Set the corresponding bit in the Extended ID Register to "0" if a standard frame is to be received, or "1" if an extended frame is to be received.

(5) Setting CAN Message Slot Control Register

Write H’40 to the CAN Message Slot Control Register to set the RR (Receive Request) bit to "1".

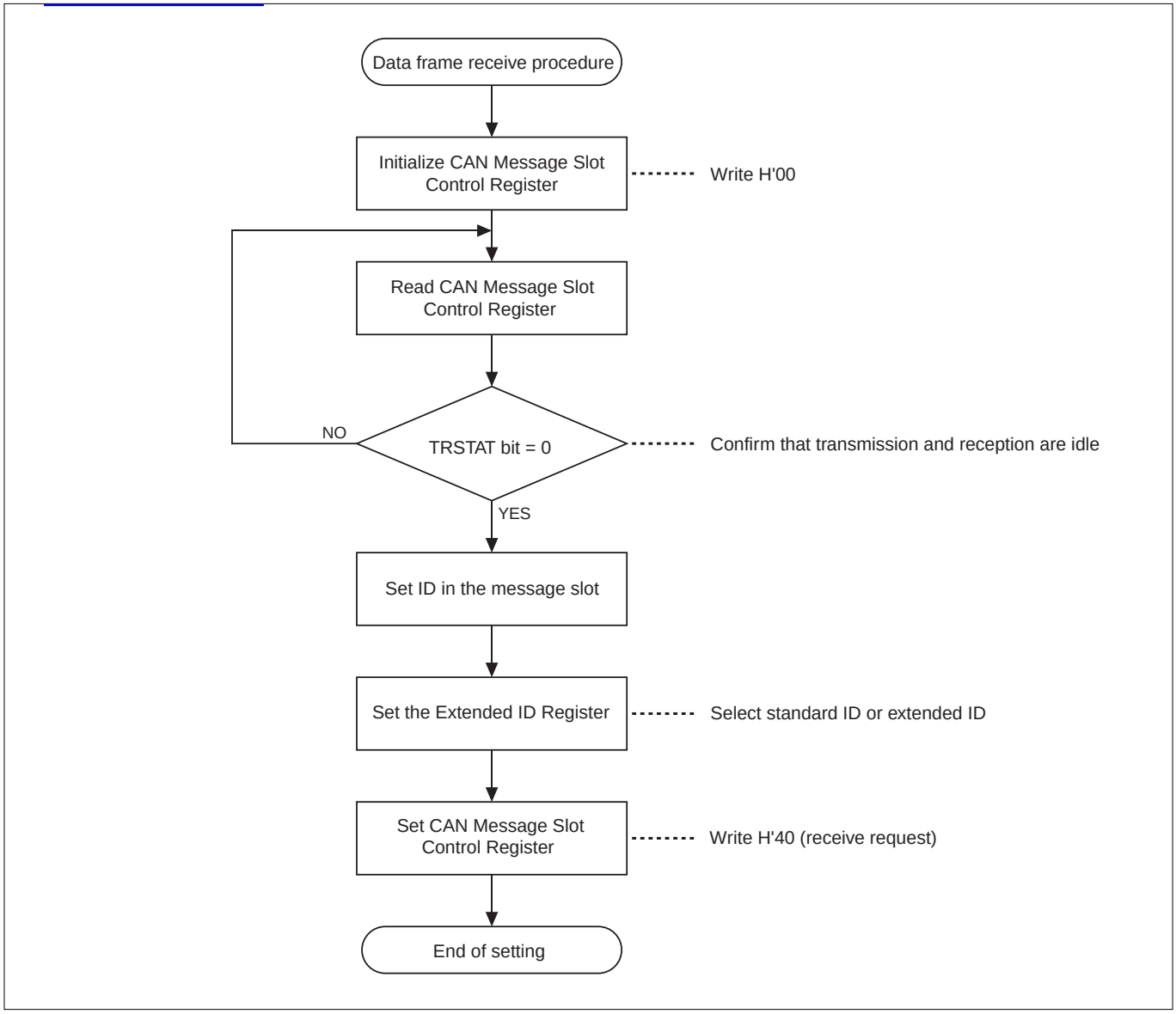


Figure 13.6.1 Data Frame Receive Procedure

13.6.2 Data Frame Receive Operation

The following describes data frame receive operation. The operations described below are automatically performed in hardware.

(1) Acceptance filtering

When the CAN module finished receiving data, it starts searching for the slot that satisfies the conditions for receiving the received message, sequentially from slot 0 (up to slot 15). The following shows receive conditions for the slots that have been set for data frame reception.

[Conditions]

- The received frame is a data frame.
- The receive ID and the slot ID are identical, assuming the ID Mask Register bits set to "0" are “Don’t care.”
- The standard and extended frame types are the same.

Note: • In BasicCAN mode, slots 14 and 15 while being set for data frame reception can also receive remote frames.

(2) When the receive conditions are met

When the receive conditions in (1) above are met, the CAN module sets the CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit and TRFIN (Transmit/Receive Finished) bit to "1" while at the same time writing the received data to the message slot. If the TRFIN (Transmit/Receive Finished) bit is already set to "1" at this time, the CAN module also sets the ML (Message Lost) bit to "1", indicating that the message slot has been overwritten. The message slot has both of its ID and DLC fields entirely overwritten and has an undefined value written in its unused area (e.g., extended ID field during standard frame reception and an unused data field).

Furthermore, a timestamp count value at which the message was received is written to the CAN Message Slot Timestamp (C0MSLnTSP, C1MSLnTSP) along with the received data. When the CAN module finished writing to the message slot, it sets the CAN Slot Interrupt Request Status bit to "1". If the interrupt request for the slot has been enabled, the CAN module generates an interrupt request and enters a wait state for the next reception.

(3) When the receive conditions are not met

The received frame is discarded, and the CAN module goes to the next transmit/receive operation without writing to the message slot.

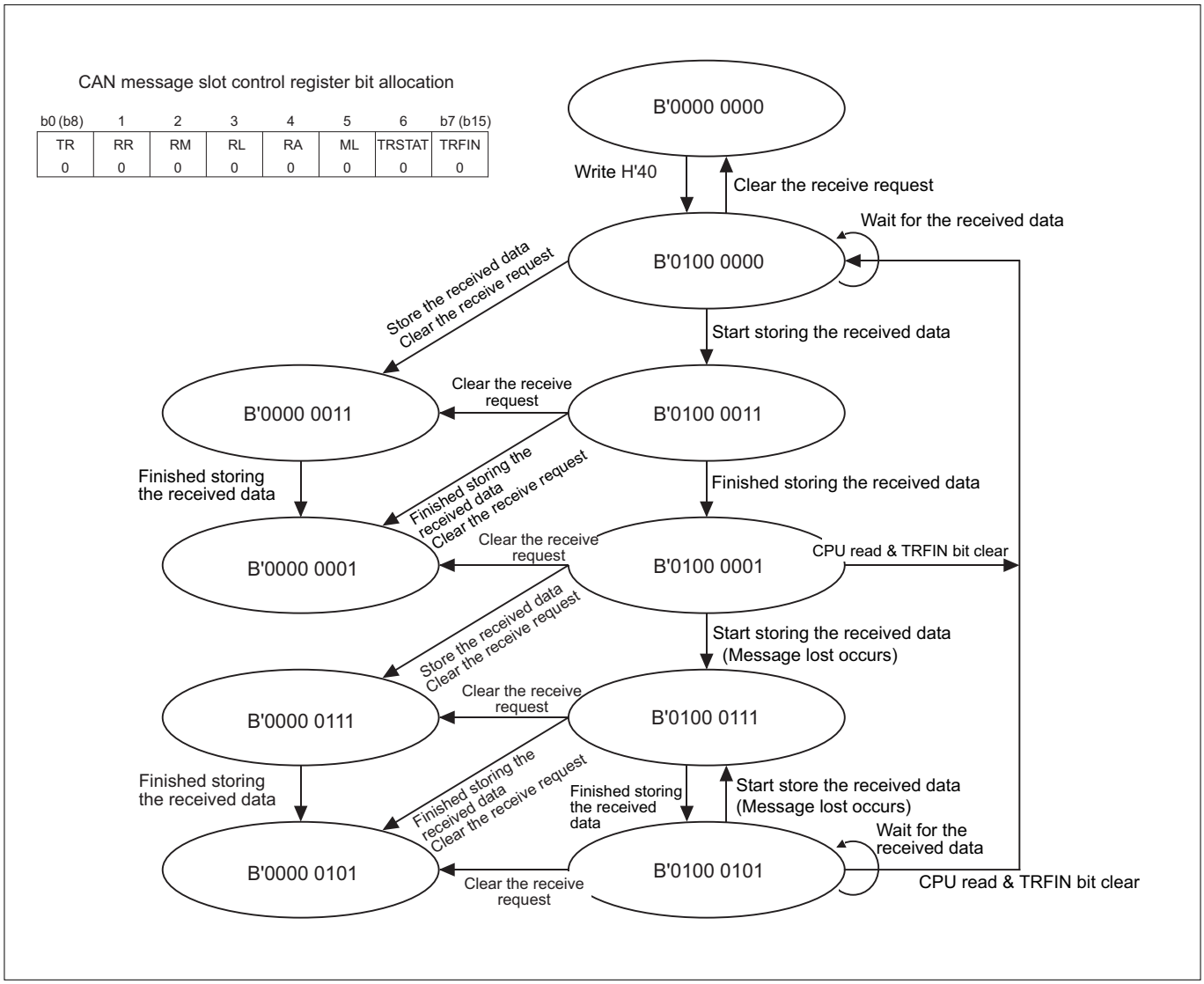


Figure 13.6.2 Operation of CAN Message Slot Control Register during Data Frame Reception

13.6.3 Reading Out Received Data Frames

The following shows the procedure for reading out received data frames from the slot.

(1) Clearing TRFIN (Transmit/Receive Finished) bit

Write H'4E, H'40 or H'00 to the CAN Message Slot Control Register (C0MSLnCNT, C1MSLnCNT) to clear the TRFIN bit to "0". After this write, the slot operates as follows:

Values Written to C0MSLnCNT, C1MSLnCNT	Slot Operation after Write
H'4E	Operates as a data frame receive slot. Whether overwritten can be verified by ML bit.
H'40	Operates as a data frame receive slot. Whether overwritten cannot be verified by ML bit.
H'00 (Note 1)	The slot stops transmit/receive operation.

Note 1: When the CAN Message Slot Control Register (C0MSLnCNT, C1MSLnCNT) RR (Receive Request) bit is cleared to "0" by writing H'00, and the receive operation has started until just before the bit is cleared, the transmit/ receive control will be performed until the receive operation is finished.

Note: • To conduct message lost check by ML bit, write H'4E and clear TRFIN bit.

(2) Reading out from the message slot

Read out a message from the message slot.

(3) Checking TRFIN (Transmit/Receive Finished) bit

Read the CAN Message Slot Control Register to check the TRFIN (Transmit/Receive Finished) bit.

1) If TRFIN (Transmit/Receive Finished) bit = "1"

It means that new data was stored in the slot while still reading out a message from it in (2) above. In this case, the data read out in (2) may contain an undefined value. Therefore, reexecute the above procedure beginning with clearing of the TRFIN (Transmit/Receive Finished) bit in (1).

2) If TRFIN (Transmit/Receive Finished) bit = "0"

It means that the CAN module finished reading out from the slot normally.

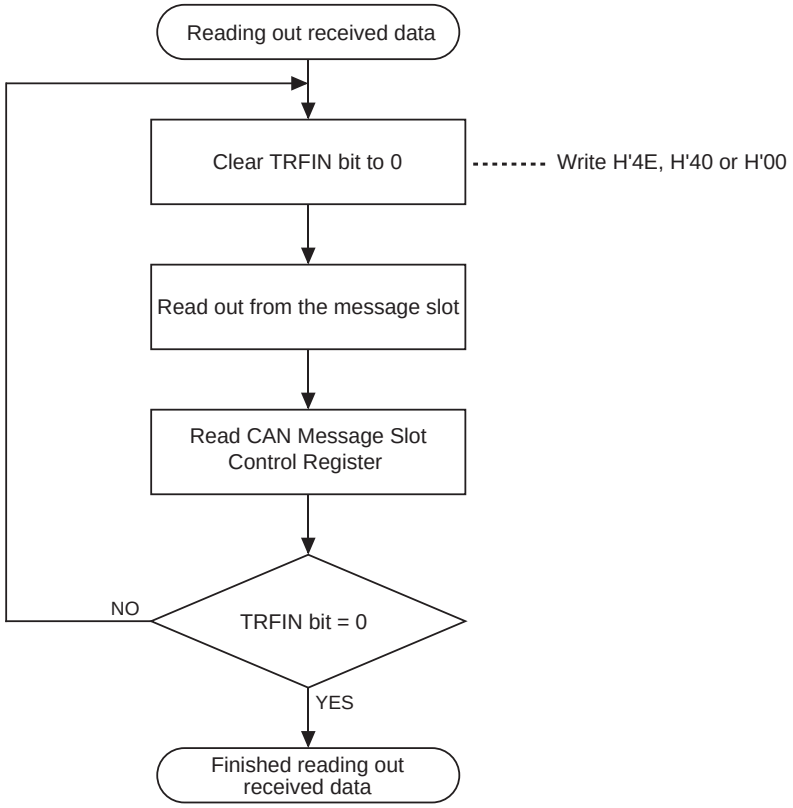


Figure 13.6.3 Procedure for Reading Out Received Data

13.7 Transmitting Remote Frames

13.7.1 Remote Frame Transmit Procedure

The following describes the procedure for transmitting remote frames.

(1) Initializing CAN Message Slot Control Register

Initialize the CAN Message Slot Control Register for the slot to be transmitted by writing H’00 to the register.

(2) Confirming that transmission is idle

Read the CAN Message Slot Control Register that has been initialized and check the TRSTAT (Transmit/ Receive Status) bit to see that transmission/reception has stopped and remains idle. If this bit = "1", it means that the CAN module is accessing the message slot. Therefore, wait until the bit is cleared to "0".

(3) Setting transmit ID

Set the ID to be transmitted in the message slot.

(4) Setting the Extended ID Register

Set the corresponding bit in the Extended ID Register to "0" if the data is to be transmitted as a standard frame, or "1" if the data is to be transmitted as an extended frame.

(5) Setting CAN Message Slot Control Register

Write H’A0 to the CAN Message Slot Control Register to set the TR (Transmit Request) bit and RM (Remote) bit to "1".

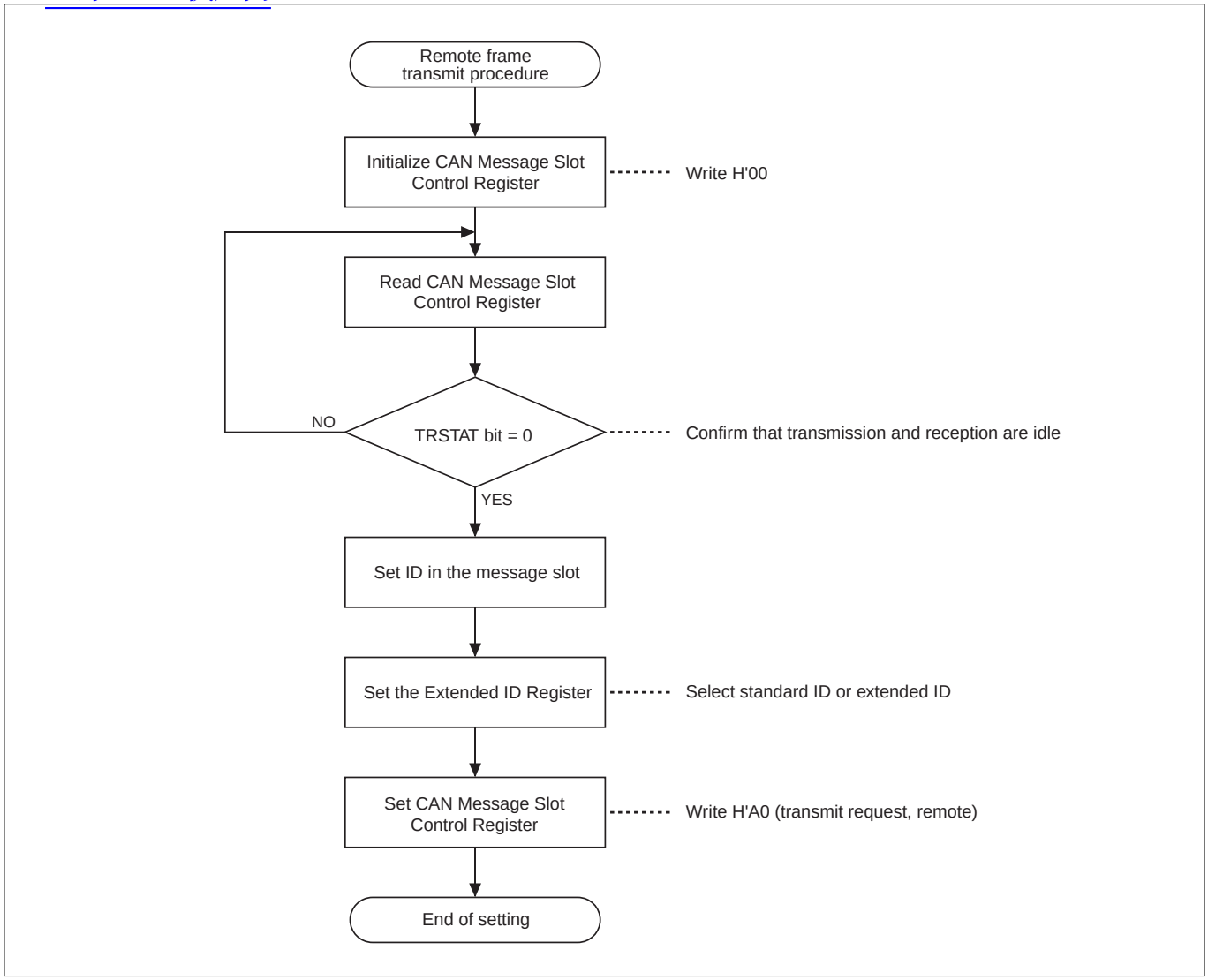


Figure 13.7.1 Remote Frame Transmit Procedure

13.7.2 Remote Frame Transmit Operation

The following describes remote frame transmit operation. The operations described below are automatically performed in hardware.

(1) Setting RA (Remote Active) bit

The RA (Remote Active) bit is set to "1" at the same time H'A0 (Transmit Request, Remote) is written to the CAN Message Slot Control Register, indicating that the corresponding slot is to handle remote frames.

(2) Selecting a transmit frame

The CAN module checks slots which have transmit requests (including data frame transmit slots) every intermission to determine the frame to transmit. If two or more transmit slots exist, frames are transmitted in order of slot numbers beginning with the smallest.

(3) Transmitting a remote frame

After determining the transmit slot, the CAN module sets the corresponding CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit to "1" and starts transmitting.

(4) If lost in CAN bus arbitration or a CAN bus error occurs

If the CAN module lost in CAN bus arbitration or a CAN bus error occurs in the middle of transmission, the CAN module clears the CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit to "0". If the CAN module requested a transmit abort, the transmit abort is accepted and the message slot is enabled for write.

(5) Completion of remote frame transmission

When remote frame transmission finishes, the timestamp count value at which transmission finished is written to the CAN Message Slot Timestamp (C0MSLnTSP, C1MSLnTSP) and the CAN Message Slot Control Register's RA (Remote Active) bit is cleared to "0". In addition, the CAN Slot Interrupt Request Status bit is set to "1" by completion of transmission, but the CAN Message Slot Control Register's TRFIN (Transmit/Receive Finished) bit is not set to "1". If the CAN slot interrupt request has been enabled, an interrupt request is generated when transmission has finished.

(6) Receiving a data frame

When remote frame transmission finishes, the slot automatically starts functioning as a data frame receive slot.

(7) Acceptance filtering

When the CAN module finished receiving data, it starts searching for the slot that satisfies the conditions for receiving the received message, sequentially from slot 0 (up to slot 15). The following shows receive conditions for the slots that have been set for data frame reception.

[Conditions]

- The received frame is a data frame.
- The receive ID and the slot ID are identical, assuming the ID Mask Register bits set to "0" are "Don't care."
- The standard and extended frame types are the same.

Note: • In BasicCAN mode, slots 14 and 15 cannot be used as a transmit slot.

(8) When the receive conditions are met

When the receive conditions in (7) above are met, the CAN module sets the CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit and TRFIN (Transmit/Receive Finished) bit to "1" while at the same time writing the received data to the message slot. If the TRFIN (Transmit/Receive Finished) bit is already set to "1" at this time, the CAN module also sets the ML (Message Lost) bit to "1", indicating that the message slot has been overwritten. The message slot has both of its ID and DLC fields entirely overwritten and has an undefined value written in its unused area (e.g., extended ID field during standard frame reception and an unused data field).
Furthermore, a timestamp count value at which the message was received is written to the CAN Message Slot Timestamp (C0MSLnTSP, C1MSLnTSP) along with the received data. When the CAN module finished writing to the message slot, it sets the CAN Slot Interrupt Request Status bit to "1". If the interrupt request for the slot has been enabled, the CAN module generates an interrupt request and enters a wait state for the next reception.

Note: • If the CAN module receives a corresponding data frame before sending a remote frame, it stores the received data frame in the slot and does not transmit the remote frame.

(9) When the receive conditions are not met

The received frame is discarded, and the CAN module goes to the next transmit/receive operation without writing to the message slot.

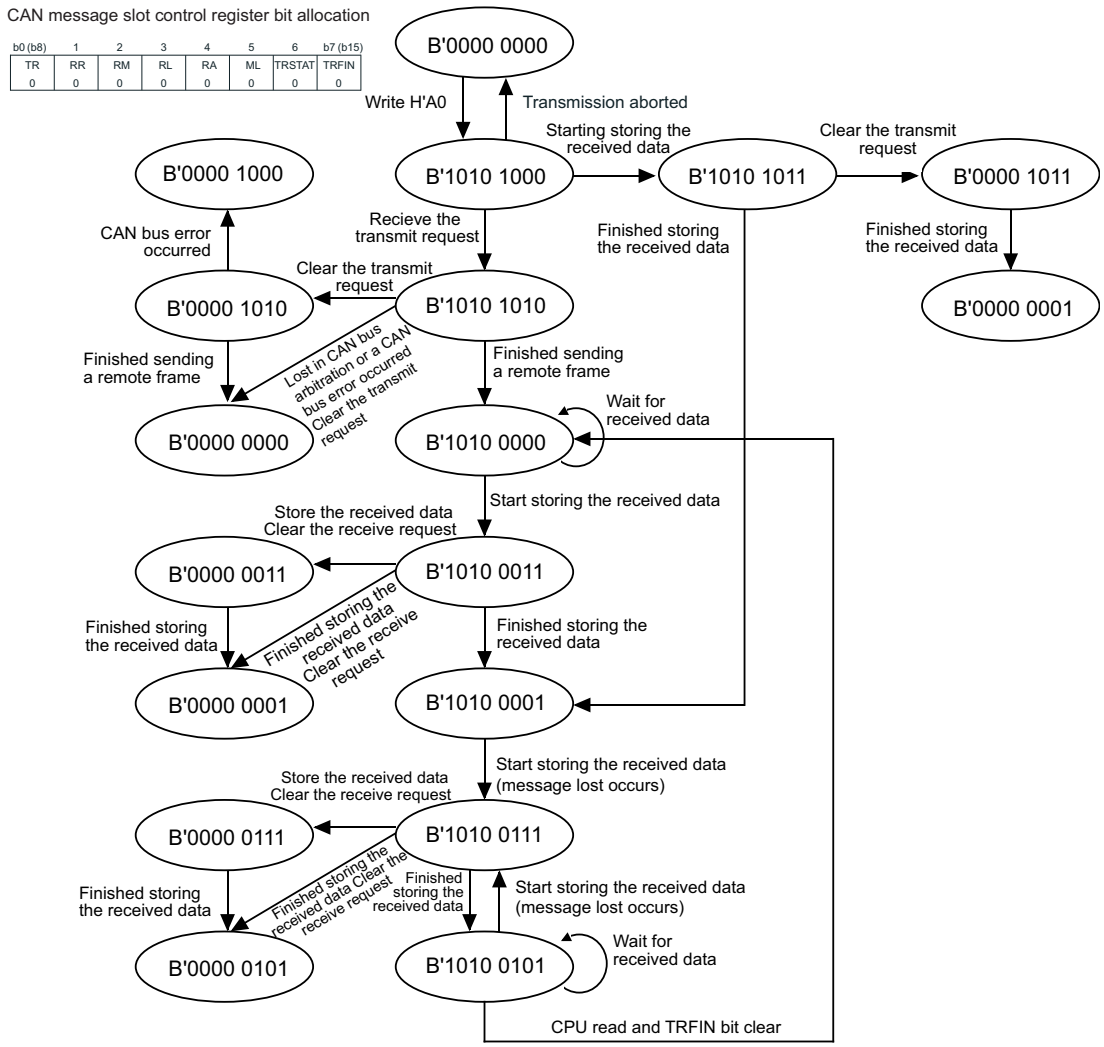


Figure 13.7.2 Operation of the CAN Message Slot Control Register during Remote Frame Transmission

13.7.3 Reading Out Received Data Frames when Set for Remote Frame Transmission

The following shows the procedure for reading out the data frames that have been received in the slot when it is set for remote frame transmission.

(1) Clearing TRFIN (Transmit/Receive Finished) bit

Write H'AE or H'00 to the CAN Message Slot Control Register (C0MSLnCNT, C1MSLnCNT) to clear the TRFIN bit to "0". After this write, the slot operates as follows:

Values Written to C0MSLnCNT, C1MSLnCNT	Slot Operation after Write
H'AE	Operates as a data frame receive slot. Whether overwritten can be verified by ML bit.
H'00	The slot stops transmit/receive operation.

Notes: • If message-lost check by the ML bit is needed, write H'AE to clear the TRFIN bit.
• If the TRFIN bit was cleared by writing H'AE or H'00, it is possible that new data will be stored in the slot while still reading out a message from it.
• The received data frame cannot be read out by writing H'A0 to the register. If the TRFIN bit is cleared by writing H'A0, the slot performs remote frame transmit operation.

(2) Reading out from the message slot

Read out a message from the message slot.

(3) Checking TRFIN (Transmit/Receive Finished) bit

Read the CAN Message Slot Control Register to check the TRFIN (Transmit/Receive Finished) bit.

1) If TRFIN (Transmit/Receive Finished) bit = "1"

It means that new data was stored in the slot while still reading out a message from it in (2) above. In this case, the data read out in (2) may contain an undefined value. Therefore, reexecute the above procedure beginning with clearing of the TRFIN (Transmit/Receive Finished) bit in (1).

2) If TRFIN (Transmit/Receive Finished) bit = "0"

It means that the CAN module finished reading out from the slot normally.

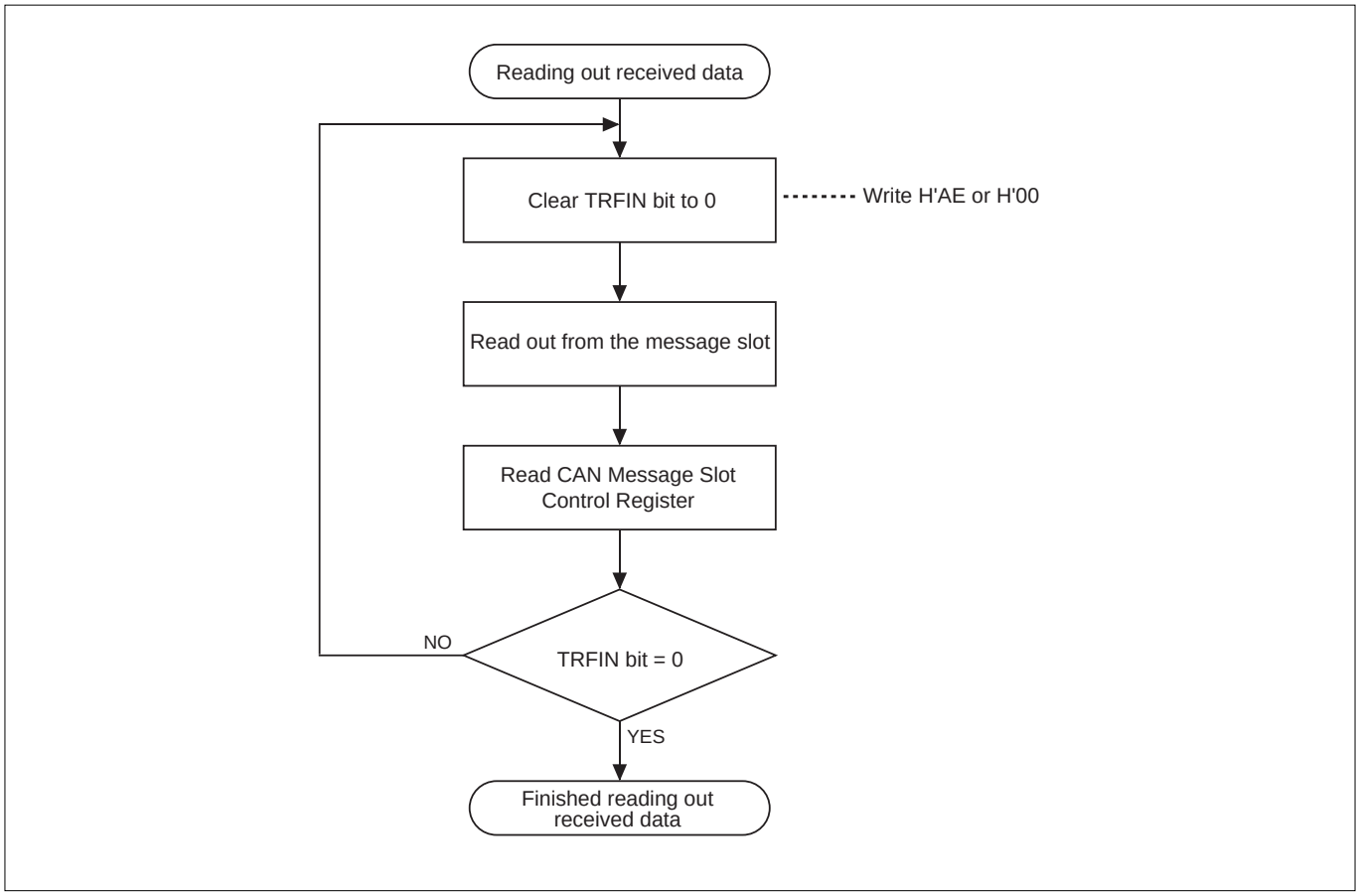


Figure 13.7.3 Procedure for Reading Out Received Data when Set for Remote Frame Transmission

13.8 Receiving Remote Frames

13.8.1 Remote Frame Receive Procedure

The following describes the procedure for receiving remote frames.

(1) Initializing CAN Message Slot Control Register

Initialize the CAN Message Slot Control Register for the slot to be received by writing H'00 to the register.

(2) Confirming that reception is idle

Read the CAN Message Slot Control Register that has been initialized and check the TRSTAT (Transmit/ Receive Status) bit to see that reception has stopped and remains idle. If this bit = "1", it means that the CAN module is accessing the message slot. Therefore, wait until the bit is cleared to "0".

(3) Setting the receive ID

Set the desired receive ID in the message slot.

(4) Setting the Extended ID Register

Set the corresponding bit in the Extended ID Register to "0" if a standard frame is to be received, or "1" if an extended frame is to be received.

(5) Setting CAN Message Slot Control Register

1) When automatic response (data frame transmission) for remote frame reception is desired

Write H'60 to the CAN Message Slot Control Register to set the RR (Receive Request) bit and RM (Remote) bit to "1".

2) When automatic response (data frame transmission) for remote frame reception is to be disabled

Write H'70 to the CAN Message Slot Control Register to set the RR (Receive Request) bit, RM (Remote) bit and RL (Automatic Response Enable) bit to "1".

Note: • During BasicCAN mode, slots 14 and 15, although capable of receiving remote frames, cannot automatically respond to remote frame reception.

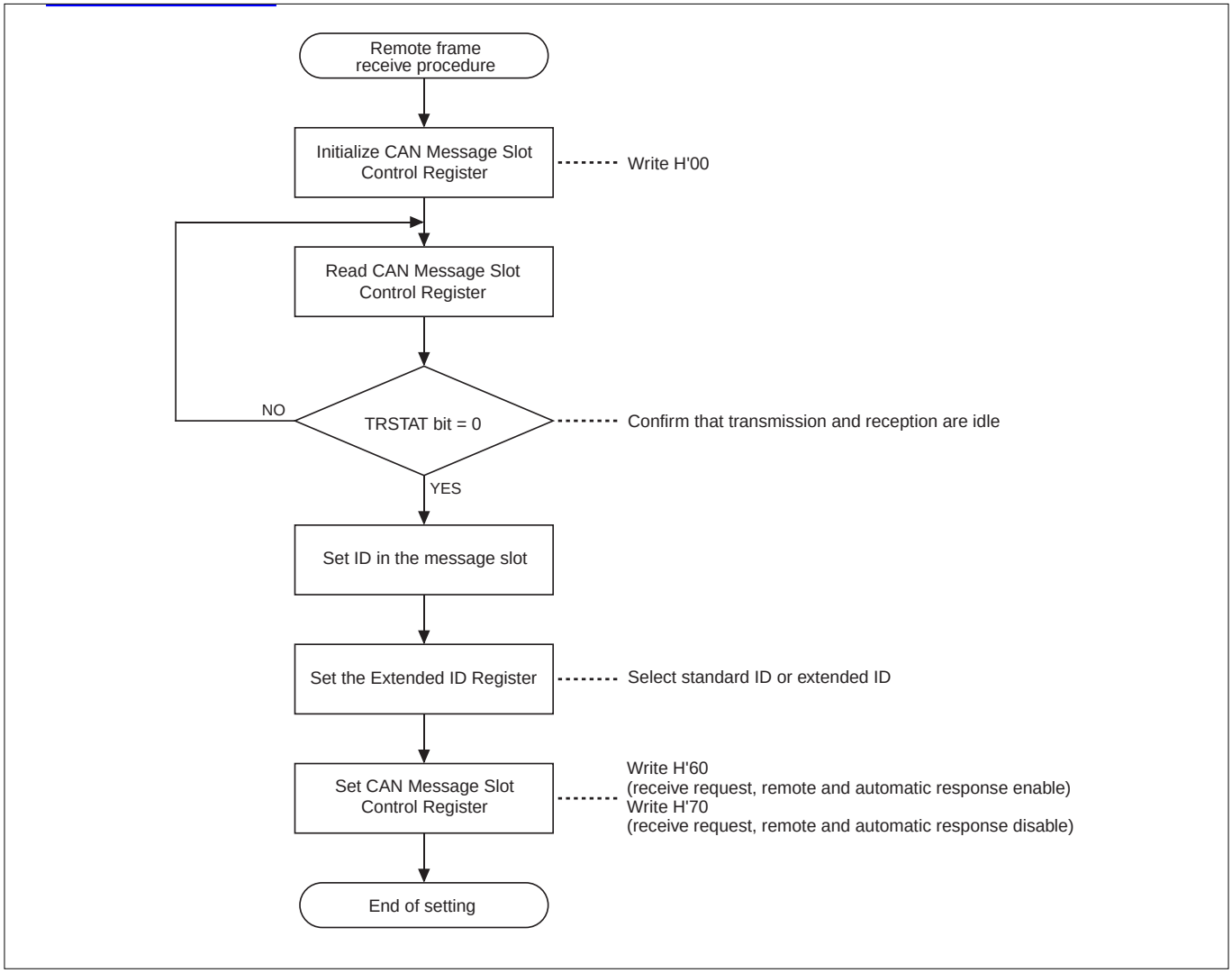


Figure 13.8.1 Remote Frame Receive Procedure

13.8.2 Remote Frame Receive Operation

The following describes remote frame receive operation. The operations described below are automatically performed in hardware.

(1) Setting RA (Remote Active) bit

The RA (Remote Active) bit indicating that the corresponding slot is to handle remote frames is set to "1" at the same time H'60 (Receive Request, Remote, Automatic Response Enable) or H'70 (Receive Request, Remote, Automatic Response Disable) is written to the CAN Message Slot Control Register.

(2) Acceptance filtering

When the CAN module finished receiving data, it starts searching for the slot that satisfies the conditions for receiving the received message, sequentially from slot 0 (up to slot 15). The following shows receive conditions for the slots that have been set for remote frame reception.

[Conditions]

- The received frame is a remote frame.
- The receive ID and the slot ID are identical, assuming the ID Mask Register bits set to "0" are “Don’t care.”
- The standard and extended frame types are the same.

(3) When the receive conditions are met

When the receive conditions in (2) above are met, the CAN module sets the CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit and TRFIN (Transmit/Receive Finished) bit to "1" while at the same time writing the received data to the message slot. In addition, a timestamp count value at which the message was received is written to the CAN Message Slot Timestamp (C0MSLnTSP, C1MSLnTSP) along with the received data. When the CAN module finished writing to the message slot, it sets the CAN Slot Interrupt Request Status bit to "1". If the interrupt request for the slot has been enabled, the CAN module generates an interrupt request.

Notes: • The ID field and DLC value are written to the message slot.
• An undefined value is written to the extended ID area when receiving standard format frames.
• The data field is not written to.
• The RA and TRFIN bits are cleared to "0" after writing the received remote frame data.

(4) When the receive conditions are not met

The received data is discarded, and the CAN module waits for the next receive frame. No data is written to the message slot.

(5) Operation after receiving a remote frame

The operation performed after receiving a remote frame differs depending on how automatic response is set.

1) When automatic response is disabled

The slot which has had reception completed goes to an inactive state and remains inactive (neither transmit nor receive) until it is newly set in software.

2) When automatic response is enabled

After receiving a remote frame, the slot automatically changes to a data frame transmit slot and performs the transmit operation described below. In this case, the transmitted data conforms to the ID and DLC of the received remote frame.

• Selecting a transmit frame

The CAN module checks slots which have transmit requests (including remote frame transmit slots) every intermission to determine the frame to transmit. If two or more transmit slots exist, frames are transmitted in order of slot numbers beginning with the smallest.

• Transmitting a data frame

After determining the transmit slot, the CAN module sets the corresponding CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit to "1" and starts transmitting.

• If lost in CAN bus arbitration or a CAN bus error occurs

If the CAN module lost in CAN bus arbitration or a CAN bus error occurs in the middle of transmission, the CAN module clears the CAN Message Slot Control Register's TRSTAT (Transmit/Receive Status) bit to "0". If the CAN module requested a transmit abort, the transmit abort is accepted and the message slot is enabled for write.

• Completion of data frame transmission

When data frame transmission has finished, the CAN Message Slot Control Register's TRFIN (Transmit/Receive Finished) bit and the CAN Slot Interrupt Request Status Register are set to "1". Also, a timestamp count value at which transmission has finished is written to the CAN Message Slot Timestamp (C0MSLnTSP, C1MSLnTSP), and the transmit operation is thereby completed.
If the CAN slot interrupt request has been enabled, an interrupt request is generated at completion of transmit operation. The slot which has had transmission completed goes to an inactive state and remains inactive (neither transmit nor receive) until it is newly set in software.

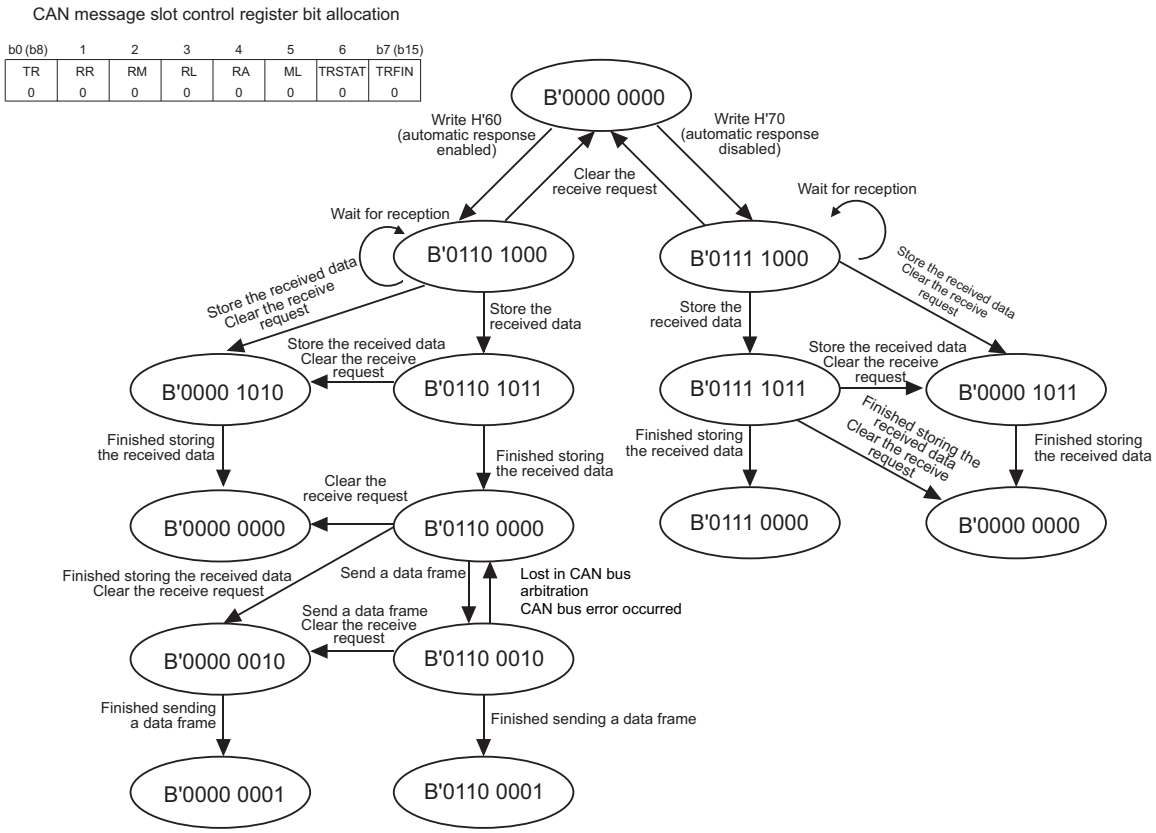


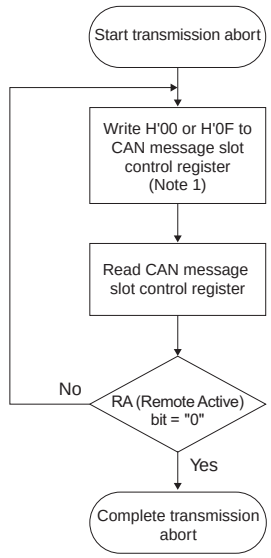
Figure 13.8.2 Operation of CAN Message Slot Control Register during Remote Frame Reception

13.9 Notes on CAN Module

• Note for cancelation of transmit and receive CAN remote frame

When aborting remote frame transmission or canceling remote frame receiving, make sure that the RA (Remote Active) bit is cleared to "0" after writing "H'00" or "H'0F" to the CAN Message Slot Control Register.

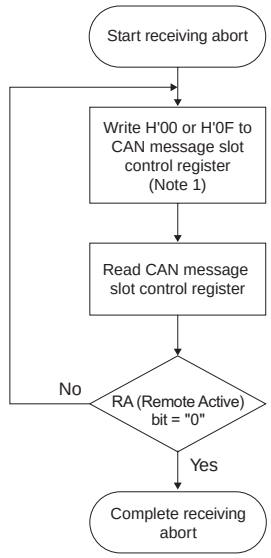
(1) When aborting remote frame transmission



Note 1: H'00 or H'0F can be used.

Figure 13.9.1 Opertion Flow when Aborting Remote Frame Transmission

(2) When canceling remote frame receiving



Note 1: H'00 or H'0F can be used.

Figure 13.9.2 Opertion Flow when Canceling Remote Frame Receiving

CHAPTER 14

REAL TIME DEBUGGER (RTD)

14.1 Outline of the Real-Time Debugger (RTD)

The Real-Time Debugger (RTD) is a serial interface through which to read or write to any location in the entire area of the internal RAM by using commands from outside the microcomputer. Because data transfers between the RTD and internal RAM are performed via a dedicated internal bus independently of the M32R, RTD operation can be controlled without the need to stop the M32R.

Table 14.1.1 Outline of the Real-Time Debugger (RTD)

Item	Description
Transfer method	Clock-synchronous serial interface
Generation of transfer clock	Generated by external host
RAM access area	Entire area of the internal RAM (controlled by A16–A29)
Transmit/receive data length	32 bits (fixed)
Bit transfer sequence	LSB first
Maximum transfer rate	2 Mbits/second
Input/output pins	4 pins (RTDTXD, RTDRXD, RTDACK, RTDCLK)
Number of commands	Following five functions • Monitor continuously • Output real-time RAM content • Forcibly rewrite RAM content (with verify) • Recover from runaway condition • Request RTD interrupt

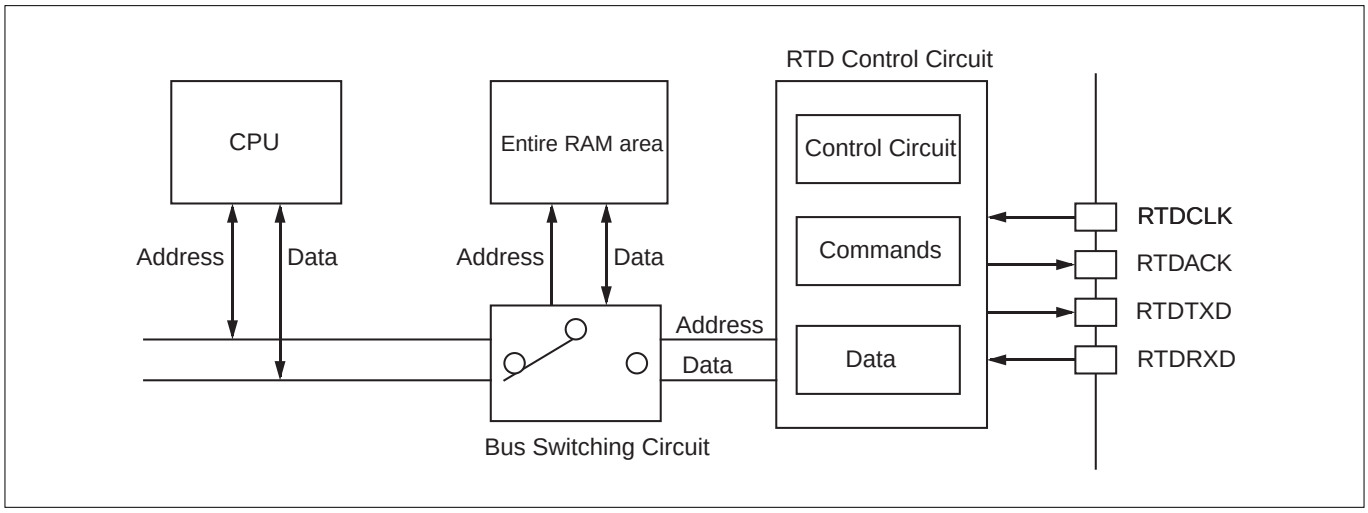


Figure 14.1.1 Block Diagram of the Real-Time Debugger (RTD)

14.2 Pin Functions of RTD

Pin Functions of the RTD are shown below.

Table 14.2.1 Pin Functions of RTD

Pin Name	Type	Function
RTDTXD	Output	RTD serial data output
RTDRXD	Input	RTD serial data input
RTDACK	Output	Output a "L" level pulse synchronously with the beginning clock edge of the output data word. The width of this pulse indicates the type of instruction or data the RTD has received. 1 clock period: VER (continuous monitor) command 1 clock period: VEI (RTD interrupt request) command 2 clock periods: RDR (real-time RAM content output) command 3 clock periods: WRR (RAM content forcible rewrite) command or the data to rewrite 4 clock periods or more: RCV (recover from runaway) command
RTDCLK	Input	RTD transfer clock input

14.3 RTD Related Register

The following shows an RTD related register map.

RTD Related Register Map

Address	+0 address b0 b7	+1 address b8 b15	See page
H'0080 077A	(Use inhibited area)	RTD write function disable register (WRRDIS)	14-3

14.3.1 RTD Write Function Disable Register

RTD Write Function Disable Register (WRRDIS) <Address: H'0080 077B>

b8	9	10	11	12	13	14	b15
0	0	0	0	0	0	0	RTDWRDIS 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–14	No function assigned. Fix to "0"		0	0
15	RTDWRDIS Write to RAM by RTD disable bit	0: Write to RAM by RTD enabled 1: Write to RAM by RTD disabled	R	W

This register is used to select whether to enable or disable a write to RAM by the RTD.
Setting the RTDWRDIS bit to 1 disables a write to RAM by the RTD, so that even when the RTD receives a command for write to RAM, the command is ignored and no write operation to RAM is executed.

Notes • Do not alter settings while using the RTD.

14.4 Functional Description of RTD

14.4.1 Outline of RTD Operation

Operation of the RTD is specified by a command entered from devices external to the chip. A command is indicated by bits 16–19 (Note 1) of the RTD received data.

Table 14.4.1 RTD Commands

RTD Received Data				Command	
b19	b18	b17	b16	Mnemonic	RTD Function
0	0	0	0	VER (VERify)	Continuous monitor
0	1	0	0		
0	1	0	1		
0	1	1	0	VEI (VErify Interrupt request)	RTD interrupt request
0	0	1	0	RDR (ReaD RAM)	Real-time RAM content output
0	0	1	1	WRR (WRite RAM)	RAM content forcible rewrite (with verify)
1	1	1	1	RCV (ReCoVer)	Recover from runaway condition (Note 2), (Note 3)
0	0	0	1	System reserved (use inhibited)	

↑ (Note 1)

Note 1: The RTD received data bit 19 actually is not stored in the command register, and except for the RCV command, handled as a “Don’t care” bit. (Bits 16–18 are effective for the command specified.)

Note 2: The RCV command must always be transmitted twice in succession.

Note 3: For the RCV command, all bits, not just 16–19, (i.e., bits 0–15 and bits 20–31) must be set to "1".

14.4.2 Operation of RDR (Real-time RAM Content Output)

When the RDR (real-time RAM content output) command is issued, the RTD is enabled to transfer the contents of the internal RAM to external devices without causing the CPU’s internal bus to stop. Because the RTD reads data from the internal RAM while there are no transfers performed between the CPU and internal RAM, no extra CPU load is incurred.

Only the 32-bit word-aligned addresses can be specified for read from the internal RAM. (The two low-order address bits specified by a command are ignored.) Data are read out and transferred from the internal RAM in 32-bit units.

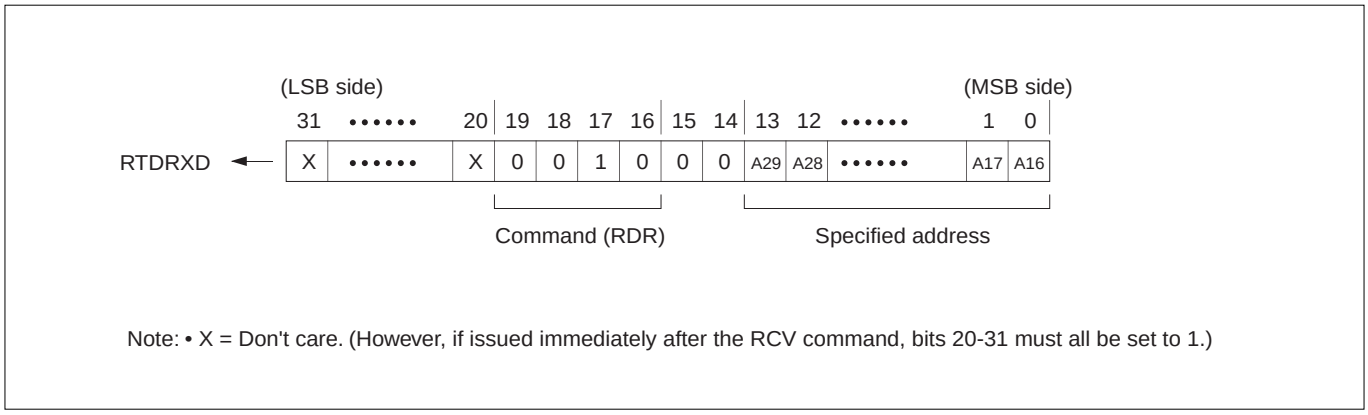
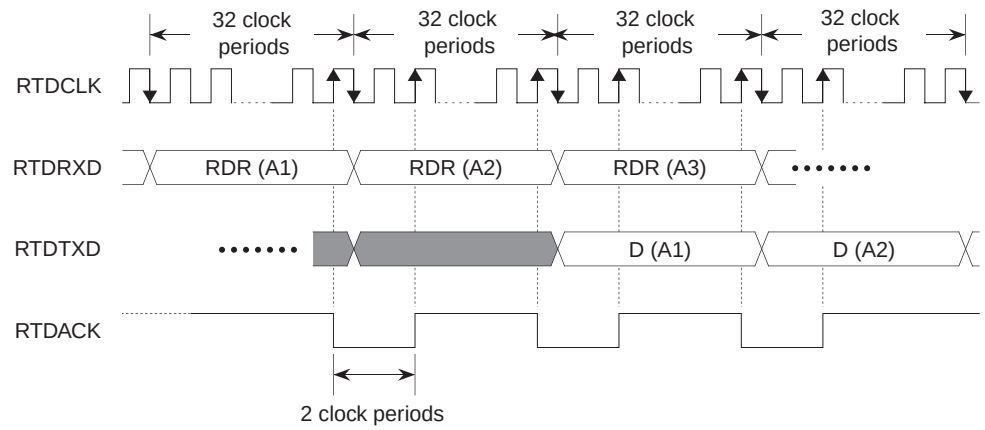
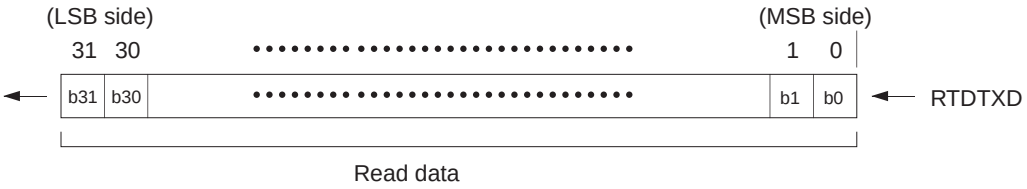


Figure 14.4.1 RDR Command Data Format



Note: • (An) = Specified address
• D(An) = Data at specified address (An)

Figure 14.4.2 Operation of RDR Command

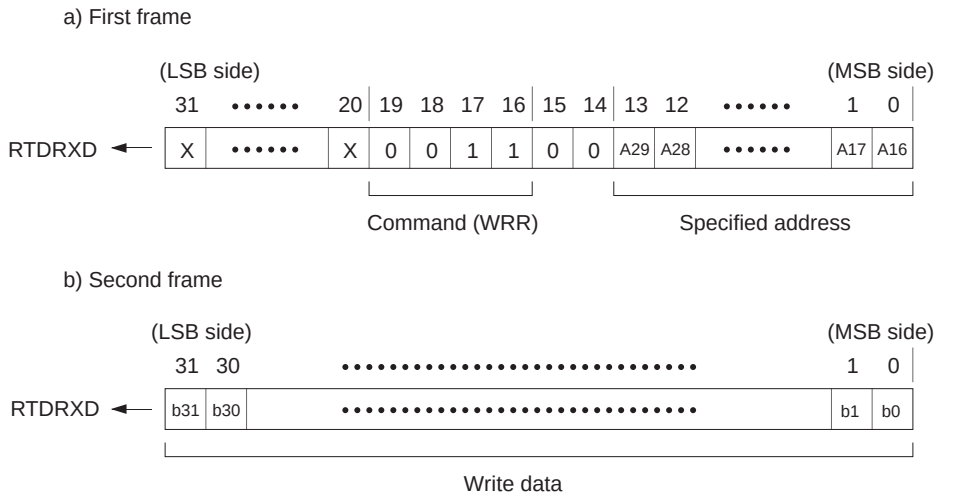


Note: • The read data is transferred LSB-first.

Figure 14.4.3 Read Data Transfer Format

14.4.3 Operation of WRR (RAM Content Forcible Rewrite)

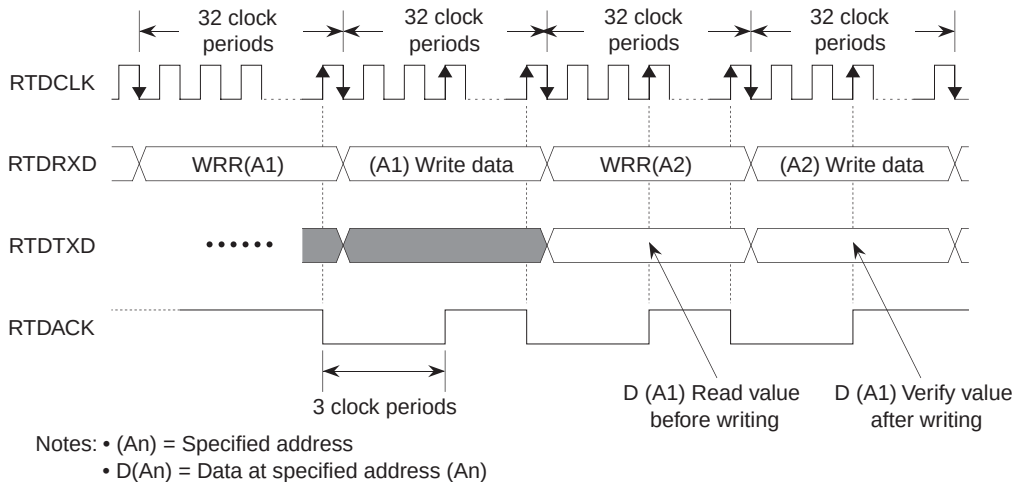
When the WRR (RAM content forcible rewrite) command is issued, the RTD forcibly rewrites the contents of the internal RAM without causing the CPU's internal bus to stop. Because the RTD writes data to the internal RAM while there are no transfers performed between the CPU and internal RAM, no extra CPU load is incurred. Only the 32-bit word-aligned addresses can be specified for read from the internal RAM. (The two low-order address bits specified by a command are ignored.) Data are written to the internal RAM in 32-bit units.
The external host should transmit the command and address in the first frame and then the write data in the second frame. The RTD writes to the internal RAM in the third frame after receiving the write data.



Notes: • X = Don't care. (However, if issued immediately after the RCV command, bits 20-31 must all be set to 1.)
• The specified address and write data are transferred LSB-first.

Figure 14.4.4 WRR Command Data Format

The RTD reads out data from the specified address before writing to the internal RAM and again reads out data from the same address immediately after writing to the internal RAM (this helps to verify the data written to the internal RAM). The read data is output at the timing shown below.

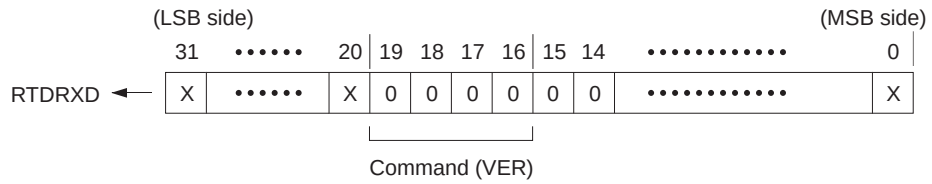


Notes: • (An) = Specified address
• D(An) = Data at specified address (An)

Figure 14.4.5 Operation of WRR Command

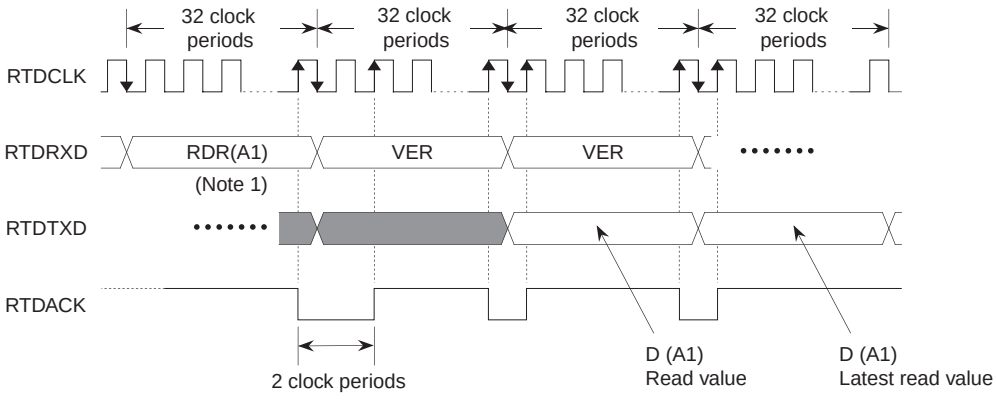
14.4.4 Operation of VER (Continuous Monitor)

When the VER (continuous monitor) command is issued, the RTD outputs the data from the address that has been accessed by an instruction (either read or write) immediately before receiving the VER command.



Notes: • X = Don't care.
(However, if issued immediately after the RCV command, bits 20-31 must all be set to 1.)

Figure 14.4.6 VER (Continuous Monitor) Command Data Format

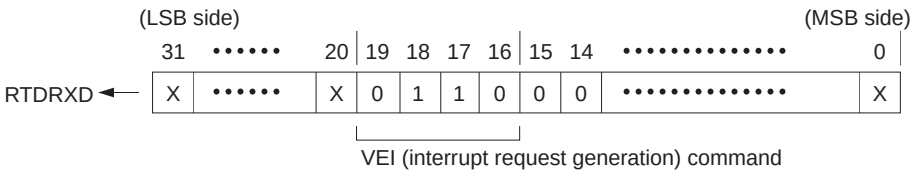


Note 1: WRR command can also be used.
Notes: • (An) = Specified address
• D(An) = Data at specified address (An)

Figure 14.4.7 Operation of VER (Continuous Monitor) Command

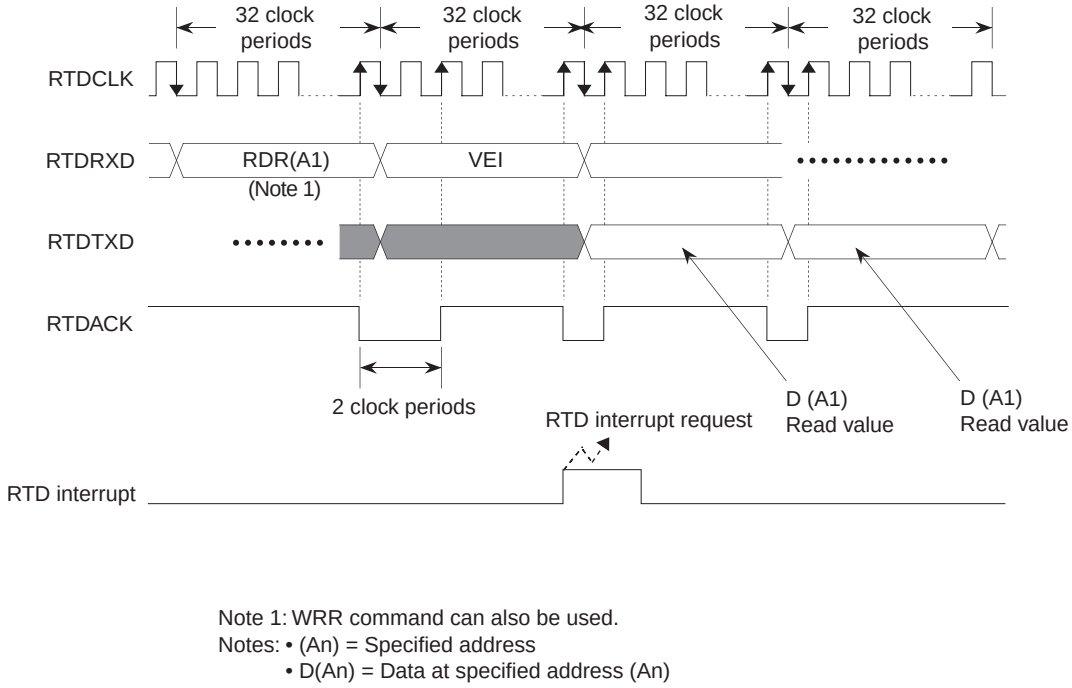
14.4.5 Operation of VEI (Interrupt Request)

When the VEI (interrupt request) command is issued, an RTD interrupt request is generated. Furthermore, the RTD outputs the data from the address that has been accessed by an instruction (either read or write) immediately before receiving the VEI command.



Note: • X = Don't care.
(However, if issued immediately after the VEI command, bits 20-31 must all be set to 1.)

Figure 14.4.8 VEI (Interrupt Request) Command Data Format

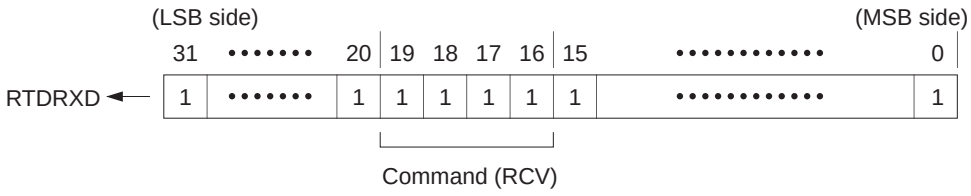


Note 1: WRR command can also be used.
Notes: • (An) = Specified address
• D(An) = Data at specified address (An)

Figure 14.4.9 Operation of VEI (Interrupt Request) Command

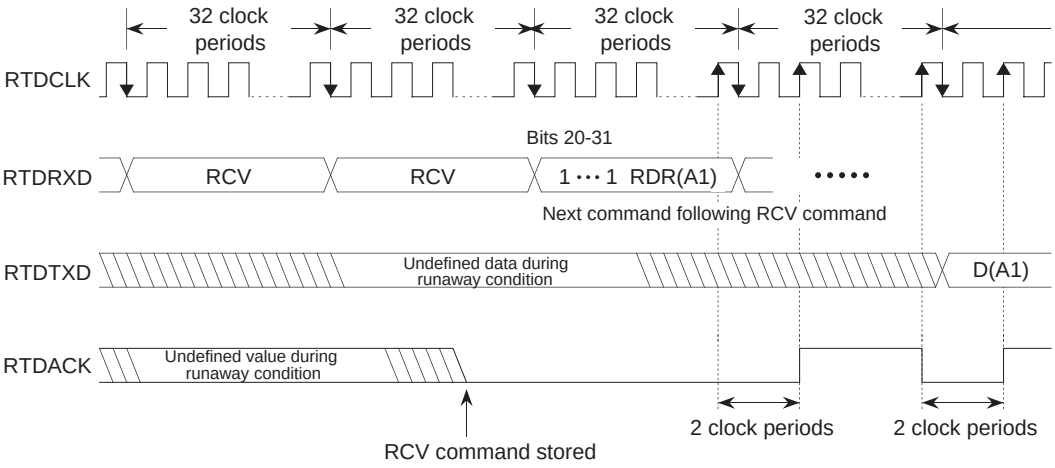
14.4.6 Operation of RCV (Recover from Runaway)

If the RTD runs out of control, the RCV (recover from runaway) command may be issued to recover from the runaway condition without the need to reset the system. The RCV command must always be issued twice in succession. Also, any command issued immediately following the RCV command must have all of its bits 20–31 set to 1.



Notes: • All of 32 data bits are 1's.
• The RCV command must always be issued twice in succession.

Figure 14.4.10 RCV Command Data Format



Note: • The next command following the RCV command must have all of its bits 20-31 set to 1.

Figure 14.4.11 Operation of RCV Command

14.4.7 Method for Setting a Specified Address when Using RTD

In the Real-Time Debugger (RTD), the low-order 16-bit addresses of the internal RAM can be specified. Because the internal RAM is located in a 24-KB area ranging from H'0080 4000 to H'0080 9FFF, the low-order 16-bit address of that area (H'4000 to H'FFFF) can be set. However, to access any area other than RAM is inhibited.

Note also that two least significant address bits, A31 and A30, area always 0 because data are read and written to and from the internal RAM in a fixed length of 32 bits.

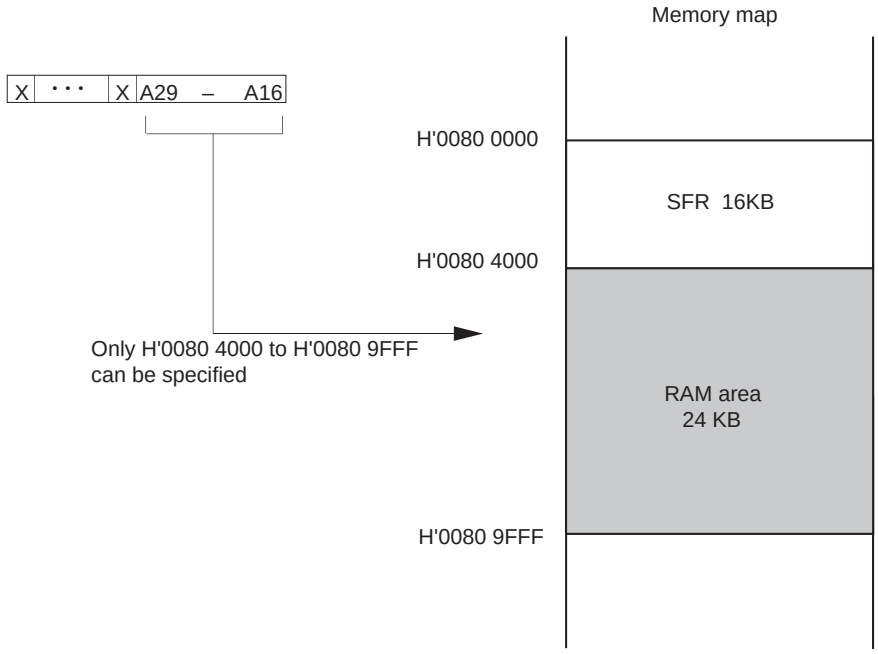


Figure 14.4.12 Setting Addresses in the Real-Time Debugger

14.4.8 Resetting RTD

The RTD is reset by applying a system reset (i.e., RESET# signal input). The status of the RTD related output pins after a system reset are shown below.

Table 14.4.2 RTD Pin Status after System Reset

Pin Name	Status
RTDACK	"H" level output
RTDTXD	"H" level output

The first command transfer to the RTD after being reset is initiated by transferring data to the RTDRXD pin synchronously with the falling edge of RTDCLK.

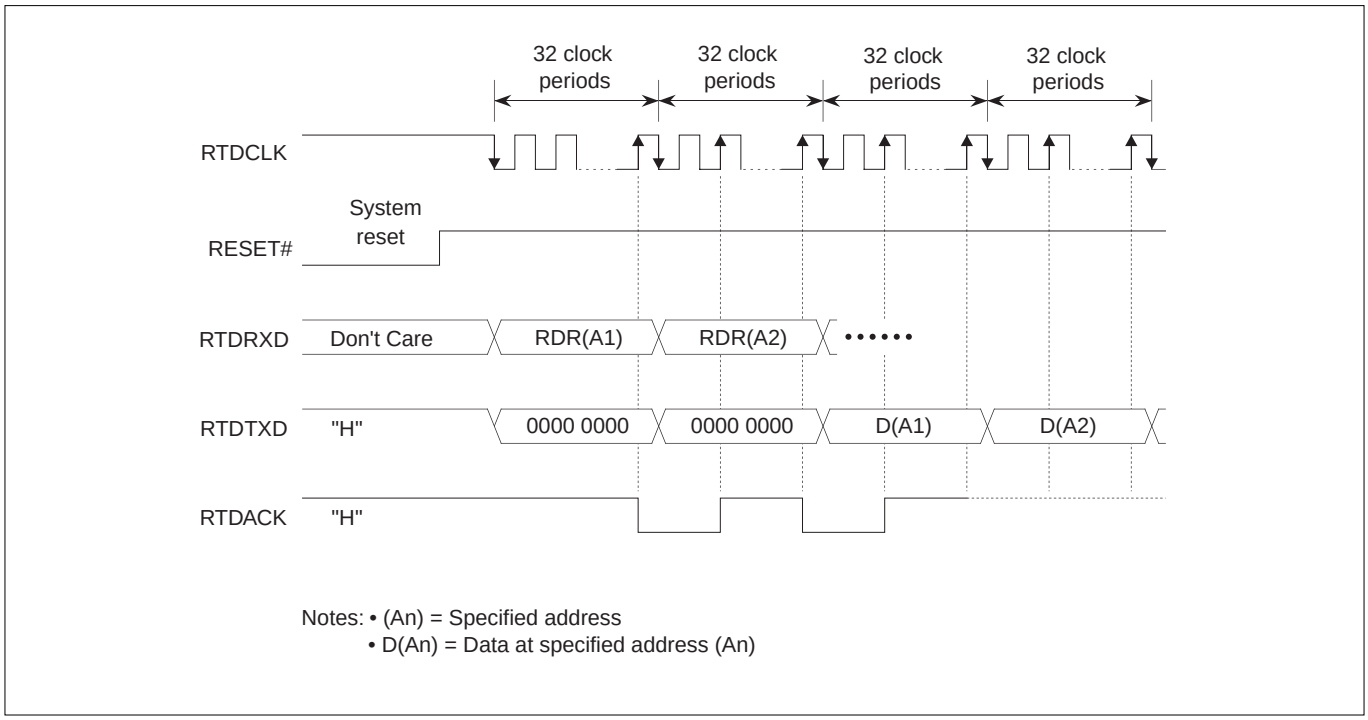
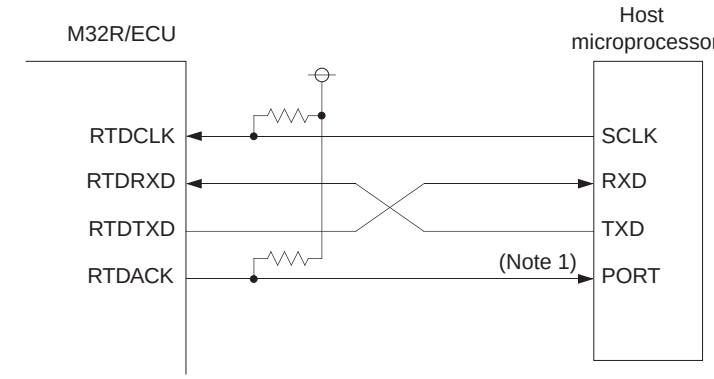


Figure 14.4.13 Command Transfer to RTD after System Reset

14.5 Typical Connection with the Host

The host uses a serial synchronous interface to transfer data. The clock for synchronous communication should be generated by the host. An example for connecting the RTD and host is shown below.



Note 1: This applies to the case where the RTDACK level is checked between transfer frames.

Figure 14.5.1 Connecting the RTD and Host

The RTD communication is performed in a fixed length of 32 bits per frame. Because serial interfaces generally handle data in 8-bit units, data is transferred separately in four operations, 8 bits at a time. The RTDACK signal is used to verify that communication is performed normally.

The RTDACK signal goes "L" after a command is sent, providing a means of verifying the communication status. When issuing the VER command, the RTDACK signal is pulled "L" for only one clock period. Therefore, after sending 32 bits in one frame via a serial interface, turn off RTDCLK output and check that RTDACK is "L." That way, it is possible to know whether the RTD is communicating normally.

If it is desirable to identify the type of transmitted command by the width of RTDACK, use the microcomputer's internal measurement timer (to count RTDCLK pulses while RTDACK is "L"), or design a dedicated circuit.

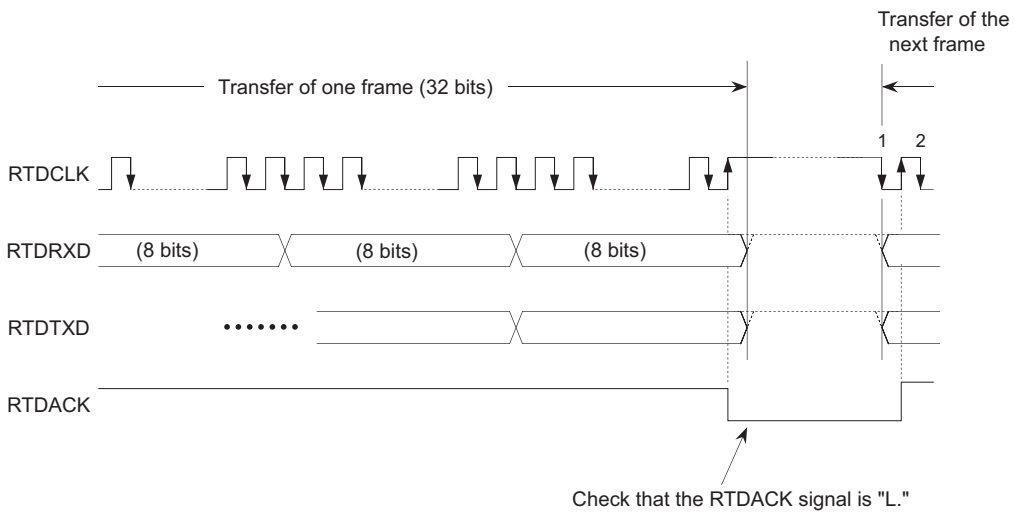


Figure 14.5.2 Example of Communication with the Host (when Using VER Command)

This page is blank for reasons of layout.

CHAPTER 15

EXTERNAL BUS INTERFACE

15.1 External Bus Interface Related Signals

The 32176 has the external bus interface related signals described below. These signals can be used in external extension and processor modes.
The symbol “#” suffixed to the signal names (or pin names) means that the signals (or pins) are active "L."

(1) Address

The 32176 outputs a 19-bit address (A12–A30) for addressing any location in a 1-Mbyte space. The least significant A31 is not output.

(2) Chip select (CS0#, CS1#)

The CS0# and CS1# signals are output for external extension areas divided in 2-Mbyte units. The CS0# signal points to a 2-Mbyte area during processor mode or a 1-Mbyte area during external extension mode. (For details, see Chapter 3, “Address Space.”)

(3) Read strobe (RD#)

Output during an external read cycle, this signal indicates the timing at which to read data from the bus. This signal is driven "H" when writing to the bus or accessing the internal area.

(4) Byte High Write/Byte High Enable (BHW#/BHE#)

The pin function changes depending on the Bus Mode Control Register (BUSMODC).
When BUSMOD = "0" and this signal is Byte High Write (BHW#), during external write access it indicates that the upper byte (DB0–DB7) of the data bus is the valid data transferred. During external read and when accessing the internal area it outputs a "H."
When BUSMOD = "1" and this signal is Byte High Enable (BHE#), during external access (for read or write) it indicates that the upper byte (DB0–DB7) of the data bus is the valid data transferred. When accessing the internal area it outputs a "H."

(5) Byte Low Write/Byte Low Enable (BLW#/BLE#)

The pin function changes depending on the Bus Mode Control Register (BUSMODC).
When BUSMOD = "0" and this signal is Byte Low Write (BLW#), during external write access it indicates that the lower byte (DB8–DB15) of the data bus is the valid data transferred. During external read and when accessing the internal area it outputs a "H."
When BUSMOD = "1" and this signal is Byte Low Enable (BLE#), during external access (for read or write) it indicates that the lower byte (DB8–DB15) of the data bus is the valid data transferred. When accessing the internal area it outputs a "H."

(6) Data bus (DB0–DB15)

This is the 16-bit data bus used to access external devices. During external read access, data is latched from the bus synchronously with the rising edge of the read strobe. Even during 8-bit read, the microcomputer always reads in 16 bits of data, with only the valid byte part of data transferred into the internal circuit. During external write access, data is output from the bus. During 8-bit write, the microcomputer outputs the valid byte part of data to be written as BHW#/BLW#. When accessing the internal area, the bus functions as an input bus.

(7) System clock/write (BCLK/WR#)

The pin function changes depending on the Bus Mode Control Register (BUSMODC).
When BUSMOD = "0" and this signal is System Clock (BCLK), it outputs the system clock necessary to synchronize operations in an external system. When the CPU clock = 40 MHz, a 20 MHz clock is output from BCLK. When not using the BCLK/WR function, this pin can be used as P70 by clearing the P7 Operation Mode Register P70MOD bit to "0".
When BUSMOD = "1" and this signal is Write (WR#), during external write access it indicates the valid data transferred on the data bus. During external read cycle and when accessing the internal area it outputs a "H."

(8) Wait (WAIT#)

When the 32176 started an external bus cycle, it automatically inserts wait states while the WAIT# input signal is asserted. For details, see Chapter 16, “Wait Controller.” When not using the WAIT function, this pin can be used as P71 by clearing the P7 Operation Mode Register P71MOD bit to "0".
For external access, one or more wait cycles always need to be inserted. Therefore, the shortest possible access to an external device is equal to one wait cycle (2 BCLK periods).

(9) Hold control (HREQ#, HACK#)

The hold state means that internal bus and external bus stop accessing the bus and the bus interface related pins are tristated (high impedance). While the microcomputer is in a hold state, any bus master external to the chip can use the system bus to transfer data. Even during hold status the command in which command que is done though, if the command with access to bus is done, the command performance operation is stopped at that time.
A "L" signal input on the HREQ# pin places the microcomputer into a hold state. While the microcomputer remains in a hold state after accepting the hold request and during a transition to the hold state, the HACK# pin outputs a "L" level signal. To exit the hold state and return to normal operating state, release the HREQ# signal back "H." Furthermore, when not using the HREQ and HACK functions, these pins can be used as P72 and P73 by clearing P72MOD and P73MOD in the P7 Operation Mode Register to 0.
The status of each pin during hold are shown below.

Table 15.1.1 Pin State during Hold Period

Pin Name	Pin State or Operation
A12–A30, DB0–DB15, CS0#, CS1#, RD#, BHW#, BLW#, BHE#, BLE#, WR#	High impedance
HACK#	Output a "L"
Other pins (e.g., ports and timer output)	Normal operation

15.2 External Bus Interface Related Registers

The following describes the external bus interface related registers.

15.2.1 Port Operation Mode Register

Ports P70–P73 can be switched for external access signal pins at any time irrespective of the CPU operation mode.

P7 Operation Mode Register (P7MOD) <Address: H'0080 0747>

b8	9	10	11	12	13	14	b15
P70MOD	P71MOD	P72MOD	P73MOD	P74MOD	P75MOD	P76MOD	P77MOD
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	P70MOD Port P70 operation mode bit (Note 2)	0: P70 1: BCLK/WR#	R	W
9	P71MOD Port P71 operation mode bit	0: P71 1: WAIT#	R	W
10	P72MOD Port P72 operation mode bit	0: P72 1: HREQ#	R	W
11	P73MOD Port P73 operation mode bit	0: P73 1: HACK#	R	W
12	P74MOD Port P74 operation mode bit	0: P74 1: RTDTXD/TXD3 (Note 1)	R	W
13	P75MOD Port P75 operation mode bit	0: P75 1: RTDRXD/RXD3 (Note 1)	R	W
14	P76MOD Port P76 operation mode bit	0: P76 1: RTDACK/CTX1 (Note 1)	R	W
15	P77MOD Port P77 operation mode bit	0: P77 1: RTDCLK/CRX1 (Note 1)	R	W

Note 1: Either of the functions is selected using the P7 Peripheral Function Select Register.

Note 2: During external extension mode, when BUSMOD bit of the BUSMODC register is set to 1 (Byte enable separate mode), P70/BCLK/WR# pin functions to output WR# signal regardless of the settings for the P7 Operation Mode Register.

15.2.2 Bus Mode Control Register

Bus Mode Control Register (BUSMODC) <Address: H'0080 077F>

b8	9	10	11	12	13	14	b15
0	0	0	0	0	0	0	BUSMOD 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–14	No function assigned. Fix to "0"		0	0
15	BUSMOD Bus mode control bit (Note 1)	0: WR signal separate mode 1: Byte enable separate mode	R	W

Note 1: During external extension mode, when BUSMOD bit of the BUSMODC register is set to 1 (Byte enable separate mode), P70/BCLK/WR# pin functions to output WR# signal regardless of the settings for the P7 Operation Mode Register.

This register is used to facilitate memory connections during processor mode and external extension mode.

When the Bus Mode Control bit (BUSMOD) = "0", the WR# signal is output separately for each byte area. Signals RD#, BHW#, BLW#, BCLK# and WAIT# can be used. For memory connection in boot mode, the Bus Mode Control Register has no effect, and the microcomputer operates in the same way as when the Bus Mode Control bit (BUSMOD) is cleared to "0".

When the Bus Mode Control bit (BUSMOD) = "1", the byte enable signal is output separately for each byte area. Signals RD#, BHE#, BLE#, WR# and WAIT# can be used. In a WAIT control circuit configuration, because BCLK output is not available, timing must be controlled external to the chip.

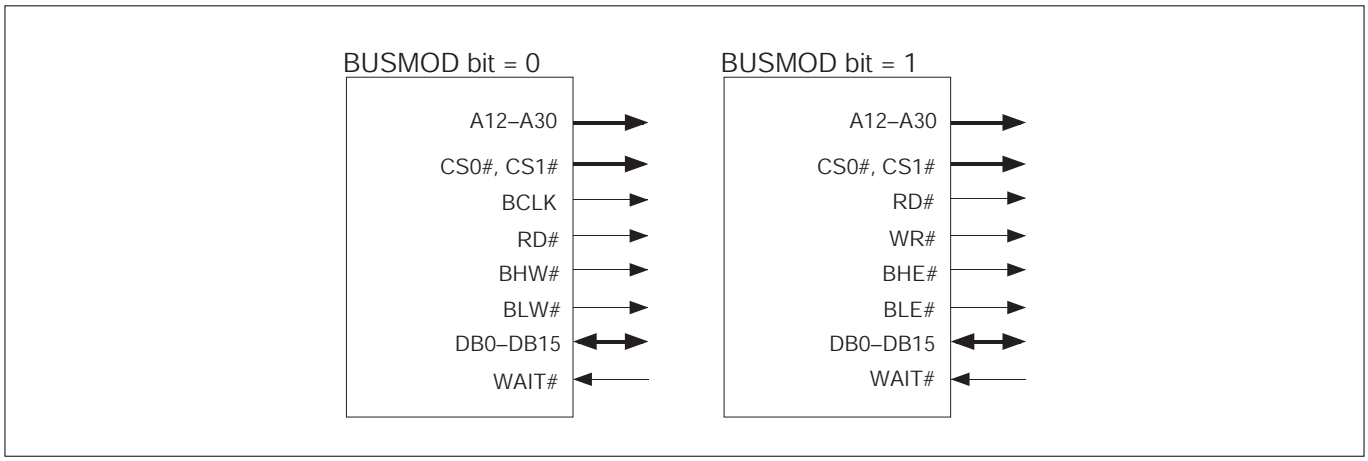


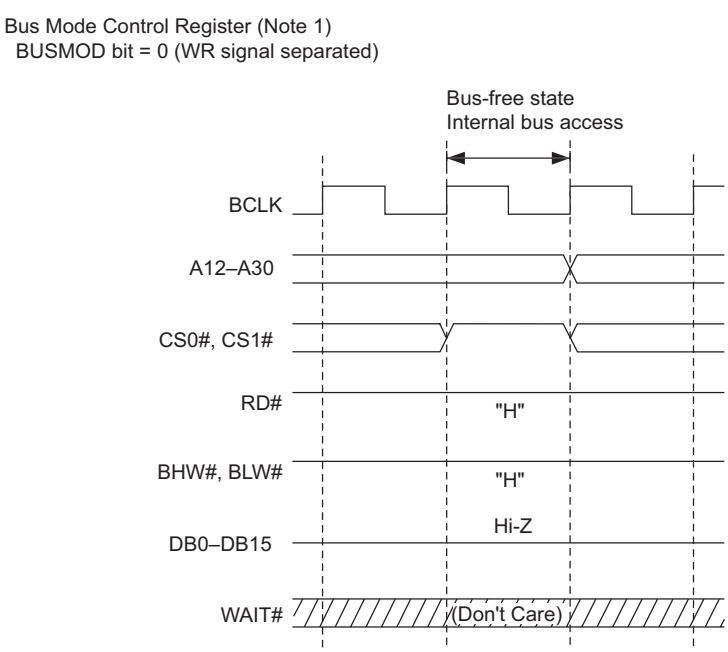
Figure 15.2.1 Pin Functions when External Bus Modes are Changed

15.3 Read/Write Operations

(1) When the Bus Mode Control Register is set to "0"

External read/write operations are performed using the address bus, data bus and the signals CS0#,CS1#, RD#, BHW#, BLW#, WAIT# and BCLK. In the external read cycle, the RD# signal is "L" while BHW# and BLW# both are "H", with data read in from only the necessary byte position. In the external write cycle, the BHW# or BLW# signal output for the byte position to write is asserted "L" as data is written to the bus.

When an external bus cycle starts, wait states are inserted as long as the WAIT# signal is "L." Unless necessary, the WAIT# signal must always be held "H." One wait cycle always need to be inserted even for the shortest external access. (The shortest possible bus cycle is 2 BCLK periods).

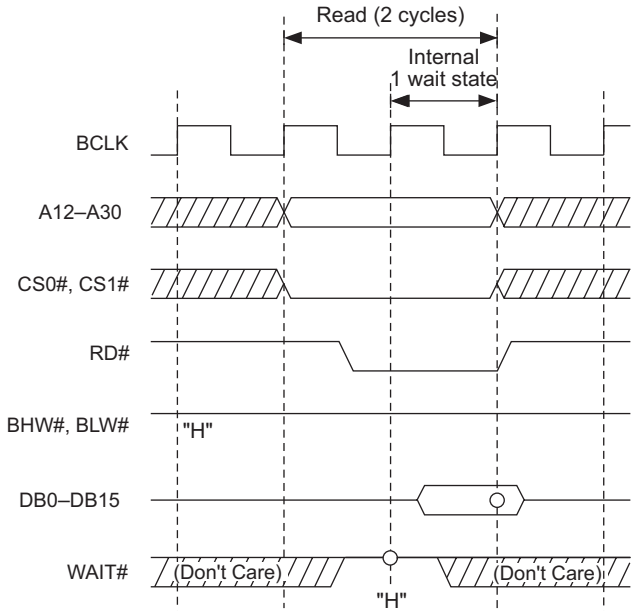


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note: • Hi-Z denotes a high-impedance state.

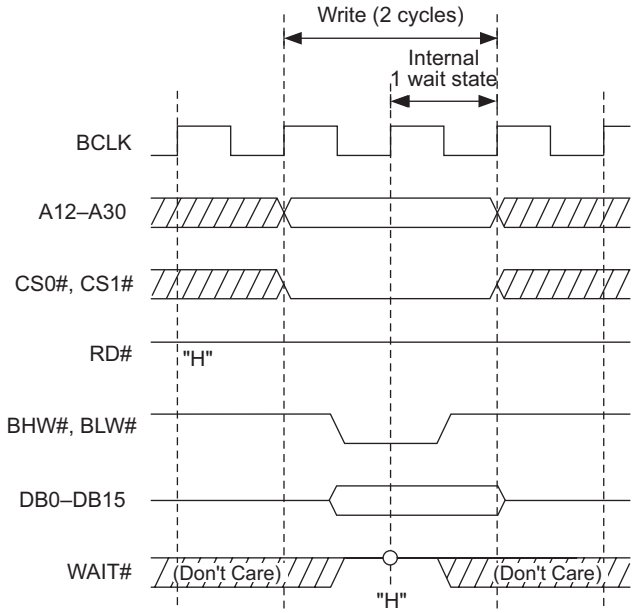
Figure 15.3.1 Internal Bus Access during Bus Free State

Bus Mode Control Register (Note 1)
BUSMOD bit = 0 (WR signal separated)

Read



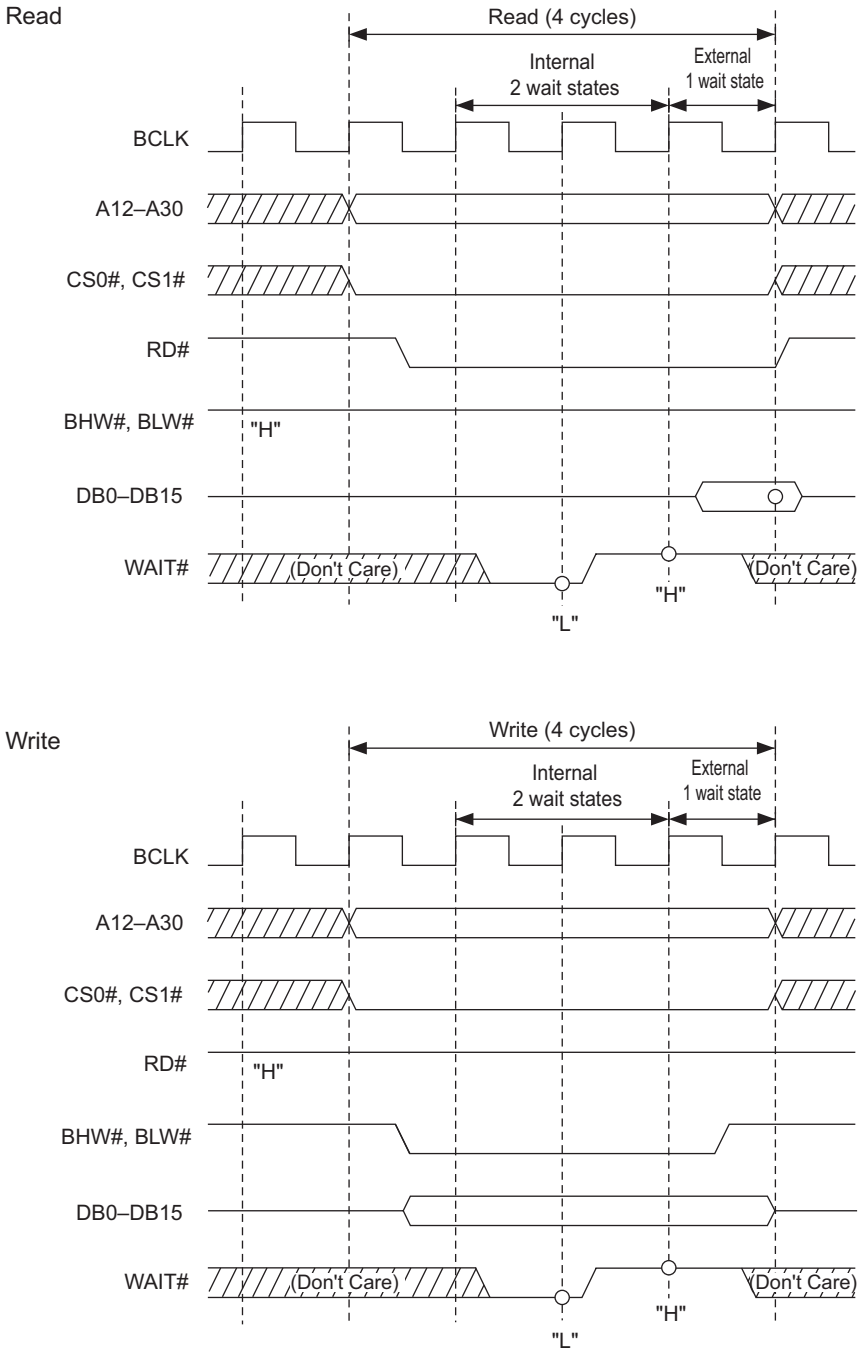
Write



Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note: • Circles in the above diagram denote the sampling timing.

Figure 15.3.2 Read/Write Timing (for Shortest External Access)

Bus Mode Control Register (Note 1)
BUSMOD bit = 0 (WR signal separated)



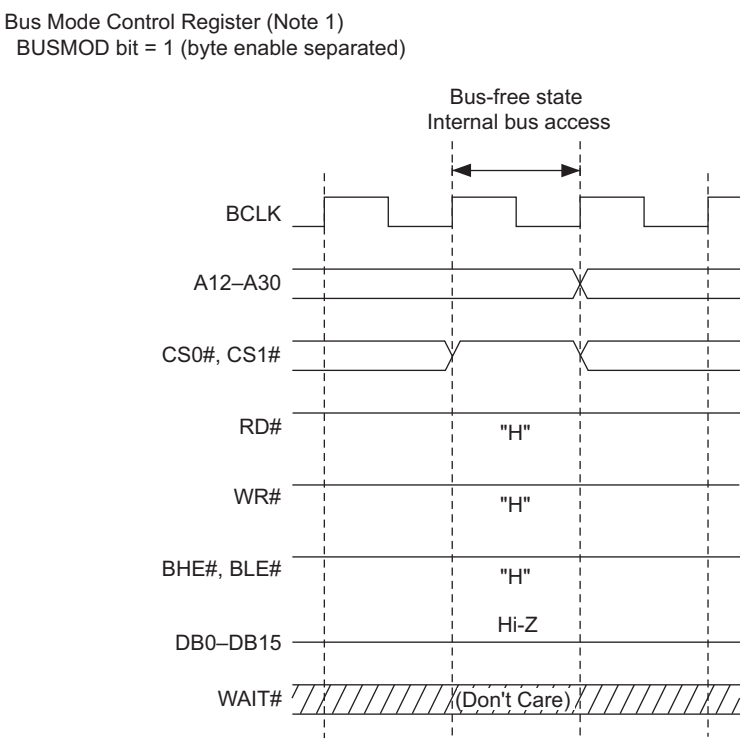
Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note: • Circles in the above diagram denote the sampling timing.

Figure 15.3.3 Read/Write Timing (for Access with Internal 2 and External 1 Wait States)

(2) When the Bus Mode Control Register is set to "1"

External read/write operations are performed using the address bus, data bus and the signals CS0#, CS1#, RD#, BHE#, BLE#, WAIT# and WR#. In the external read cycle, the RD# signal is "L" and the BHE# or BLE# signal output for the byte position from which to read is asserted "L", with data read in from only the necessary byte position of the bus. In the external write cycle, the WR# signal goes "L" and the BHE# or BLE# signal output for the byte position to write is asserted "L", with data written to the necessary byte position.

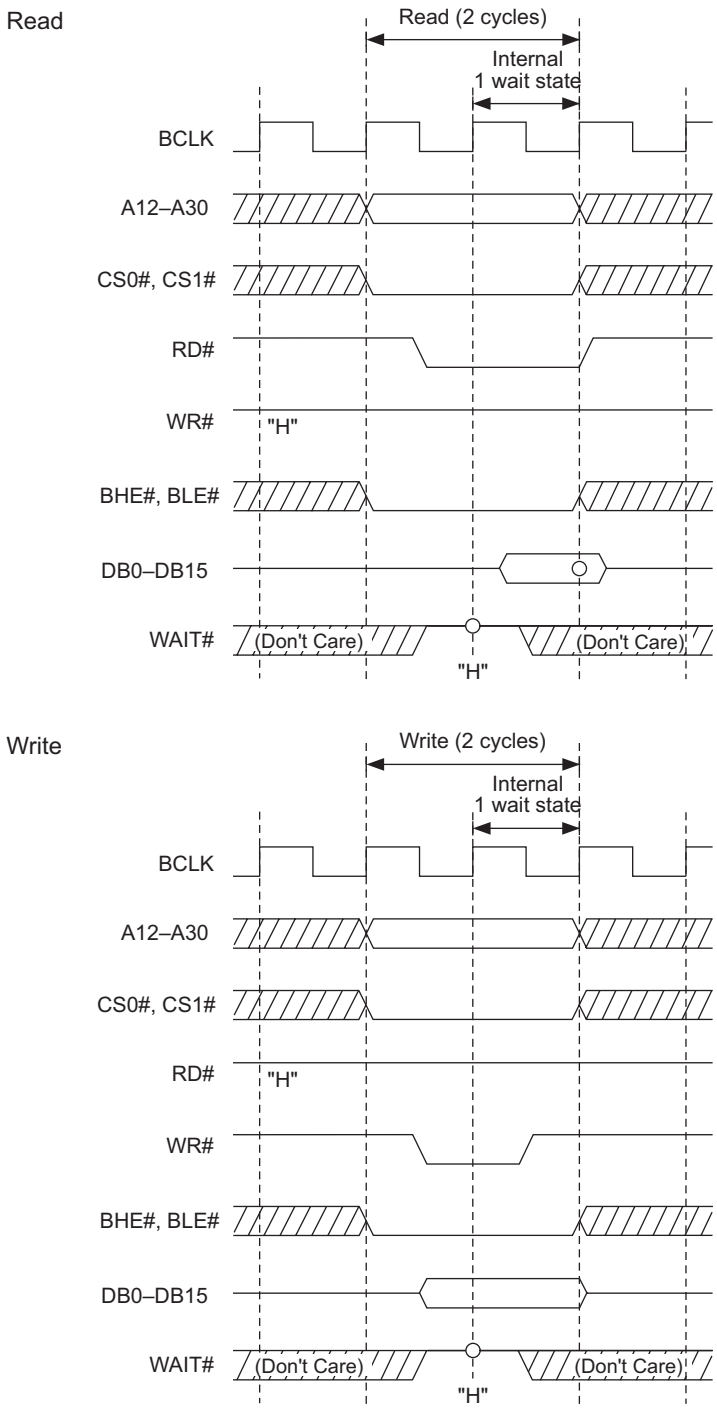
When an external bus cycle starts, wait states are inserted as long as the WAIT# signal is "L." Unless necessary, the WAIT# signal must always be held "H." One wait cycle always need to be inserted even for the shortest external access. (The shortest possible bus cycle is 2 BCLK periods). When not using the WAIT function, this pin can be used as P71 by clearing the P7 Operation Mode Register P71MOD bit to "0".



Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Notes: • Hi-Z denotes a high-impedance state.
• BCLK is not output.

Figure 15.3.4 Internal Bus Access during Bus Free State

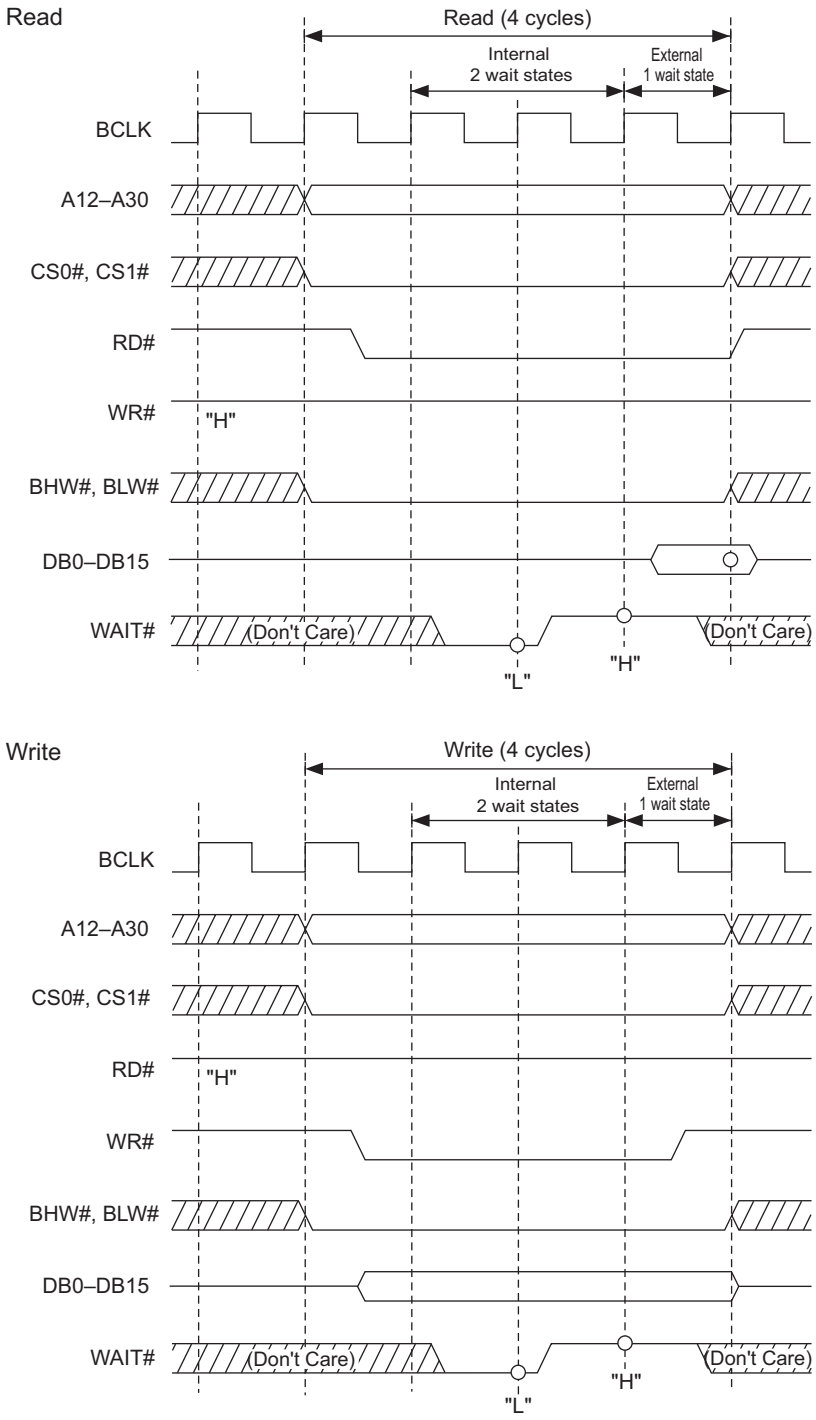
Bus Mode Control Register (Note 1)
BUSMOD bit = 1 (byte enable separated)



Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Notes: • Circles in the above diagram denote the sampling timing.
• BCLK is not output.

Figure 15.3.5 Read/Write Timing (for Shortest External Access)

Bus Mode Control Register (Note 1)
BUSMOD bit = 1 (byte enable separated)



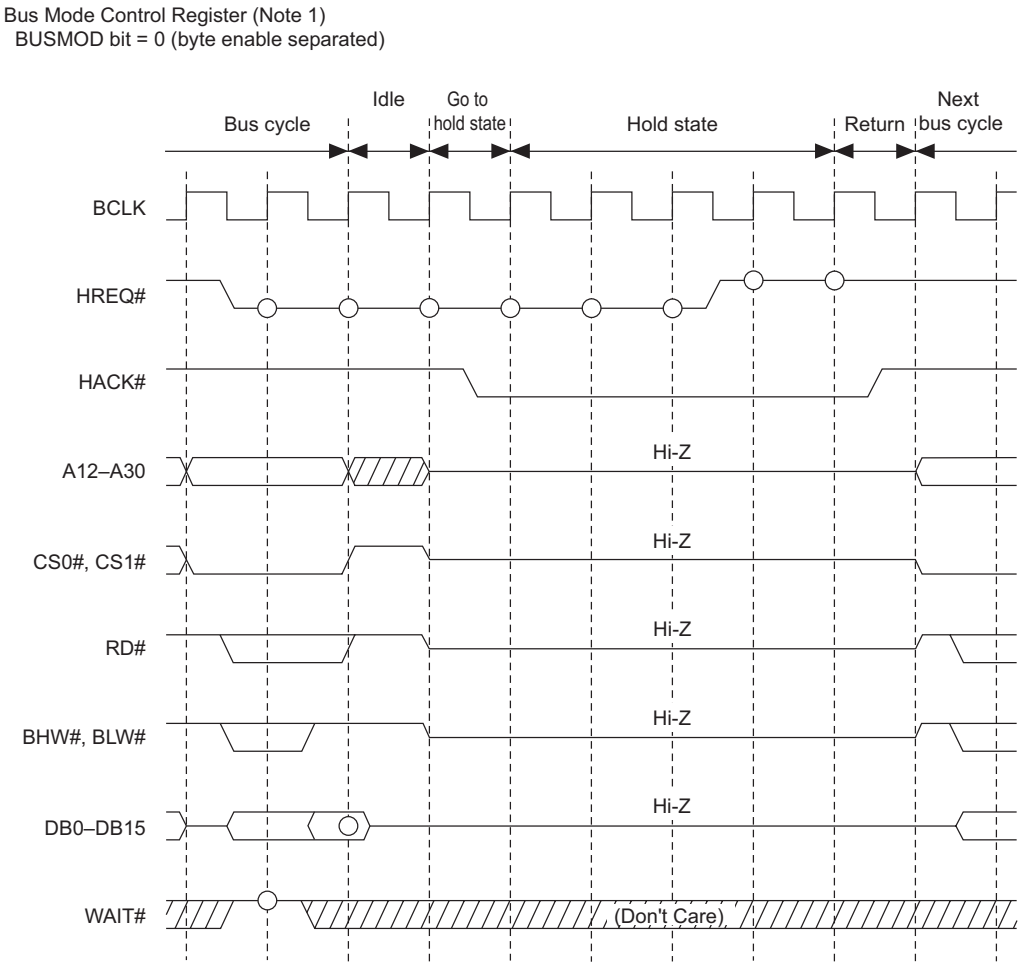
Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Notes: • Circles in the above diagram denote the sampling timing.
• BCLK is not output.

Figure 15.3.6 Read/Write Timing (for Access with Internal 2 and External 1 Wait States)

15.4 Bus Arbitration

(1) When the Bus Mode Control Register is set to "0"

When the input signal on the HREQ# pin is pulled "L" and the hold request is accepted, the microcomputer goes to a hold state and outputs a "L" from the HACK# pin. During hold state, all bus related pins are placed in the high-impedance state, allowing data to be transferred on the system bus. To exit the hold state and return to normal operating state, release the HREQ# signal back "H."

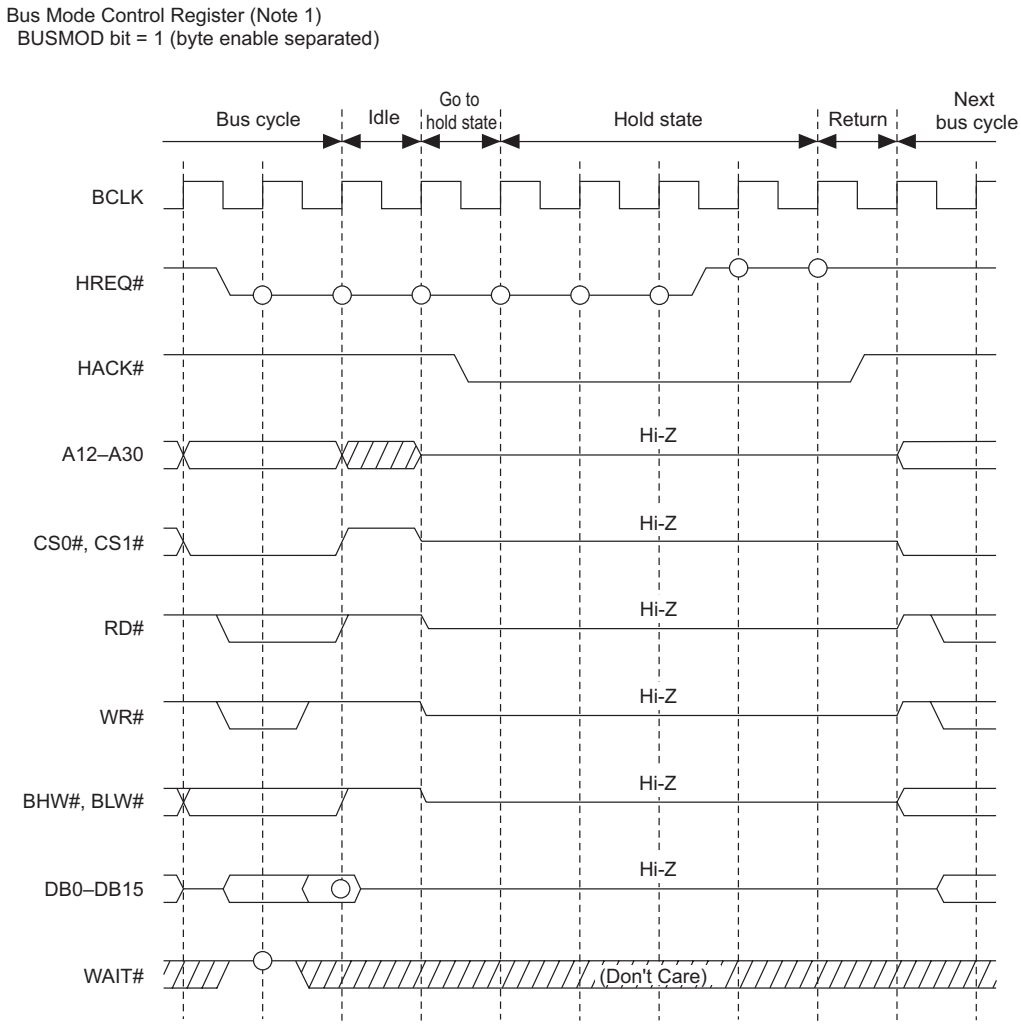


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Notes: • Circles in the above diagram denote the sampling timing.
• Hi-Z denotes a high-impedance state.
• Idle cycles are inserted only when a hold state is entered immediately following an external read access.
• The number of cycles before a state transition to idle (recovery) of hold state occurs after input to the HREQ# pin is pulled "L" differs depending on the status of the bus access being executed then.

Figure 15.4.1 Bus Arbitration Timing

(2) When the Bus Mode Control Register is set to "1"

When the input signal on the HREQ# pin is pulled "L" and the hold request is accepted, the microcomputer goes to a hold state and outputs a "L" from the HACK# pin. During hold state, all bus related pins are placed in the high-impedance state, allowing data to be transferred on the system bus. To exit the hold state and return to normal operating state, release the HREQ# signal back "H."



Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Notes: • Circles in the above diagram denote the sampling timing.
• Hi-Z denotes a high-impedance state.
• Idle cycles are inserted only when a hold state is entered immediately following an external read access.
• The number of cycles before a state transition to idle (recovery) of hold state occurs after input to the HREQ# pin is pulled "L" differs depending on the status of the bus access being executed then.

Figure 15.4.2 Bus Arbitration Timing

15.5 Typical Connection of External Extension Memory

(1) When the Bus Mode Control Register is set to "0"

A typical memory connection when using external extension memory is shown in Figure 15.5.1. (External extension memory can only be used in external extension mode and processor mode.)

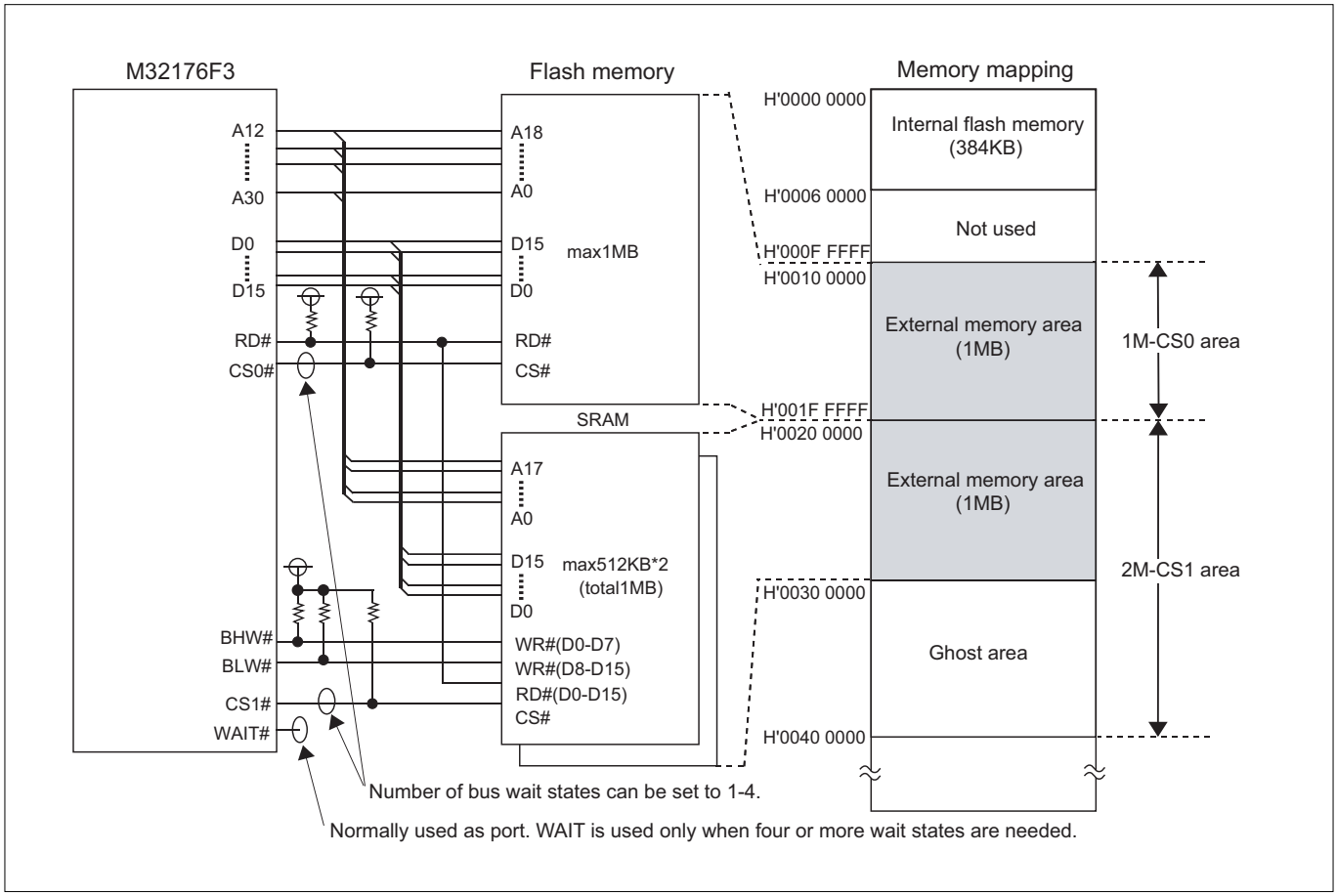


Figure 15.5.1 Typical Connection of External Extension Memory (when BUSMOD bit = "0")

Note: • The address and data are connected in such a way that pin 0 is the MSB and pin 15 is the LSB. When connecting external extension memory, connections of the MSB and LSB sides must be reversed.

(2) When the Bus Mode Control Register is set to "1"

A typical memory connection when using external extension memory is shown in Figure 15.5.2. (External extension memory can only be used in external extension mode and processor mode.)

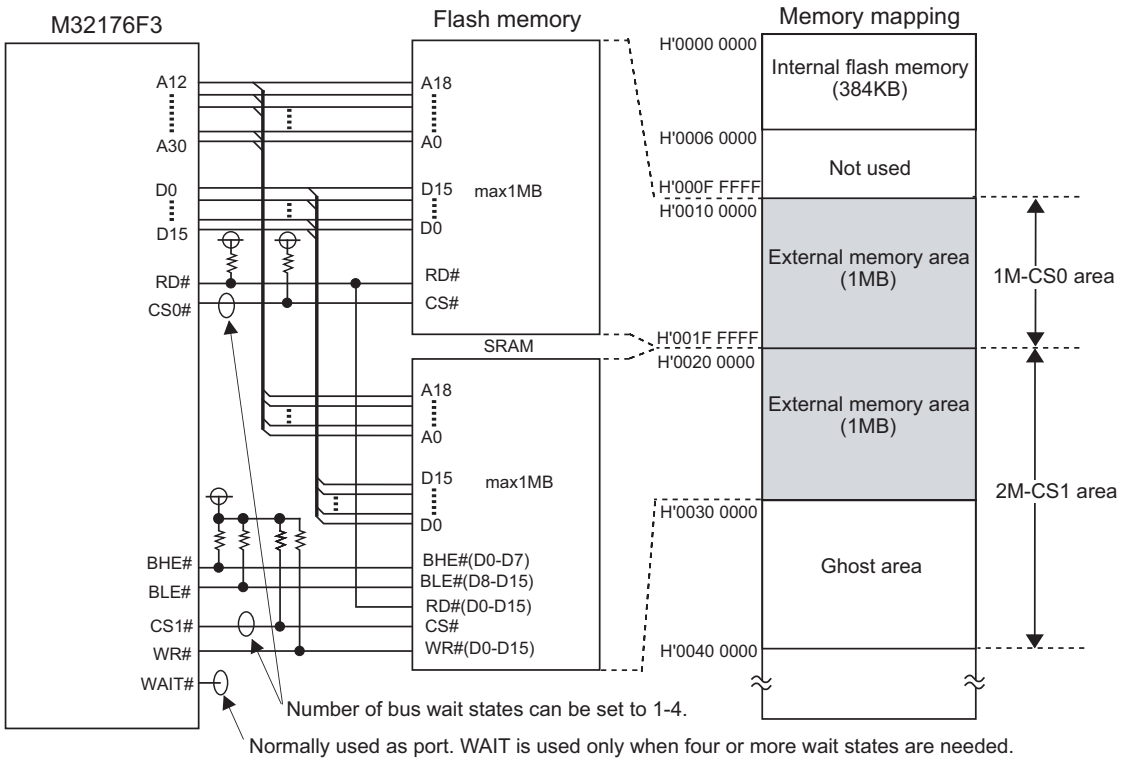


Figure 15.5.2 Typical Connection of External Extension Memory (when BUSMOD bit = "1")

Note: • The address and data are connected in such a way that pin 0 is the MSB and pin 15 is the LSB. When connecting external extension memory, connections of the MSB and LSB sides must be reversed.

(3) When the Bus Mode Control Register is set to "1" using a combination of 8/16-bit data bus memories

The diagram below shows a typical connection of external extension memory, with an 8-bit data bus memory located in the CS0 area, and a 16-bit data bus memory located in the CS1 area. (External extension memory can only be used in external extension mode and processor mode.)

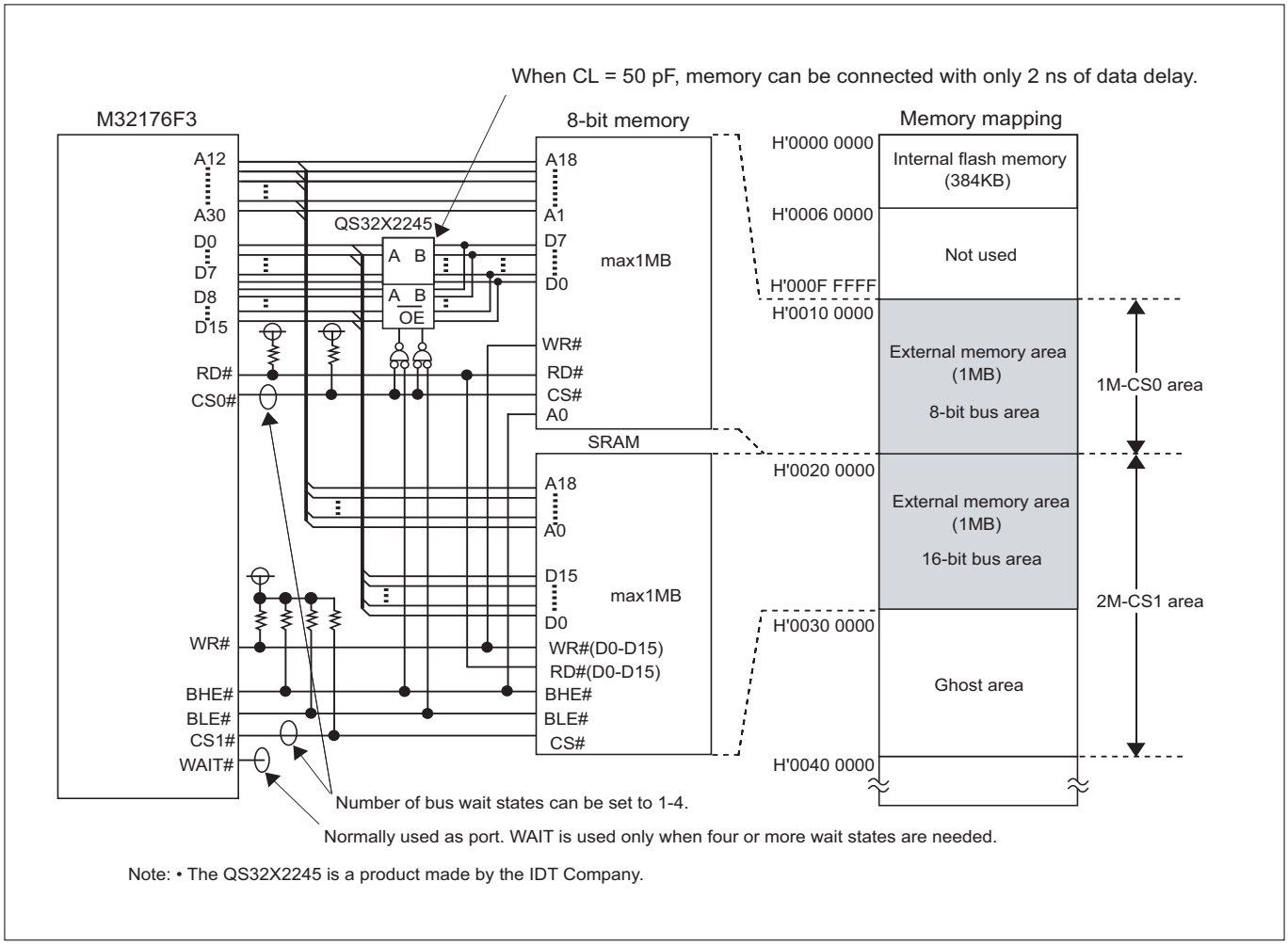


Figure 15.5.3 Typical Connection of External Extension Memory (when BUSMOD bit = "1" using a combination of 8/16-Bit Memories)

Note: • The address and data are connected in such a way that pin 0 is the MSB and pin 15 is the LSB. When connecting external extension memory, connections of the MSB and LSB sides must be reversed.

CHAPTER 16

WAIT CONTROLLER

16.1 Outline of the Wait Controller

The Wait Controller controls the number of wait states inserted in bus cycles when accessing an external extension area. The Wait Controller is outlined in the table below.

Table 16.1.1 Outline of the Wait Controller

Item	Description
Target space	Control is applied to the following address spaces depending on operation mode: Single-chip mode: No target space (Settings of the Wait Controller have no effect) External extension mode: CS0 area (1 Mbyte), CS1 area (1 Mbyte), Processor mode: CS0 area (1 Mbyte), CS1 area (1 Mbyte)
Number of wait states that can be inserted	1–4 wait states set by software + any number of wait states set from the WAIT# pin (The shortest possible bus cycle during external access is equal to one wait cycle inserted.)

During external extension and processor modes, two chip select signals (CS0#, CS1#) are output, each corresponding to one of the two external extension areas referred to as CS0 and CS1.

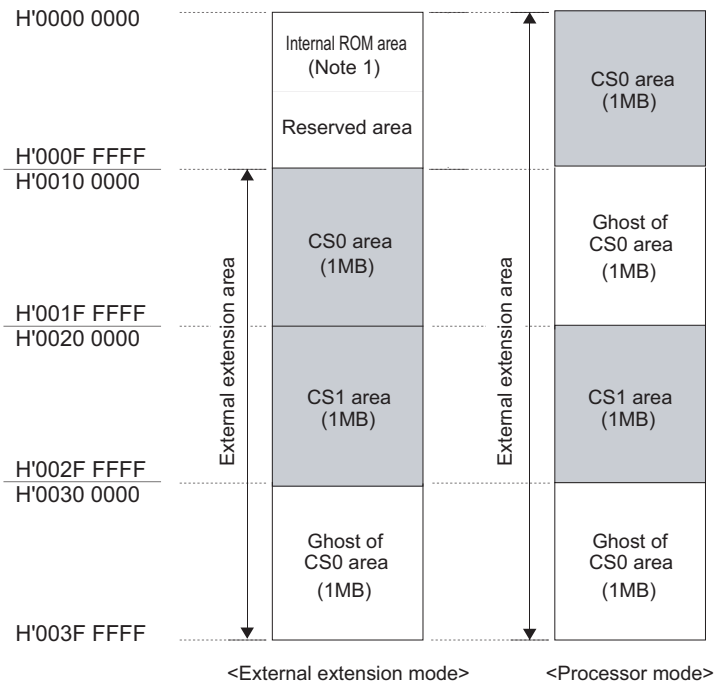


Figure 16.1.1 CS0 and CS1 Area Address Map

When accessing the external extension area, the Wait Controller controls the number of wait states inserted in bus cycles based on the number of wait states set by software and those entered from the WAIT# pin.

The number of wait states that can be controlled in software is 1 to 4. (The shortest possible bus cycle during external access is equal to one wait cycle inserted.)

When the input signal on the WAIT# pin is sampled "L" in the last cycle of internal wait state, the wait state is extended as long as the WAIT# input signal is held "L." Then when the WAIT# input signal is released back "H", the wait state is terminated and the next new bus cycle is entered into.

Table 16.1.2 Number of Wait States that Can Be Set by the Wait Controller

External Extension Area	Address	Number of Wait States Inserted
CS0 area	H'0010 0000 to H'001F FFFF (external extension mode) H'0000 0000 to H'001F FFFF (processor mode) (Note 1)	1 to 4 wait states set by software + any number of wait states entered from the WAIT# pin (However, software settings have priority.)
CS1 area	H'0020 0000 to H'002F FFFF (external extension and processor modes) (Note 2)	1 to 4 wait states set by software + any number of wait states entered from the WAIT# pin (However, software settings have priority.)

Note 1: During processor mode, a ghost of the CS0 area (1 Mbytes) will appear in the H'0010 0000–H'001F FFFF area.
Note 2: A ghost of the CS1 area (1 Mbytes) will appear in the H'0030 0000–H'003F FFFF area.

16.2 Wait Controller Related Register

A Wait Controller related register map is shown below.

Wait Controller Related Register Map

Address	+0 address b0 – b7	+1 address b8 – b15	See page
H'0080 0180	Wait Cycles Control Register (WTCCR)	(Use inhibited area)	16-4

16.2.1 Wait Cycles Control Register

Wait Cycles Control Register (WTCCR) <Address: H'0080 0180>

b0	1	2	3	4	5	6	b7
		CS0WTC				CS1WTC	
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

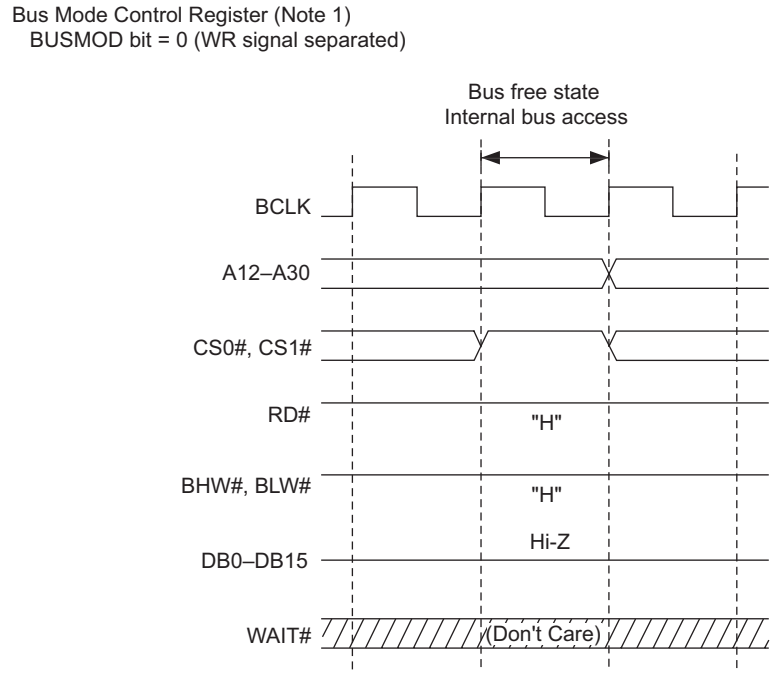
b	Bit Name	Function	R	W
0, 1	No function assigned. Fix to "0".		0	0
2, 3	CS0WTC CS0 wait cycles select bit	00: 4 wait states (Upon exiting reset) 01: 3 wait states 10: 2 wait states 11: 1 wait state	R	W
4, 5	No function assigned. Fix to "0".		0	0
6, 7	CS1WTC CS1 wait cycles select bit	00: 4 wait states (Upon exiting reset) 01: 3 wait states 10: 2 wait states 11: 1 wait state	R	W

16.3 Typical Operation of the Wait Controller

The following shows a typical operation of the Wait Controller. The Wait Controller can control bus access in 2 to 5 cycles. If more access cycles than that are needed, use the WAIT function in combination with the Wait Controller.

(1) When the Bus Mode Control Register is set to 0

External read/write operations are performed using the address bus, data bus and the signals CS0#, CS1#, RD#, BHW#, BLW#, WAIT# and BCLK.



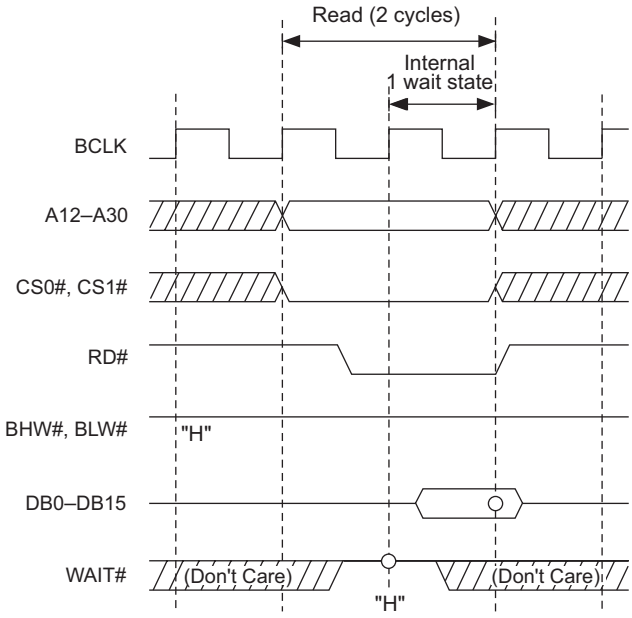
Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note: • Hi-Z denotes a high-impedance state.

Figure 16.3.1 Internal Bus Access during Bus Free State

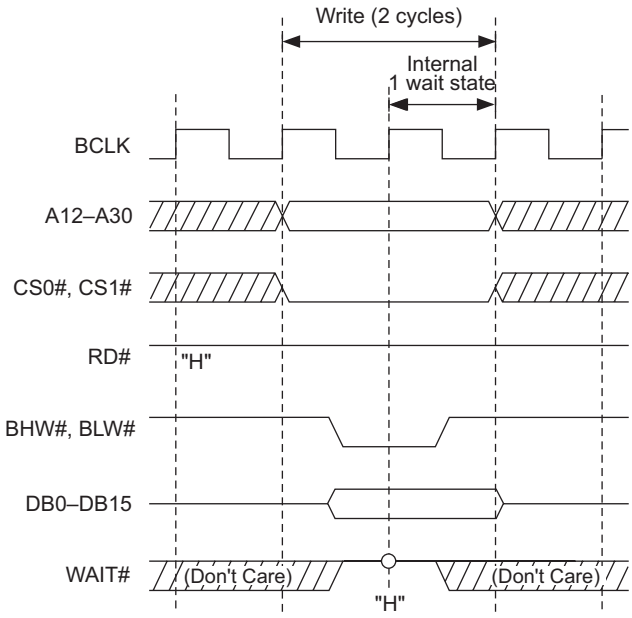
Bus Mode Control Register (Note 1)
BUSMOD bit = 0 (WR signal separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 11 (1 wait)

Read



Write

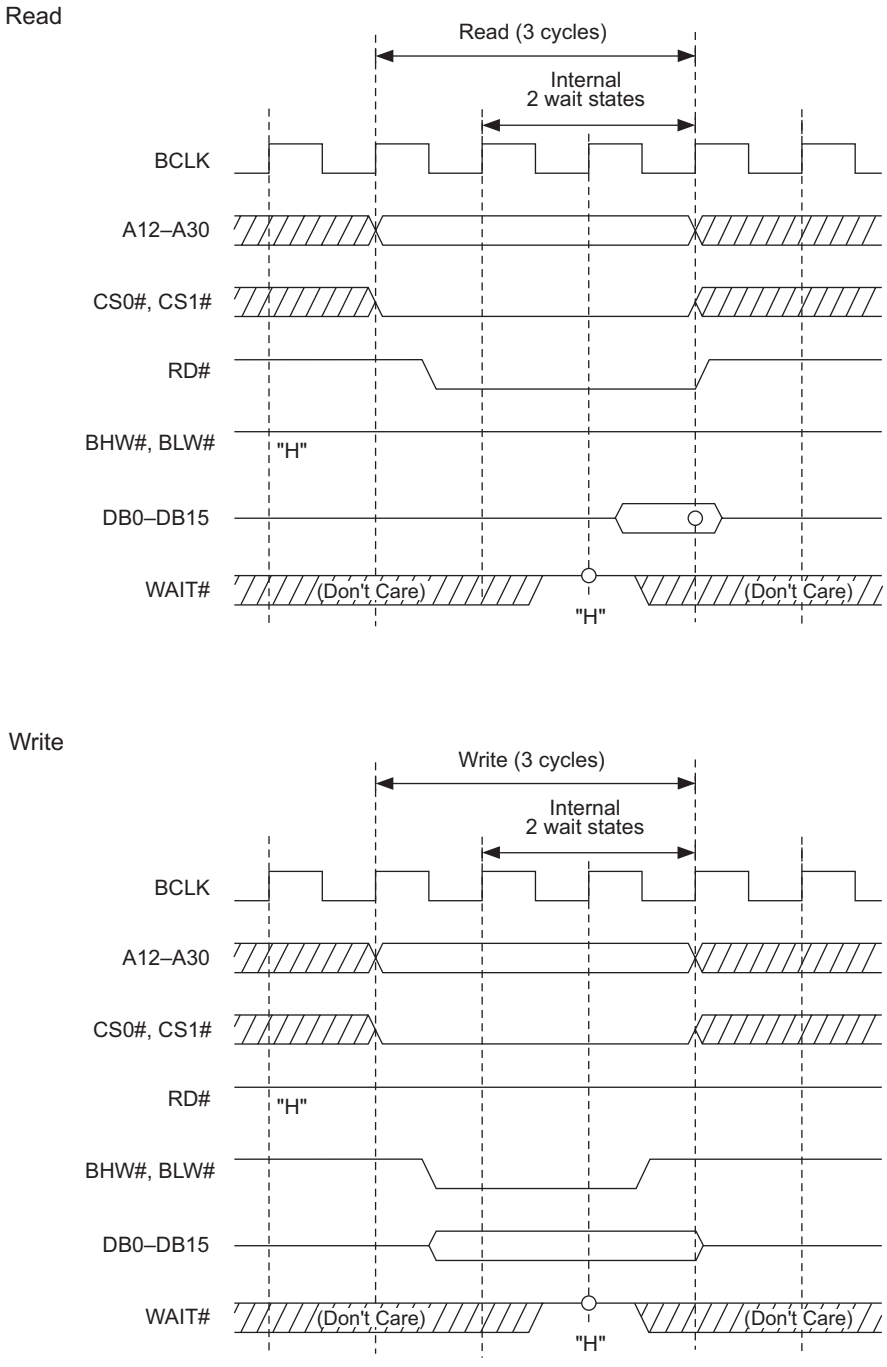


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Note: • Circles in the above diagram indicate the sampling timing.

Figure 16.3.2 Read/Write Timing (for Access with Internal 1 Wait State)

Bus Mode Control Register (Note 1)
BUSMOD bit = 0 (WR signal separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 10 (2 waits)

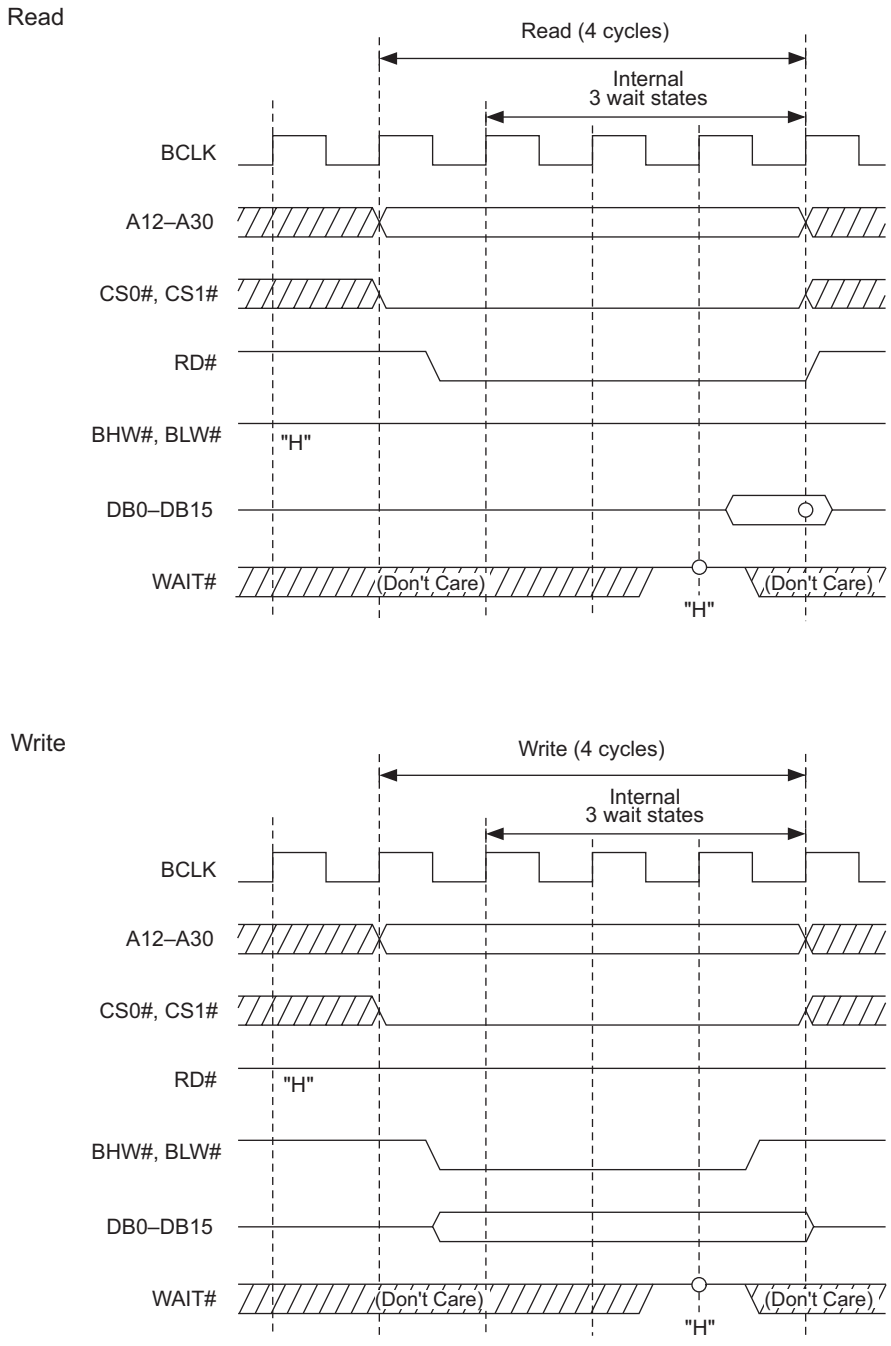


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Note: • Circles in the above diagram indicate the sampling timing.

Figure 16.3.3 Read/Write Timing (for Access with Internal 2 Wait States)

Bus Mode Control Register (Note 1)
BUSMOD bit = 0 (WR signal separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 01 (3 waits)

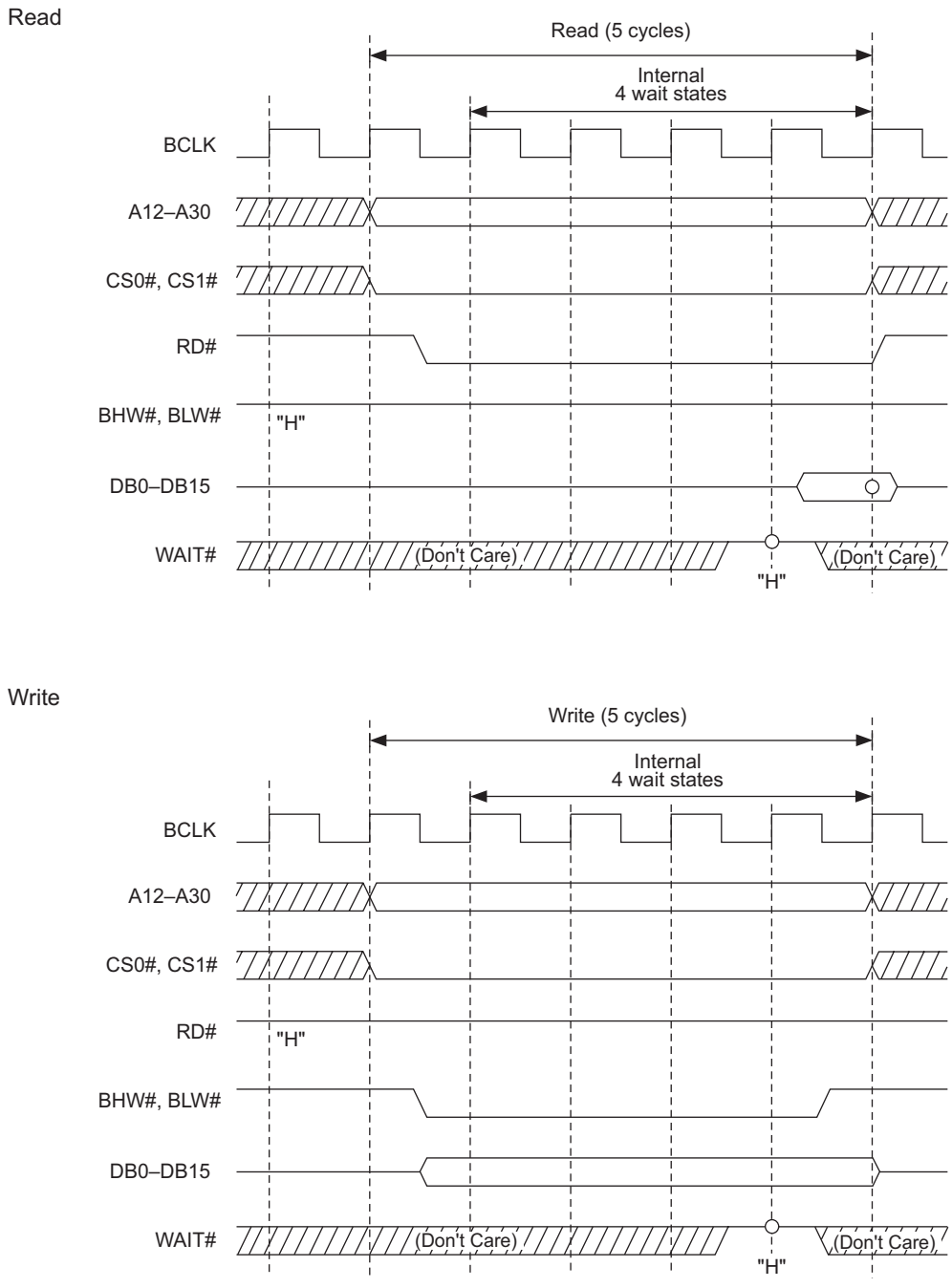


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Note: • Circles in the above diagram indicate the sampling timing.

Figure 16.3.4 Read/Write Timing (for Access with Internal 3 Wait States)

Bus Mode Control Register (Note 1)
BUSMOD bit = 0 (WR signal separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 00 (4 waits)

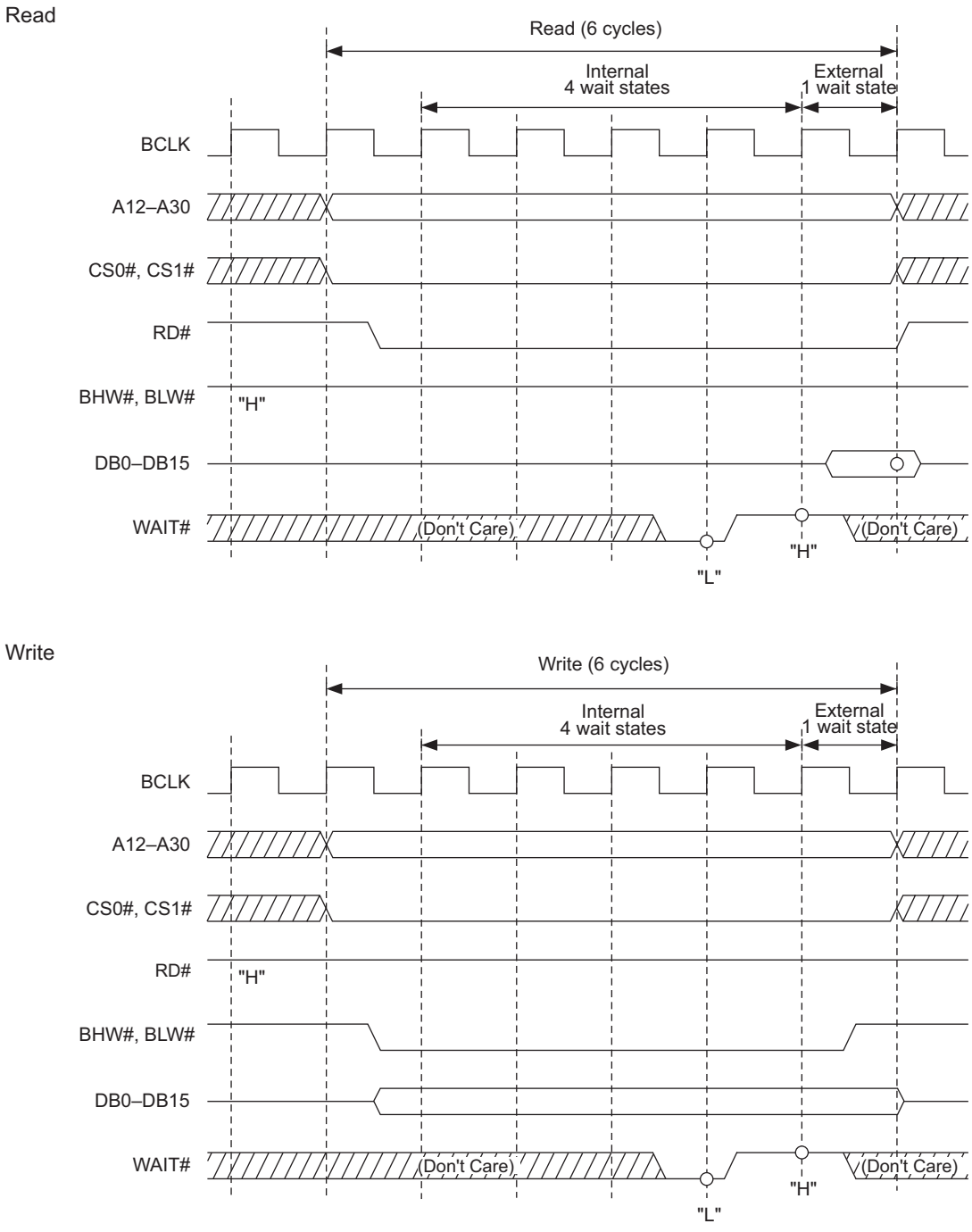


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Note: • Circles in the above diagram indicate the sampling timing.

Figure 16.3.5 Read/Write Timing (for Access with Internal 4 Wait States)

Bus Mode Control Register (Note 1)
BUSMOD bit = 0 (WR signal separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 00 (4 waits)

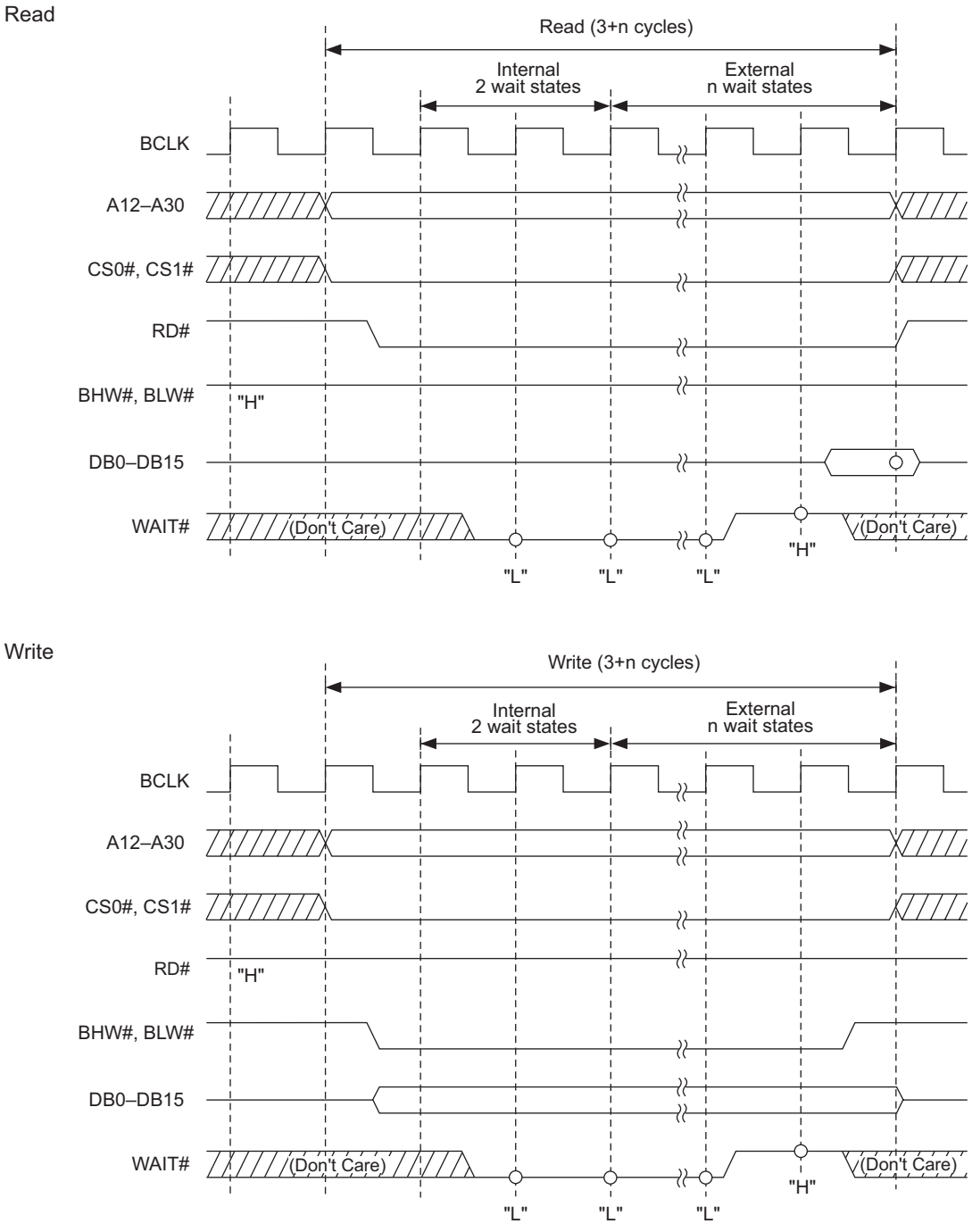


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Note: • Circles in the above diagram indicate the sampling timing.

Figure 16.3.6 Read/Write Timing (for Access with Internal 4 and External 1 Wait States)

Bus Mode Control Register (Note 1)
BUSMOD bit = 0 (WR signal separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 10 (2 waits)

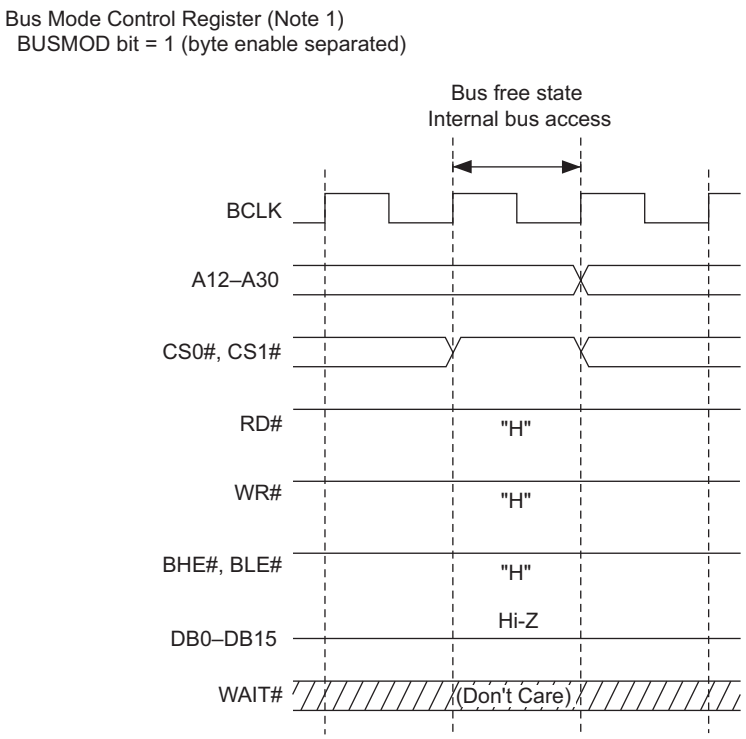


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Note: • Circles in the above diagram indicate the sampling timing.

Figure 16.3.7 Read/Write Timing (for Access with Internal 2 and External n Wait States)

(2) When the Bus Mode Control Register is set to 1

External read/write operations are performed using the address bus, data bus and the signals CS0#, CS1#, RD#, BHE#, BLE#, WAIT# and WR#.

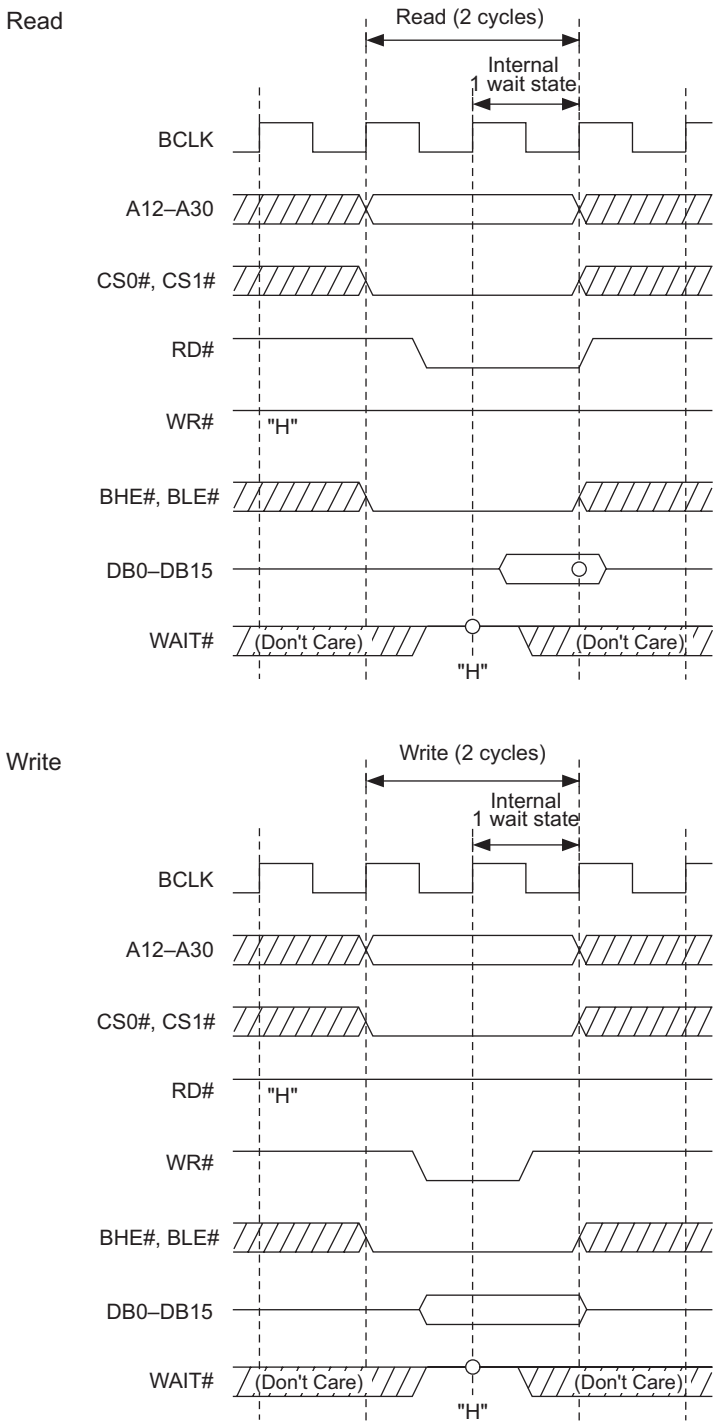


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Notes: • Hi-Z denotes a high-impedance state.
• BCLK is not output.

Figure 16.3.8 Internal Bus Access during Bus Free State

Bus Mode Control Register (Note 1)
BUSMOD bit = 1 (byte enable separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 11 (1 wait)

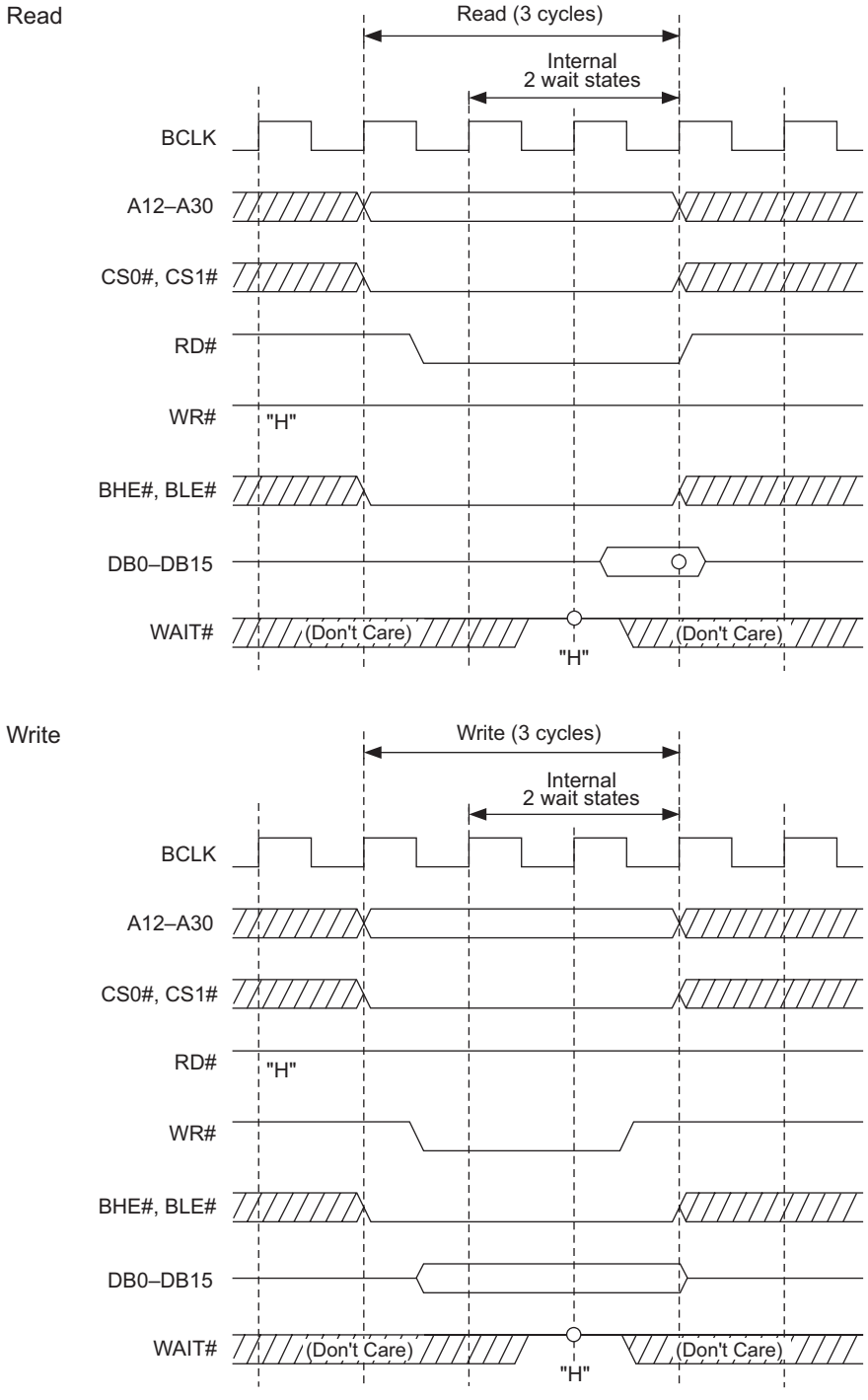


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Notes: • Circles in the above diagram indicate the sampling timing.
• BCLK is not output.

Figure 16.3.9 Read/Write Timing (for Access with Internal 1 Wait State)

Bus Mode Control Register (Note 1)
BUSMOD bit = 1 (byte enable separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 10 (2 waits)

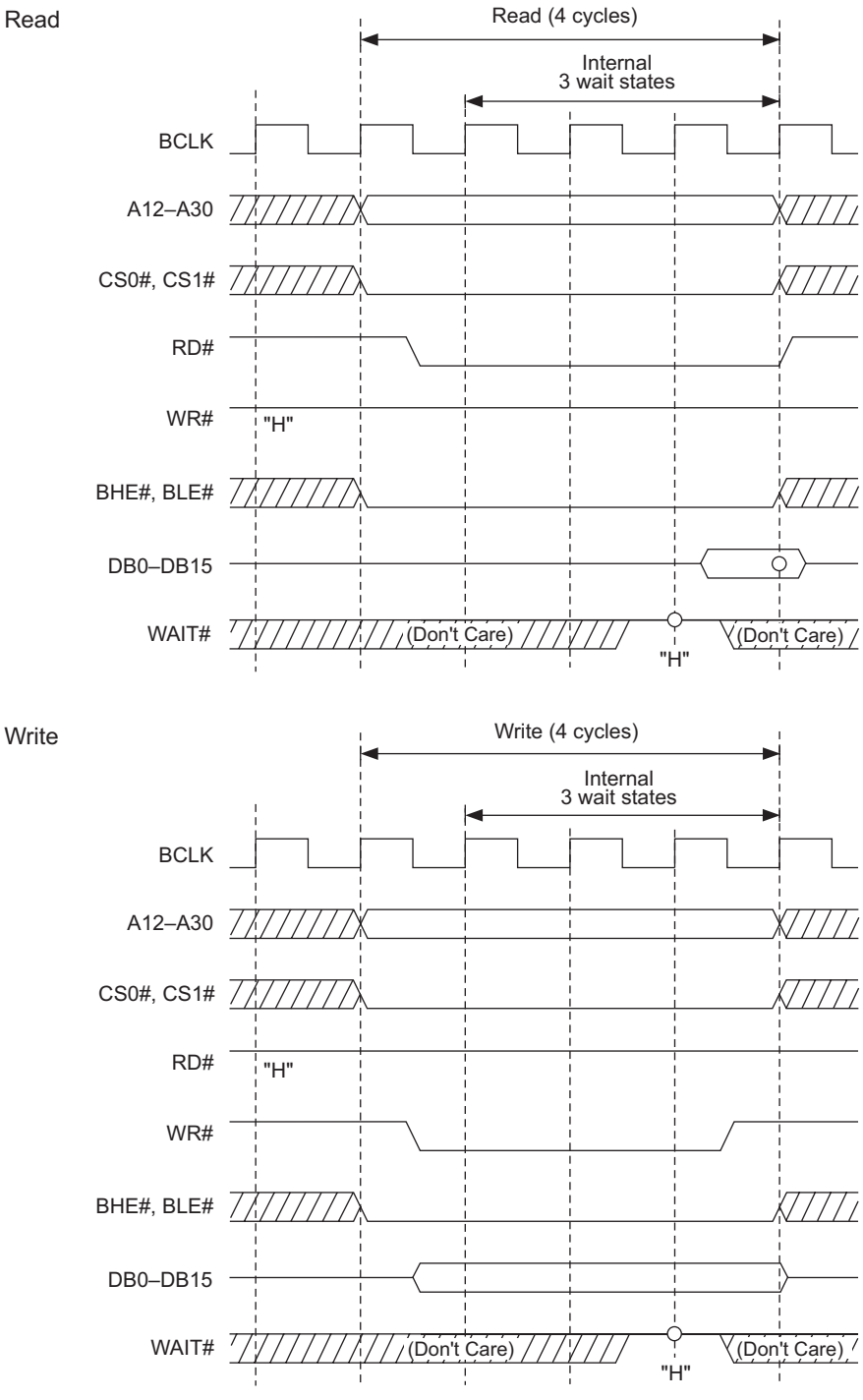


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Notes: • Circles in the above diagram indicate the sampling timing.
• BCLK is not output.

Figure 16.3.10 Read/Write Timing (for Access with Internal 2 Wait States)

Bus Mode Control Register (Note 1)
BUSMOD bit = 1 (byte enable separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 01 (3 waits)

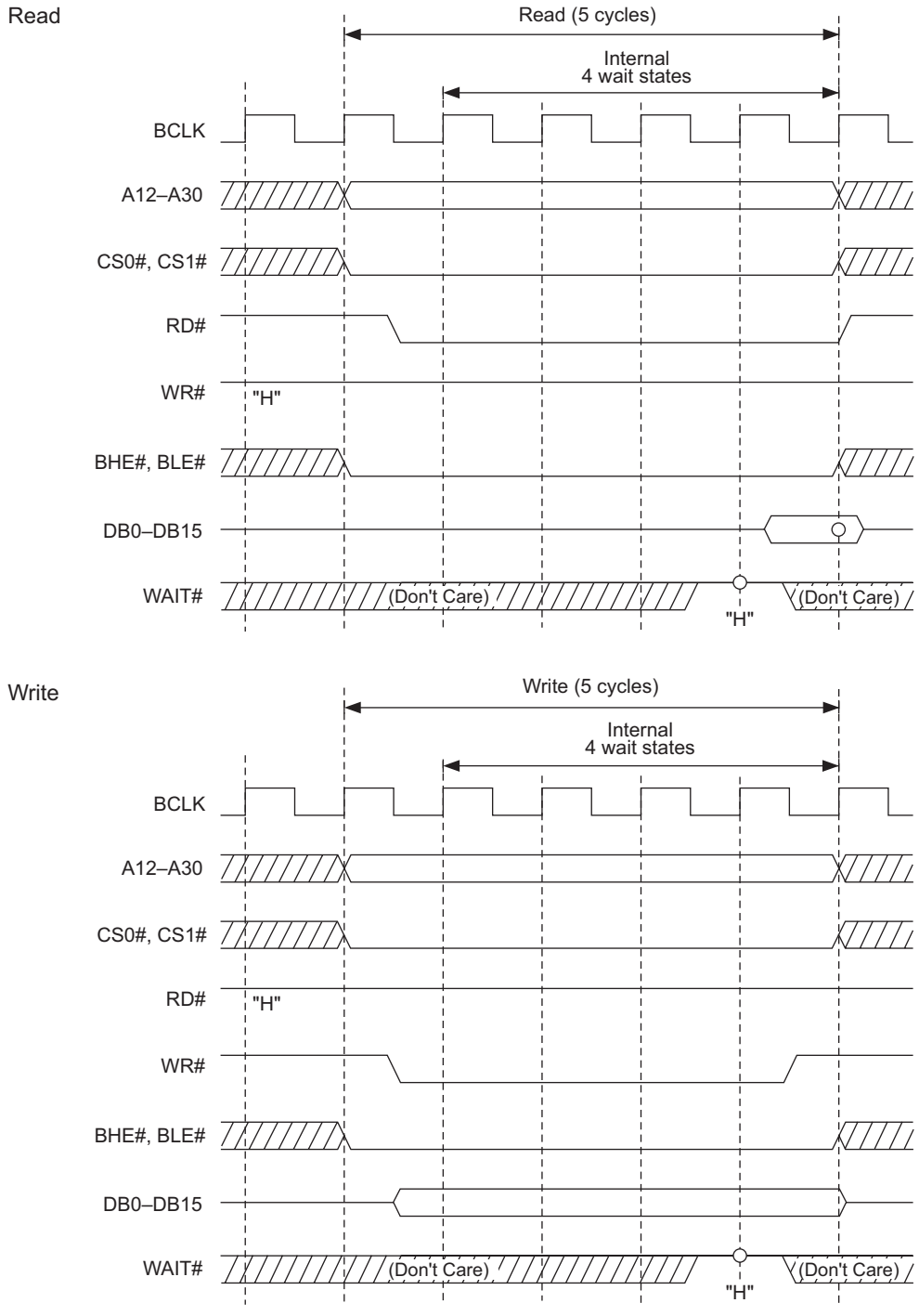


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Notes: • Circles in the above diagram indicate the sampling timing.
• BCLK is not output.

Figure 16.3.11 Read/Write Timing (for Access with Internal 3 Wait States)

Bus Mode Control Register (Note 1)
BUSMOD bit = 1 (byte enable separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 00 (4 waits)

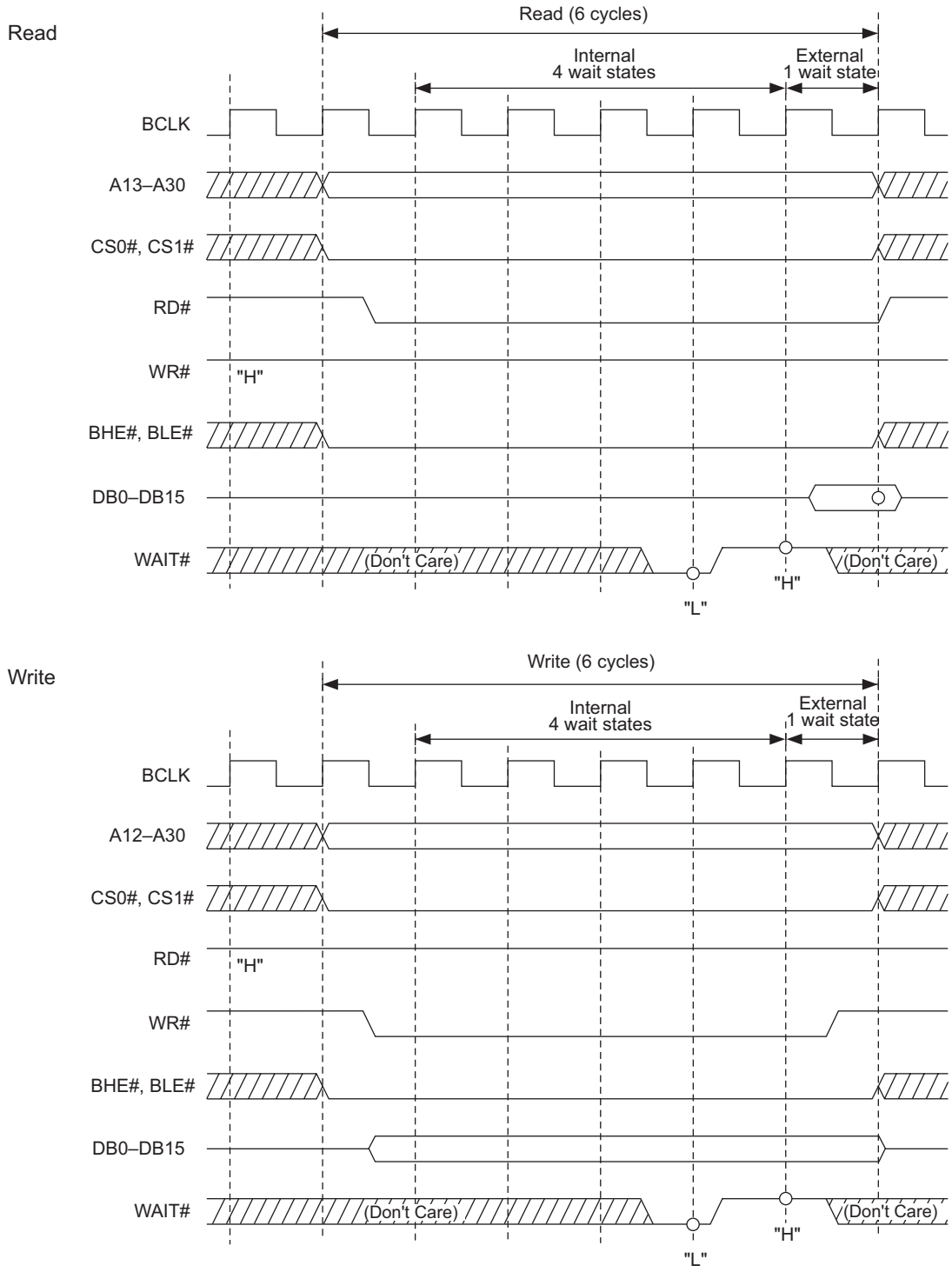


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Notes: • Circles in the above diagram indicate the sampling timing.
• BCLK is not output.

Figure 16.3.12 Read/Write Timing (for Access with Internal 4 Wait States)

Bus Mode Control Register (Note 1)
BUSMOD bit = 1 (byte enable separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 00 (4 waits)

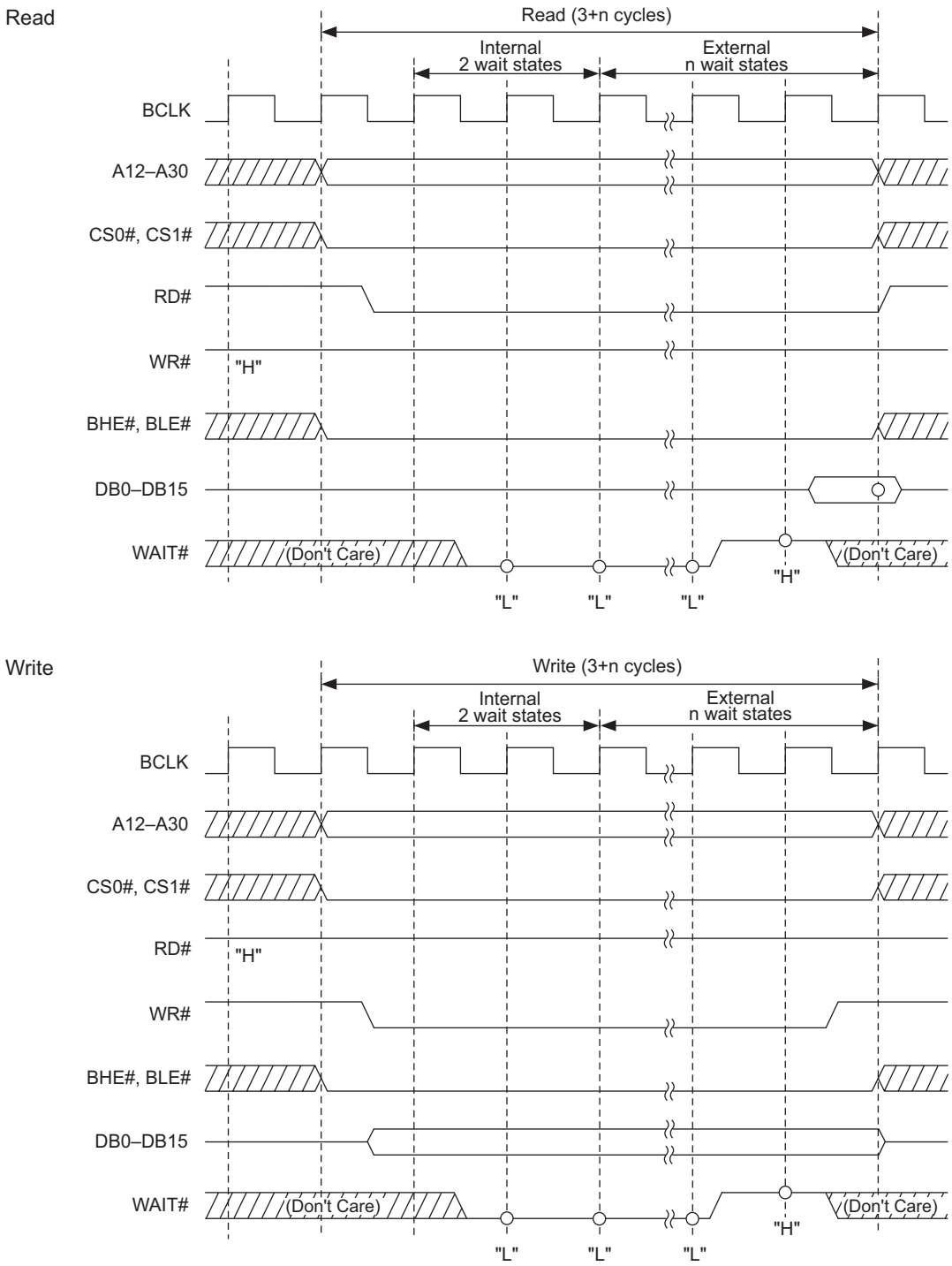


Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Notes: • Circles in the above diagram indicate the sampling timing.
• BCLK is not output.

Figure 16.3.13 Read/Write Timing (for Access with Internal 4 and External 1 Wait States)

Bus Mode Control Register (Note 1)
BUSMOD bit = 1 (byte enable separated)

Wait Cycles Control Register (Note 2)
CSnWTC bit = 10 (2 waits)



Note 1: For details about the Bus Mode Control Register, see Section 15.2.2, "Bus Mode Control Register."
Note 2: For details about the Wait Cycles Control Register, see Section 16.2.1, "Wait Cycles Control Register."
Notes: • Circles in the above diagram indicate the sampling timing.
• BCLK is not output.

Figure 16.3.14 Read/Write Timing (for Access with Internal 2 and External n Wait States)

CHAPTER 17

RAM BACKUP MODE

17.1 Outline of RAM Backup Mode

In RAM backup mode, the contents of the internal RAM are retained while the power is turned off. RAM backup mode is used for the following two purposes.
RAM backup area for 32176 is from H'0080 4000 to H'0080 9FFF (24KB).

- Back up the internal RAM data when the power is forcibly turned off from the outside (RAM backup when the power is off)
- For the M32R/ECU to turn off the power to the CPU at any time as needed to reduce the system's power consumption while retaining the internal RAM data (RAM backup for saving the power consumption)

The M32R/ECU is placed in RAM backup mode by applying a voltage of 3.0–5.5 V to the VDDE pin (provided for RAM backup) and 0 V to all other pins.
During RAM backup mode, the contents of the internal RAM are retained, while the CPU and internal peripheral I/O remain idle. Because all pins except VDDE are held "L" during RAM backup mode, the power consumption in the system can effectively be reduced.

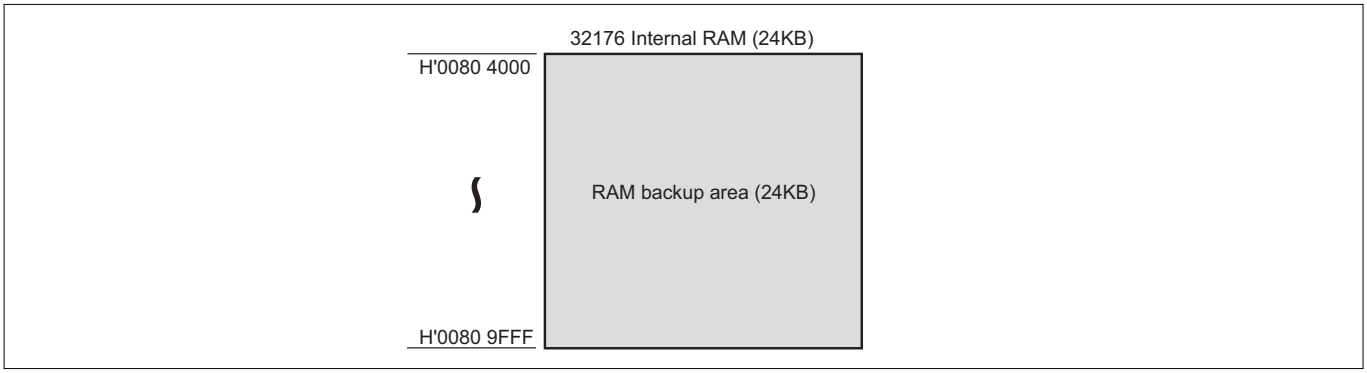


Figure 17.1.1 RAM Backup Area

17.2 Example of RAM Backup when Power is Off

A typical circuit for RAM backup at power outage is shown in Figure 17.2.1. The following explains how the RAM can be backed up by using this circuit as an example.

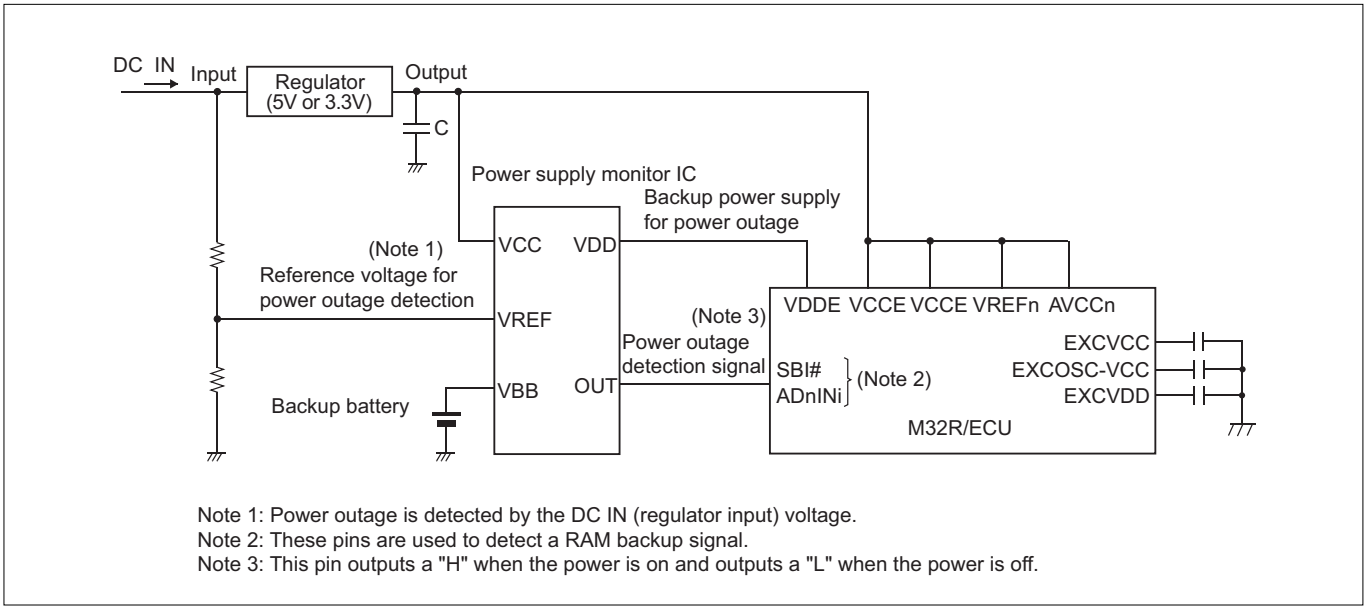
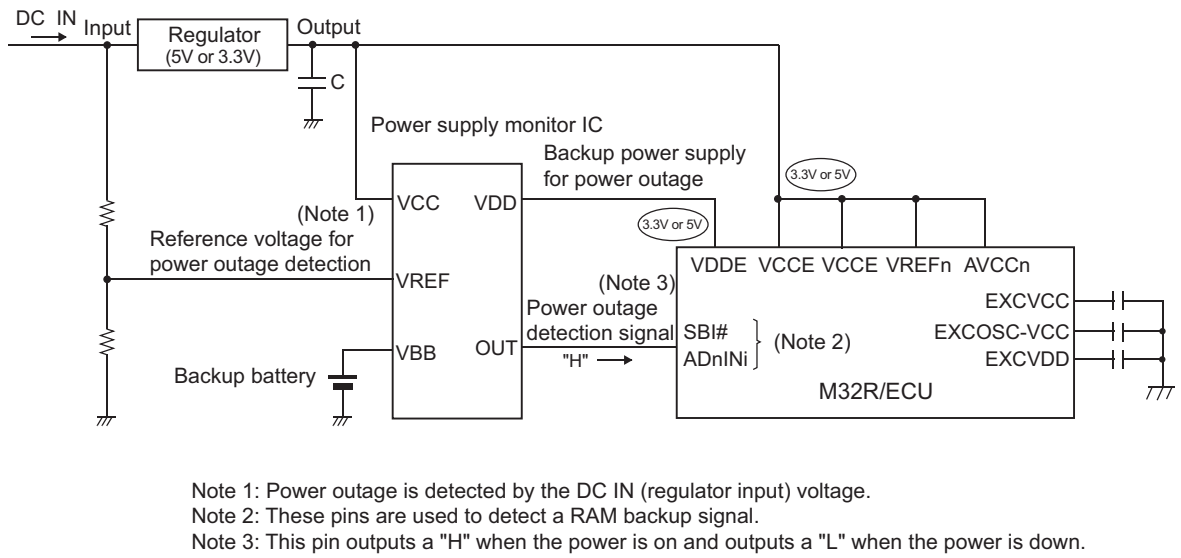


Figure 17.2.1 Typical Circuit for RAM Backup at Power Outage

17.2.1 Normal Operating State

Figure 17.2.2 shows the normal operating state of the M32R/ECU. During normal operation, input on the SBI# pin or ADnINi (i = 0–15) pin which is used to detect a RAM backup signal remains "H."



Note 1: Power outage is detected by the DC IN (regulator input) voltage.
Note 2: These pins are used to detect a RAM backup signal.
Note 3: This pin outputs a "H" when the power is on and outputs a "L" when the power is down.

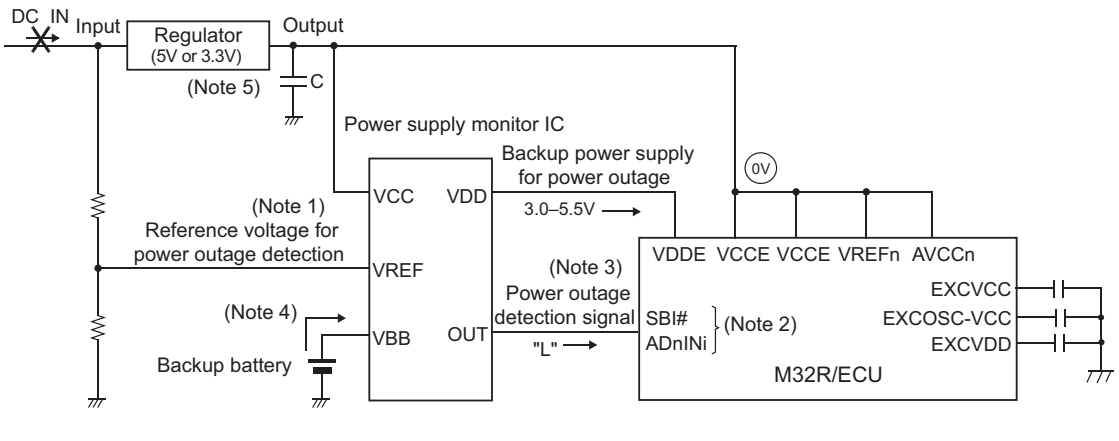
Figure 17.2.2 Normal Operating State

17.2.2 RAM Backup State

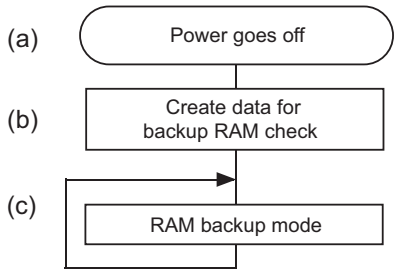
Figure 17.2.3 shows the power outage RAM backup state of the M32R/ECU. When the power supply goes off, the power supply monitor IC starts feeding current from the backup battery to the M32R/ECU. Also, the power supply monitor IC's power outage detection pin outputs a "L", causing the SBI# pin or ADnINi pin to go "L", which generates a RAM backup signal ((a) in Figure 17.2.3). Determination of whether the power is off must be made with respect to the DC IN (regulator input) voltage in order to allow for a software processing time at power outage.
To enable RAM backup mode, make the following setting:

(1) Create data for RAM check to verify whether the RAM data has been retained normally after returning from RAM backup mode to normal mode ((b) in Figure 17.2.3).

If the power supply to VCCE goes off after making above setting, the VDDE pin voltage goes to 3.0–3.3 V and all other pin voltages drop to 0 V, and the M32R/ECU is thereby placed in RAM backup mode ((c) in Figure 17.2.3).



Example of RAM backup processing

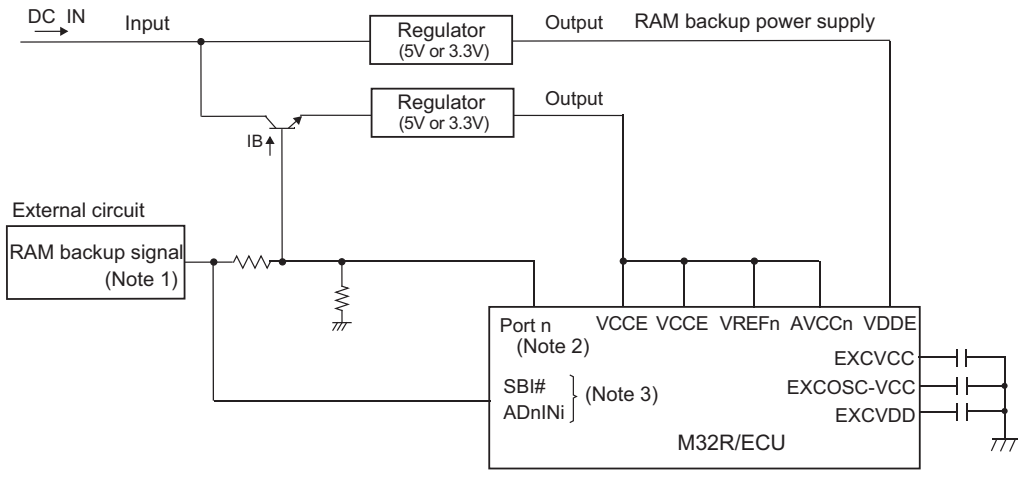


Note 1: Power outage is detected by the DC IN (regulator input) voltage.
Note 2: These pins are used to detect a RAM backup signal.
Note 3: This pin outputs a "H" when the power is on and outputs a "L" when the power is off.
Note 4: Determined by the input level on SBI# pin or ADnINi pin.
Note 5: The time needed for processing in (b) is secured by adjusting the capacitance.

Figure 17.2.3 Power Outage RAM Backup State

17.3 Example of RAM Backup for Saving Power Consumption

A typical RAM backup circuit for saving the microcomputer's power consumption is shown in Figure 17.3.1. The following explains how the RAM is backed up for the purpose of low-power operation by using this circuit as an example.

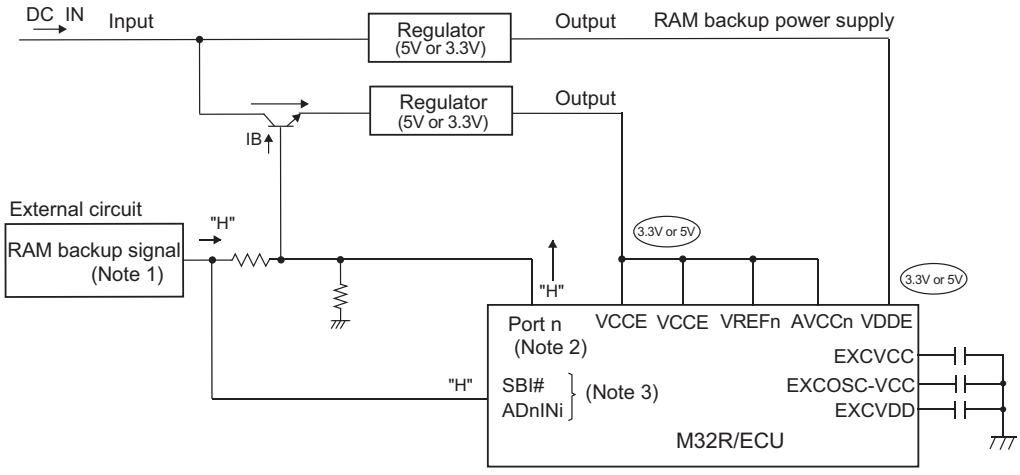


Note 1: This circuit outputs a "L" during RAM backup.
Note 2: This port outputs a "H" when the power is on, and is set for input mode when in RAM backup mode.
Note 3: These pins are used to detect a RAM backup signal.

Figure 17.3.1 Typical RAM Backup Circuit for Saving Power Consumption

17.3.1 Normal Operating State

Figure 17.3.2 shows the normal operating state of the M32R/ECU. During normal operation, the RAM backup signal output by the external circuit is "H." Also, input on the SBI# pin or ADnINi (i = 0–15) pin which is used to detect a RAM backup signal remains "H."
Port n, which connects to the transistor's base, should output a "H." This causes the transistor's base voltage, IB, to go "H" so that current is fed from the power supply to the VCCE pin via the transistor.



Note 1: This circuit outputs a "L" during RAM backup.
Note 2: This port outputs a "H" when the power is on, and is set for input mode when in RAM backup mode (one of the port pins selected).
Note 3: These pins are used to detect a RAM backup signal.

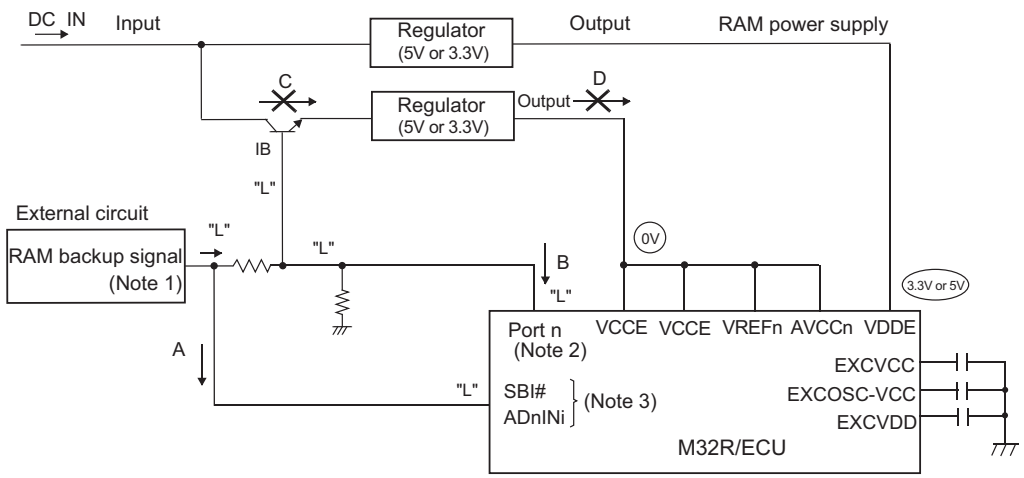
Figure 17.3.2 Normal Operating State

17.3.2 RAM Backup State

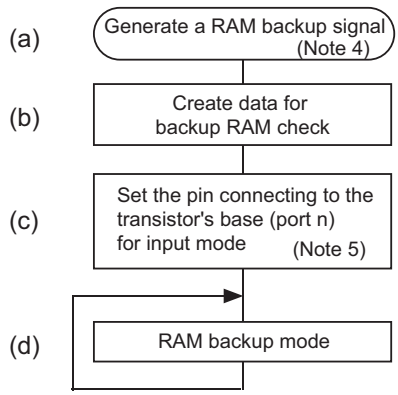
Figure 17.3.3 shows the RAM backup state of the M32R/ECU. Figure 17.3.4 shows a RAM backup sequence. When the external circuit outputs a "L", input on the SBI# or ADnINi pin is pulled "L." A "L" on these input pins generates a RAM backup signal (A and (a) in Figure 17.3.3). To enable RAM backup mode, make the following settings:

(1) Create data for RAM check to verify after returning from RAM backup mode to normal mode whether the RAM data has been retained normally ((b) in Figure 17.3.3).
(2) To materialize low-power operation, set all programmable input/output pins except port n for input mode (or for output mode, with the output level fixed "L") ((c) in Figure 17.3.3).
(3) Set port n for input mode (B and (d) in Figure 17.3.3). This causes the transistor's base voltage, IB, to go "L", so that the power to all power supply pins except VDDE is shut off (C and D in Figure 17.3.3).

By settings in (1) to (3), the VDDE pin voltage goes to 3.0–5.5 V and all other pin voltages drop to 0 V, and the M32R/ECU is thereby placed in RAM backup mode ((d) in Figure 17.3.3).



Example of RAM backup processing



Note 1: This circuit outputs a "L" during RAM backup.
Note 2: This port outputs a "H" when the power is on, and is set for input mode when in RAM backup mode (one of the port pins selected).
Note 3: These pins are used to detect a RAM backup signal.
Note 4: Determined by the input level on SBI# pin or ADnINi pin.
Note 5: Base voltage IB = 0 causes the power to all power supply pins except VDDE to stop.
See A to D in the above explanation.

Figure 17.3.3 RAM Backup State for Low Power Operation

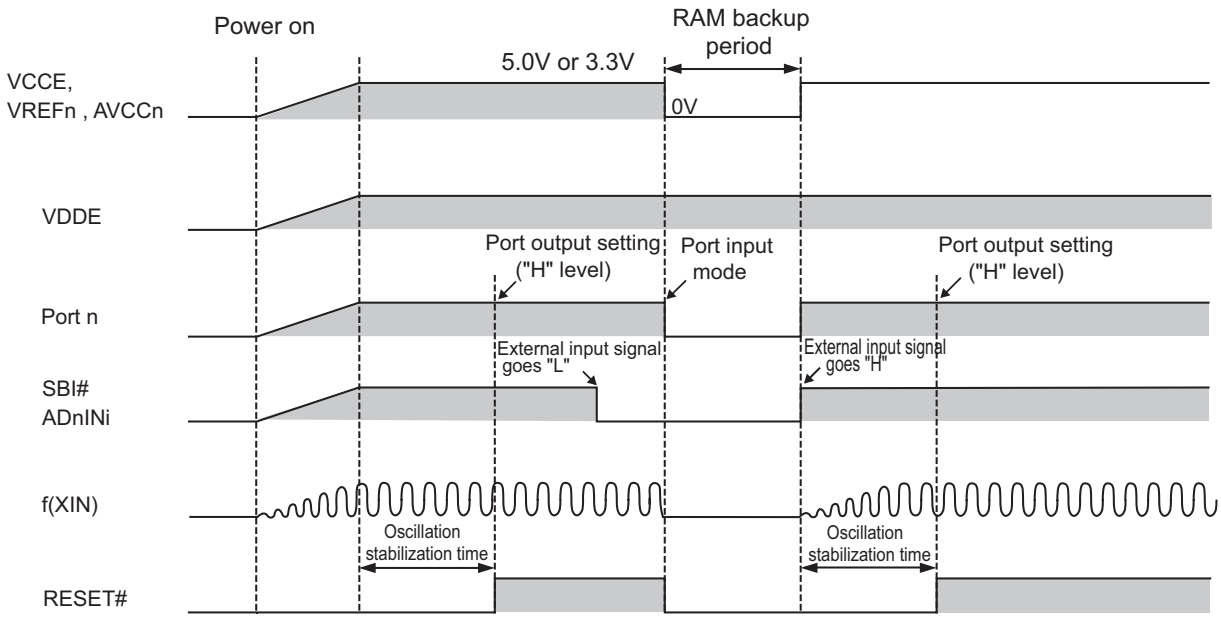


Figure 17.3.4 Example of a RAM Backup Sequence for Low Power Operation

17.3.3 Precautions to Be Observed at Power-On

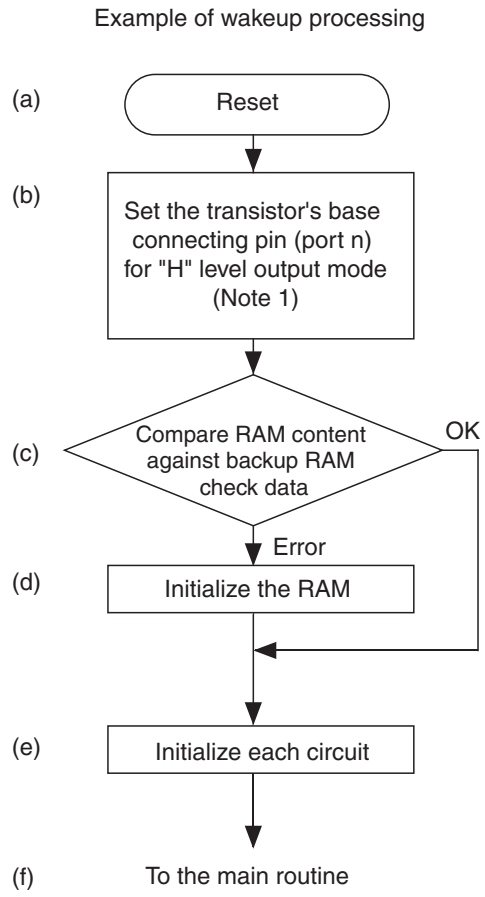
When changing port n from input mode to output mode after power-on, pay attention to the following.
If port n is set for output mode while no data is set in the Port n Data Register, the port's initial output level is instable. Therefore, before changing port n for output mode, make sure the Port n Data Register is set to output a "H."
Unless this precaution is followed, port output may go "L" at the same time the port is set for output after the oscillation has stabilized, causing the microcomputer to enter RAM backup mode.

17.4 Exiting RAM Backup Mode (Wakeup)

The processing to place the M32R/ECU out of RAM backup mode and return it to normal operation mode is referred to as “wakeup” processing. Figure 17.4.1 shows an example of wakeup processing.
Wakeup processing is initiated by applying a reset. The following shows how to execute wakeup processing.

(1) Reset the microcomputer ((a) in Figure 17.4.1).
(2) Set port n for output mode and output a "H" from the port ((b) in Figure 17.4.1) (Note 1)
(3) Compare the RAM content against the RAM check data created before entering RAM backup mode ((c) in Figure 17.4.1).
(4) If the comparison in (3) did not match, initialize the RAM ((d) in Figure 17.4.1).
If the comparison in (3) matched, use the retained data in the program.
(5) Initialize each internal circuit ((e) in Figure 17.4.1) before returning to the main routine ((f) in Figure 17.4.1).

Note 1: For wakeup from power outage RAM backup mode, port n settings are unnecessary.



Note 1: For wakeup from power outage RAM backup mode, port n settings are unnecessary.

Figure 17.4.1 Wakeup Processing

This page is blank for reasons of layout.

CHAPTER 18

OSCILLATOR CIRCUIT

18.1 Oscillator Circuit

The 32176 contains an oscillator circuit that supplies operating clocks for the CPU core, internal peripheral I/O and internal memory. The frequency supplied to the clock input pin (XIN) is multiplied by 4 by an internal PLL circuit to produce the CPU clock, which is the operating clock for the CPU core and internal memory. The frequency of this clock is divided by 2 in the subsequent circuit to produce the peripheral clock, which is the operating clock for the internal peripheral I/O and external data bus.

18.1.1 Example of an Oscillator Circuit

An oscillator circuit can be configured by connecting a ceramic (or crystal) resonator between the XIN and XOUT pins external to the chip. Figure 18.1.1 shows an example of a system clock generating circuit illustrating a resonator connected external to the chip. For the constants Rf, Cin, Cout and Rd, the resonator manufacturer should be consulted to determine the appropriate values.
To use an externally sourced clock signal without using an internal oscillator circuit, connect the external clock signal to the XIN pin and leave the XOUT pin open.

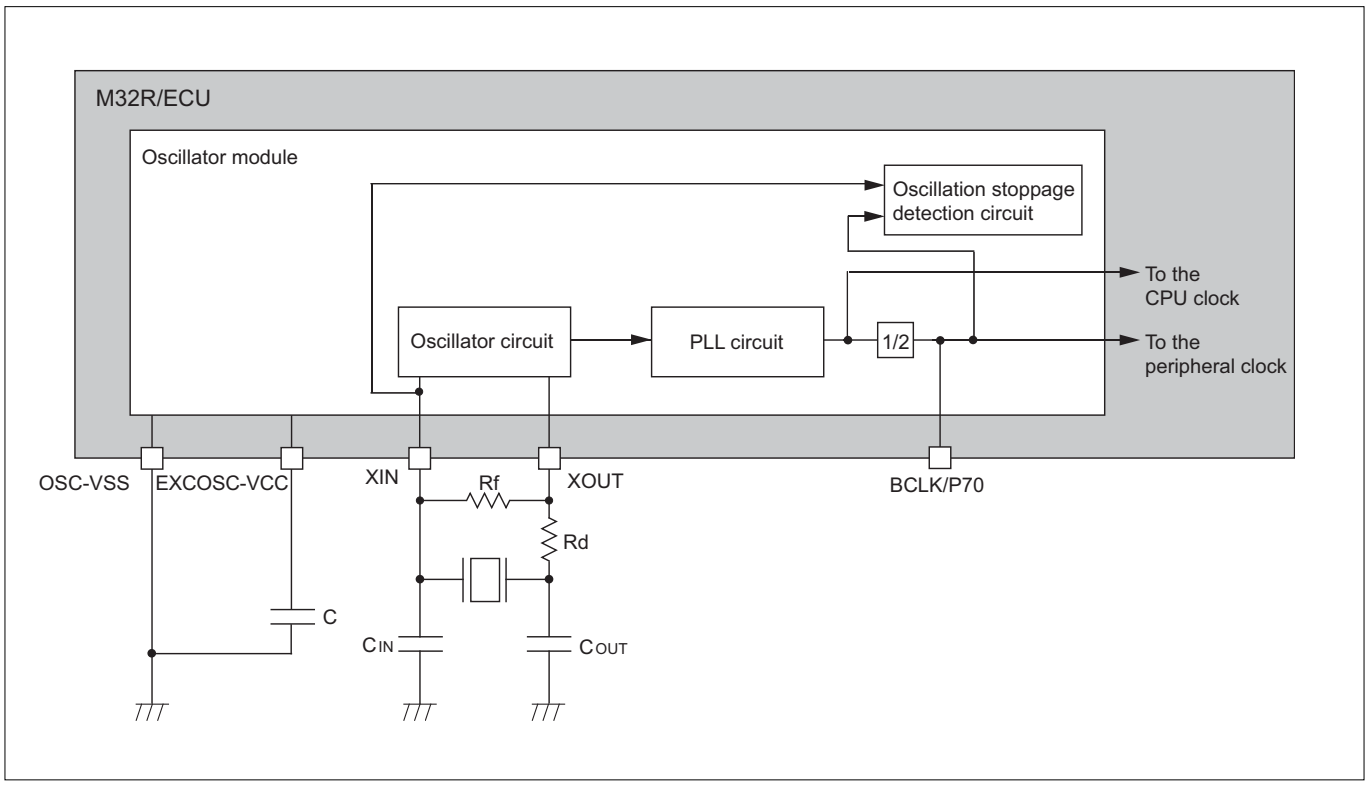


Figure 18.1.1 Example of an Oscillator Circuit

18.1.2 XIN Oscillation Stoppage Detection Function

The 32176 contains a detection circuit to find whether oscillation input to the PLL circuit has stopped. The PLL circuit oscillates with the frequency of its specific vibration in the absence of the reference oscillation input.
The XIN oscillation input is sampled at the peripheral clock and when the XIN oscillation is found to be at the same level, the XSTAT bit is set. Because the CPU continues operating with the PLL circuit's natural frequency even when the XIN oscillation has stopped, error handling for the stoppage of XIN oscillation can be accomplished by inspecting XSTAT in software.

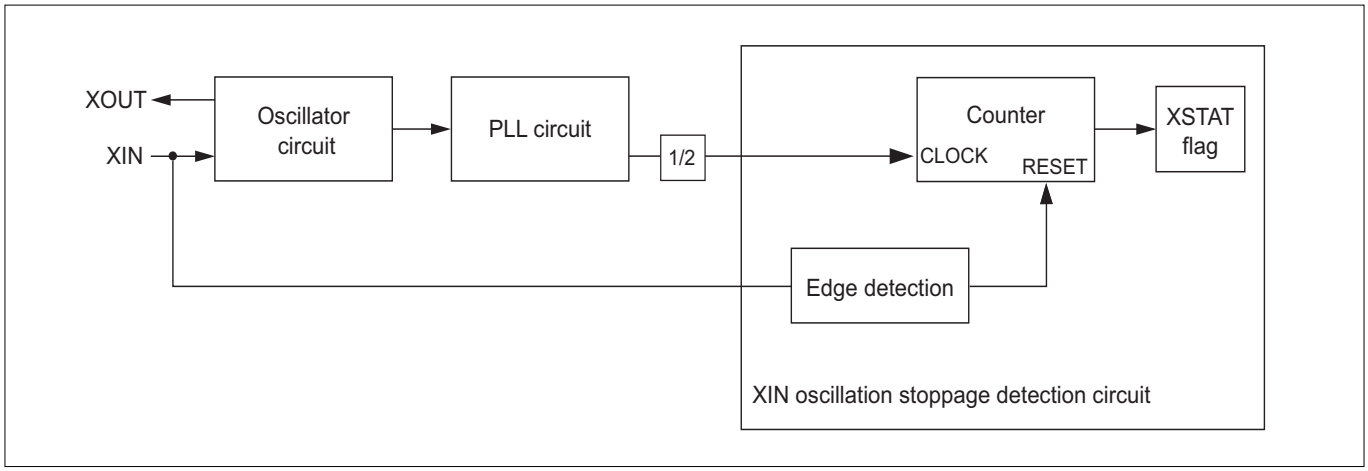


Figure 18.1.2 Block Diagram of the XIN Oscillation Stoppage Detection Circuit

Port Input Special Function Control Register (PICNT) <Address: H'0080 0745>

b8	9	10	11	12	13	14	b15
0	0	0	XSTAT 0	0	0	PISEL 0	PIEN0 0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8–10	No function assigned. Fix to "0".		0	0
11	XSTAT XIN oscillation status bit	0: XIN oscillating 1: XIN inactive	R	(Note1)
12–13	No function assigned. Fix to "0".		0	0
14	PISEL Port input data select bit	0: Content of port output latch 1: Port pin level	R	W
15	PIEN0 Port input enable bit	0: Disable input 1: Enable input	R	W

Note 1: Only writing "0" is effective. Writing "1" has no effect; the bit retains the value it had before the write.

For details about the function of the port input data select bit (PISEL) and port input enable bit (PIEN0), see Section 8.3.5, "Port Input Special Function Control Register."

(1) XSTAT (XIN oscillation status) bit (Bit 11)

1) Conditions under which XSTAT is set to "1"

XSTAT is set to "1" upon detecting that XIN oscillation has stopped. When XIN remains at the same level for a predetermined time (3 BCLK periods up to 4 BCLK periods), XIN oscillation is assumed to have stopped. When operating normally, XIN changes state (high or low) once every BCLK period.

2) Conditions under which XSTAT is cleared to "0"

XSTAT is cleared to "0" by a system reset or by writing "0". If XSTAT is cleared at the same time it is set to "1" in 1) above, the former has priority so that XSTAT is cleared. Writing "1" to XSTAT is ignored.

3) Method for detecting XIN oscillation stoppage by using XSTAT

Because the M32R/ECU internally contains a PLL, the internal clock remains active even when XIN oscillation has stopped.
By reading XSTAT without clearing it after exiting the reset state, it is possible to know whether XIN has stopped since the reset signal was deasserted. Similarly, by reading XSTAT after clearing it by writing 0, it is possible to know the current oscillating status of XIN. (However, there must be an interval of at least 10 BCLK periods (20 CPU clock periods) between read and write.)
To carry out this function when XSTAT bit is set to 1, make sure to reconfirm after clearing XSTAT bit once, and pay extra attention before use.

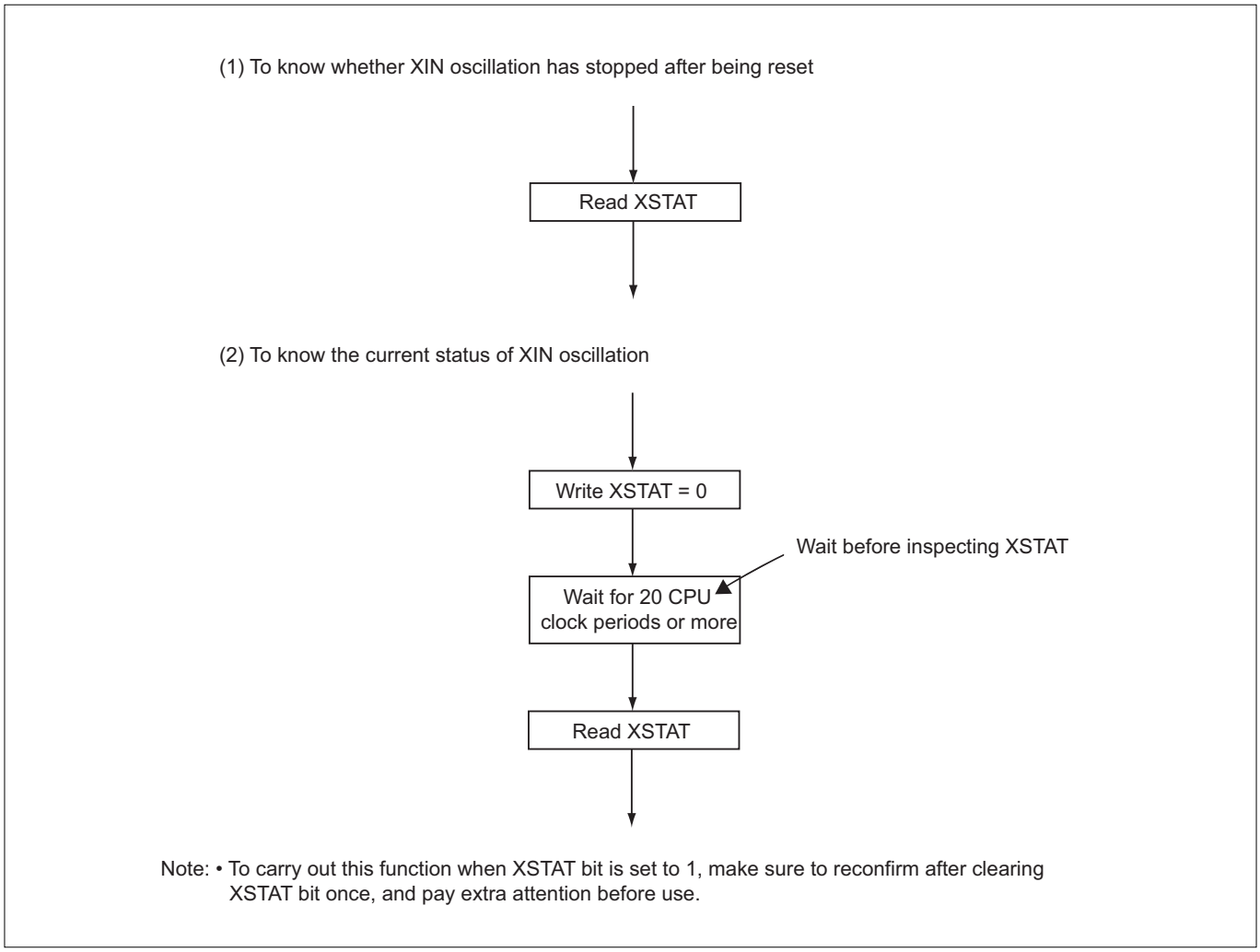


Figure 18.1.3 Procedure for Setting XSTAT

18.1.3 Oscillation Drive Capability Select Function

The microcomputer incorporates a four-stage oscillation drive capability select function.
Once the oscillation of the oscillator circuit has stabilized, the XIN-XOUT drive capability can be lowered. The lower the drive capability, the smaller the amount of power consumption.

Clock Control Register (CLKCR) <Address: H'0080 0786>

b0	1	2	3	4	5	6	b7
					XDRVP	XDRV	
0	0	0	0	0	0	1	1

<Upon exiting reset: H'03>

b	Bit Name	Function	R	W
0–4	No function assigned. Fix to "0".		0	0
5	XDRVP XDRV write control bit		0	W
6–7	XDRV XIN-XOUT drive capability select bit	XIN-XOUT drive capability (performance ratio) 00: Low 0.25 01: 0.50 10: 0.75 11: High 1.00	R	W

(1) XDRV write control bit (XDRVP) (Bit 5)

This bit controls writing to the XIN-XOUT drive capability select bits.

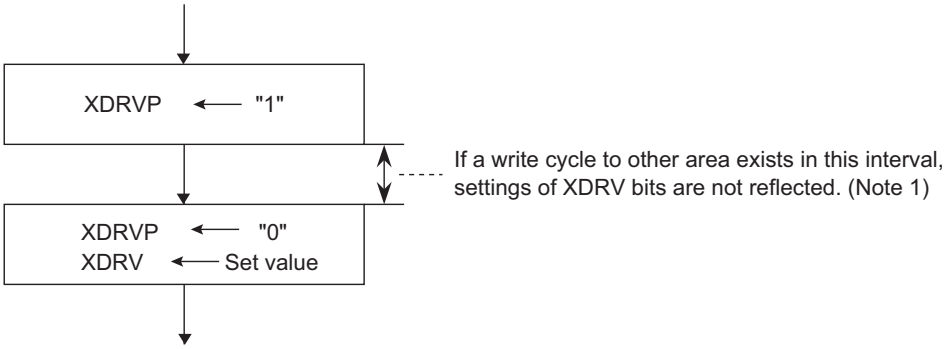
(2) XIN-XOUT drive capability select bits (Bits 6, 7)

The following shows the procedure for writing to these bits.

1. Set the write control bit (XDRVP) to "1".
2. Immediately following the above, reset the write control bit (XDRVP) to "0" and write the appropriate value to the XIN-XOUT drive capability select bits.

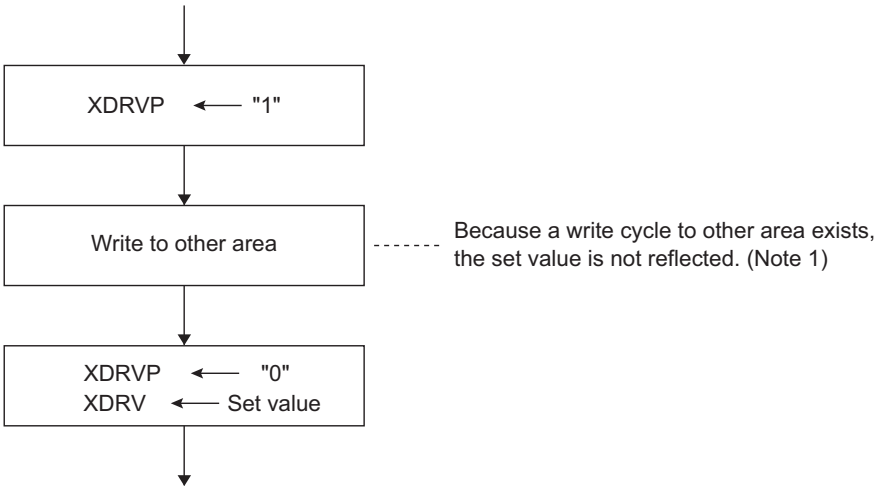
Note: • If a write cycle to any other area occurs between 1 and 2, write to XDRV has no effect and the written value is not reflected. Therefore, disable interrupts and DMA transfers before setting the drive capability control bits. Note that a pair of two consecutive writes comprise a write operation.

• Example of correct setting

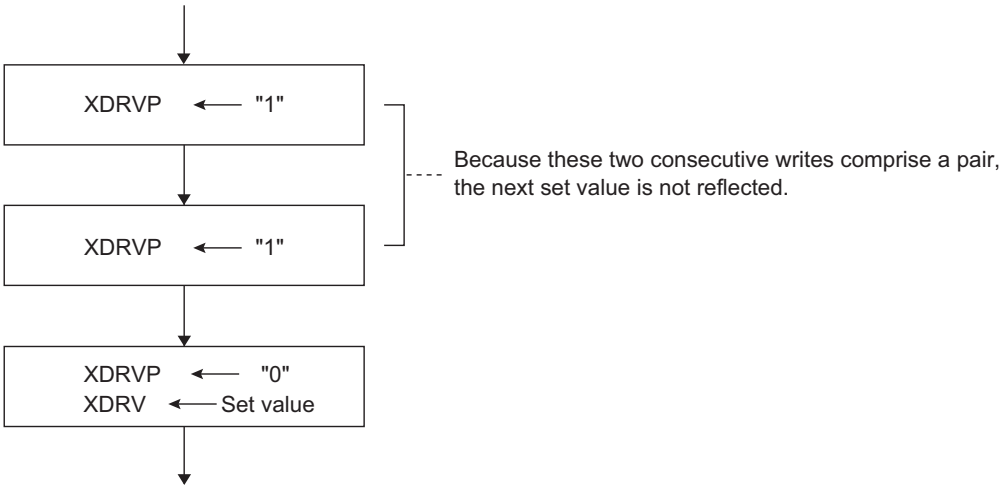


• Settings that do not have effect

(1)



(2)



Note 1: The writing cycle to the other area is the writing cycle from CPU, DMA, and SDI (tool) to any other area. There is no effect on the writing cycle from RTD.

Figure 18.1.4 Procedure for Setting the Oscillation Drive Capability

18.1.4 System Clock Output Function

A clock whose frequency is twice that of the input clock (i.e., the peripheral clock) can be output from the BCLK pin. The BCLK pin is shared with port P70. To use this pin to output the peripheral clock, set the P7 Operation Mode Register (P7MOD) bit 8 to "1".
Configuration of the P7 Operation Mode Register is shown below.

P7 Operation Mode Register (P7MOD) <Address: H'0080 0747>

b8	9	10	11	12	13	14	b15
P70MD	P71MD	P72MD	P73MD	P74MD	P75MD	P76MD	P77MD
0	0	0	0	0	0	0	0

<Upon exiting reset: H'00>

b	Bit Name	Function	R	W
8	P70MD Port P70 operation mode bit	0:P70 1:BCLK	R	W
9	P71MD Port P71 operation mode bit	0:P71 1:WAIT#	R	W
10	P72MD Port P72 operation mode bit	0:P72 1:HREQ#	R	W
11	P73MD Port P73 operation mode bit	0:P73 1:HACK#	R	W
12	P74MD Port P74 operation mode bit	0:P74 1:RTDTXD	R	W
13	P75MD Port P75 operation mode bit	0:P75 1:RTDRXD	R	W
14	P76MD Port P76 operation mode bit	0:P76 1:RTDACK	R	W
15	P77MD Port P77 operation mode bit	0:P77 1:RTDCLK	R	W

18.1.5 Oscillation Stabilization Time at Power-On

The oscillator circuit comprised of a ceramic (or crystal) resonator requires a finite time before its oscillation stabilizes after being powered on. Therefore, there must be a certain amount of oscillation stabilization time that suits the oscillator circuit used.
Figure 18.1.5 shows an oscillation stabilization time required at power-on.

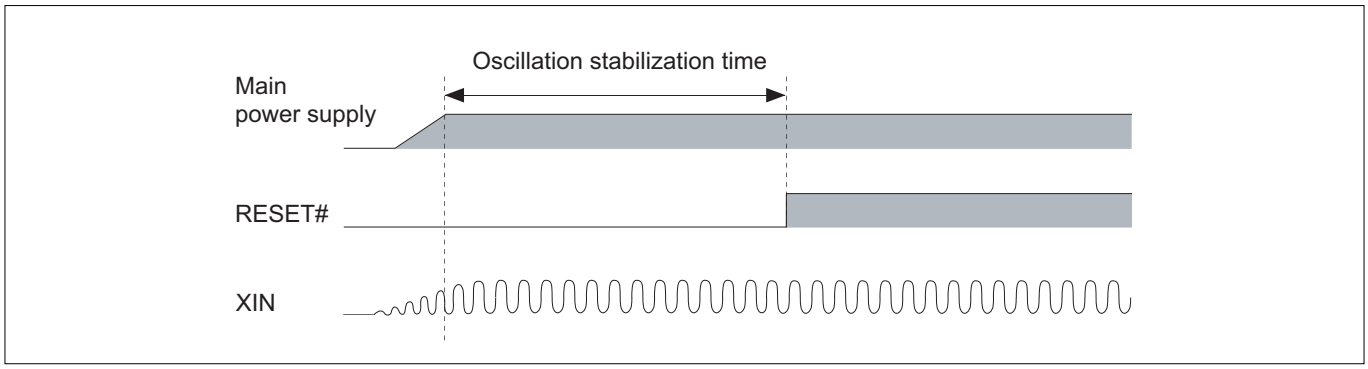


Figure 18.1.5 Oscillation Stabilization Time at Power-On

18.2 Clock Generator Circuit

Supply independent clocks to the CPU and the internal peripheral circuit.

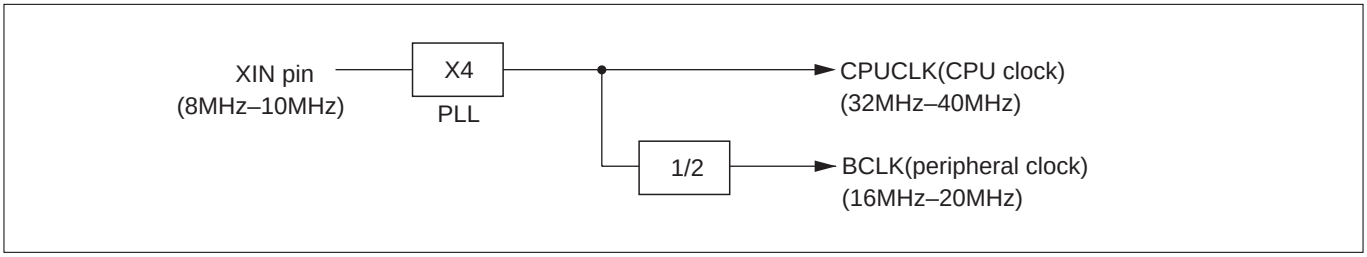


Figure 18.2.1 Conceptual Diagram of Clock Generation

CHAPTER 19

JTAG

19.1 Outline of JTAG

The M32R/ECU contains a JTAG (Joint Test Action Group) interface compliant with IEEE Standard Test Access Port and Boundary-Scan Architecture (IEEE Std. 1149.1a-1993). This JTAG interface can be used as an input/output path for boundary-scan test (boundary-scan path). For details about IEEE 1149.1 JTAG test access ports, see IEEE Std. 1149.1a-1993 documentation.

Note: • The JTAG interface in the M32R/ECU is used to connect a JTAG emulator during debugging as well. In this chapter, the JTAG interface is explained assuming its use as an input/output path for boundary-scan test.

Functions of the JTAG interface-related pins mounted on the M32R/ECU are shown below.

Table 19.1.1 JTAG Pin Functions

Type	Pin Name	Signal Name	I/O	Function
TAP	JTCK	Test clock	Input	Clock input to the test circuit.
(Note 1)	JTDI	Test data Input	Input	Synchronous serial data input pin used to supply the test instruction code and test data. This input is sampled on the rising edge of JTCK.
	JTDO	Test data Output	Output	Synchronous serial data output pin used to output the test instruction code and test data. This signal changes state on the falling edge of JTCK, and is output in only the Shift-IR or Shift-DR state. Otherwise, it goes to a high-impedance state.
	JTMS	Test mode select	Input	Test mode select input to control the test circuit's state transition. This input is sampled on the rising edge of JTCK.
	JTRST	Test reset	Input	Active "L" test reset input to initialize the test circuit asynchronously. To ensure that the test circuit is reset without fail, JTMS input signal must be held "H" while this signal changes state from "L" to "H."

Note: TAP stands for Test Access Port (JTAG interface specified in IEEE 1149.1).

19.2 Configuration of JTAG Circuit

The JTAG circuit consists of the following circuit blocks.

- Instruction register to hold the instruction code that is fetched through the boundary-scan path
- A set of registers which are accessed through the boundary-scan path
- Test access port (abbreviated TAP) controller to control the JTAG unit's state transition
- Control logic to select input, output, etc.

The figure below shows the configuration of the JTAG circuit.

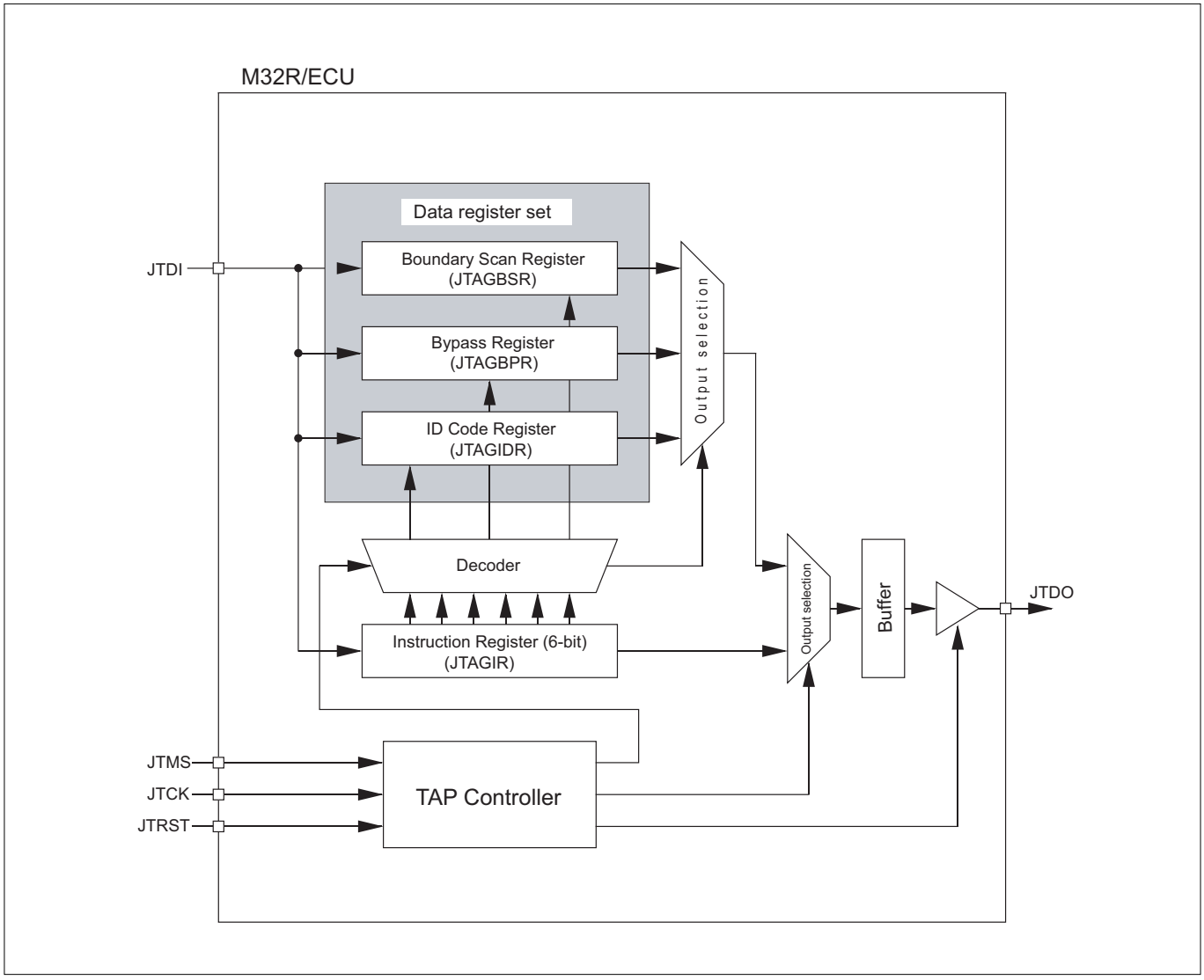


Figure 19.2.1 Configuration of the JTAG Circuit

19.3 JTAG Registers

19.3.1 Instruction Register (JTAGIR)

The Instruction Register is a 6-bit register to hold instruction code. This register is set in the IR path sequence. The instructions set in this register determine the data register to be selected in the subsequent DR path sequence.

The initial value of this register after test is reset (to initialize the test circuit) is b'000010 (IDCODE instruction). After a test reset, the ID Code Register is selected as the data register until instruction code is set by an external device. In the Capture-IR state, this register always has b'110001 (fixed value) loaded into it. Therefore, when in the Shift-IR state, no matter what value was set in this register, the value b'110001 is always output from the JTDO pin (sequentially beginning with the LSB). However, this value normally is not handled as instruction code.

Shown below is outside the scope of guaranteed operations. If this operation is attempted, the microcomputer may handle b'110001 as instruction code, which makes the microcomputer unable to operate normally.

Capture-IR → Exit1-IR → Update-IR

Following instructions are supported for the JTAG interface of the M32R/ECU:

- Three instructions specified as essential in IEEE 1149.1 (EXTEST, SAMPLE/PRELOAD, BYPASS)
- Device identification register access instruction (IDCODE)

Table 19.3.1 JTAG Instruction List

Instruction Code	Abbreviation	Operation
b'000000	EXTEST	Test the circuit/board-level connections external to the chip.
b'000001	SAMPLE/PRELOAD	Sample the operating status of the circuit and output the sampled status from the JTDO pin, while at the same time supplying the data used for boundary-scan test from the JTDI pin and preset it in the Boundary Scan Register.
b'000010	IDCODE	Select the ID Code Register to output the device and manufacturer identification data from the JTDO pin.
b'111111	BYPASS	Select the Bypass Register to inspect or set data.

Notes: • Do not set any other instruction code.
• For details about the IR path sequence, DR path sequence, test reset, Capture-IR state, Shift-IR state, Exit1-IR state and Update-IR state, see Section 19.4, "Basic Operation of JTAG."

19.3.2 Data Register

(1) Boundary Scan Register (JTAGBSR)

The Boundary Scan Register is a 475-bit register used to perform boundary-scan test. The bits in this register are assigned to each pin on the microcomputer.

Connected between the JTDI and JTDO pins, this register is selected when issuing EXTEST or SAMPLE/PRELOAD instruction. In the Capture-DR state, this register captures the status of input pins or internal logic outputs. In the Shift-DR state, while outputting the sampled value, this register receives the input data for boundary-scan test to set pin functions (direction of input/output and tristate output pins) and output values.

(2) Bypass Register (JTAGBPR)

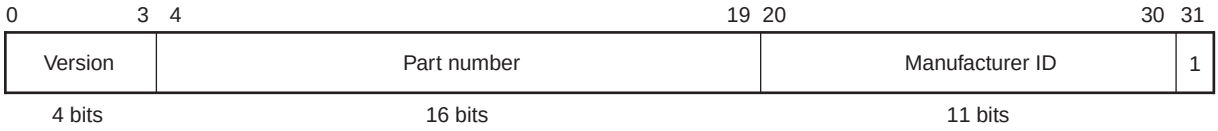
The Bypass Register is a 1-bit register used to bypass the boundary-scan path when the microcomputer is not the target of boundary-scan test. Connected between the JTDI and JTDO pins, this register is selected when issuing BYPASS instruction. This register is loaded with b'0 (fixed value) in the Capture-DR state.

(3) ID Code Register (JTAGIDR)

The ID Code Register is a 32-bit register used to identify the device and manufacturer. It holds the following information:

- Version information (4 bits) : b'0000
- Part number (16 bits) : b'0011 0010 0010 0100
- Manufacturer ID (11 bits) : b'000 0001 1100

This register is connected between the JTDI and JTDO pins, and is selected when issuing IDCODE instruction. This register is loaded with said IDCODE data in the Capture-DR state, and outputs it from the JTDO pin in the Shift-DR state.
The ID Code Register is a read-only register. Because the data written from the JTDI pin during DR path sequence is ignored, make sure JTDI input = "L" while in the Shift-DR state.



Note: • For details about the Capture-DR and Shift-DR states, see Section 19.4, "Basic Operation of JTAG."

19.4 Basic Operation of JTAG

19.4.1 Outline of JTAG Operation

The instruction and data registers basically are accessed in conjunction with the following three operations, which are performed based on the TAP Controller's state transition. The TAP Controller changes state according to JTMS input, and generates control signals required for operation in each state.

- **Capture operation**

  The result of boundary-scan test or the fixed data defined for each register is sampled. As a register operation, data input is latched into the shift register stage.

- **Shift operation**

  The register is accessed from outside through the boundary-scan path. The sample value is output to the outside at the same time data is set from the outside. As a register operation, the bits are shifted right between each shift register stage.

- **Update operation**

  The data set from the outside during shifting is driven. As a register operation, the value set in the shift register stage is transferred to the parallel output stage.
  The JTAG interface undergoes transition of the internal state depending on JTMS input and on such state transition, it performs the following two operations. In either case, the operation basically is performed in order of Capture → Shift → Update.

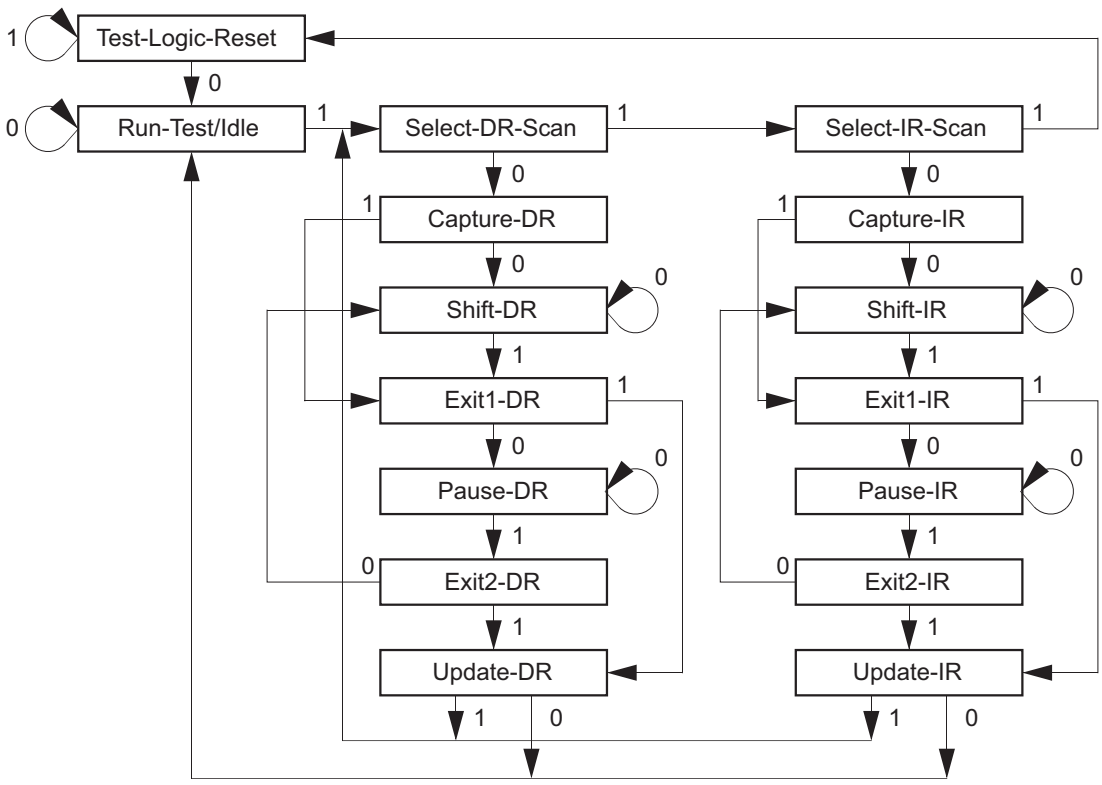
- **IR path sequence**

  Instruction code is set in the instruction register to select the data register to be operated on in the subsequent DR path sequence.

- **DR path sequence**

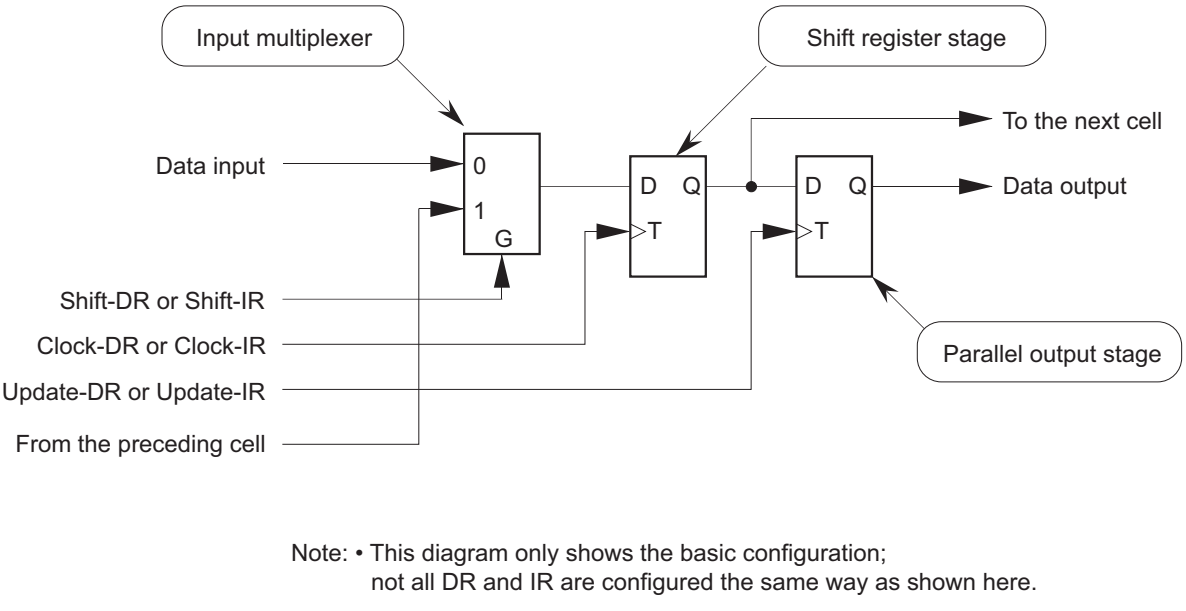
  Data inspection or setting is performed for the selected data register.

The state transition of the TAP Controller and the basic configuration of the JTAG related registers are shown below.



Note: • The values (0 or 1) in this diagram denote the state of JTMS input signal.

Figure 19.4.1 TAP Controller State Transition



Note: • This diagram only shows the basic configuration;
not all DR and IR are configured the same way as shown here.

Figure 19.4.2 Basic Configuration of the JTAG Related Registers

19.4.2 IR Path Sequence

Instruction code is set in the Instruction Register (JTAGIR) to select the data register to be accessed in the subsequent DR path sequence. The IR path sequence is performed following the procedure described below.

(1) From the Run-Test/Idle state, apply JTMS = "H" for a period of 2 JTCK cycles to enter the Select-IR-Scan state.

(2) Apply JTMS = "L" to enter the Capture-IR state. At this time, b'110001 (fixed value) is set in the Instruction Register's shift register stage.

(3) Proceed and apply JTMS = "L" to enter the Shift-IR state. In the Shift-IR state, the value of the shift register stage is shifted right one bit every cycle, and the data b'110001 (fixed value) that was set in (2) is serially output from the JTDO pin. At the same time, instruction code is set in the shift register stage bit by bit as it is serially fed from the JTDI pin. Because the instruction code is set in the Instruction Register that consists of 6 bits, the Shift-IR state must be continued for a period of 6 JTCK cycles. To stop the shift operation in the middle of the execution, enter the Pause-IR state via the Exit1-IR state (by setting JTMS input from "H" to "L"). To return from the Pause-IR state, enter the Shift-IR state via the Exit2-IR state (by setting JTMS input from "H" to "L").

(4) Apply JTMS = "H" to move from the Shift-IR state to the Exit1-IR state. This completes the shift operation.

(5) Proceed and apply JTMS = "H" to enter the Update-IR state. In the Update-IR state, the instruction code that was set in the Instruction Register's shift register stage is transferred to the Instruction Register's parallel output stage, and decoding of JTAG instruction is thereby started.

(6) Proceed and apply JTMS = "H" to enter the Select-DR-Scan state or JTMS = "L" to enter the Run-Test/Idle state.

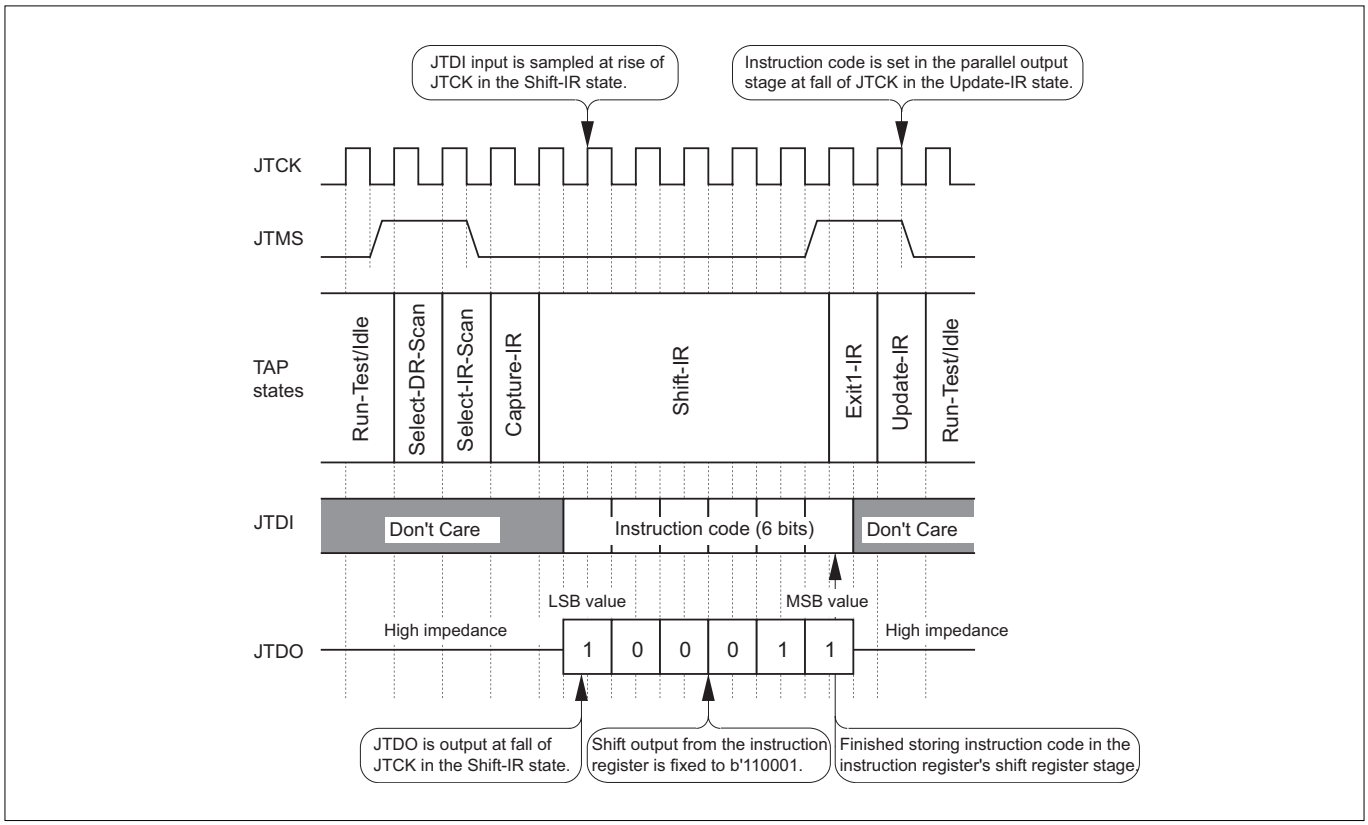


Figure 19.4.3 IR Path Sequence

19.4.3 DR Path Sequence

Data inspection or setting is performed for the data register selected in the IR path sequence prior to the DR path sequence. The DR path sequence is performed following the procedure described below.

(1) From the Run-Test/Idle state, apply JTMS = "H" for a period of 1 JTCK cycle to enter the Select-DR-Scan state. Which data register will be selected at this time depends on the instruction that was set during the IR path sequence performed prior to the DR path sequence.

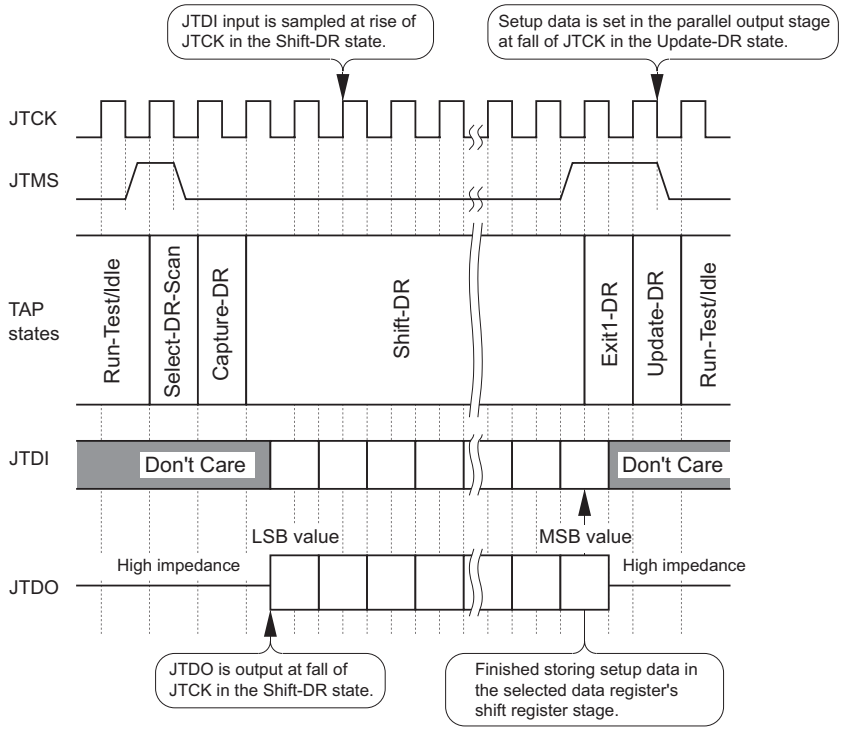
(2) Apply JTMS = "L" to enter the Capture-DR state. At this time, the result of boundary-scan test or the fixed data defined for each register is set in the data register's shift register stage.

(3) Proceed and apply JTMS = "L" to enter the Shift-DR state. In the Shift-DR state, the DR value is shifted right one bit every cycle, and the data that was set in (2) is serially output from the JTDO pin. At the same time, setup data is set in the data register's shift register stage bit by bit as it is serially fed from the JTDI pin. By continuing the Shift-IR state as long as the number of bits that comprise the selected data register (by applying JTMS = "L"), all bits of data can be set in and read out from the shift register stage. To stop the shift operation in the middle of the execution, enter the Pause-DR state via the Exit1-DR state (by setting JTMS input from "H" to "L"). To return from the Pause-DR state, enter the Shift-DR state via the Exit2-DR state (by setting JTMS input from "H" to "L").

(4) Apply JTMS = "H" to move from the Shift-DR state to the Exit1-DR state. This completes the shift operation.

(5) Proceed and apply JTMS = "H" to enter the Update-DR state. In the Update-DR state, the data that was set in the data register's shift register stage is transferred to the parallel output stage, and the setup data is thereby made ready for use.

(6) Proceed and apply JTMS = "H" to enter the Select-DR-Scan state or JTMS = "L" to enter the Run-Test/Idle state.



Note: • Because all bits in the data register's shift register stage also are shifted right, data is output from JTDO beginning with the LSB. Similarly, data is supplied to JTDI beginning with the LSB.

Figure 19.4.4 DR Path Sequence

19.4.4 Inspecting and Setting Data Registers

To inspect or set the data register, follow the procedure described below.

(1) To access the test access port (JTAG) for the first time, apply a test reset (to initialize the test circuit). One of the following two methods may be used to apply a test reset:

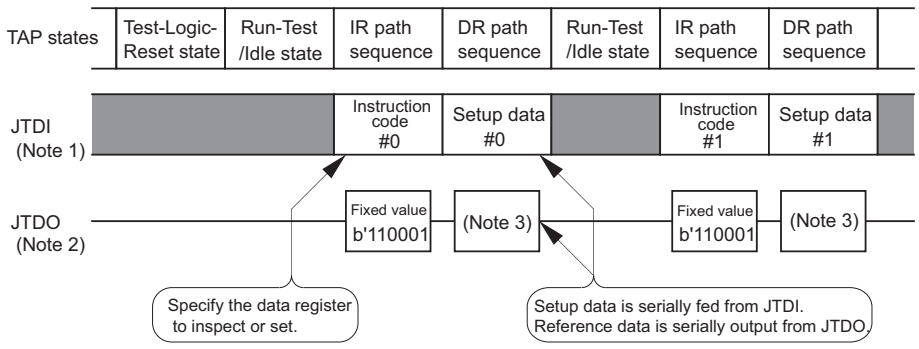
- Pull the JTRST pin "L."
- Drive the JTMS pin "H" to apply 5 or more JTCK cycles

(2) Apply JTMS = "L" to enter the Run-Test/Idle state. To continue the idle state, hold JTMS input "L."

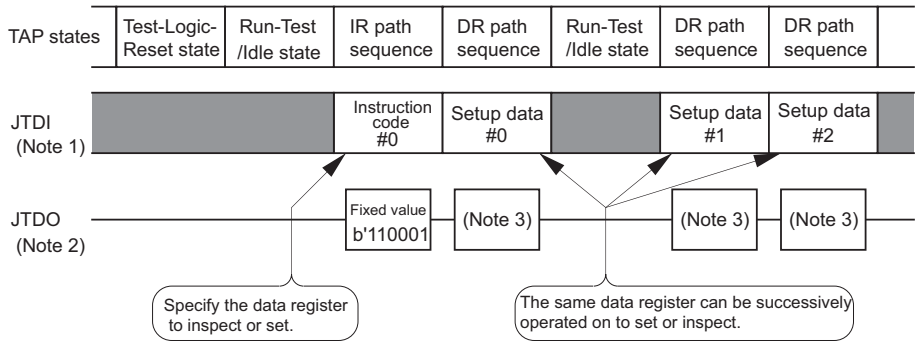
(3) Apply JTMS = "H" to exit the Run-Test/Idle state and perform IR path sequence. In the IR path sequence, specify the data register to inspect or set.

(4) Proceed to perform DR path sequence. Feed setup data from the JTDI pin into the data register specified in the IR path sequence, and read out reference data from the JTDO pin.

(5) To proceed to perform IR path or DR path sequence after the DR path sequence is completed, apply JTMS = "H" to return to the Select-DR-Scan state.
To wait for the next processing after a series of IR and DR sequence processing is completed, apply JTMS = "L" to enter the Run-Test/Idle state and keep that state.



(1) Basic access



(2) Successive accesses to the same data register

Note 1: The setup value for each register must be supplied to the JTDI pin beginning with the LSB.
Note 2: The value in each register is output to the JTDO pin beginning with the LSB. It is only in the Shift-IR state of IR path sequence and the Shift-DR state of DR path sequence that valid data is output from the JTDO pin. In all other states, the JTDO pin goes to a high-impedance state.
Note 3: This shows readout from the data register selected by the instruction that was set in the immediately preceding IR path sequence. The value sampled during Capture-DR state is output at the shift register stage of the selected data register.

Figure 19.4.5 Successive JTAG Access

19.5 Boundary Scan Description Language

The Boundary Scan Description Language (abbreviated BSDL) is described in the supplements to the Standard Test Access Port and Boundary-Scan Architecture of IEEE 1149.1-1990 and IEEE 1149.1a-1993. BSDL is a subset of IEEE 1076-1993 Standard VHSIC Hardware Description Language (VHDL). BSDL allows to precisely describe the functions of conforming components to be tested. For package connection test, this language is used by Automated Test Pattern Generation tools, and for synthesized test logic and verification, this language is used by Electronic Design Automation tools. BSDL provides powerful extended functions usable in internal test generation and necessary to write hardware debug and diagnostics software.
The primary section of BSDL has statements of logical port description, physical pin map, instruction set and boundary register description.

- **Logical port description**

  The logical port description assigns meaningful symbol names to each pin on the chip. The logic type of each pin, whether input, output, input/output, buffer or link, that defines the logical direction of signal flow is determined here.

- **Physical pin map**

  The physical pin map correlates the chip's logical ports to the physical pins on each package. By using separate names for each map, it is possible to define two or more physical pin maps in one BSDL description.

- **Instruction set statement**

  The instruction set statement writes bit patterns to be shifted in into the chip's instruction register. This bit pattern is necessary to place the chip into each test mode defined in standards. Instructions exclusive to the chip can also be written.

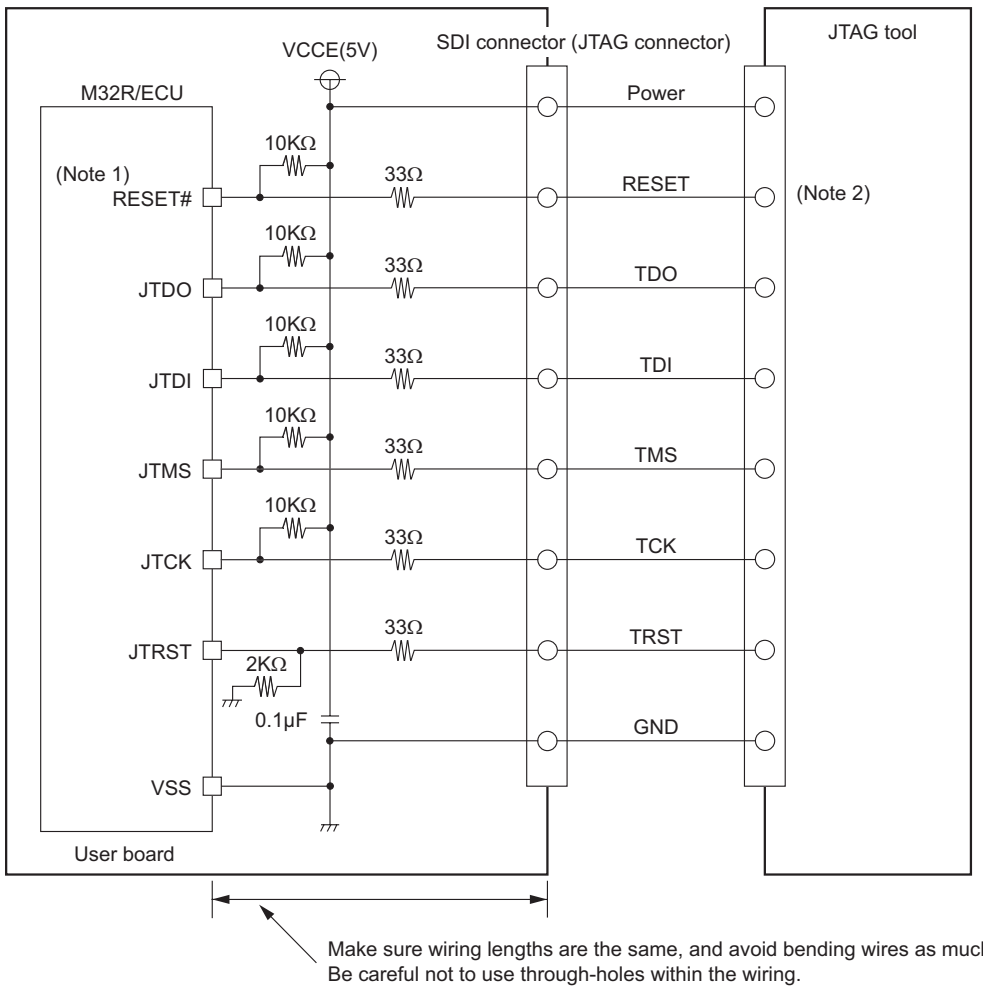
- **Boundary register description**

  The boundary register description is a list of boundary register cells or shift stages. Each cell is assigned a separate number. The cell with number 0 is located nearest to the test data output (JTDO) pin, and the cell with the largest number is located nearest to the test data input (JTDI) pin. Cells also contain related other information which includes cell type, logical port corresponding to the cell, logical function of the cell, safety value, control cell number, disable value and result value.

Note: • Boundary Scan Description Language (BSDL) can be downloaded from Renesas Technology Website after mass production.

19.6 Notes on Board Design when Connecting JTAG

To materialize fast and highly reliable communication with JTAG tools, make sure wiring lengths of JTAG pins are matched during board design.

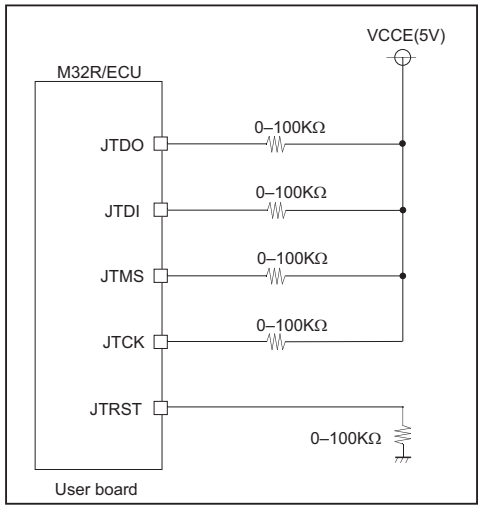


Note 1: The RESET# related circuit and resistance-capacitance values must be determined depending on the user board's system design conditions and the microcomputer's operating conditions.
Note 2: N-channel open-drain output is recommended for the RESET output of JTAG tools. For details, see JTAG tool specifications.
Notes: • Only if the JTRST pin is firmly tied to ground, the JTDO, JTDI, JTMS and JTCLK pins can be processed by either pullup or pulldown.
• Each of these pins must always be processed even when not using JTAG tools.
The same pullup/pulldown resistance values as when using JTAG tools may be used.

Figure 19.6.1 Notes on Board Design when Connecting JTAG Tools

19.7 Processing Pins when Not Using JTAG

The following shows how the pins on the chip should be processed when not using JTAG tools.



Note: • Only if the JTRST pin is firmly tied to ground, the JTDO, JTDI, JTMS and JTCLK pins can be processed by either pullup or pulldown.

Figure 19.7.1 Processing Pins when Not Using JTAG

This page is blank for reasons of layout.

CHAPTER 20

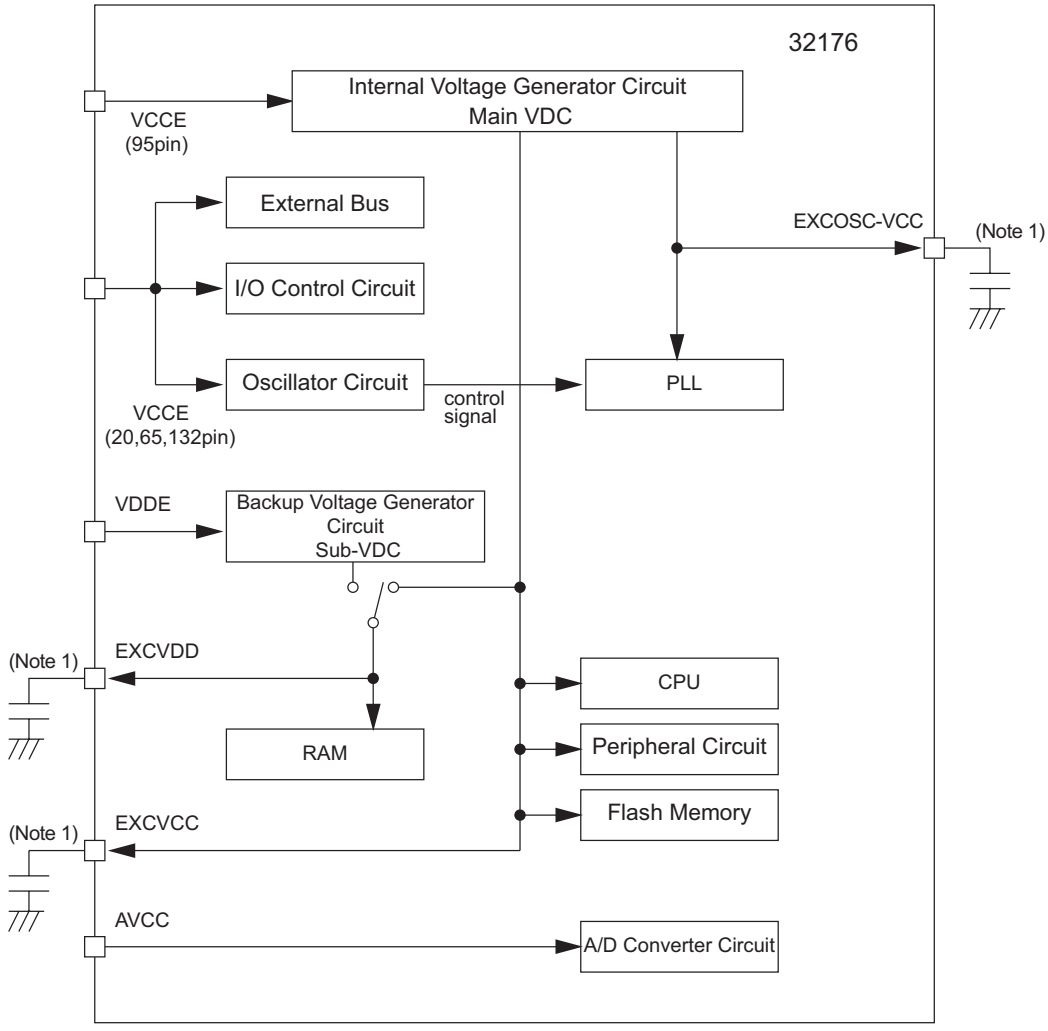
POWER SUPPLY CIRCUIT

20.1 Configuration of the Power Supply Circuit

The 32176 operates with a single 5 V ± 0.5 V or 3.3 V ± 0.3 V single power supply.
Unless otherwise noted, 5 V ± 0.5 V and 3.3 V ± 0.3 V in this chapter are referred to simply by 5 V and 3.3 V, respectively.

Table 20.1.1 Power Supply Functions

Power Supply Type	Pin Name	Function
5.0V or 3.3V	VCCE	Main power supply
	AVCC0	Power supply for the A/D converter
	VREF0	Reference voltage for the A/D converter
	VDDE	Power supply for the internal RAM backup
	EXCVCC	External capacitor connection pin
	EXCVDD	External capacitor connection pin
	EXCOSC-VCC	External capacitor connection pin



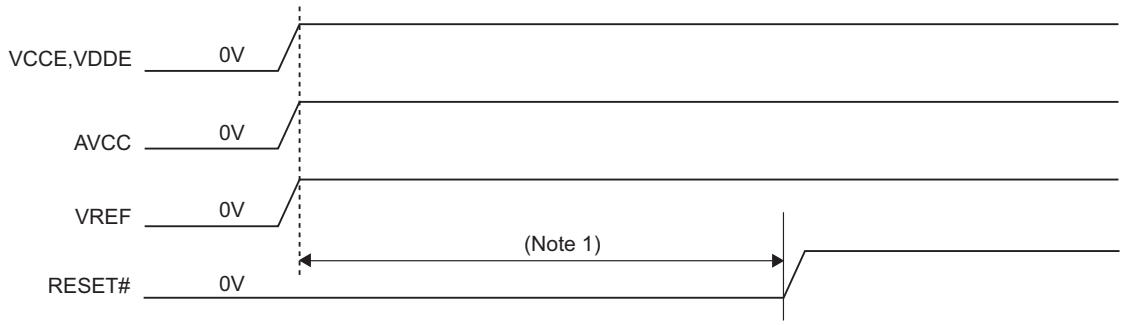
Note 1: Do not apply power voltage to the EXCVDD, EXCVCC and EXCOSC-VCC pins. And External Capacitance for Power Supplies refer to Chapter 21 ELECTRICAL CHARACTERISTICS.

Figure 20.1.1 Configuration of the Power Supply Circuit (VCCE = 5.0 V or 3.3 V)

20.2 Power-On Sequence

20.2.1 Power-On Sequence when Not Using RAM Backup

The diagram below shows a turn-on sequence of the power supply (5.0 V or 3.3 V) when not using RAM backup.



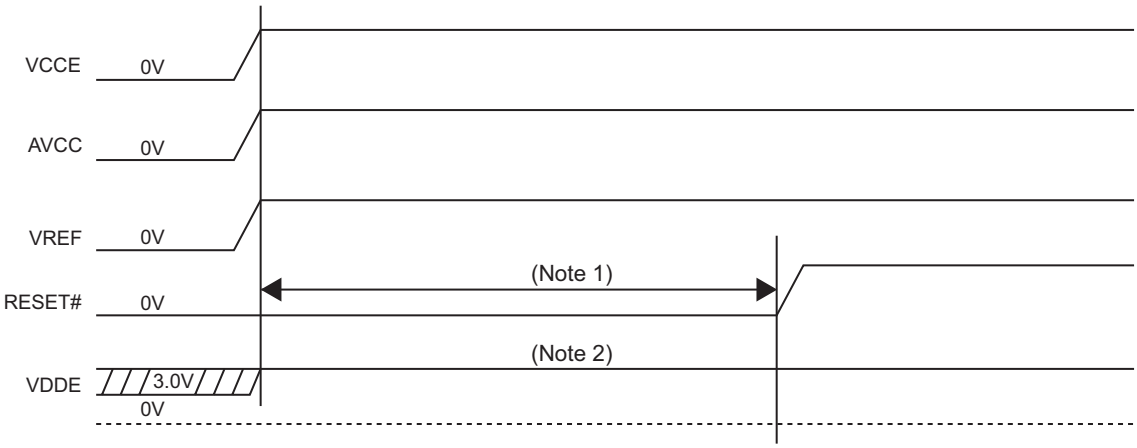
Note 1: After turning on all power supplies and holding the RESET# pin "L" for an oscillation stabilization time, release the RESET# pin back "H" (to exit the reset state).

Notes: • Power-on limitation
VDDE ≥ VCCE
In addition, when VDDE is more than 3.0 V, it is not problem even if it cannot fulfill restrictions (VDDE≥VCCE).
• However, if it cannot fulfill restrictions (VDDE≥VCCE) when Power-On, sufficient evaluation must be made during system design in order to ensure that no power will be applied to the microcomputer with a potential difference of 1 V or more.
For potential differences 0 V to 0.6 V, there is almost no in-flow current. The amount of in-flow current begins to increase when the potential difference exceeds 0.6 V.

Figure 20.2.1 Power-On Sequence when Not Using RAM Backup

20.2.2 Power-On Sequence when Using RAM Backup

The diagram below shows a turn-on sequence of the power supply (5.0 V or 3.3 V) when using RAM backup.



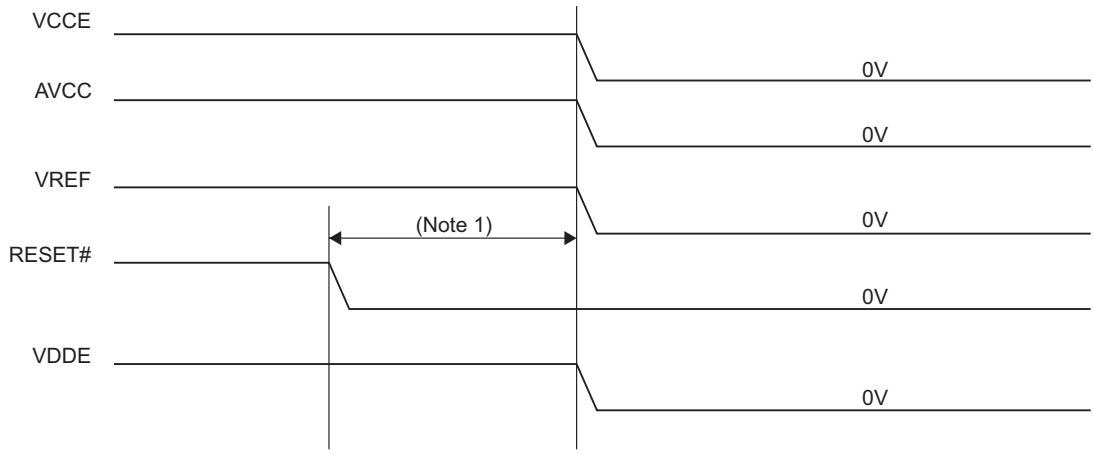
Note 1: After turning on all power supplies and holding the RESET# pin "L" for an oscillation stabilization time, release the RESET# pin back "H" (to exit the reset state).
Note 2: Because of RAM backup, it is assumed that VDDE is 3.0 V or more. The diagram here is shown for the VCCE = 5 V case.
Notes: • Power-on limitation
VDDE ≥ VCCE
In addition, when VDDE is more than 3.0 V, it is not problem even if it cannot fulfill restrictions (VDDE≥VCCE).
• However, if it cannot fulfill restrictions (VDDE≥VCCE) when Power-On, sufficient evaluation must be made during system design in order to ensure that no power will be applied to the microcomputer with a potential difference of 1 V or more.
For potential differences 0 V to 0.6 V, there is almost no in-flow current. The amount of in-flow current begins to increase when the potential difference exceeds 0.6 V.

Figure 20.2.2 Power-On Sequence when Using RAM Backup

20.3 Power-Off Sequence

20.3.1 Power-Off Sequence when Not Using RAM Backup

The diagram below shows a turn-off sequence of the power supply (5.0 V or 3.3 V) when using RAM backup.

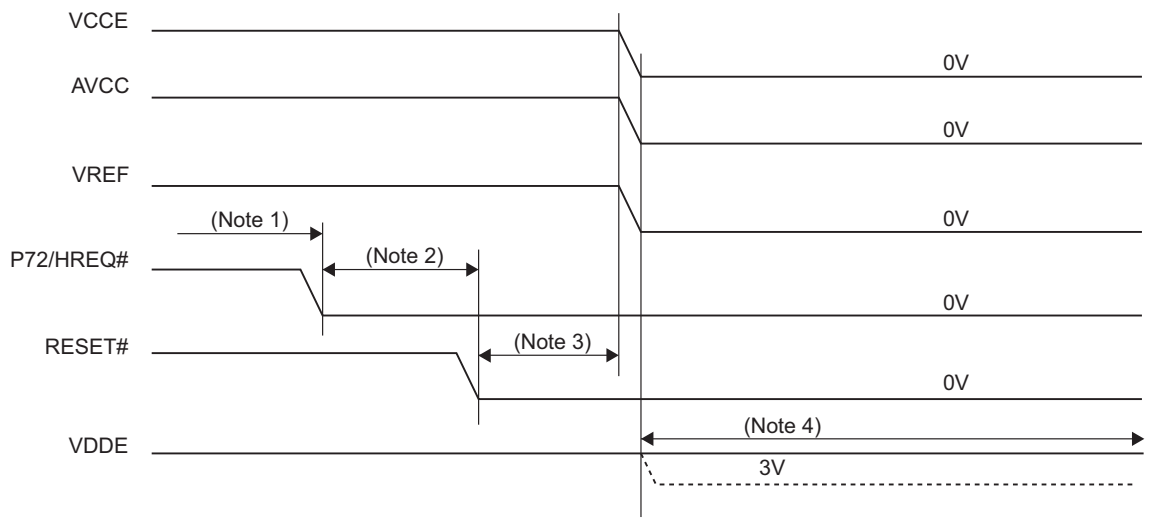


Note 1: Wait until the RESET# pin goes "L" before turning the power supply off.
Notes: • Power-off limitation
VDDE ≥ VCCE
In addition, when VDDE is more than 3.0 V, it is not problem even if it cannot fulfill restrictions (VDDE≥VCCE).
• However, if it cannot fulfill restrictions (VDDE≥VCCE) when Power-Off, sufficient evaluation must be made during system design in order to ensure that no power will be applied to the microcomputer with a potential difference of 1 V or more.
For potential differences 0 V to 0.6 V, there is almost no in-flow current. The amount of in-flow current begins to increase when the potential difference exceeds 0.6 V.

Figure 20.3.1 Power-Off Sequence when Not Using RAM Backup

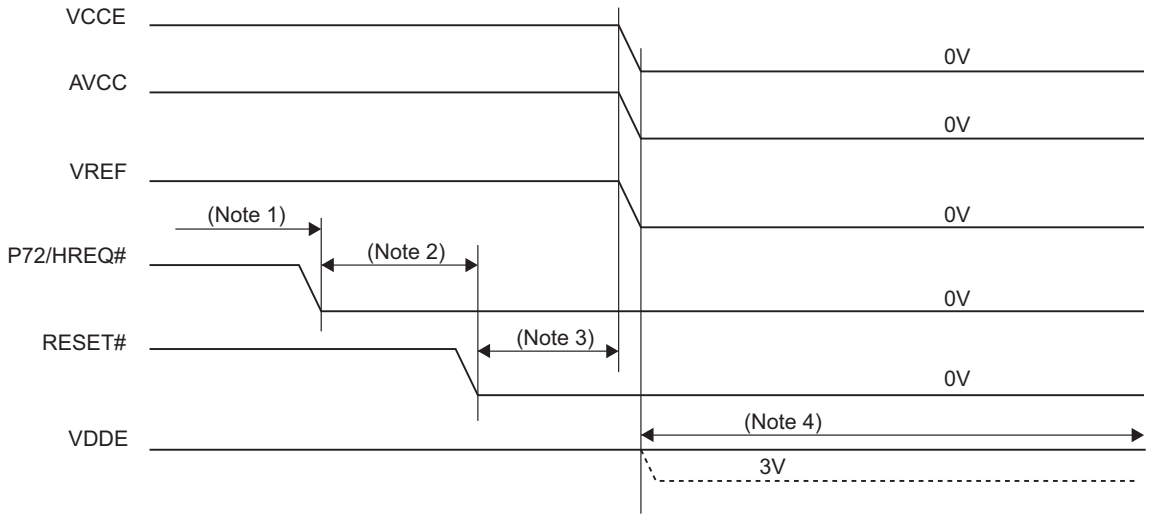
20.3.2 Power-Off Sequence when Using RAM Backup

The diagram below shows a turn-off sequence of the power supply (5.0 V or 3.3 V) when using RAM backup with HREQ function.



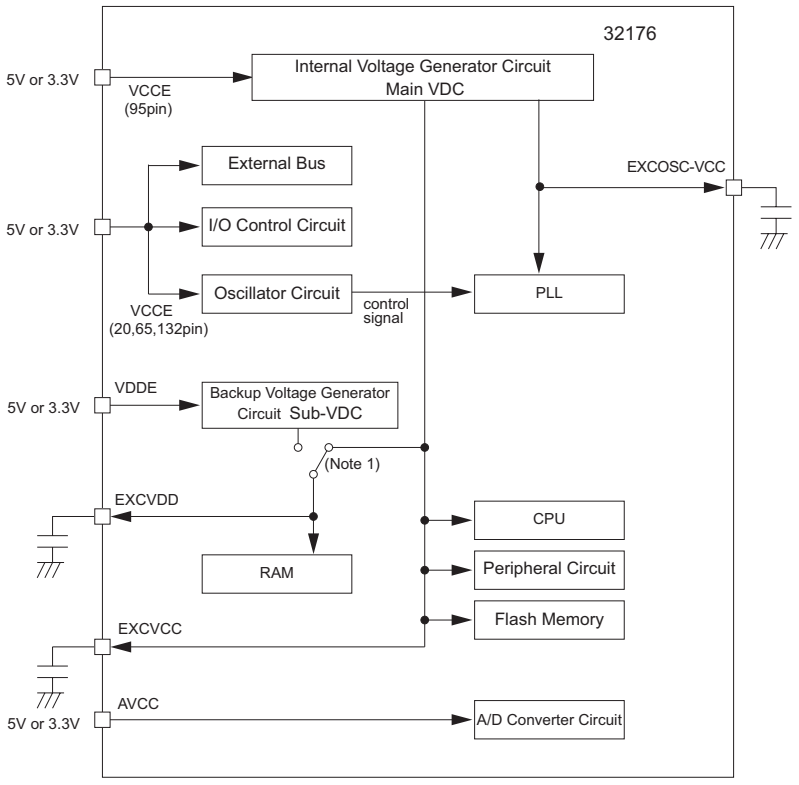
Note 1: Pull the HREQ# input pin "L" to halt the CPU at the end of the bus cycle.
Or disable RAM access in software. P72 can be used as HREQ# irrespective of the operation mode.
However, HREQ# must be selected with the Port Operation Mode Register for P72.
Note 2: Pull the RESET# input pin "L" while the CPU is halted or RAM access is disabled.
Note 3: Wait until the RESET# pin goes "L" before turning the power supply off.
Note 4: Lower the VDDE voltage from 5.0 V to 3.0 V as necessary.
Notes: • Power-off limitation
VDDE ≥ VCCE
In addition, when VDDE is more than 3.0 V, it is not problem even if cannot fulfill restrictions (VDDE≥VCCE).
• However, if it cannot fulfill restrictions (VDDE≥VCCE) when Power-Off, sufficient evaluation must be made during system design in order to ensure that no power will be applied to the microcomputer with a potential difference of 1 V or more.
For potential differences 0 V to 0.6 V, there is almost no in-flow current. The amount of in-flow current begins to increase when the potential difference exceeds 0.6 V.

Figure 20.3.2 Power-Off Sequence when Using RAM Backup (VCCE = 5.0 V or 3.3 V)



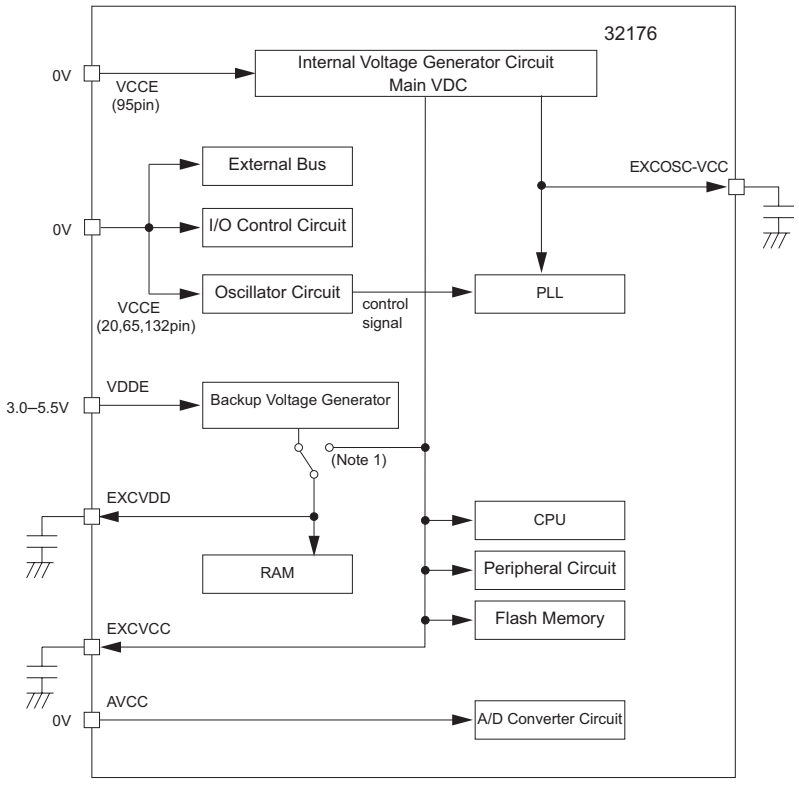
Note 1: Pull the HREQ# input pin "L" to halt the CPU at the end of the bus cycle.
Or disable RAM access in software. P72 can be used as HREQ# irrespective of the operation mode.
However, HREQ# must be selected with the Port Operation Mode Register for P72.
Note 2: Pull the RESET# input pin "L" while the CPU is halted or RAM access is disabled.
Note 3: Wait until the RESET# pin goes "L" before turning the power supply off.
Note 4: Lower the VDDE voltage from 5.0 V to 3.0 V as necessary.
Notes: • Power-off limitation
VDDE ≥ VCCE
In addition, when VDDE is more than 3.0 V, it is not problem even if it cannot fulfill restrictions (VDDE≥VCCE).
• However, if it cannot fulfill restrictions (VDDE≥VCCE) when Power-Off, sufficient evaluation must be made during system design in order to ensure that no power will be applied to the microcomputer with a potential difference of 1 V or more.
For potential differences 0 V to 0.6 V, there is almost no in-flow current. The amount of in-flow current begins to increase when the potential difference exceeds 0.6 V.

Figure 20.3.3 Power-Off Sequence when Using RAM Backup (VCCE = 5.0 V, VDDE = 3.3 V)



Note 1: When the microcomputer is ready to operate, it automatically changes to the main VDC.

Figure 20.3.4 Microcomputer Ready to Operate State (VCCE = 5.0 V or 3.3 V)



Note 1: During RAM backup mode, it automatically changes to the sub VDC, allowing to save on power consumption in the chip.

Figure 20.3.5 SRAM Data Backup State (VCCE = 5.0 V or 3.3 V)

CHAPTER 21

ELECTRICAL CHARACTERISTICS

21.1 Absolute Maximum Ratings

Absolute Maximum Ratings

Symbol	Parameter	Test Condition	Rated Value	Unit
VCCE	Main Power Supply		-0.3–6.5	V
VDDE	RAM Power Supply		-0.3–6.5	V
AVCC	Analog Power Supply	VCCE≥AVCC≥VREF	-0.3–6.5	V
VREF	Reference Voltage Input	VCCE≥AVCC≥VREF	-0.3–6.5	V
VI	XIN		-0.3–VCCE+0.3	V
	Other		-0.3–VCCE+0.3	V
VO	XOUT		-0.3–VCCE+0.3	V
	Other		-0.3–VCCE+0.3	V
Pd	Power Dissipation	Ta=-40–85°C	500	mW
		Ta=-40–125°C	400	mW
TOPR	Operating Ambient Temperature (Note 1)		-40–125	°C
Tstg	Storage Temperature		-65–150	°C

Note 1: This does not guarantee that the microcomputer can operate continuously at 125°C. Consult Renesas if the microcomputer is going to be used for 125°C applications.

21.2 Electrical Characteristics when VCCE = 5 V, f(XIN) = 10 MHz

21.2.1 Recommended Operating Conditions (when VCCE = 5 V, f(XIN) = 10 MHz)

Recommended Operating Conditions (Referenced to VCCE, VDDE = 5 V ± 0.5 V, Ta = –40°C to 85°C Unless Otherwise Noted)

Symbol	Parameter				Rated Value			Unit
					MIN	TYP	MAX	
VCCE	Main Power Supply (Note 1)				4.5	5.0	5.5	V
VDDE	RAM Power Supply (Note 1)				4.5	5.0	5.5	V
AVCC	Analog Power Supply (Note 1)				4.5	5.0	5.5	V
VREF	Reference Voltage Input (Note 1)				4.5	5.0	5.5	V
VIH	Input "H" Voltage	When threshold switching function is used	When CMOS input is selected	Threshold selection : 0.35 VCCE	0.45VCCE		VCCE	V
				Threshold selection : 0.5VCCE	0.6VCCE		VCCE	V
				Threshold selection : 0.7VCCE	0.8VCCE		VCCE	V
			When Schmitt input is selected	VT+/VT- : 0.5VCCE/0.35VCCE	0.6VCCE		VCCE	V
				VT+/VT- : 0.7VCCE/0.35VCCE	0.8VCCE		VCCE	V
				VT+/VT- : 0.7VCCE/0.5VCCE	0.8VCCE		VCCE	V
		FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#			0.8VCCE		VCCE	V
		Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1			0.8VCCE		VCCE	V
		Standard input for the following pins: DB0–15, WAIT#			0.43VCCE		VCCE	V
		Standard input for the following pins: SBI#, HREQ#			0.6VCCE		VCCE	V
VIL	Input "L" Voltage	When threshold switching function is used	When CMOS input is selected	Threshold selection : 0.35VCCE	0		0.25VCCE	V
				Threshold selection : 0.5VCCE	0		0.4VCCE	V
				Threshold selection : 0.7VCCE	0		0.6VCCE	V
			When Schmitt input is selected	VT+/VT- : 0.5VCCE/0.35VCCE	0		0.25VCCE	V
				VT+/VT- : 0.7VCCE/0.35VCCE	0		0.25VCCE	V
				VT+/VT- : 0.7VCCE/0.5VCCE	0		0.4VCCE	V

Symbol	Parameter		Rated Value			Unit
			MIN	TYP	MAX	
VIL	Input "L" Voltage	FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#	0		0.2VCCE	V
		Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1	0		0.25VCCE	V
		Standard input for the following pins: DB0–15, WAIT#	0		0.16VCCE	V
		Standard input for the following pins: SBI#, HREQ#	0		0.25VCCE	V
IOH(peak)	"H" State Peak Output Current P0–P22 (Note 2)				-10	mA
IOH(avg)	"H" State Average Output Current P0–P22 (Note 3)				-5	mA
IOL(peak)	"L" State Peak Output Current P0–P22 (Note 2)				10	mA
IOL(avg)	"L" State Average Output Current P0–P22 (Note 3)				5	mA
CL	Output Load Capacitance	JTDO, JTMS			80	pF
		Other than above	15		50	pF
f(XIN)	External Clock Input Frequency		5		10	MHz

Note 1: Subject to conditions VCCE ≥ AVCC ≥ VREF
Note 2: Make sure the total output current (peak) of ports are
| ports P0 + P1 + P2 | ≤ 80 mA
| ports P3 + P4 + P13 + P15 + P22 | ≤ 80 mA
| ports P6 + P7 + P8 + P9 + P17 | ≤ 80 mA
| ports P10 + P11 + P12 | ≤ 80 mA
Note 3: The average output current is a value averaged during a 100 ms period.

21.2.2 D.C. Characteristics (when VCCE = 5 V, f(XIN) = 10 MHz)

Electrical Characteristics (Referenced to VCCE, VDDE = 5 V ± 0.5 V, Ta = –40°C to 85°C Unless Otherwise Noted)

Symbol	Parameter		Test Condition	Rated Value			Unit
				MIN	TYP	MAX	
VOH	Output "H" Voltage		IOH≥-5mA	VCCE+0.165 ×IOH(mA)		VCCE	V
VOL	Output "L" Voltage		IOL≤5mA	0		0.15×IOL (mA)	V
VDDE	RAM Retention Power Supply Voltage		When operating	4.5		5.5	V
			During backup	3.0		5.5	V
IIH	"H" State Input Current		VI=VCCE	-5		5	µA
IIL	"L" State Input Current		VI=0V	-5		5	µA
ICC	Total Power Supply Current (Note 1)		f(XIN)=10.0MHz, When reset			50	mA
			f(XIN)=10.0MHz, When operating		65	90	
IDDEhold	RAM Retention Power Supply Current	VDDE=5.5V	Ta=25C°		0.1	2	µA
			Ta=85C°			50	
		VDDE=3.0V	Ta=25C°		0.05	1	
			Ta=85C°			25	
VT+- VT-	FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#			1.0			V
	Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1			1.0			
	Standard input for the following pins: SBI#, HREQ#			0.3			
	When threshold switching function is used (VT+ / VT–)	0.7VCCE/0.35VCCE		1.0			
		0.7VCCE/0.5VCCE		0.3			
		0.5VCCE/0.35VCCE		0.3			

Note 1: Total amount of current when VCCE = VDDE = AVCC = VREF in single-chip mode

Electrical Characteristics of Each Power Supply Pin

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
ICCE	VCCE Power Supply Current When Operating	f(XIN)=10.0MHz			90	mA
IDDE	VDDE Power Supply Current When Operating	f(XIN)=10.0MHz			1	mA
IAVCC	AVCC Power Supply Current When Operating	f(XIN)=10.0MHz			3	mA
IVREF	VREF Power Supply Current When Operating	f(XIN)=10.0MHz			1	mA

21.2.3 A/D Conversion Characteristics (when VCCE = 5 V, f(XIN) = 10 MHz)

A/D Conversion Characteristics (Referenced to VCCE, VDDE = 5.12 V, Ta = –40°C to 85°C Unless Otherwise Noted)

Symbol	Parameter				Test Condition	Rated Value			Unit
						MIN	TYP	MAX	
–	Resolution				VREF=VCCE=AVCC			10	bits
–	Absolute Accuracy (Note 1)	Without sample-and-hold or during normal sample-and-hold	Slow mode	Normal speed				±2	LSB
				Double speed				±2	
			Fast mode	Normal speed				±3	
				Double speed				±3	
		During fast sample-and-hold	Slow mode	Normal speed				±3	
				Double speed				±3	
			Fast mode	Normal speed				±3	
				Double speed				±8	
TCONV	Conversion Time	Without sample-and-hold or during normal sample-and-hold	Slow mode	Normal speed		14.95			µs
				Double speed		8.65			
			Fast mode	Normal speed		6.55			
				Double speed		4.45			
		During fast sample-and-hold	Slow mode	Normal speed		9.55			
				Double speed		5.05			
			Fast mode	Normal speed		4.75			
				Double speed		2.65			
IIAN	Analog Input Leakage Current (Note 2)				AVSS≤AD0INi≤AVCC	-5		5	µA

Note 1: Absolute accuracy refers to the accuracy of output code relative to the analog input including all error sources (including quantization error) in an A/D converter, and is calculated using the equation below.

Absolute accuracy = output code – (analog input voltage AD0INi / 1 LSB)

When AVCC = AVREF = 5.12 V, 1 LSB = 5 mV.

Note 2: This refers to the input leakage current on AD0INi while the A/D converter remains idle.

21.3 Electrical Characteristics when VCCE = 5 V, f(XIN) = 8 MHz

21.3.1 Recommended Operating Conditions (when VCCE = 5 V, f(XIN) = 8 MHz)

Recommended Operating Conditions (Referenced to VCCE, VDDE = 5 V ± 0.5 V, Ta = –40°C to 125°C Unless Otherwise Noted)

Symbol	Parameter				Rated Value			Unit
					MIN	TYP	MAX	
VCCE	Main Power Supply (Note 1)				4.5	5.0	5.5	V
VDDE	RAM Power Supply (Note 1)				4.5	5.0	5.5	V
AVCC	Analog Power Supply (Note 1)				4.5	5.0	5.5	V
VREF	Reference Voltage Input (Note 1)				4.5	5.0	5.5	V
VIH	Input "H" Voltage	When threshold switching function is used	When CMOS input is selected	Threshold selection : 0.35VCCE	0.45VCCE		VCCE	V
				Threshold selection : 0.5VCCE	0.6VCCE		VCCE	V
				Threshold selection : 0.7VCCE	0.8VCCE		VCCE	V
			When Schmitt input is selected	VT+/VT- : 0.5VCCE/0.35VCCE	0.6VCCE		VCCE	V
				VT+/VT- : 0.7VCCE/0.35VCCE	0.8VCCE		VCCE	V
				VT+/VT- : 0.7VCCE/0.5VCCE	0.8VCCE		VCCE	V
		FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#			0.8VCCE		VCCE	V
		Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1			0.8VCCE		VCCE	V
		Standard input for the following pins: DB0–15, WAIT#			0.43VCCE		VCCE	V
		Standard input for the following pins: SBI#, HREQ#			0.6VCCE		VCCE	V
VIL	Input "L" Voltage	When threshold switching function is used	When CMOS input is selected	Threshold selection : 0.35VCCE	0		0.25VCCE	V
				Threshold selection : 0.5VCCE	0		0.4VCCE	V
				Threshold selection : 0.7VCCE	0		0.6VCCE	V
			When Schmitt input is selected	VT+/VT- : 0.5VCCE/0.35VCCE	0		0.25VCCE	V
				VT+/VT- : 0.7VCCE/0.35VCCE	0		0.25VCCE	V
				VT+/VT- : 0.7VCCE/0.5VCCE	0		0.4VCCE	V

Symbol	Parameter		Rated Value			Unit
			MIN	TYP	MAX	
VIL	Input "L" Voltage	FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#	0		0.2VCCE	V
		Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1	0		0.25VCCE	V
		Standard input for the following pins: DB0–15, WAIT#	0		0.16VCCE	V
		Standard input for the following pins: SBI#, HREQ#	0		0.25VCCE	V
IOH(peak)	"H" State Peak Output Current P0–P22 (Note 2)				-10	mA
IOH(avg)	"H" State Average Output Current P0–P22 (Note 3)				-5	mA
IOL(peak)	"L" State Peak Output Current P0–P22 (Note 2)				10	mA
IOL(avg)	"L" State Average Output Current P0–P22 (Note 3)				5	mA
CL	Output Load Capacitance	JTDO, JTMS			80	pF
		Other than above	15		50	pF
f(XIN)	External Clock Input Frequency		5		8	MHz

Note 1: Subject to conditions VCCE ≥ AVCC ≥ VREF
Note 2: Make sure the total output current (peak) of ports are
| ports P0 + P1 + P2 | ≤ 80 mA
| ports P3 + P4 + P13 + P15 + P22 | ≤ 80 mA
| ports P6 + P7 + P8 + P9 + P17 | ≤ 80 mA
| ports P10 + P11 + P12 | ≤ 80 mA
Note 3: The average output current is a value averaged during a 100 ms period.

21.3.2 D.C. Characteristics (when VCCE = 5 V, f(XIN) = 8 MHz)

Electrical Characteristics (Referenced to VCCE, VDDE = 5 V ± 0.5 V, Ta = –40°C to 125°C Unless Otherwise Noted)

Symbol	Parameter		Test Condition	Rated Value			Unit
				MIN	TYP	MAX	
VOH	Output "H" Voltage		IOH≥-5mA	VCCE+0.165 ×IOH(mA)		VCCE	V
VOL	Output "L" Voltage		IOL≤5mA	0		0.15×IOL (mA)	V
VDDE	RAM Retention Power Supply Voltage		When operating	4.5		5.5	V
			During backup	3.0		5.5	V
IIH	"H" State Input Current		VI=VCCE	-5		5	µA
IIL	"L" State Input Current		VI=0V	-5		5	µA
ICC	Total Power Supply Current (Note 1)		f(XIN)=8.0MHz, When reset			40	mA
			f(XIN)=8.0MHz, When operating		55	70	
IDDEhold	RAM Retention Power Supply Current	VDDE=5.5V	Ta=25°C		0.1	2	µA
			Ta=125°C			200	
		VDDE=3.0V	Ta=25°C		0.05	1	
			Ta=125°C			100	
VT+- VT-	FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#			1.0			V
	Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1			1.0			
	Standard input for the following pins: SBI#, HREQ#			0.3			
	When threshold switching function is used (VT+ / VT–)	0.7VCCE/0.35VCCE		1.0			
		0.7VCCE/0.5VCCE		0.3			
		0.5VCCE/0.35VCCE		0.3			

Note 1: Total amount of current when VCCE = VDDE = AVCC = VREF in single-chip mode

Electrical Characteristics of Each Power Supply Pin

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
ICCE	VCCE Power Supply Current When Operating	f(XIN)=8.0MHz			70	mA
IDDE	VDDE Power Supply Current When Operating	f(XIN)=8.0MHz			1	mA
IAVCC	AVCC Power Supply Current When Operating	f(XIN)=8.0MHz			3	mA
IVREF	VREF Power Supply Current When Operating	f(XIN)=8.0MHz			1	mA

21.3.3 A/D Conversion Characteristics (when VCCE = 5 V, f(XIN) = 8 MHz)

A/D Conversion Characteristics (Referenced to VCCE, VDDE = 5.12 V, Ta = –40°C to 125°C Unless Otherwise Noted)

Symbol	Parameter				Test Condition	Rated Value			Unit
						MIN	TYP	MAX	
–	Resolution				VREF=VCCE=AVCC			10	bits
–	Absolute Accuracy (Note 1)	Without sample-and-hold or during normal sample-and-hold	Slow mode	Normal speed				±2	LSB
				Double speed				±2	
			Fast mode	Normal speed				±3	
				Double speed				±3	
		During fast sample-and-hold	Slow mode	Normal speed				±3	
				Double speed				±3	
			Fast mode	Normal speed				±3	
				Double speed				±8	
TCONV	Conversion Time	Without sample-and-hold or during normal sample-and-hold	Slow mode	Normal speed		18.6875			µs
				Double speed		10.8125			
			Fast mode	Normal speed		8.1875			
				Double speed		5.5625			
		During fast sample-and-hold	Slow mode	Normal speed		11.9375			
				Double speed		6.3125			
			Fast mode	Normal speed		5.9375			
				Double speed		3.3125			
IIAN	Analog Input Leakage Current (Note 2)				AVSS≤AD0INi≤AVCC	-5		5	µA

Note 1: Absolute accuracy refers to the accuracy of output code relative to the analog input including all error sources (including quantization error) in an A/D converter, and is calculated using the equation below.

Absolute accuracy = output code – (analog input voltage AD0INi / 1 LSB)

When AVCC = AVREF = 5.12 V, 1 LSB = 5 mV.

Note 2: This refers to the input leakage current on AD0INi while the A/D converter remains idle.

21.4 Electrical Characteristics when VCCE = 3.3 V, f(XIN) = 10 MHz

21.4.1 Recommended Operating Conditions (when VCCE = 3.3 V ± 0.3 V, f(XIN) = 10 MHz)

Recommended Operating Conditions (Referenced to VCCE, VDDE = 3.3 V ± 0.3 V, Ta = –40°C to 85°C Unless Otherwise Noted)

Symbol	Parameter				Rated Value			Unit
					MIN	TYP	MAX	
VCCE	Main Power Supply (Note 1)				3.0	3.3	3.6	V
VDDE	RAM Power Supply (Note 1)				3.0	VCCE	3.6	V
AVCC	Analog Power Supply (Note 1)				3.0	VCCE	3.6	V
VREF	Reference Voltage Input (Note 1)				3.0	VCCE	3.6	V
VIH	Input "H" Voltage	When threshold switching function is used	When CMOS input is selected	Threshold selection : 0.35VCCE	0.5VCCE		VCCE	V
				Threshold selection : 0.5VCCE	0.65VCCE		VCCE	V
				Threshold selection : 0.7VCCE	0.8VCCE		VCCE	V
			When Schmitt input is selected	VT+/VT- : 0.5VCCE/0.35VCCE	0.65VCCE		VCCE	V
				VT+/VT- : 0.7VCCE/0.35VCCE	0.8VCCE		VCCE	V
				VT+/VT- : 0.7VCCE/0.5VCCE	0.8VCCE		VCCE	V
		FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#			0.8VCCE		VCCE	V
		Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1			0.8VCCE		VCCE	V
		Standard input for the following pins: DB0–15, WAIT#			0.5VCCE		VCCE	V
		Standard input for the following pins: SBI#, HREQ#			0.65VCCE		VCCE	V
VIL	Input "L" Voltage	When threshold switching function is used	When CMOS input is selected	Threshold selection : 0.35VCCE	0		0.2VCCE	V
				Threshold selection : 0.5VCCE	0		0.35VCCE	V
				Threshold selection : 0.7VCCE	0		0.5VCCE	V
			When Schmitt input is selected	VT+/VT- : 0.5VCCE/0.35VCCE	0		0.2VCCE	V
				VT+/VT- : 0.7VCCE/0.35VCCE	0		0.2VCCE	V
				VT+/VT- : 0.7VCCE/0.5VCCE	0		0.35VCCE	V

Symbol	Parameter		Rated Value			Unit
			MIN	TYP	MAX	
VIL	Input "L" Voltage	FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#	0		0.2VCCE	V
		Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN1, TIN16–TIN23, CRX0, CRX1	0		0.2VCCE	V
		Standard input for the following pins: DB0–15, WAIT#	0		0.2VCCE	V
		Standard input for the following pins: SBI#, HREQ#	0		0.2VCCE	V
IOH(peak)	"H" State Peak Output Current P0–P22 (Note 2)				-10	mA
IOH(avg)	"H" State Average Output Current P0–P22 (Note 3)				-5	mA
IOL(peak)	"L" State Peak Output Current P0–P22 (Note 2)				10	mA
IOL(avg)	"L" State Average Output Current P0–P22 (Note 3)				5	mA
CL	Output Load Capacitance	JTDO, JTMS			80	pF
		Other than above	15		50	pF
f(XIN)	External Clock Input Frequency		5		10	MHz

Note 1: Subject to conditions VCCE ≥ AVCC ≥ VREF
Note 2: Make sure the total output current (peak) of ports are
| ports P0 + P1 + P2 | ≤ 80 mA
| ports P3 + P4 + P13 + P15 + P22 | ≤ 80 mA
| ports P6 + P7 + P8 + P9 + P17 | ≤ 80 mA
| ports P10 + P11 + P12 | ≤ 80 mA
Note 3: The average output current is a value averaged during a 100 ms period.

21.4.2 D.C. Characteristics (when VCCE = 3.3 V ± 0.3 V, f(XIN) = 10 MHz)

Electrical Characteristics (Referenced to VCCE, VDDE = 3.3 V ± 0.3 V, Ta = –40°C to 85°C Unless Otherwise Noted)

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
VOH	Output "H" Voltage	IOH≥-2mA	VCCE+0.5 ×IOH(mA)		VCCE	V
VOL	Output "L" Voltage	IOL≤2mA	0		0.225×IOL (mA)	V
VDDE	RAM Retention Power Supply Voltage	When operating	3.0		3.6	V
		During backup	3.0		3.6	V
IIH	"H" State Input Current	VI=VCCE	-5		5	µA
IIL	"L" State Input Current	VI=0V	-5		5	µA
ICC	Total Power Supply Current (Note 1)	f(XIN)=10.0MHz, When reset			50	mA
		f(XIN)=10.0MHz, When operating		65	90	
IDDEhold	RAM Retention Power Supply Current	Ta=25°C		0.05	1	µA
		Ta=85°C			25	
VT+- VT-	FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#		0.65			V
	Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1		0.5			
	Standard input for the following pins: SBI#, HREQ#		0.2			
	When threshold switching function is used (VT+ / VT–)	0.7VCCE/0.35VCCE	0.45			
		0.7VCCE/0.5VCCE	0.2			
		0.5VCCE/0.35VCCE	0.15			

Note 1: Total amount of current when VCCE = VDDE = AVCC = VREF in single-chip mode

Electrical Characteristics of Each Power Supply Pin

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
ICCE	VCCE Power Supply Current When Operating	f(XIN)=10.0MHz			90	mA
IDDE	VDDE Power Supply Current When Operating	f(XIN)=10.0MHz			1	mA
IAVCC	AVCC Power Supply Current When Operating	f(XIN)=10.0MHz			2	mA
IVREF	VREF Power Supply Current When Operating	f(XIN)=10.0MHz			1	mA

21.4.3 A/D Conversion Characteristics (when VCCE = 3.3 V ± 0.3 V, f(XIN) = 10 MHz)

A/D Conversion Characteristics (Referenced to VCCE, VDDE = 3.3 V, Ta = –40°C to 85°C Unless Otherwise Noted)

Symbol	Parameter				Test Condition	Rated Value			Unit
						MIN	TYP	MAX	
–	Resolution				VREF=VCCE=AVCC			10	bits
–	Absolute Accuracy (Note 1)	Without sample-and-hold or during normal sample-and-hold	Slow mode	Normal speed				±4	LSB
				Double speed				±4	
			Fast mode	Normal speed				±6	
				Double speed				±6	
		During fast sample-and-hold	Slow mode	Normal speed				±4	
				Double speed				±4	
			Fast mode	Normal speed				±6	
				Double speed				±16	
TCONV	Conversion Time	Without sample-and-hold or during normal sample-and-hold	Slow mode	Normal speed		14.95			µs
				Double speed		8.65			
			Fast mode	Normal speed		6.55			
				Double speed		4.45			
		During fast sample-and-hold	Slow mode	Normal speed		9.55			
				Double speed		5.05			
			Fast mode	Normal speed		4.75			
				Double speed		2.65			
IIAN	Analog Input Leakage Current (Note 2)				AVSS≤AD0INi≤AVCC	-5		5	µA

Note 1: Absolute accuracy refers to the accuracy of output code relative to the analog input including all error sources (including quantization error) in an A/D converter, and is calculated using the equation below.
Absolute accuracy = output code – (analog input voltage AD0INi / 1 LSB)
When AVCC = AVREF = 3.072 V, 1 LSB = 3 mV.

Note 2: This refers to the input leakage current on AD0INi while the A/D converter remains idle.

21.5 Electrical Characteristics when VCCE = 3.3 V, f(XIN) = 8 MHz

21.5.1 Recommended Operating Conditions (when VCCE = 3.3 V ± 0.3 V f(XIN) = 8 MHz)

Recommended Operating Conditions (Referenced to VCCE, VDDE = 3.3 V ± 0.3 V, Ta = –40°C to 125°C Unless Otherwise Noted)

Symbol	Parameter				Rated Value			Unit
					MIN	TYP	MAX	
VCCE	Main Power Supply (Note 1)				3.0	3.3	3.6	V
VDDE	RAM Power Supply (Note 1)				3.0	VCCE	3.6	V
AVCC	Analog Power Supply (Note 1)				3.0	VCCE	3.6	V
VREF	Reference Voltage Input (Note 1)				3.0	VCCE	3.6	V
VIH	Input "H" Voltage	When threshold switching function is used	When CMOS input is selected	Threshold selection : 0.35VCCE	0.5VCCE		VCCE	V
				Threshold selection : 0.5VCCE	0.65VCCE		VCCE	V
				Threshold selection : 0.7VCCE	0.8VCCE		VCCE	V
			When Schmitt input is selected	VT+/VT- : 0.5VCCE/0.35VCCE	0.65VCCE		VCCE	V
				VT+/VT- : 0.7VCCE/0.35VCCE	0.8VCCE		VCCE	V
				VT+/VT- : 0.7VCCE/0.5VCCE	0.8VCCE		VCCE	V
		FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#			0.8VCCE		VCCE	V
		Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1			0.8VCCE		VCCE	V
		Standard input for the following pins: DB0–15, WAIT#			0.5VCCE		VCCE	V
		Standard input for the following pins: SBI#, HREQ#			0.65VCCE		VCCE	V
VIL	Input "L" Voltage	When threshold switching function is used	When CMOS input is selected	Threshold selection : 0.35VCCE	0		0.2VCCE	V
				Threshold selection : 0.5VCCE	0		0.35VCCE	V
				Threshold selection : 0.7VCCE	0		0.5VCCE	V
			When Schmitt input is selected	VT+/VT- : 0.5VCCE/0.35VCCE	0		0.2VCCE	V
				VT+/VT- : 0.7VCCE/0.35VCCE	0		0.2VCCE	V
				VT+/VT- : 0.7VCCE/0.5VCCE	0		0.35VCCE	V

Symbol	Parameter		Rated Value			Unit
			MIN	TYP	MAX	
VIL	Input "L" Voltage	FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#	0		0.2VCCE	V
		Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLK1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1	0		0.2VCCE	V
		Standard input for the following pins: DB0–15, WAIT#	0		0.2VCCE	V
		Standard input for the following pins: SBI#, HREQ#	0		0.2VCCE	V
IOH(peak)	"H" State Peak Output Current P0–P22 (Note 2)				-10	mA
IOH(avg)	"H" State Average Output Current P0–P22 (Note 3)				-5	mA
IOL(peak)	"L" State Peak Output Current P0–P22 (Note 2)				10	mA
IOL(avg)	"L" State Average Output Current P0–P22 (Note 3)				5	mA
CL	Output Load	JTDO, JTMS			80	pF
	Capacitance	Other than above	15		50	pF
f(XIN)	External Clock Input Frequency		5		8	MHz

Note 1: Subject to conditions VCCE ≥ AVCC ≥ VREF
Note 2: Make sure the total output current (peak) of ports are
| ports P0 + P1 + P2 | ≤ 80 mA
| ports P3 + P4 + P13 + P15 + P22 | ≤ 80 mA
| ports P6 + P7 + P8 + P9 + P17 | ≤ 80 mA
| ports P10 + P11 + P12 | ≤ 80 mA
Note 3: The average output current is a value averaged during a 100 ms period.

21.5.2 D.C. Characteristics (when VCCE = 3.3 V ± 0.3 V, f(XIN) = 8 MHz)

Electrical Characteristics (Referenced to VCCE, VDDE = 3.3 V ± 0.3 V, Ta = –40°C to 125°C Unless Otherwise Noted)

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
VOH	Output "H" Voltage	IOH≥-2mA	VCCE+0.5 ×IOH(mA)		VCCE	V
VOL	Output "L" Voltage	IOL≤2mA	0		0.225×IOL (mA)	V
VDDE	RAM Retention Power Supply Voltage	When operating	3.0		3.6	V
		During backup	3.0		3.6	V
IIH	"H" State Input Current	VI=VCCE	-5		5	µA
IIL	"L" State Input Current	VI=0V	-5		5	µA
ICC	Total Power Supply Current (Note 1)	f(XIN)=8.0MHz, When reset			40	mA
		f(XIN)=8.0MHz, When operating		55	70	
IDDEhold	RAM Retention Power Supply Current	Ta=25°C		0.05	1	µA
		Ta=125°C			25	
VT+- VT-	FP, MOD0, MOD1, JTMS, JTRST, JTDI, RESET#		0.65			V
	Standard input for the following pins: RTDCLK, RTDRXD, SCLKI0, SCLKI1, RXD0–RXD3, TCLK0–TCLK3, TIN0, TIN3, TIN16–TIN23, CRX0, CRX1		0.5			
	Standard input for the following pins: SBI#, HREQ#		0.2			
	When threshold switching function is used (VT+ / VT–)	0.7VCCE/0.35VCCE	0.45			
		0.7VCCE/0.5VCCE	0.2			
		0.5VCCE/0.35VCCE	0.15			

Note 1: Total amount of current when VCCE = VDDE = AVCC = VREF in single-chip mode

Electrical Characteristics of Each Power Supply Pin

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
ICCE	VCCE Power Supply Current When Operating	f(XIN)=8.0MHz			70	mA
IDDE	VDDE Power Supply Current When Operating	f(XIN)=8.0MHz			1	mA
IAVCC	AVCC Power Supply Current When Operating	f(XIN)=8.0MHz			2	mA
IVREF	VREF Power Supply Current When Operating	f(XIN)=8.0MHz			1	mA

21.5.3 A/D Conversion Characteristics (when VCCE = 3.3 V ± 0.3 V, f(XIN) = 8 MHz)

A/D Conversion Characteristics (Referenced to VCCE, VDDE = 3.3 V, Ta = –40°C to 125°C Unless Otherwise Noted)

Symbol	Parameter				Test Condition	Rated Value			Unit
						MIN	TYP	MAX	
–	Resolution				VREF=VCCE=AVCC			10	bits
–	Absolute Accuracy (Note 1)	Without sample-and-hold or during normal sample-and-hold	Slow mode	Normal speed				±4	LSB
				Double speed				±4	
			Fast mode	Normal speed				±6	
				Double speed				±6	
		During fast sample-and-hold	Slow mode	Normal speed				±4	
				Double speed				±4	
			Fast mode	Normal speed				±6	
				Double speed				±16	
TCONV	Conversion Time	Without sample-and-hold or during normal sample-and-hold	Slow mode	Normal speed		18.6875			µs
				Double speed		10.8125			
			Fast mode	Normal speed		8.1875			
				Double speed		5.5625			
		During fast sample-and-hold	Slow mode	Normal speed		11.9375			
				Double speed		6.3125			
			Fast mode	Normal speed		5.9375			
				Double speed		3.3125			
IIAN	Analog Input Leakage Current (Note 2)				AVSS≤AD0INi≤AVCC	-5		5	µA

Note 1: Absolute accuracy refers to the accuracy of output code relative to the analog input including all error sources (including quantization error) in an A/D converter, and is calculated using the equation below.

Absolute accuracy = output code – (analog input voltage AD0INi / 1 LSB)

When AVCC = AVREF = 3.072 V, 1 LSB = 3 mV.

Note 2: This refers to the input leakage current on AD0INi while the A/D converter remains idle.

21.6 Flash Memory Related Characteristics

Symbol	Parameter		Test Condition	Rated Value			Unit
				MIN	TYP	MAX	
Topr	Flash Rewrite Ambient Temperature		T version	-40		85	°C
			V version	-40		125	°C
cycle	Flash Rewrite	Standard product		100			times
	Durability (Note 1)	10000 (10k) times rewritable	4-Kbyte block (Note 3) (Block 1, 2)	10000 (10k)			times
		product (Note 2)	Other than 4-Kbyte block	1000 (1k)			times
VCCE	VCCE power supply voltage		Within the range of recom-	3.0	3.3	3.6	V
	(when reprogramming)		mended power supply voltage	4.5	5.0	5.5	
ICCE	VCCE power supply current (when programming)		When using boot program			90	mA
ICCE	VCCE power supply current (When erasing)		When using boot program			90	mA

Note 1: The rewrite durability indicates the number of erase times for each block.
Note that more than one write operations (overwrite) on the same address cannot be performed. In that case, erase the old data before writing new.
Note 2: The 10000 (10k) times rewritable product is offered as an optional item. For details about it, please contact your nearest office of Renesas or its distributor. Please note that the standard product will be shipped if not consulted.
Note 3: Block 1: H'0000 2000 to H'0000 2FFF
Block 2: H'0000 3000 to H'0000 3FFF
Note 4: Do not rewrite within the voltage range from 3.6V to 4.5V.

(1) Standard product (Flash rewrite durability: 100 times)

Symbol	Parameter	Test Condition		Rated Value			Unit
				MIN	TYP	MAX	
tPRG	Program time (Note 1)	All blocks	up to 100 times		25	200	µs
TBERS	Block erase time	4-Kbyte block	up to 100 times		0.3	6	s
		8-Kbyte block	up to 100 times		0.3	6	s
		32-Kbyte block	up to 100 times		0.5	6	s
		64-Kbyte block	up to 100 times		0.8	6	s

Note 1: It indicates a write time per halfword.

(2) 10000 (10k) times rewritable product (Note 1)

Symbol	Parameter	Test Condition		Rated Value			Unit
				MIN	TYP	MAX	
tPRG	Program time (Note 2)	All blocks	up to 1000 (1k) times		25	200	µs
		4-Kbyte block	up to 10000 (10k) times			600	µs
TBERS	Block erase time	4-Kbyte block	up to 1000 (1k) times		0.3	6	s
			up to 10000 (10k) times			8	s
		8-Kbyte block	up to 1000 (1k) times		0.3	6	s
		32-Kbyte block	up to 1000 (1k) times		0.5	6	s
		64-Kbyte block	up to 1000 (1k) times		0.8	6	s

Note 1: The 10000 (10k) times rewritable product is offered as an optional item. For details about it, please contact your nearest office of Renesas or its distributor. Please note that the standard product will be shipped if not consulted.
Note 2: It indicates a write time per halfword.

21.7 External Capacitance for Power Supply

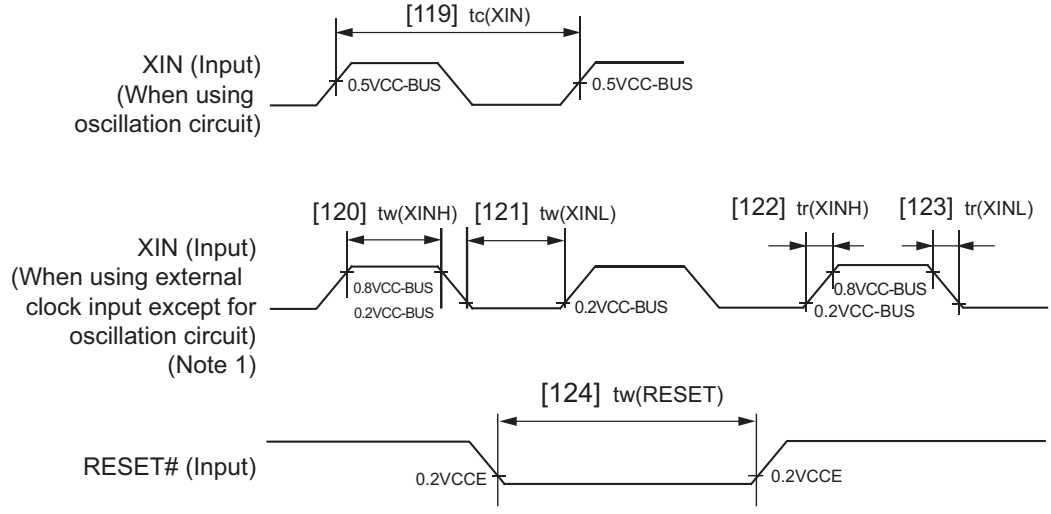
Symbol	Parameter	Rated Value			Unit
		MIN	TYP	MAX	
EXCVCC	External capacitance connecting pin	1		10	µF
EXCVDD	External capacitance connecting pin for the internal power supply of the built in RAM	1		10	µF
EXCOSC-VCC	External capacitance connecting pin for the internal power supply of the clock	1		10	µF

21.8 A.C. Characteristics (when VCCE = 5 V)

- The timing conditions are referenced to VCCE, VDDE = 5 V ± 0.5 V, Ta = –40°C to 125°C unless otherwise noted.
- The rated values below are guaranteed for the case where the output load capacitance of the measured pins are 15 pF to 50 pF (for JTAG related values, a concentrated capacitance of 80 pF).

(1) Clock and reset timing

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.1
Timing requirements	tc(XIN)	Clock Input Cycle Time	100	200	ns	[119]
	tw(XINH)	External Clock Input "H" Pulse Width	35		ns	[120]
	tw(XINL)	External Clock Input "L" Pulse Width	35		ns	[121]
	tr(XINH)	External Clock Input High-going Time		10	ns	[122]
	tr(XINL)	External Clock Input Low-going Time		10	ns	[123]
	tw(RESET)	Reset Input "L" Pulse Width	200		ns	[124]



Note1: Make XOUT pin open and set parasitic capacity as 10pF or less.
The XDRV bit of clock control register (CLKCR) should be choosen B'11 (maximum).

Figure 21.8.1 Clock and Reset Timing

(2) Input/output ports

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.2
Timing requirements	tsu(P-E)	Port Input Setup Time	100		ns	[1]
	th(E-P)	Port Input Hold Time	0		ns	[2]
Switching characteristics	td(E-P)	Port Data Output Delay Time		100	ns	[3]

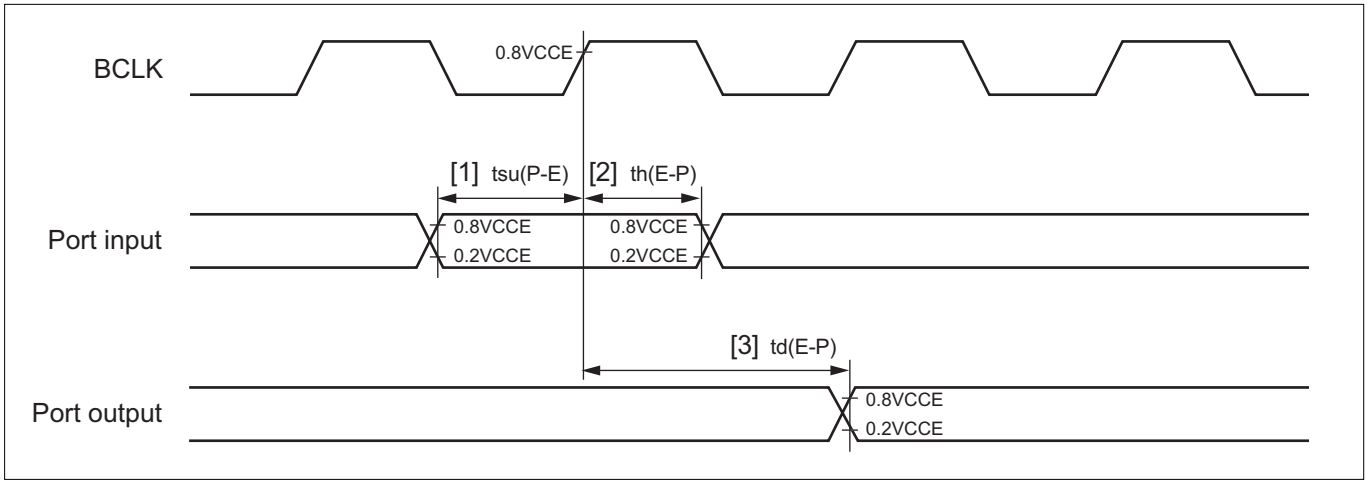


Figure 21.8.2 Input/Output Port Timing

(3) Serial interface

a) CSIO mode, with internal clock selected

	Symbol	Parameter	Rated Value		Unit	See Fig. 21.8.3
			MIN	MAX		
Timing requirements	tsu(D-CLK)	RXD Input Setup Time	150		ns	[4]
	th(CLK-D)	RXD Input Hold Time	50		ns	[5]
Switching characteristics	td(CLK-D)	TXD Output Delay Time		60	ns	[6]
	th(CLK-D)	TXD Hold Time	0		ns	[82]

b) CSIO mode, with external clock selected

	Symbol	Parameter	Rated Value		Unit	See Fig. 21.8.3
			MIN	MAX		
Timing requirements	tc(CLK)	CLK Input Cycle Time	640		ns	[7]
	tw(CLKH)	CLK Input "H" Pulse Width	300		ns	[8]
	tw(CLKL)	CLK Input "L" Pulse Width	300		ns	[9]
	tsu(D-CLK)	RXD Input Setup Time	60		ns	[10]
	th(CLK-D)	RXD Input Hold Time	100		ns	[11]
Switching characteristics	td(CLK-D)	TXD Output Delay Time		160	ns	[12]

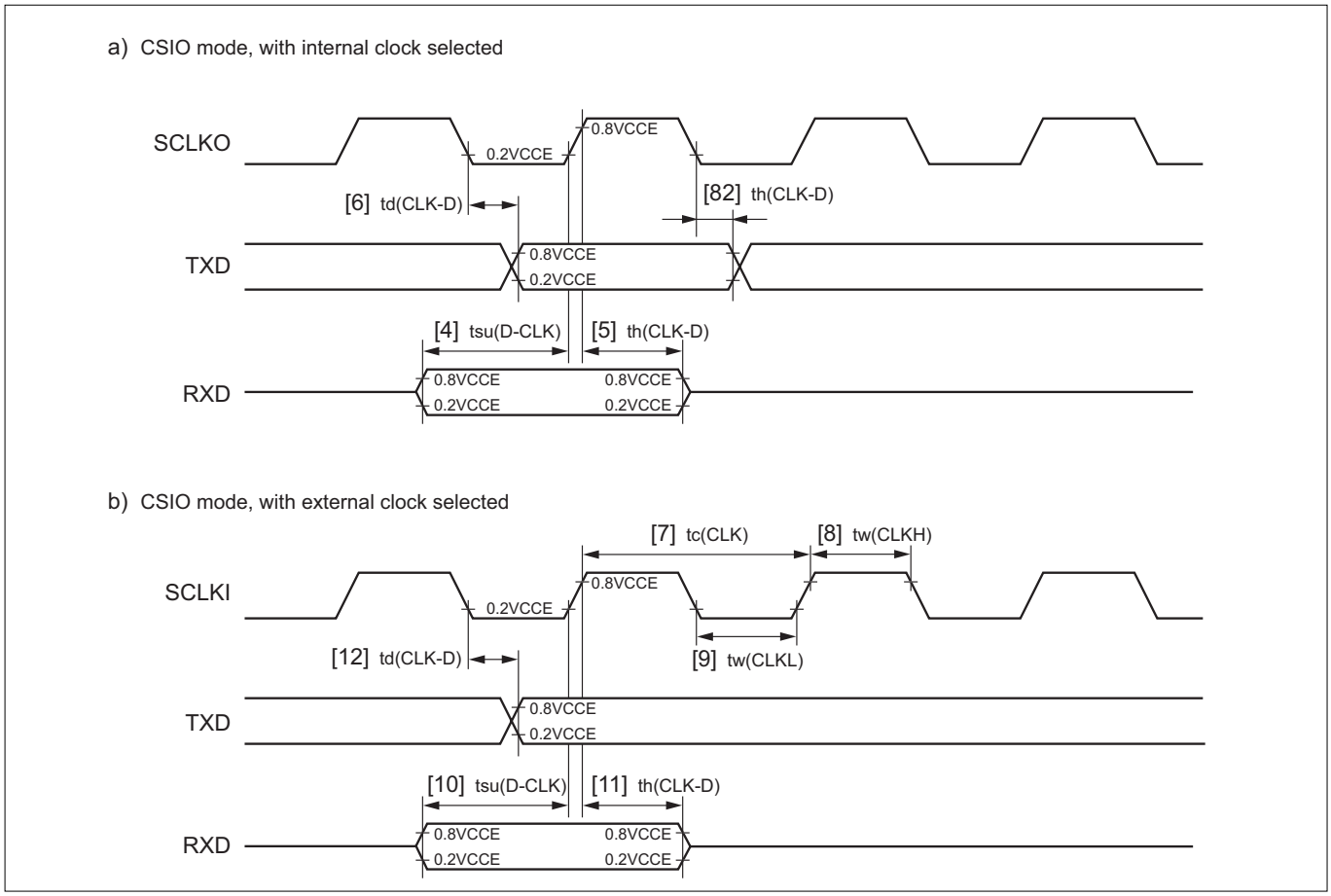


Figure 21.8.3 Serial Interface Timing

(4) SBI

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.4
Timing requirements	tw(SBIL)	SBI# Input Pulse Width	$5\times\frac{tc(BCLK)}{2}$		ns	[13]

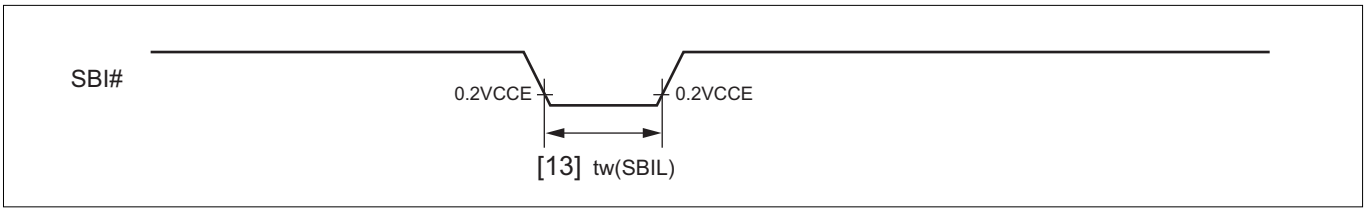


Figure 21.8.4 SBI Timing

(5) TIN

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.5
Timing requirements	tw(TIN)	TIN Input Pulse Width	$7\times\frac{tc(BCLK)}{2}$		ns	[14]

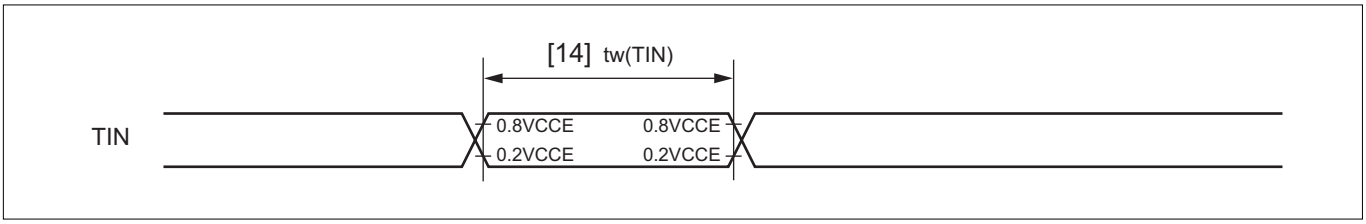


Figure 21.8.5 TIN Timing

(6) TO

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.6
Switching characteristics	td(BCLK-TO)	TO Output Delay Time		100	ns	[15]

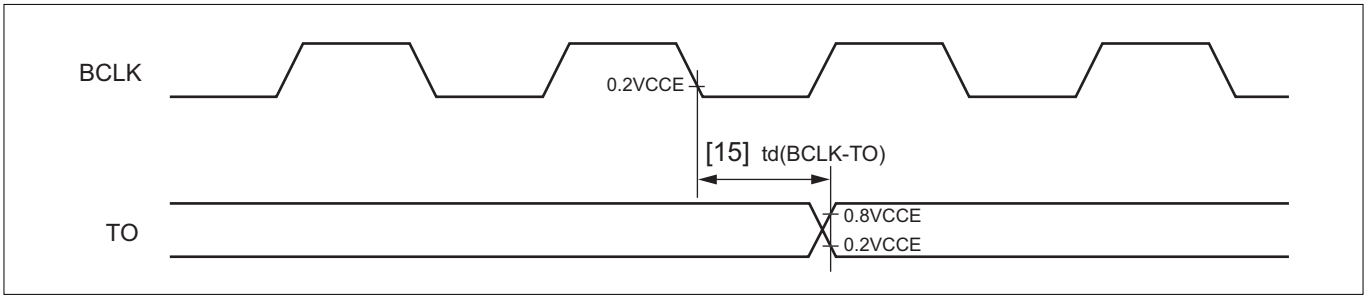


Figure 21.8.6 TO Timing

(7) TCLK

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.7
Timing requirements	tw(TCLKH)	TCLK Input "H" Pulse Width	$7\times\frac{tc(BCLK)}{2}$		ns	[99]
	tw(TCLKL)	TCLK Input "L" Pulse Width	$7\times\frac{tc(BCLK)}{2}$		ns	[100]

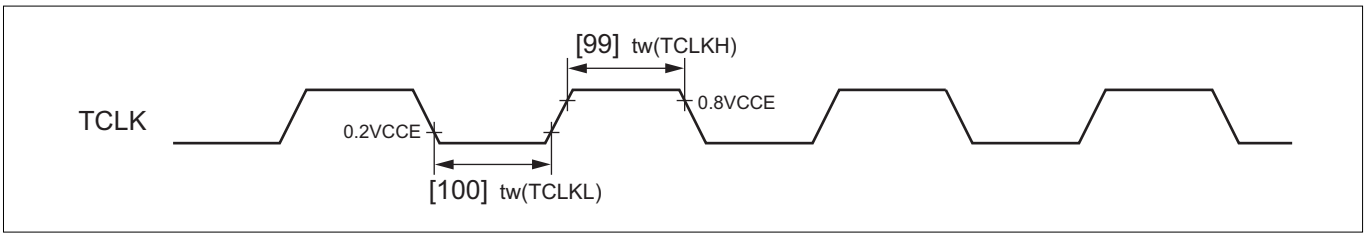
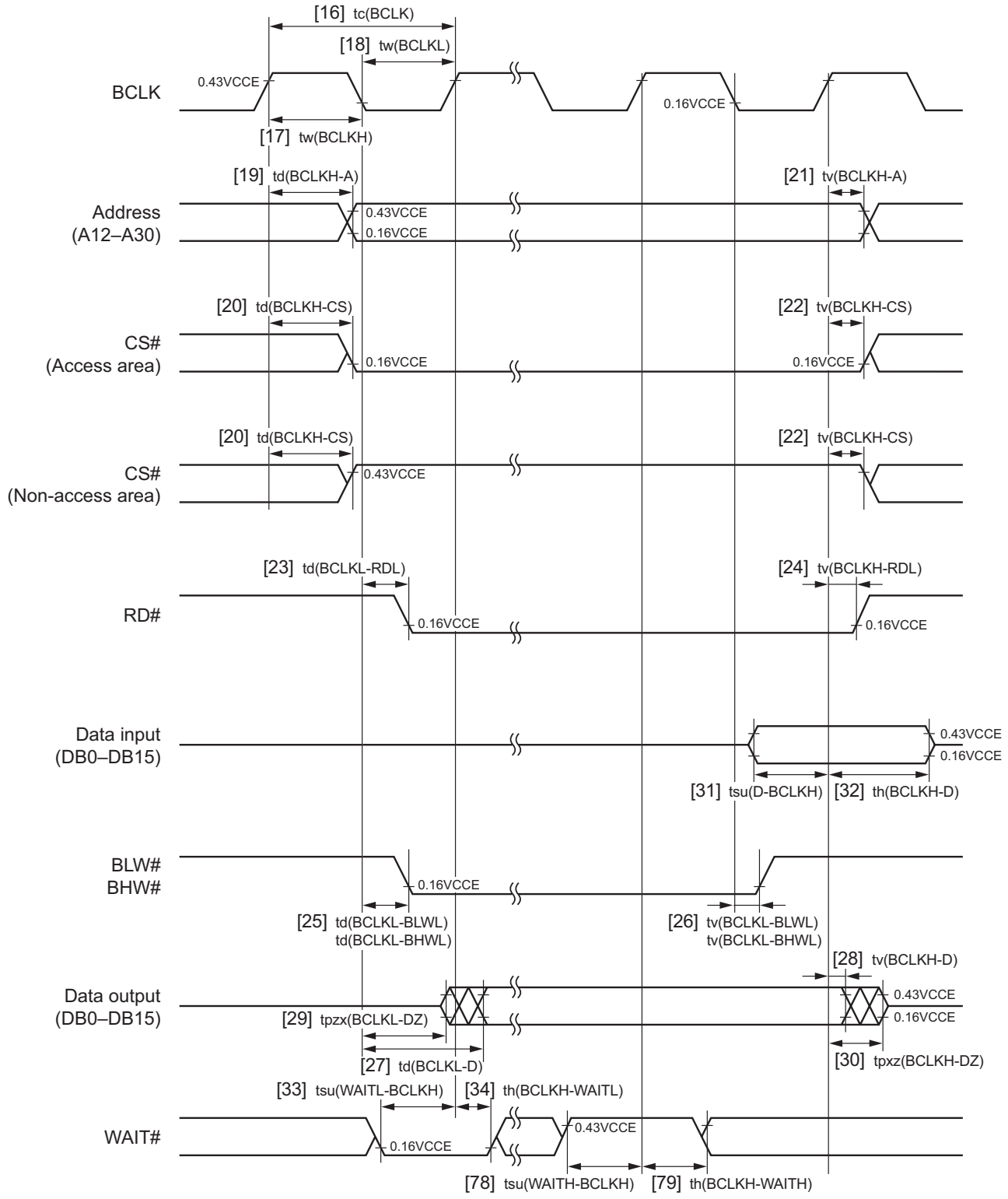


Figure 21.8.7 TCLK Timing

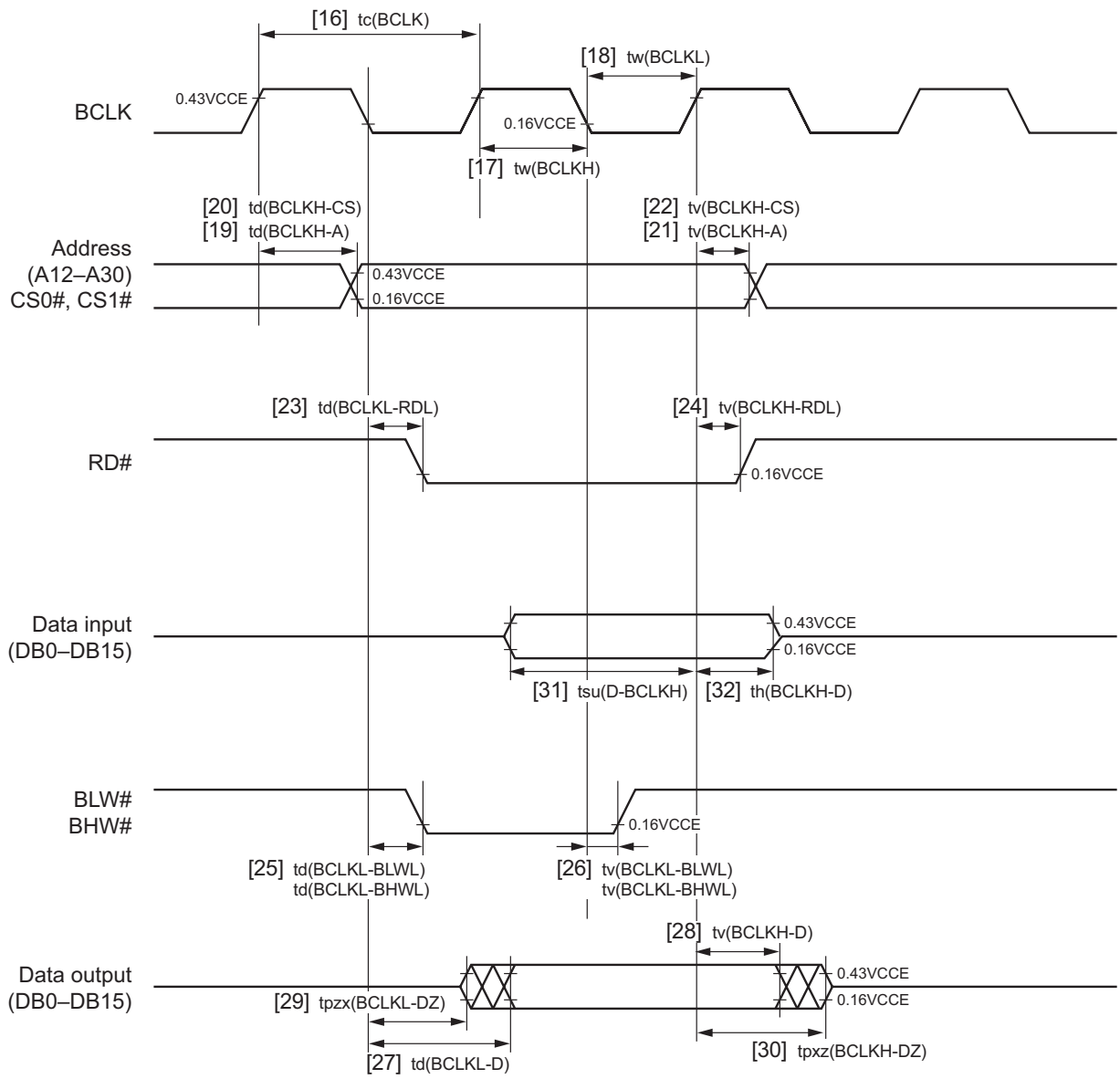
(8) Read and write timing (1/4)

	Symbol	Parameter	Rated Value		Unit	See Figs.
			MIN	MAX		21.8.8 21.8.9
Timing requirements	tsu(D-BCLKH)	Data Input Setup Time before BCLK	26		ns	[31]
	th(BCLKH-D)	Data Input Hold Time after BCLK	0		ns	[32]
	tsu(WAITL-BCLKH)	WAIT# Input Setup Time before BCLK	26		ns	[33]
	th(BCLKH-WAITL)	WAIT# Input Hold Time after BCLK	0		ns	[34]
	tsu(WAITH-BCLKH)	WAIT# Input Setup Time before BCLK	26		ns	[78]
	th(BCLKH-WAITH)	WAIT# Input Hold Time after BCLK	0		ns	[79]
Switching characteristics	tc(BCLK)	BCLK Output Cycle Time		$\frac{tc(XIN)}{2}$	ns	[16]
	tw(BCLKH)	BCLK Output "H" Pulse Width	$\frac{tc(BCLK)}{2}$ -5		ns	[17]
	tw(BCLKL)	BCLK Output "L" Pulse Width	$\frac{tc(BCLK)}{2}$ -5		ns	[18]
	td(BCLKH-A)	Address Delay Time after BCLK		24	ns	[19]
	td(BCLKH-CS)	Chip Select Delay Time after BCLK		24	ns	[20]
	tv(BCLKH-A)	Address Valid Time after BCLK	-11		ns	[21]
	tv(BCLKH-CS)	Chip Select Valid Time after BCLK	-11		ns	[22]
	td(BCLKL-RDL)	Read Delay Time after BCLK		10	ns	[23]
	tv(BCLKH-RDL)	Read Valid Time after BCLK	-12		ns	[24]
	td(BCLKL-BLWL) td(BCLKL-BHWL)	Write Delay Time after BCLK		11	ns	[25]
	tv(BCLKL-BLWL) tv(BCLKL-BHWL)	Write Valid Time after BCLK	-12		ns	[26]
	td(BCLKL-D)	Data Output Delay Time after BCLK		18	ns	[27]
	tv(BCLKH-D)	Data Output Valid Time after BCLK	-16		ns	[28]
	tpzx(BCLKL-DZ)	Data Output Enable Time after BCLK	-19		ns	[29]
	tpxz(BCLKH-DZ)	Data Output Disable Time after BCLK		5	ns	[30]



Notes: • For signal-to-signal timing, see Figure 21.8.10, "Read Timing (Relative to Read Pulse)," and Figure 21.8.11, "Write Timing (Relative to Write Pulse)."
• When using the threshold switching function, the data input and WAIT# voltage levels are determined with respect to the rated minimum and maximum values for VIH and VIL.

Figure 21.8.8 Read and Write Timing (Relative to BCLK) with 1 or more External WAIT States

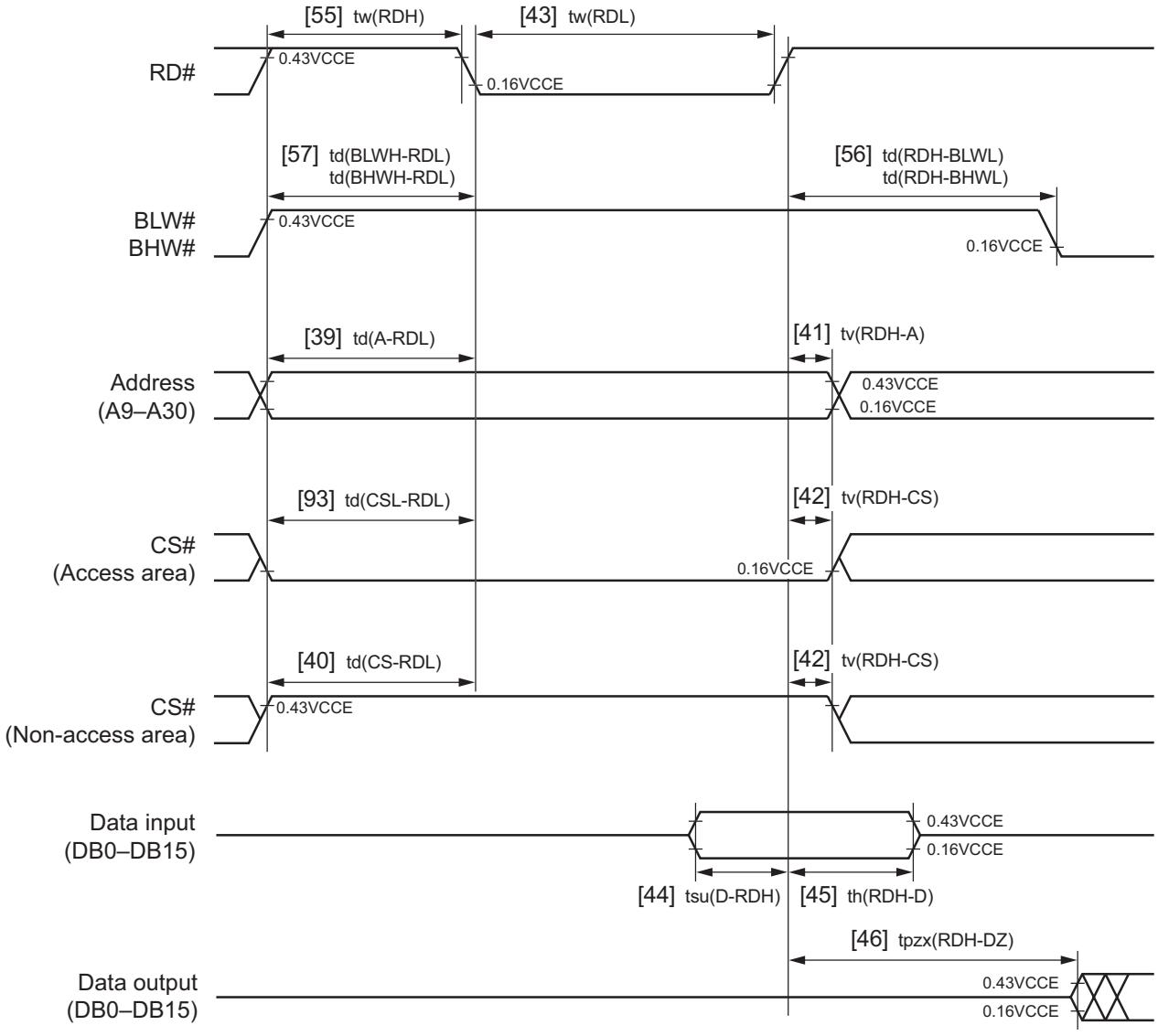


Note: • When using the threshold switching function, the data input voltage levels are determined with respect to the rated minimum and maximum values for VIH and VIL.

Figure 21.8.9 Read and Write Timing (Relative to BCLK) with 1 WAIT State

(8) Read and write timing (2/4)

	Symbol	Parameter	Rated Value		Unit	See Figs. 21.8.10 21.8.11
			MIN	MAX		
Timing requirements	tsu(D-RDH)	Data Input Setup Time before Read	30		ns	[44]
	th(RDH-D)	Data Input Hold Time after Read	0		ns	[45]
	tsu(WAITH-RDL) tsu(WAITL-RDL)	WAIT Input Setup Time before Read	tc(BCLK)+21		ns	[132]
	tw(WAITH)	WAIT "H" Pulse Width	26		ns	[133]
	tw(WAITL)	WAIT "L" Pulse Width	26		ns	[134]
	tsu(WAITH-BLWL) tsu(WAITH-BHWL) tsu(WAITL-BLWL) tsu(WAITL-BHWL)	WAIT Input Setup Time before Write (byte write mode)	$\frac{tc(BCLK)}{2}$ +21		ns	[135]
Switching characteristics	td(A-RDL)	Address Delay Time before Read	$\frac{tc(BCLK)}{2}$ -15		ns	[39]
	td(CS-RD)	Chip Select Delay Time before Read	$\frac{tc(BCLK)}{2}$ -15		ns	[40]
	tv(RDH-A)	Address Valid Time after Read	0		ns	[41]
	tv(RDH-CS)	Chip Select Valid Time after Read	0		ns	[42]
	tw(RDL)	Read "L" Pulse Width	3× $\frac{tc(BCLK)}{2}$ -23		ns	[43]
	tpzx(RDH-DZ)	Data Output Enable Time after Read	$\frac{tc(BCLK)}{2}$		ns	[46]
	td(A-BLWL) td(A-BHWL)	Address Delaly Time before Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[47]
	td(CS-BLWL) td(CS-BHWL)	Chip Select Delay Time before Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[48]
	tv(BLWH-A) tv(BHWH-A)	Address Valid Time after Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[49]
	tv(BLWH-CS) tv(BHWH-CS)	Chip Select Valid Time after Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[50]
	tw(BLWL) tw(BHWL)	Write "L" Pulse Width (byte write mode)	tc(BCLK)-25		ns	[51]
	td(BLWL-D) td(BHWL-D)	Data Output Delay Time after Write (byte write mode)		15	ns	[52]
	tv(BLWH-D) tv(BHWH-D)	Data Output Valid Time after Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -13		ns	[53]
	tpxz(BLWH-DZ) tpxz(BHWH-DZ)	Data Output Disable Time after Write (byte write mode)		$\frac{tc(BCLK)}{2}$ +5	ns	[54]
	tw(RDH)	Read "H" Pulse Width	$\frac{tc(BCLK)}{2}$ -3		ns	[55]
	td(RDH-BLWL) td(RDH-BHWL)	Write Delay Time after Read	$\frac{tc(BCLK)}{2}$ -10		ns	[56]
	td(BLWH-RDL) td(BHWH-RDL)	Read Delay Time after Write	$\frac{tc(BCLK)}{2}$ -10		ns	[57]
	td(CSL-RDL)	Chip Select Delay Time before Read	$\frac{tc(BCLK)}{2}$ -20		ns	[93]
	td(CSL-BLWL) td(CSL-BHWL)	Chip Select Delay Time before Write	$\frac{tc(BCLK)}{2}$ -20		ns	[95]



Note: • When usint the threshold switching function, the data input voltage levels are determined with respect to the rated minimum and maximum values for VIH and VIL.

Figure 21.8.10 Read Timing (Relative to Read Pulse)

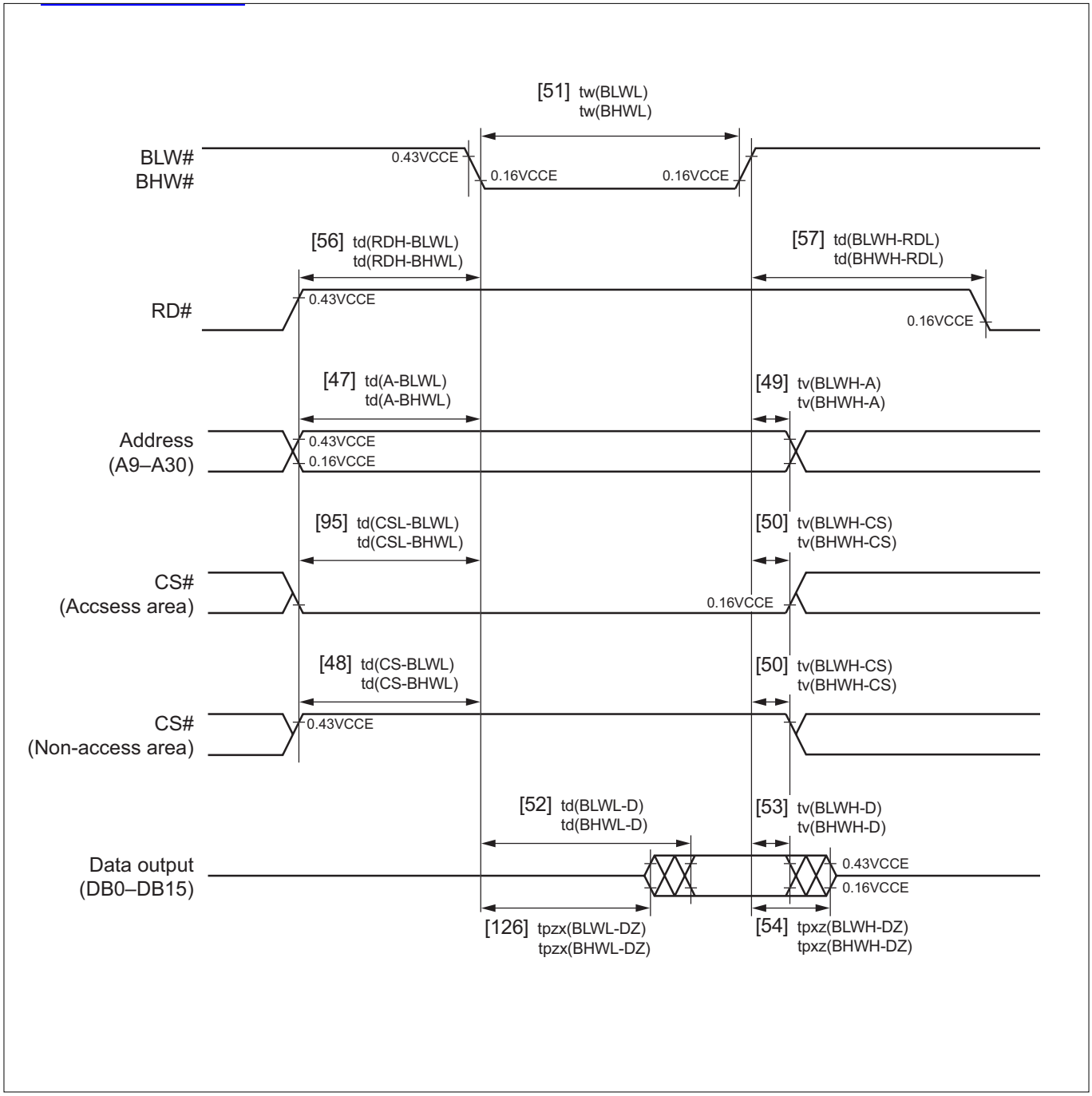


Figure 21.8.11 Write Timing (Relative to Write pulse)

(8) Read and write timing (3/4)

	Symbol	Parameter	Rated Value		Unit	See Figs.
			MIN	MAX		21.8.12
Timing requirements	tsu(D-RDH)	Data Input Setup Time before Read	30		ns	[44]
	th(RDH-D)	Data Input Hold Time before Read	0		ns	[45]
Switching characteristics	td(A-RDL)	Address Delay Time before Read	$\frac{tc(BCLK)}{2}$ -15		ns	[39]
	td(CSL-RDL)	Chip Select Delay Time before Read	$\frac{tc(BCLK)}{2}$ -15		ns	[40]
	tv(RDH-A)	Address Valid Time after Read	0		ns	[41]
	tv(RDH-CS)	Chip Select Valid Time after Read	0		ns	[42]
	tw(RDL)	Read "L" Pulse Width	$3\times \frac{tc(BCLK)}{2}$ -23		ns	[43]
	tpzx(RDH-DZ)	Data Output Enable Time after Read	$\frac{tc(BCLK)}{2}$		ns	[46]
	td(RDH-WRL)	Write Delay Time after Read (byte enable mode)	$\frac{tc(BCLK)}{2}$ -10		ns	[80]
	td(WRH-RDL)	Read Delay Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -10		ns	[81]
	td(CSL-RDL)	Chip Select Delay Time before Read	$\frac{tc(BCLK)}{2}$ -20		ns	[93]
	td(BLEL-RDL) td(BHEL-RDL)	Byte Enable Delay Time before Read (byte enable mode)	$\frac{tc(BCLK)}{2}$ -20		ns	[136]
	tv(RDH-BLEL) tv(RDH-BHEL)	Byte Enable Valid Time after Read (byte enable mode)	-5		ns	[137]

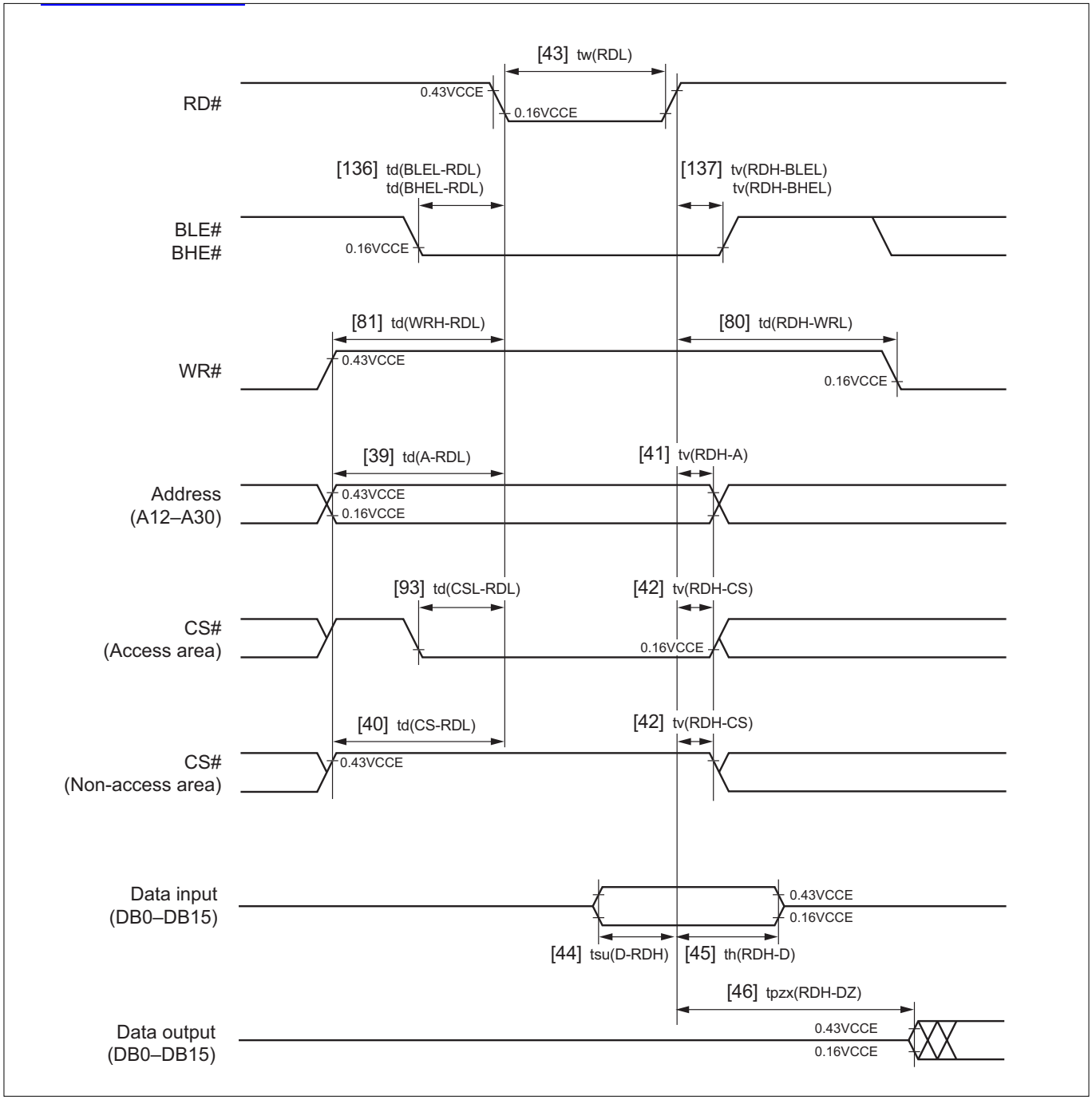


Figure 21.8.12 Read Timing (Byte Enable Mode)

(8) Read and write timing (4/4)

	Symbol	Parameter	Rated Value		Unit	see Figs.
			MIN	MAX		21.8.13
Switching characteristics	tw(WRL)	Write "L" Pulse Width (byte enable mode)	tc(BCLK) -25		ns	[68]
	td(A-WRL)	Address Delay Time before Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[69]
	td(CS-WRL)	Chip Select Delay Time before Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[70]
	tv(WRH-A)	Address Valid Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[71]
	tv(WRH-CS)	Chip Select Valid Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[72]
	td(BLEL-WRL) td(BHEL-WRL)	Byte Enable Delay Time before Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[73]
	tv(WRH-BLEL) tv(WRH-BHEL)	Byte Enable Valid Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[74]
	td(WRL-D)	Data Output Delay Time after Write (byte enable mode)		15	ns	[75]
	tv(WRH-D)	Data Output Valid Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -13		ns	[76]
	tpxz(WRH-DZ)	Data Output Disable Time after Write (byte enable mode)		$\frac{tc(BCLK)}{2}$ +5	ns	[77]
	td(RDH-WRL)	Write Delay Time after Read (byte enable mode)	$\frac{tc(BCLK)}{2}$ -10		ns	[80]
	td(WRH-RDL)	Read Delaly Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -10		ns	[81]
	td(CSL-WRL)	Chip Select Delay Time before Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[96]

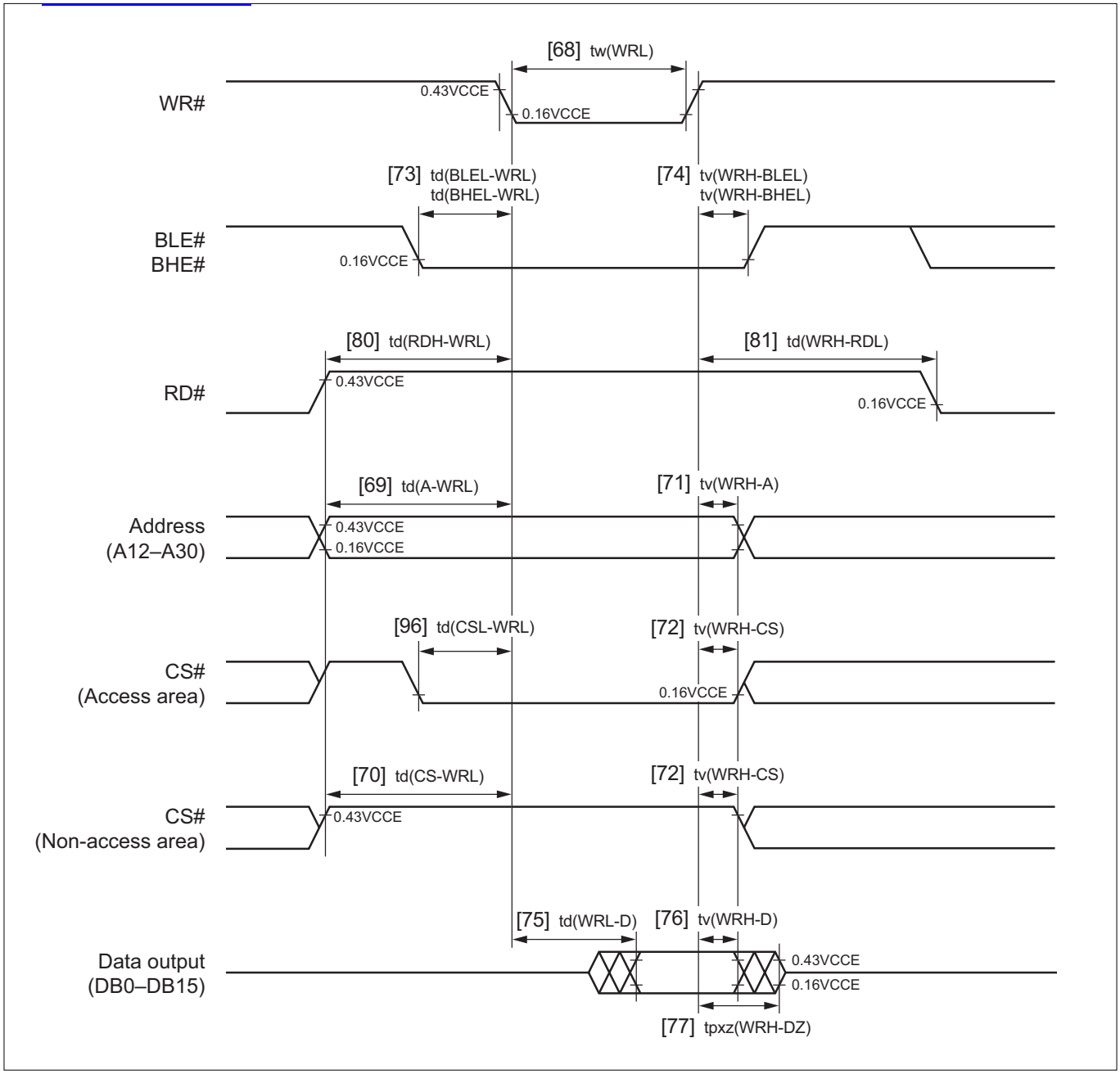


Figure 21.8.13 Write Timing (Byte Enable Mode)

(9) Bus arbitration timing

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.14
Timing requirements	tsu(HREQL-BCLKH)	HREQ# Input Setup Time before BCLK	27		ns	[35]
	th(BCLKH-HREQL)	HREQ# Input Hold Time after BCLK	0		ns	[36]
Switching characteristics	td(BCLKL-HACKL)	HACK# Delay Time after BCLK		29	ns	[37]
	tv(BCLKL-HACKL)	HACK# Valid Time after BCLK	-11		ns	[38]

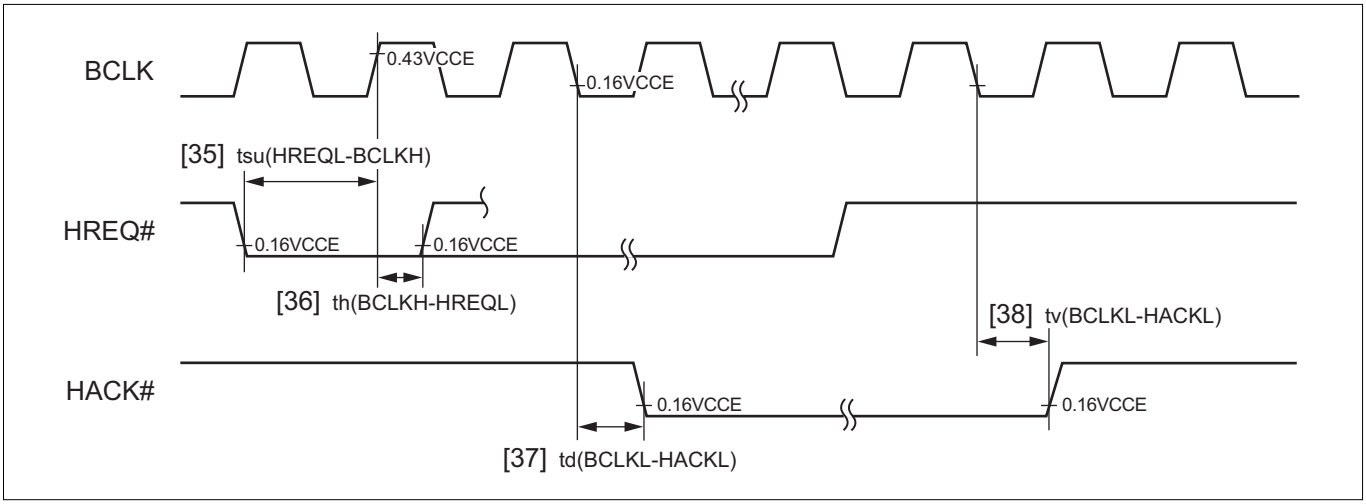
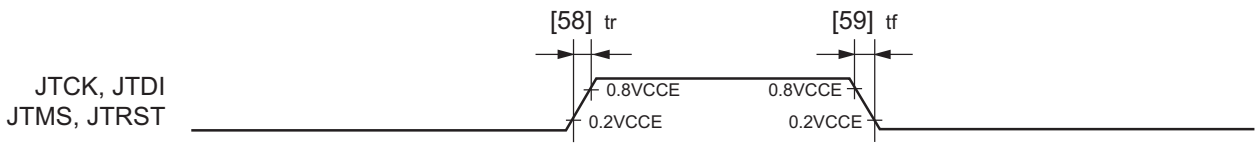


Figure 21.8.14 Bus Arbitration Timing

(10) JTAG pin input transition time

	Symbol	Parameter			Rated Value		Unit	See Fig.
					MIN	MAX		21.8.15
Timing requirements	tr	High-going Transition Time of Input	Other than JTRST pin (JTCK, JTDI, JTMS, JTDO)			10	ns	[58]
			JTRST pin	When using TAP		10	ns	
				When not using TAP		2	ms	
	tf	Low-going Transition Time of Input	Other than JTRST pin (JTCK, JTDI, JTMS, JTDO)			10	ns	[59]
			JTRST pin	When using TAP		10	ns	
				When not using TAP		2	ms	

Note: • The rated values here are guaranteed for the case where the measured pin load capacitance CL = 80pF.



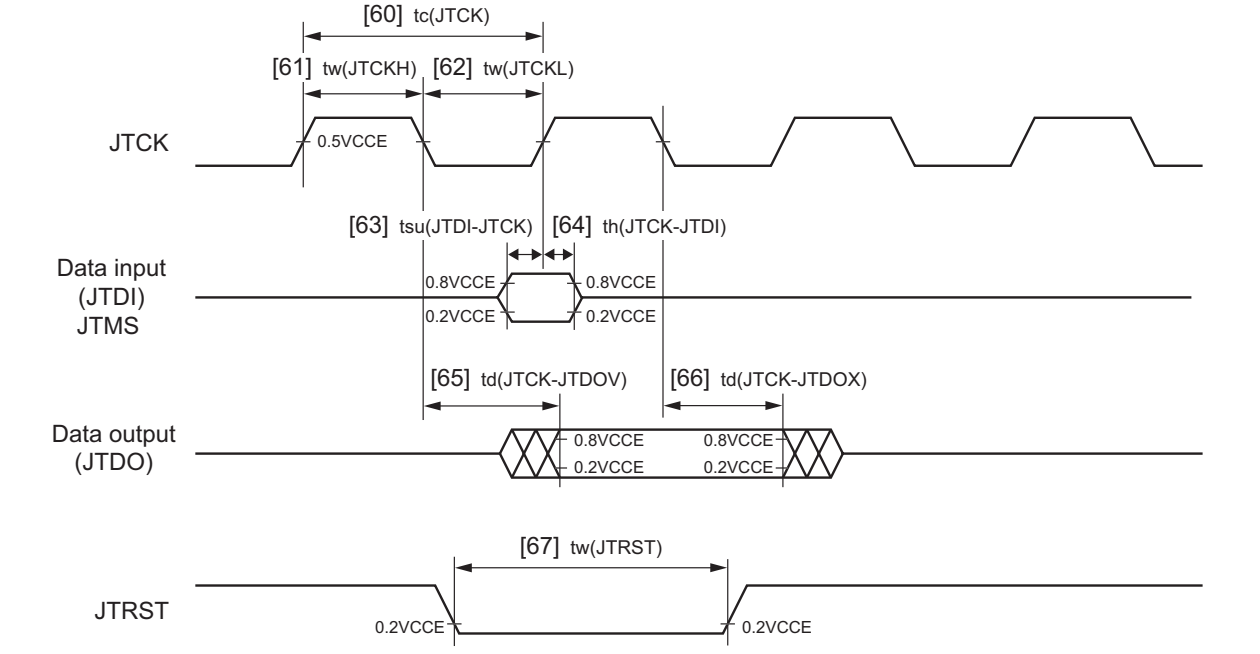
Note: • The rated values here are guaranteed for the case where the measured pin load capacitance CL = 80 pF.

Figure 21.8.15 JTAG Pin Input Transition Time

(11) JTAG interface timing

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.16
Timing requirements	tc(JTCK)	JTCK Input Cycle Time	100		ns	[60]
	tw(JTCKH)	JTCK Input "H" Pulse Width	40		ns	[61]
	tw(JTCKL)	JTCK Input "L" Pulse Width	40		ns	[62]
	tsu(JTDI-JTCK)	JTDI, JTMS Input Setup Time	15		ns	[63]
	th(JTCK-JTDI)	JTDI, JTMS Input Hold Time	20		ns	[64]
	tw(JTRST)	JTRST Input "L" Pulse Width	tc(JTCK)		ns	[67]
Switching characteristics	td(TCK-JTDOV)	JTDO Output Delay Time after JTCK Low-going		40	ns	[65]
	td(JTCK-JTDOX)	Delay Time to JTDO Output Hi-Z after JTCK Low-going		40	ns	[66]

Note: • The rated values here are guaranteed for the case where the measured pin load capacitance CL = 80pF.



Note: • The rated values here are guaranteed for the case where the measured pin load capacitance CL = 80 pF.

Figure 21.8.16 JTAG Interface Timing

(12) RDT timing

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.8.17
Timing requirements	tc(RTDCLK)	RTDCLK Input Cycle Time	500		ns	[90]
	tw(RTDCLKH)	RTDCLK Input "H" Pulse Width	230		ns	[83]
	tw(RTDCLKL)	RTDCLK Input "L" Pulse Width	230		ns	[84]
	th(RTDCLKH-RTDRXD)	RTDRXD Input Hold Time	50		ns	[88]
	tsu(RTDRXD-RTDCLKL)	RTDRXD Input Setup Time	60		ns	[89]
Switching characteristics	td(RTDCLKH-RTDACK)	RTDACK Delay Time after RTDCLK Input		160	ns	[85]
	tv(RTDCLKL-RTDACK)	RTDACK Valid Time after RTDCLK Input		160	ns	[86]
	td(RTDCLKH-RTDTXD)	RTDTXD Delay Time after RTDCLK Input		tw(RTDCLKH)+160	ns	[87]

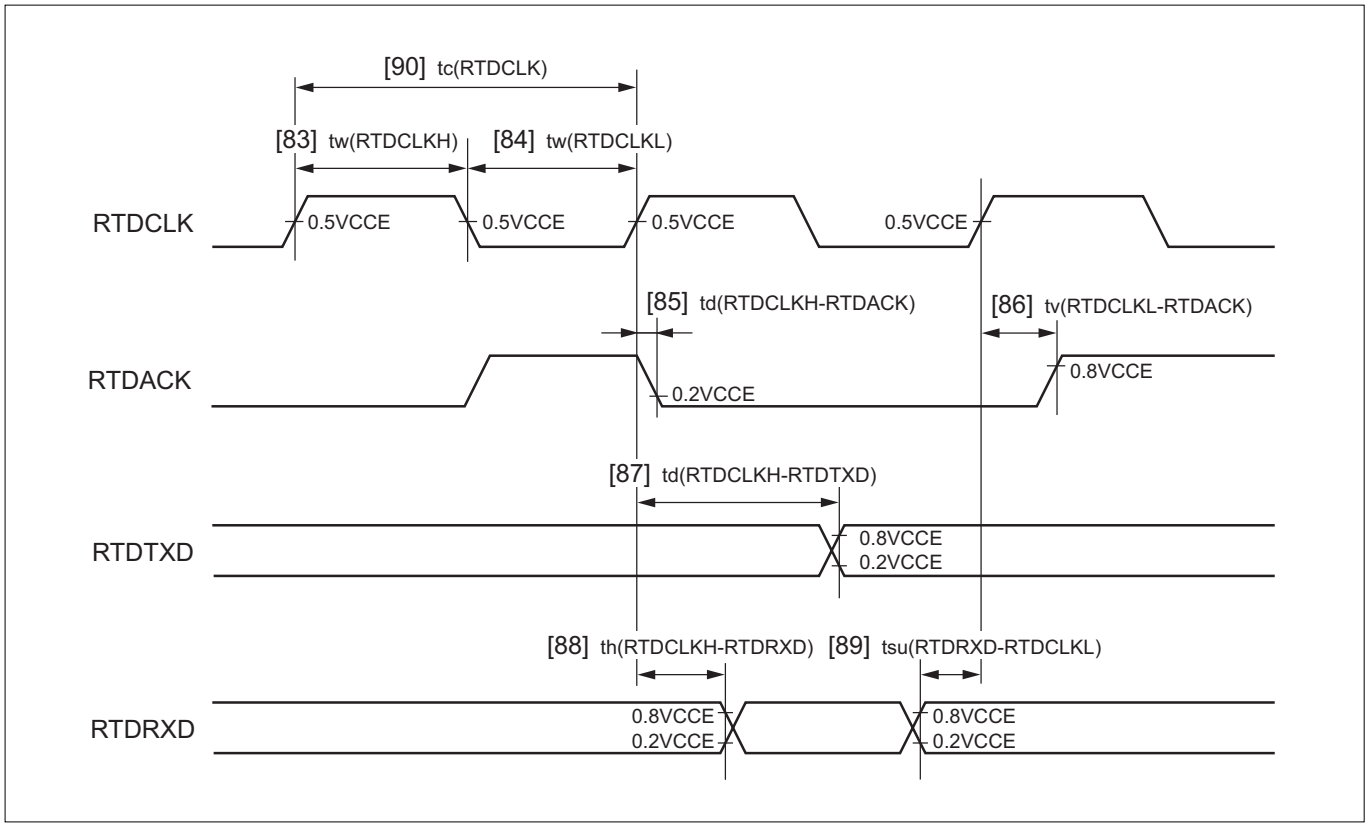


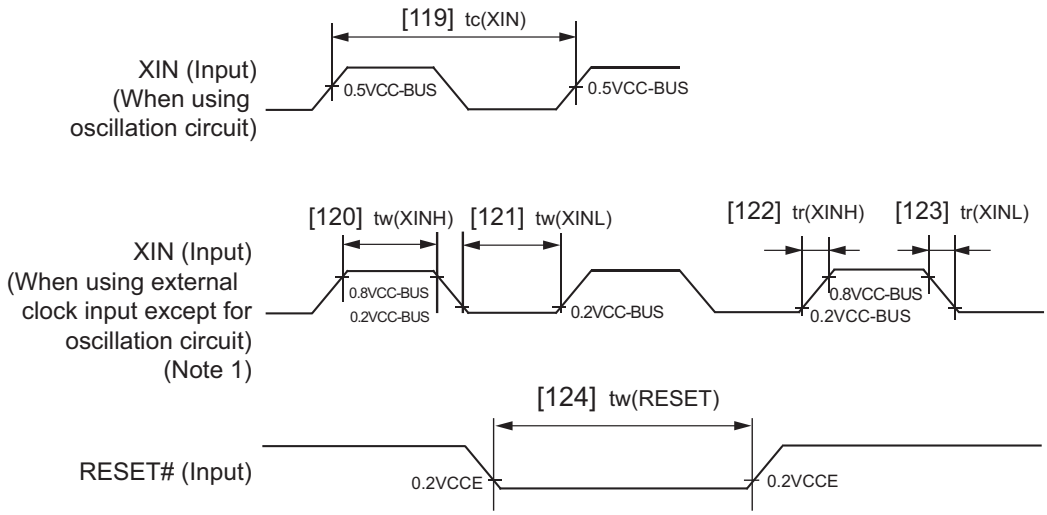
Figure 21.8.17 RTD Timing

21.9 A.C. Characteristics (when VCCE = 3.3 V)

- The timing conditions are referenced to VCCE, VDDE = 3.3V ± 0.3 V, Ta = –40°C to 125°C unless otherwise noted.
- The rated values below are guaranteed for the case where the output load capacitance of the measured pins are 15 pF to 50 pF (for JTAG related values, a concentrated capacitance of 80 pF).

(1) Clock and reset timing

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.1
Timing requirements	tc(XIN)	Clock Input Cycle Time	100	200	ns	[119]
	tw(XINH)	External Clock Input "H" Pulse Width	35		ns	[120]
	tw(XINL)	External Clock Input "L" Pulse Width	35		ns	[121]
	tr(XINH)	External Clock Input High-going Time		10	ns	[122]
	tr(XINL)	External Clock Input Low-going Time		10	ns	[123]
	tw(RESET)	Reset Input "L" Pulse Width	200		ns	[124]



Note1: Make XOUT pin open and set parasitic capacity as 10pF or less.
The XDRV bit of clock control register (CLKCR) should be choosen B'11 (maximum).

Figure 21.9.1 Clock and Reset Timing

(2) Input/output ports

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.2
Timing requirements	tsu(P-E)	Port Input Setup Time	100		ns	[1]
	th(E-P)	Port Input Hold Time	0		ns	[2]
Switching characteristics	td(E-P)	Port Data Output Delay Time		100	ns	[3]

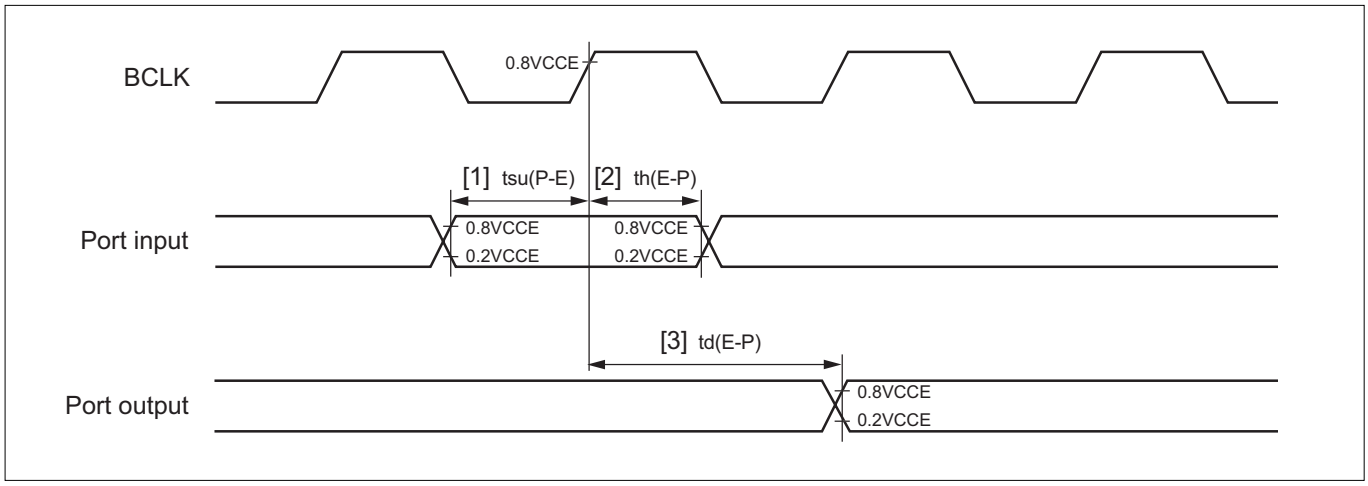


Figure 21.9.2 Input/Output Port Timing

(3) Serial interface

a) CSIO mode, with internal clock selected

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.3
Timing requirements	tsu(D-CLK)	RXD Input Setup Time	150		ns	[4]
	th(CLK-D)	RXD Input Hold Time	50		ns	[5]
Switching characteristics	td(CLK-D)	TXD Output Delay Time		60	ns	[6]
	th(CLK-D)	TXD Hold Time	0		ns	[82]

b) CSIO mode, with external clock selected

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.3
Timing requirements	tc(CLK)	CLK Input Cycle Time	640		ns	[7]
	tw(CLKH)	CLK Input "H" Pulse Width	300		ns	[8]
	tw(CLKL)	CLK Input "L" Pulse Width	300		ns	[9]
	tsu(D-CLK)	RXD Input Setup Time	60		ns	[10]
	th(CLK-D)	RXD Input Hold Time	100		ns	[11]
Switching characteristics	td(CLK-D)	TXD Output Delay Time		160	ns	[12]

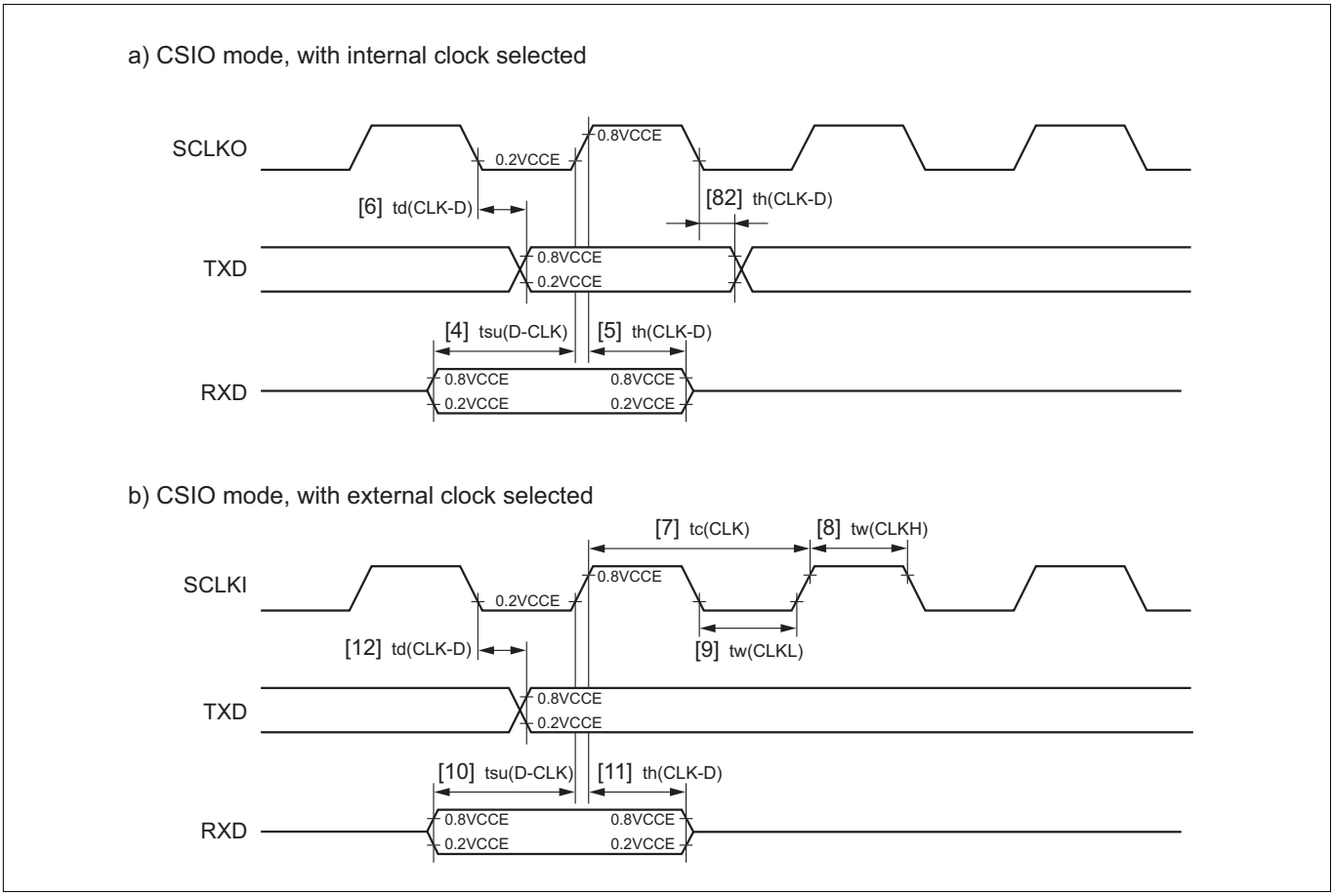


Figure 21.9.3 Serial Interface Timing

(4) SBI

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.4
Timing requirements	tw(SBIL)	SBI# Input Pulse Width	$5\times\frac{tc(BCLK)}{2}$		ns	[13]

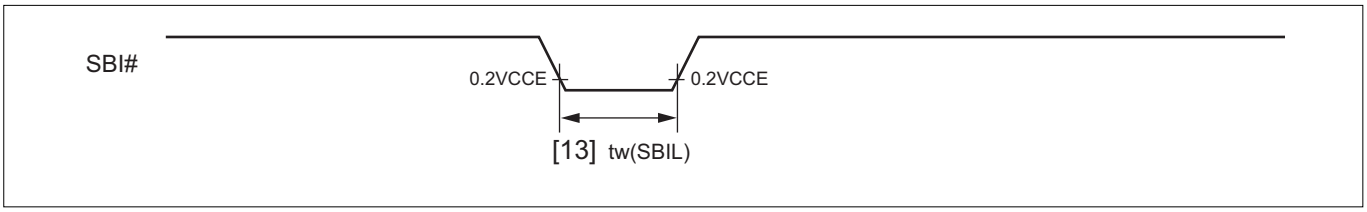


Figure 21.9.4 SBI Timing

(5) TIN

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.5
Timing requirements	tw(TIN)	TIN Input Pulse Width	$7\times\frac{tc(BCLK)}{2}$		ns	[14]

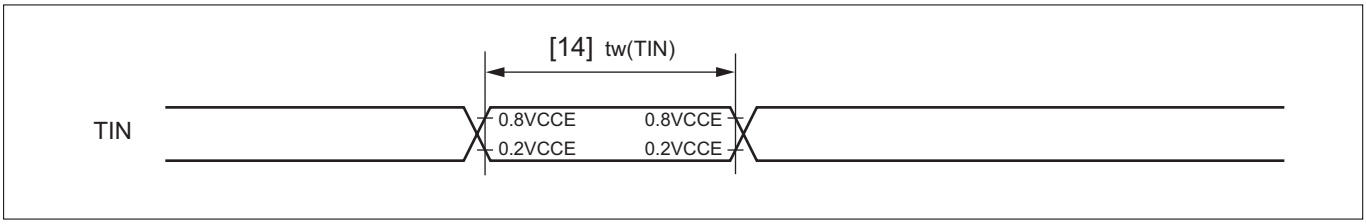


Figure 21.9.5 TIN Timing

(6) TO

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.6
Switching characteristics	td(BCLK-TO)	TO Output Delay Time		100	ns	[15]

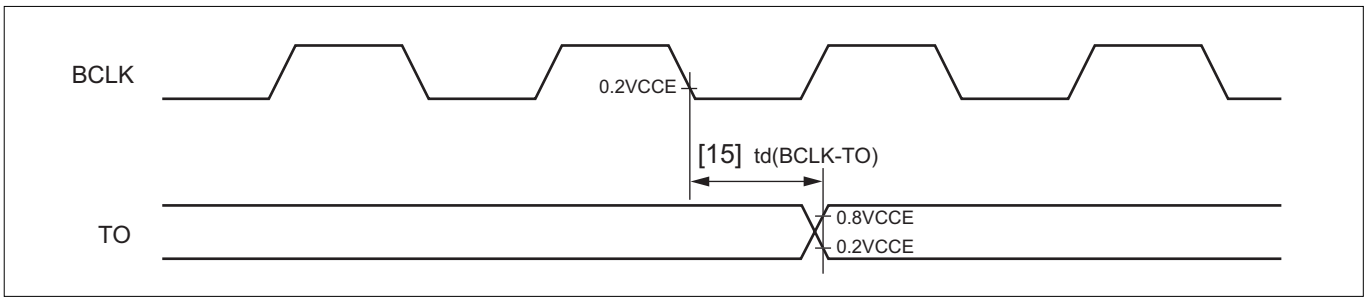


Figure 21.9.6 TO Timing

(7) TCLK

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.7
Timing requirements	tw(TCLKH)	TCLK Input "H" Pulse Width	$7\times\frac{tc(BCLK)}{2}$		ns	[99]
	tw(TCLKL)	TCLK Input "L" Pulse Width	$7\times\frac{tc(BCLK)}{2}$		ns	[100]

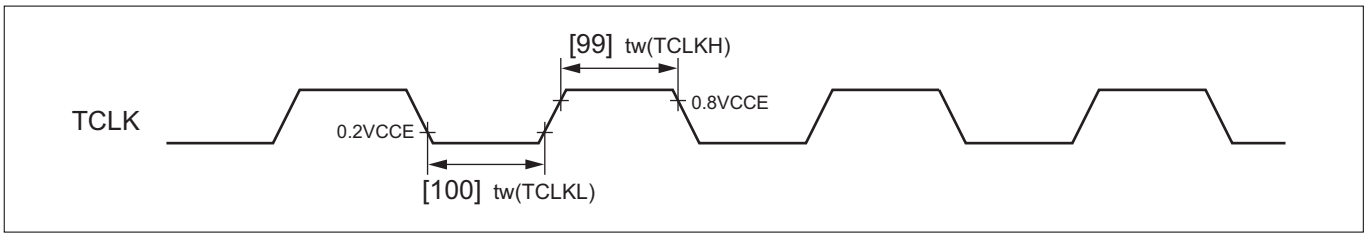
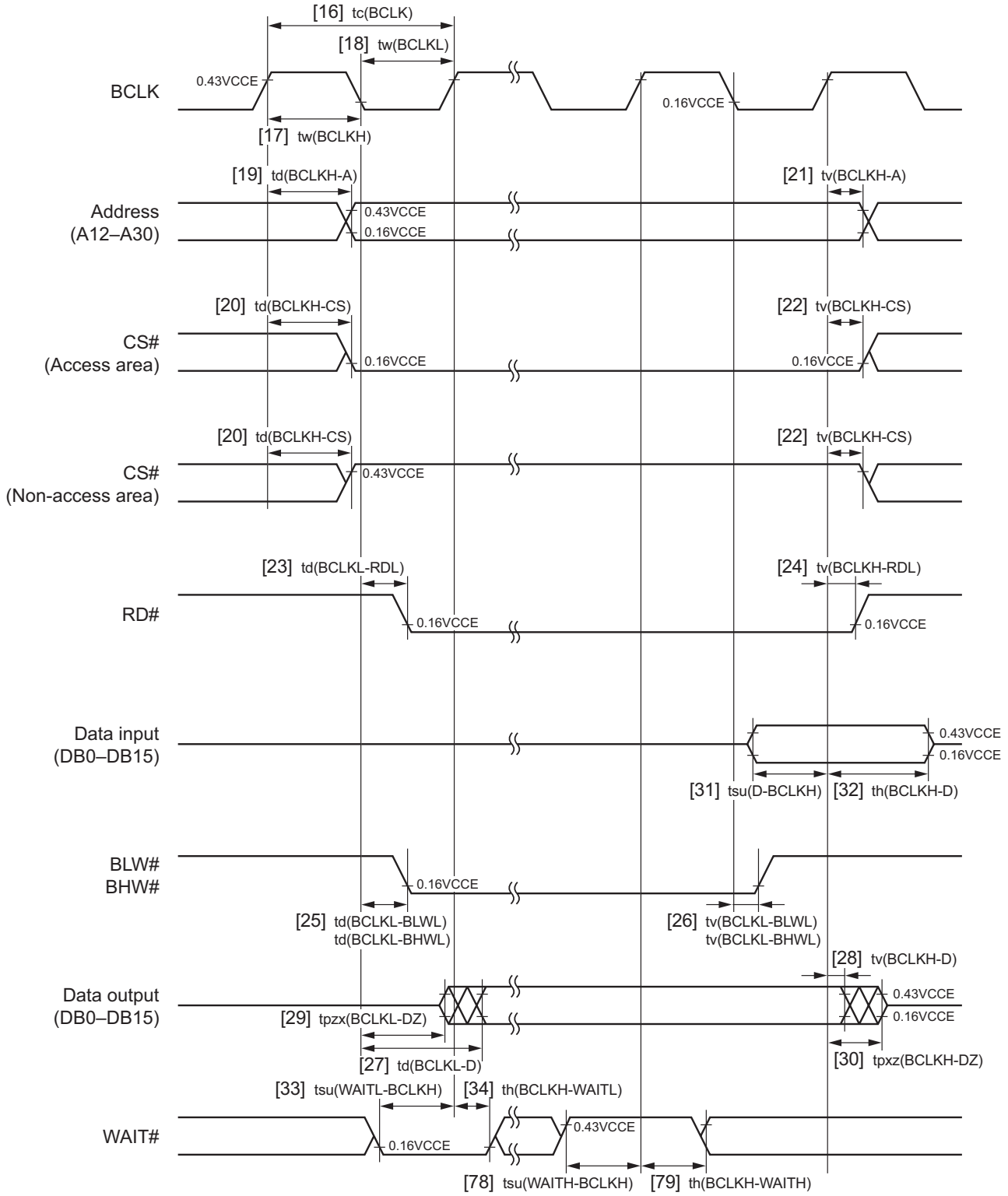


Figure 21.9.7 TCLK Timing

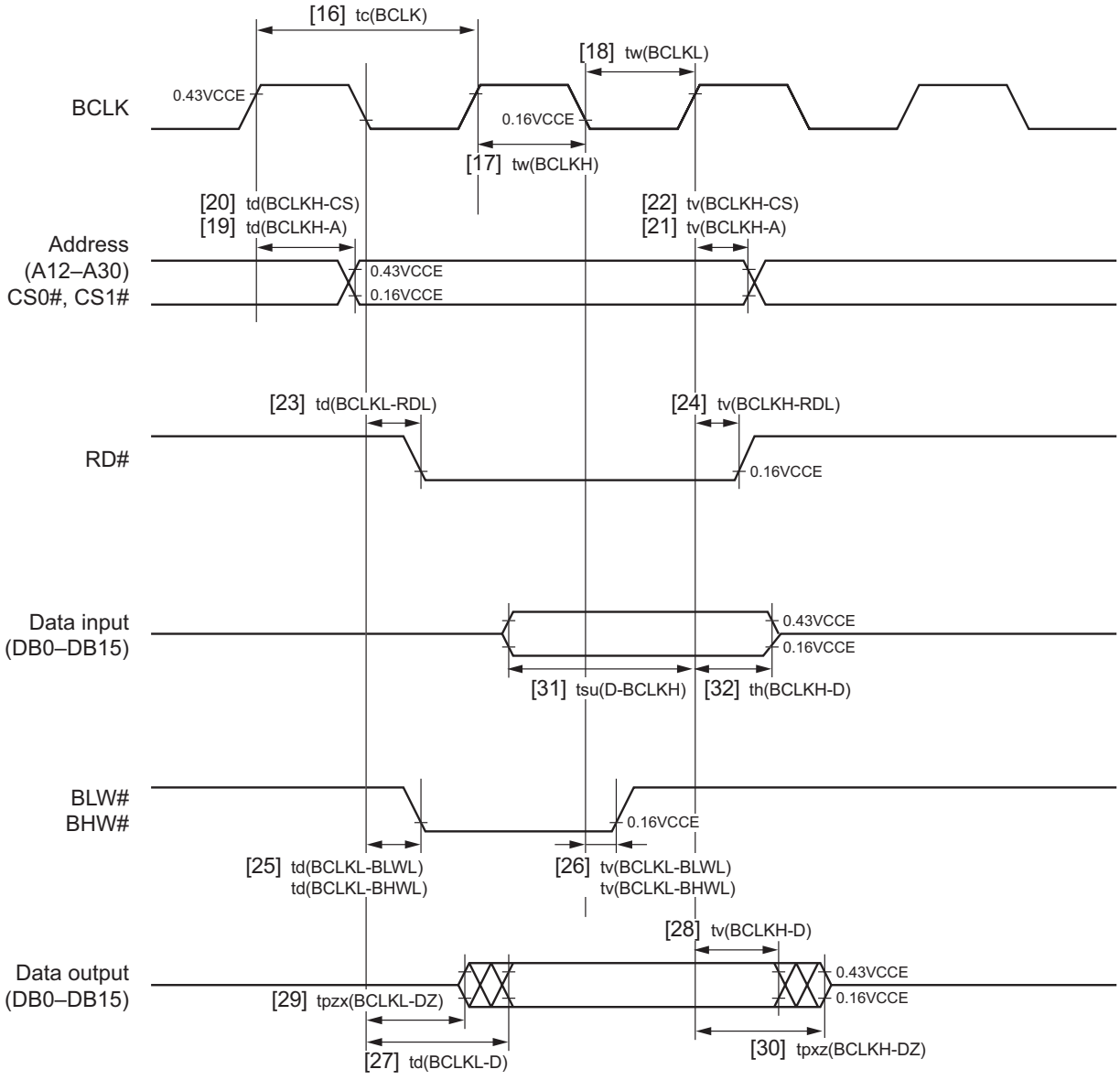
(8) Read and write timing (1/4)

	Symbol	Parameter	Rated Value		Unit	See Figs.
			MIN	MAX		21.9.8 21.9.9
Timing requirements	tsu(D-BCLKH)	Data Input Setup Time before BCLK	26		ns	[31]
	th(BCLKH-D)	Data Input Hold Time after BCLK	0		ns	[32]
	tsu(WAITL-BCLKH)	WAIT# Input Setup Time before BCLK	26		ns	[33]
	th(BCLKH-WAITL)	WAIT# Input Hold Time after BCLK	0		ns	[34]
	tsu(WAITH-BCLKH)	WAIT# Input Setup Time before BCLK	26		ns	[78]
	th(BCLKH-WAITH)	WAIT# Input Hold Time after BCLK	0		ns	[79]
Switching characteristics	tc(BCLK)	BCLK Output Cycle Time		$\frac{tc(XIN)}{2}$	ns	[16]
	tw(BCLKH)	BCLK Output "H" Pulse Width	$\frac{tc(BCLK)}{2}$ -5		ns	[17]
	tw(BCLKL)	BCLK Output "L" Pulse Width	$\frac{tc(BCLK)}{2}$ -5		ns	[18]
	td(BCLKH-A)	Address Delay Time after BCLK		29	ns	[19]
	td(BCLKH-CS)	Chip Select Delay Time after BCLK		30	ns	[20]
	tv(BCLKH-A)	Address Valid Time after BCLK	-11		ns	[21]
	tv(BCLKH-CS)	Chip Select Valid Time after BCLK	-11		ns	[22]
	td(BCLKL-RDL)	Read Delay Time after BCLK		14	ns	[23]
	tv(BCLKH-RDL)	Read Valid Time after BCLK	-12		ns	[24]
	td(BCLKL-BLWL) td(BCLKL-BHWL)	Write Delay Time after BCLK		14	ns	[25]
	tv(BCLKL-BLWL) tv(BCLKL-BHWL)	Write Valid Time after BCLK	-12		ns	[26]
	td(BCLKL-D)	Data Output Delay Time after BCLK		18	ns	[27]
	tv(BCLKH-D)	Data Output Valid Time after BCLK	-16		ns	[28]
	tpzx(BCLKL-DZ)	Data Output Enable Time after BCLK	-19		ns	[29]
	tpxz(BCLKH-DZ)	Data Output Disable Time after BCLK		5	ns	[30]



Notes: • For signal-to-signal timing, see Figure 21.9.10, "Read Timing (Relative to Read Pulse)," and Figure 21.9.11, "Write Timing (Relative to Write Pulse)."
• When using the threshold switching function, the data input and WAIT# voltage levels are determined with respect to the rated minimum and maximum values for VIH and VIL.

Figure 21.9.8 Read and Write Timing (Relative to BCLK) with 1 or more External WAIT States

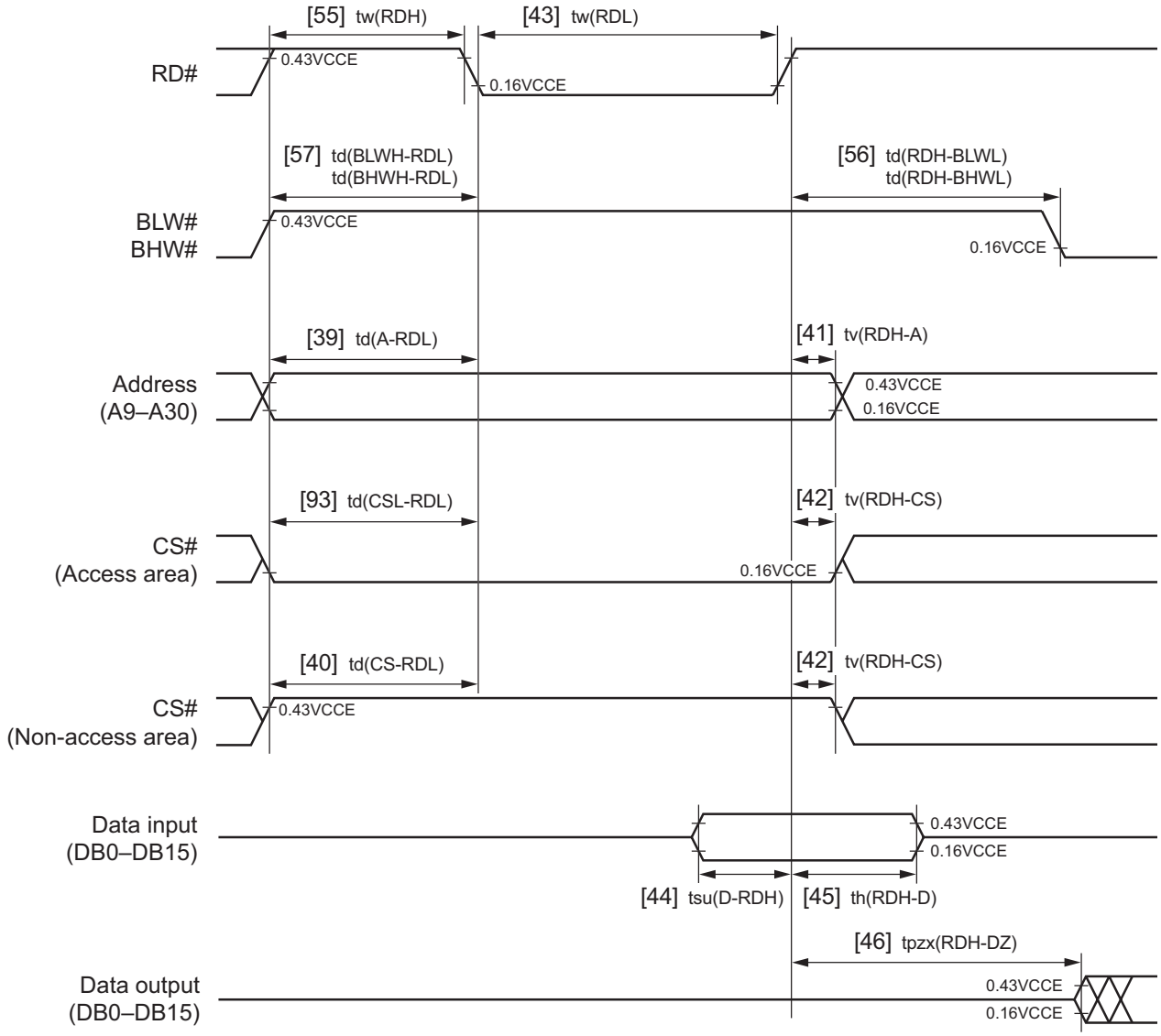


Note: • When using the threshold switching function, the data input voltage levels are determined with respect to the rated minimum and maximum values for VIH and VIL.

Figure 21.9.9 Read and Write Timing (Relative to BCLK) with 1 WAIT State

(8) Read and write timing (2/4)

	Symbol	Parameter	Rated Value		Unit	See Figs. 21.9.10 21.9.11
			MIN	MAX		
Timing requirements	tsu(D-RDH)	Data Input Setup Time before Read	30		ns	[44]
	th(RDH-D)	Data Input Hold Time after Read	0		ns	[45]
	tsu(WAITH-RDL) tsu(WAITL-RDL)	WAIT Input Setup Time before Read	tc(BCLK)+21		ns	[132]
	tw(WAITH)	WAIT "H" Pulse Width	26		ns	[133]
	tw(WAITL)	WAIT "L" Pulse Width	26		ns	[134]
	tsu(WAITH-BLWL) tsu(WAITH-BHWL) tsu(WAITL-BLWL) tsu(WAITL-BHWL)	WAIT Input Setup Time before Write (byte write mode)	$\frac{tc(BCLK)}{2}$ +21		ns	[135]
Switching characteristics	td(A-RDL)	Address Delay Time before Read	$\frac{tc(BCLK)}{2}$ -15		ns	[39]
	td(CS-RDL)	Chip Select Delay Time before Read	$\frac{tc(BCLK)}{2}$ -15		ns	[40]
	tv(RDH-A)	Address Valid Time after Read	0		ns	[41]
	tv(RDH-CS)	Chip Select Valid Time after Read	0		ns	[42]
	tw(RDL)	Read "L" Pulse Width	3× $\frac{tc(BCLK)}{2}$ -23		ns	[43]
	tpzx(RDH-DZ)	Data Output Enable Time after Read	$\frac{tc(BCLK)}{2}$		ns	[46]
	td(A-BLWL) td(A-BHWL)	Address Delaly Time before Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[47]
	td(CS-BLWL) td(CS-BHWL)	Chip Select Delay Time before Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[48]
	tv(BLWH-A) tv(BHWH-A)	Address Valid Time after Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[49]
	tv(BLWH-CS) tv(BHWH-CS)	Chip Select Valid Time after Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[50]
	tw(BLWL) tw(BHWL)	Write "L" Pulse Width (byte write mode)	tc(BCLK)-25		ns	[51]
	td(BLWL-D) td(BHWL-D)	Data Output Delay Time after Write (byte write mode)		15	ns	[52]
	tv(BLWH-D) tv(BHWH-D)	Data Output Valid Time after Write (byte write mode)	$\frac{tc(BCLK)}{2}$ -13		ns	[53]
	tpxz(BLWH-DZ) tpxz(BHWH-DZ)	Data Output Disable Time after Write (byte write mode)		$\frac{tc(BCLK)}{2}$ +5	ns	[54]
	tw(RDH)	Read "H" Pulse Width	$\frac{tc(BCLK)}{2}$ -3		ns	[55]
	td(RDH-BLWL) td(RDH-BHWL)	Write Delay Time after Read	$\frac{tc(BCLK)}{2}$ -10		ns	[56]
	td(BLWH-RDL) td(BHWH-RDL)	Read Delay Time after Write	$\frac{tc(BCLK)}{2}$ -10		ns	[57]
	td(CSL-RDL)	Chip Select Delay Time before Read	$\frac{tc(BCLK)}{2}$ -20		ns	[93]
	td(CSL-BLWL) td(CSL-BHWL)	Chip Select Delay Time before Write	$\frac{tc(BCLK)}{2}$ -20		ns	[95]



Note: • When usint the threshold switching function, the data input voltage levels are determined with respect to the rated minimum and maximum values for VIH and VIL.

Figure 21.9.10 Read Timing (Relative to Read Pulse)

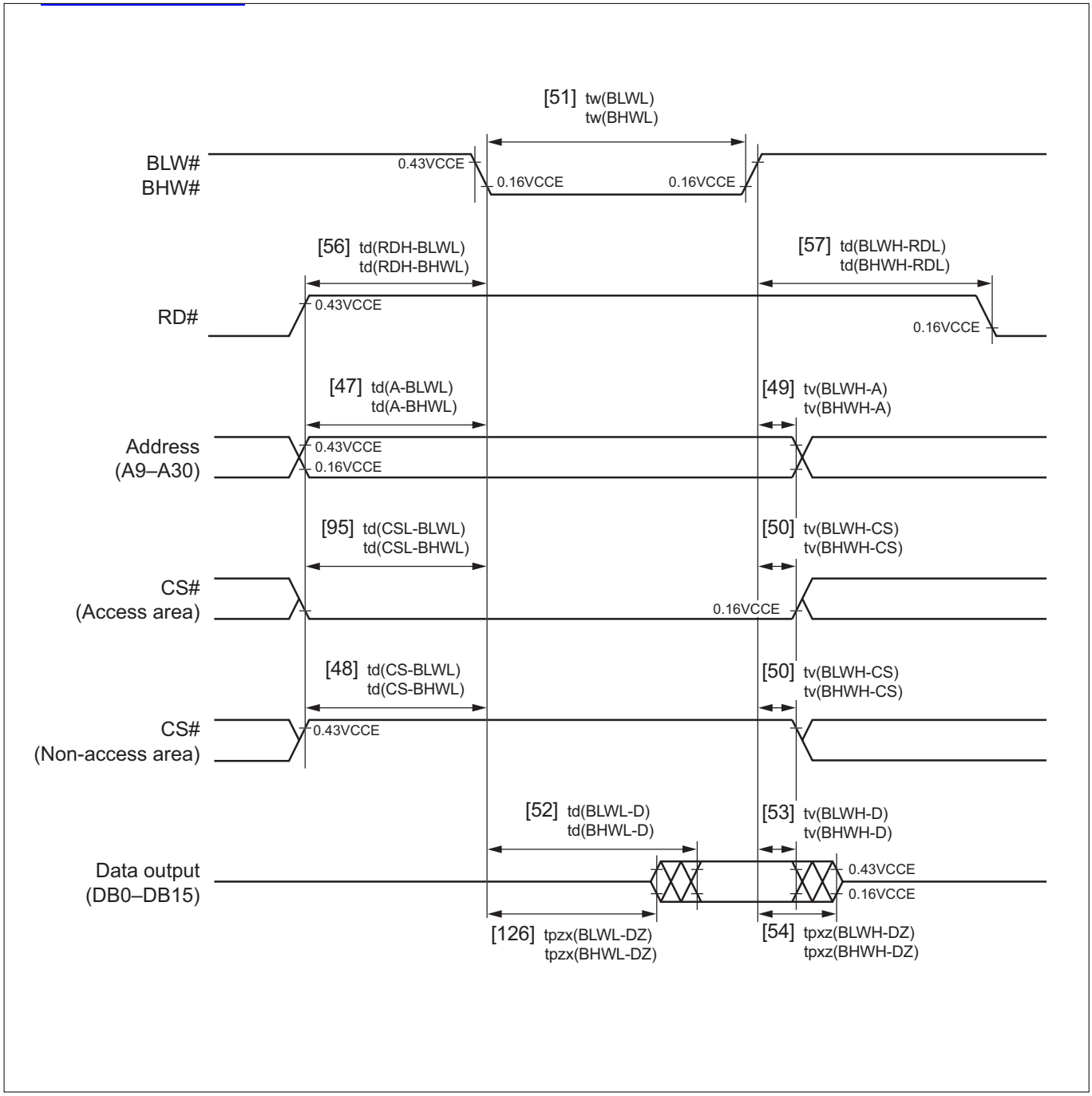


Figure 21.9.11 Write Timing (Relative to Write pulse)

(8) Read and write timing (3/4)

	Symbol	Parameter	Rated Value		Unit	See Figs.
			MIN	MAX		21.9.12
Timing requirements	tsu(D-RDH)	Data Input Setup Time before Read	30		ns	[44]
	th(RDH-D)	Data Input Hold Time before Read	0		ns	[45]
Switching characteristics	td(A-RDL)	Address Delay Time before Read	$\frac{tc(BCLK)}{2}$ -15		ns	[39]
	td(CSL-RDL)	Chip Select Delay Time before Read	$\frac{tc(BCLK)}{2}$ -15		ns	[40]
	tv(RDH-A)	Address Valid Time after Read	0		ns	[41]
	tv(RDH-CS)	Chip Select Valid Time after Read	0		ns	[42]
	tw(RDL)	Read "L" Pulse Width	$3\times \frac{tc(BCLK)}{2}$ -23		ns	[43]
	tpzx(RDH-DZ)	Data Output Enable Time after Read	$\frac{tc(BCLK)}{2}$		ns	[46]
	td(RDH-WRL)	Write Delay Time after Read (byte enable mode)	$\frac{tc(BCLK)}{2}$ -10		ns	[80]
	td(WRH-RDL)	Read Delay Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -10		ns	[81]
	td(CSL-RDL)	Chip Select Delay Time before Read	$\frac{tc(BCLK)}{2}$ -20		ns	[93]
	td(BLEL-RDL) td(BHEL-RDL)	Byte Enable Delay Time before Read (byte enable mode)	$\frac{tc(BCLK)}{2}$ -20		ns	[136]
	tv(RDH-BLEL) tv(RDH-BHEL)	Byte Enable Valid Time after Read (byte enable mode)	-5		ns	[137]

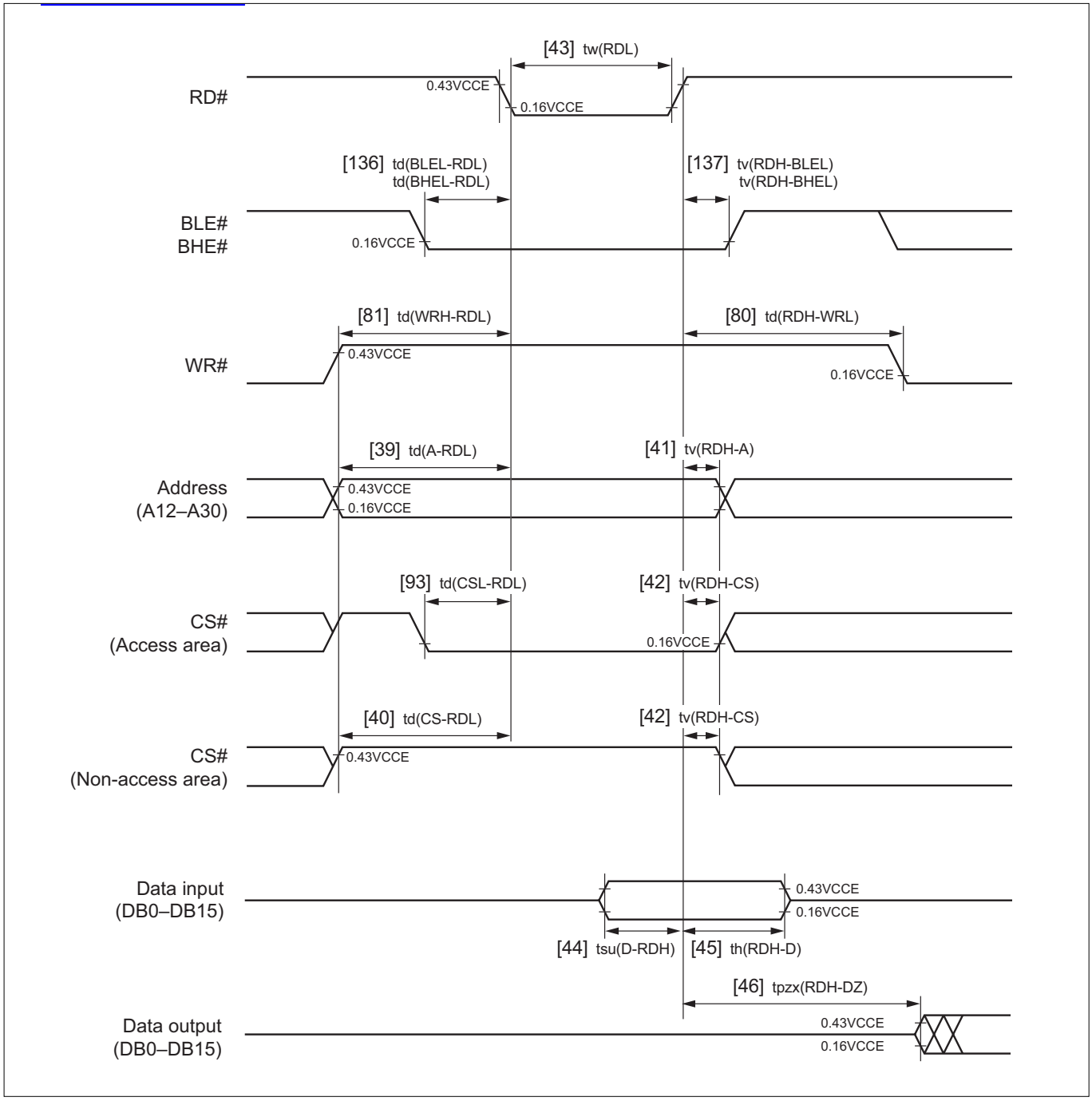


Figure 21.9.12 Read Timing (Byte Enable Mode)

(8) Read and write timing (4/4)

	Symbol	Parameter	Rated Value		Unit	see Figs.
			MIN	MAX		21.9.13
Switching characteristics	tw(WRL)	Write "L" Pulse Width (byte enable mode)	tc(BCLK)-25		ns	[68]
	td(A-WRL)	Address Delay Time before Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[69]
	td(CS-WRL)	Chip Select Delay Time before Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[70]
	tv(WRH-A)	Address Valid Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[71]
	tv(WRH-CS)	Chip Select Valid Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[72]
	td(BLEL-WRL) td(BHEL-WRL)	Byte Enable Delay Time before Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[73]
	tv(WRH-BLEL) tv(WRH-BHEL)	Byte Enable Valid Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[74]
	td(WRL-D)	Data Output Delay Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$	15	ns	[75]
	tv(WRH-D)	Data Output Valid Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -13		ns	[76]
	tpxz(WRH-DZ)	Data Output Disable Time after Write (byte enable mode)		$\frac{tc(BCLK)}{2}$ +5	ns	[77]
	td(RDH-WRL)	Write Delay Time after Read (byte enable mode)	$\frac{tc(BCLK)}{2}$ -10		ns	[80]
	td(WRH-RDL)	Read Delaly Time after Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -10		ns	[81]
	td(CSL-WRL)	Chip Select Delay Time before Write (byte enable mode)	$\frac{tc(BCLK)}{2}$ -15		ns	[96]

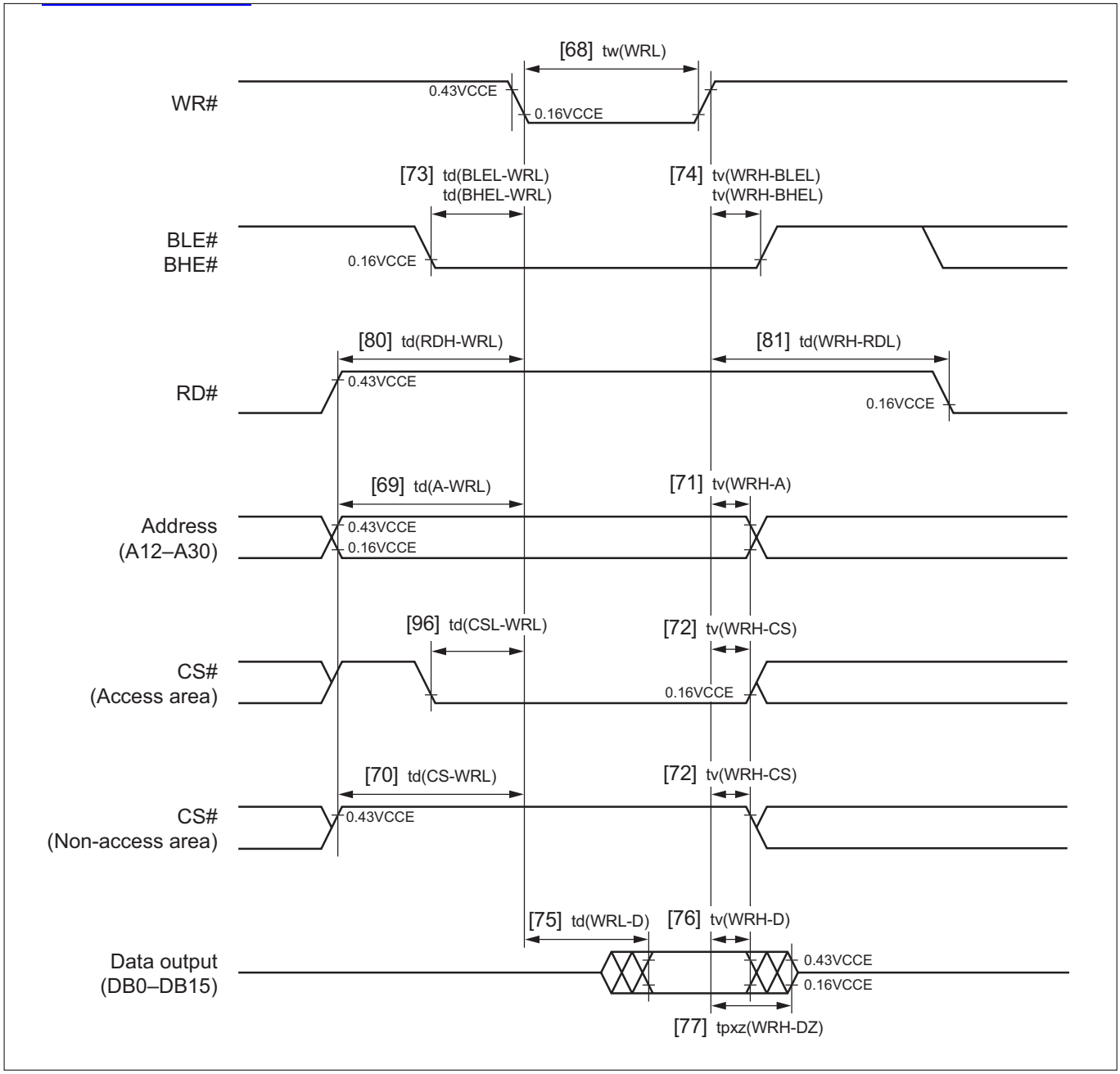


Figure 21.9.13 Write Timing (Byte Enable Mode)

(9) Bus arbitration timing

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.14
Timing requirements	tsu(HREQL-BCLKH)	HREQ# Input Setup Time before BCLK	27		ns	[35]
	th(CLKH-HREQL)	HREQ# Input Hold Time after BCLK	0		ns	[36]
Switching characteristics	td(BCLKL-HACKL)	HACK# Delay Time after BCLK		29	ns	[37]
	tv(BCLKL-HACKL)	HACK# Valid Time after BCLK	-11		ns	[38]

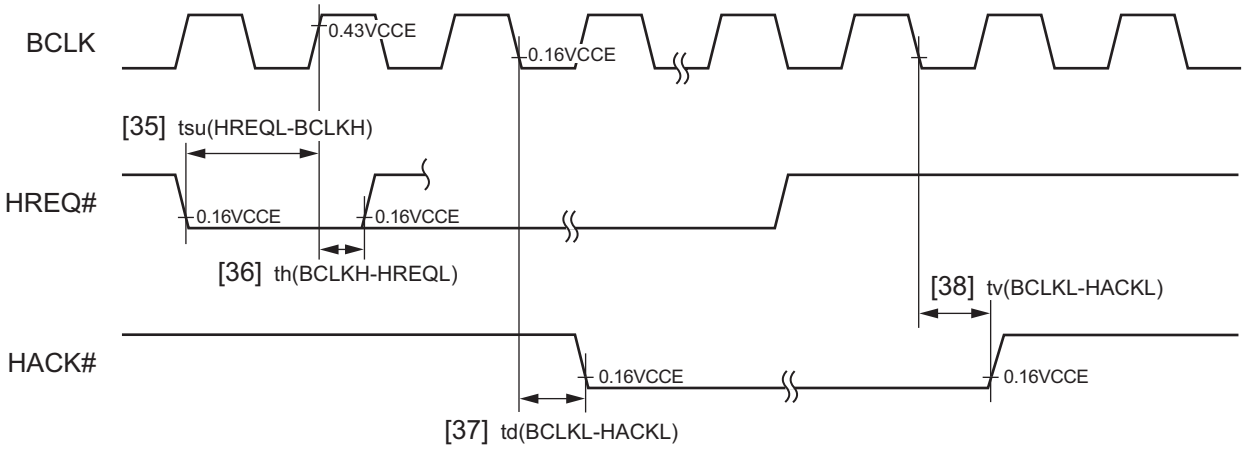
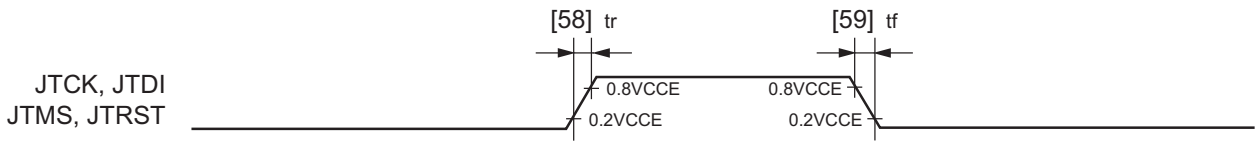


Figure 21.9.14 Bus Arbitration Timing

(10) JTAG pin input transition time

	Symbol	Parameter			Rated Value		Unit	See Fig.
					MIN	MAX		21.9.15
Timing requirements	tr	High-going Transition Time of Input	Other than JTRST pin (JTCK, JTDI, JTMS, JTDO)			10	ns	[58]
			JTRST pin	When using TAP		10	ns	
				When not using TAP		2	ms	
	tf	Low-going Transition Time of Input	Other than JTRST pin (JTCK, JTDI, JTMS, JTDO)			10	ns	[59]
			JTRST pin	When using TAP		10	ns	
				When not using TAP		2	ms	

Note: • The rated values here are guaranteed for the case where the measured pin load capacitance CL = 80pF.



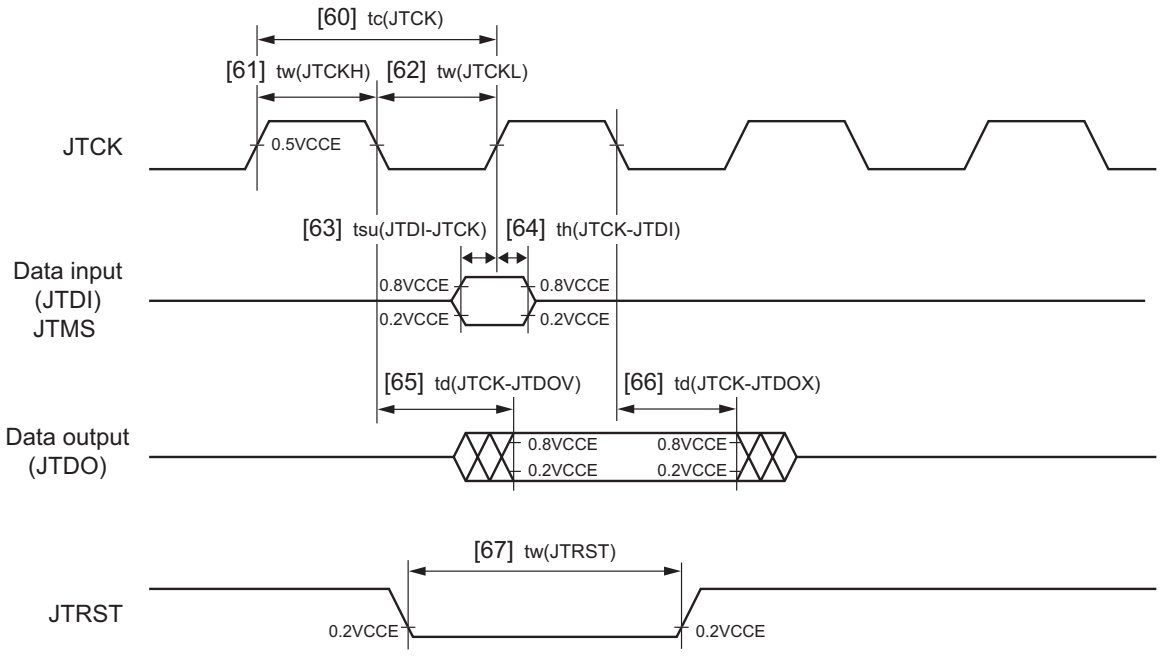
Note: • The rated values here are guaranteed for the case where the measured pin load capacitance CL = 80 pF.

Figure 21.9.15 JTAG Pin Input Transition Time

(11) JTAG interface timing

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.16
Timing requirements	tc(JTCK)	JTCK Input Cycle Time	100		ns	[60]
	tw(JTCKH)	JTCK Input "H" Pulse Width	40		ns	[61]
	tw(JTCKL)	JTCK Input "L" Pulse Width	40		ns	[62]
	tsu(JTDI-JTCK)	JTDI, JTMS Input Setup Time	15		ns	[63]
	th(JTCK-JTDI)	JTDI, JTMS Input Hold Time	20		ns	[64]
	tw(JTRST)	JTRST Input "L" Pulse Width	tc(JTCK)		ns	[67]
Switching characteristics	td(JTCK-JTDOV)	JTDO Output Delay Time after JTCK Low-going		40	ns	[65]
	td(JTCK-JTDOX)	Delay Time to JTDO Output Hi-Z after JTCK Low-going		40	ns	[66]

Note: • The rated values here are guaranteed for the case where the measured pin load capacitance CL = 80pF.



Note: • The rated values here are guaranteed for the case where the measured pin load capacitance CL = 80 pF.

Figure 21.9.16 JTAG Interface Timing

(12) RDT timing

	Symbol	Parameter	Rated Value		Unit	See Fig.
			MIN	MAX		21.9.17
Timing requirements	tc(RTDCLK)	RTDCLK Input Cycle Time	500		ns	[90]
	tw(RTDCLKH)	RTDCLK Input "H" Pulse Width	230		ns	[83]
	tw(RTDCLKL)	RTDCLK Input "L" Pulse Width	230		ns	[84]
	th(RTDCLKH-RTDRXD)	RTDRXD Input Hold Time	50		ns	[88]
	tsu(RTDRXD-RTDCLKL)	RTDRXD Input Setup Time	60		ns	[89]
Switching characteristics	td(RTDCLKH-RTDACK)	RTDACK Delay Time after RTDCLK Input		160	ns	[85]
	tv(RTDCLKL-RTDACK)	RTDACK Valid Time after RTDCLK Input		160	ns	[86]
	td(RTDCLKH-RTDTXD)	RTDTXD Delay Time after RTDCLK Input		tw(RTDCLKH)+160	ns	[87]

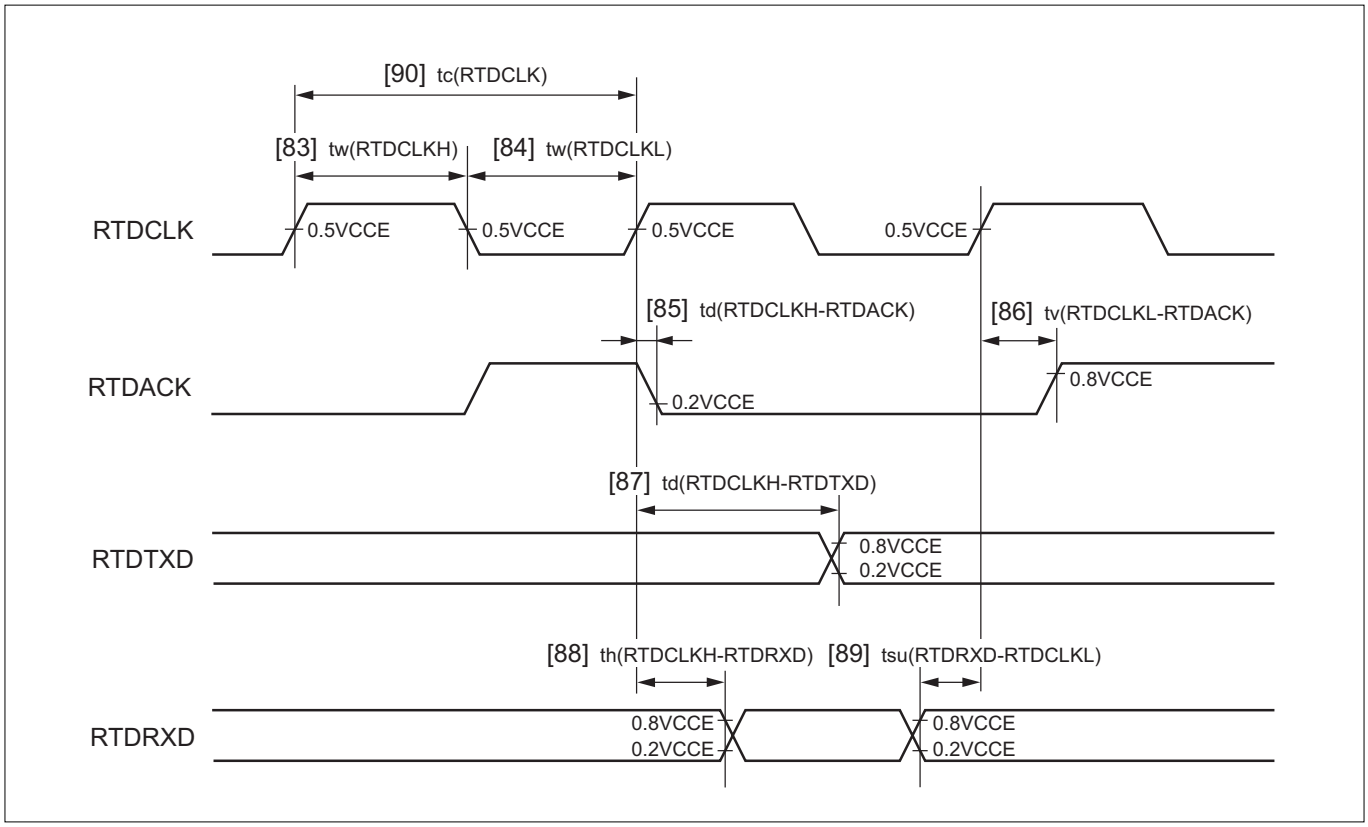


Figure 21.9.17 RTD Timing

This page is blank for reasons of layout.

APPENDIX 1

MECHANICAL SPECIFICAITONS

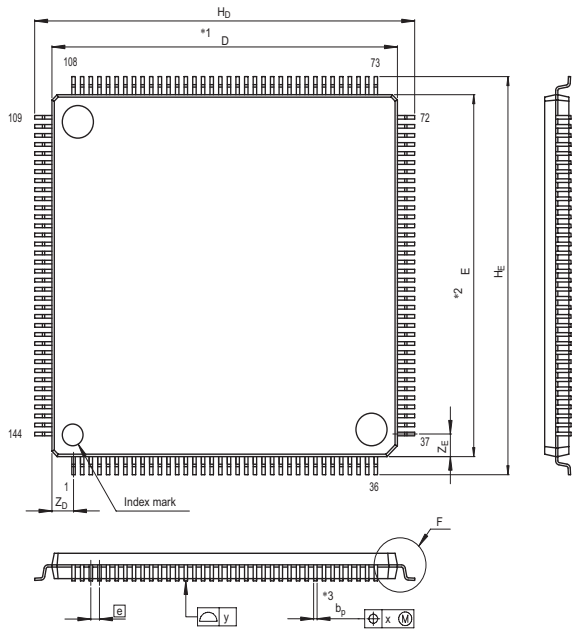
Appendix 1.1 Dimensional Outline Drawing

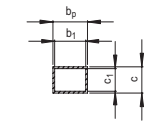
Appendix 1

Appendix 1.1 Dimensional Outline Drawing

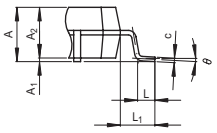
(1) 144-pin LQFP

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g





Terminal cross section



Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Note: • The latest Package Dimension is on Renesas Technology website.
Please make sure whether it is the latest or not and refer to it.

APPENDIX 2

INSTRUCTION PROCESSING TIME

Appendix 2.1 M32R/ECU Instruction Processing Time

Appendix 2.1 M32R/ECU Instruction Processing Time

For microcomputers, the number of instruction execution cycles in the E stage normally represents their instruction processing time. However, depending on pipeline operation, other stages may affect the instruction processing time. Especially when a branch instruction is executed, the processing time in each of the IF (Instruction Fetch), D (Decode) and E (Execution) stages of the next instruction must be taken into account.
The tables below show the instruction processing time in each pipelined stage of the M32R/ECU.

Appendix Table 2.1.1 Instruction Processing Time in Each Pipelined Stage

	Number of Execution Cycles in Each Stage				
Instruction	IF	D	E	MEM	WB
Load instructions (LD, LDB, LDUB, LDH, LDUH, LOCK)	R	1	1	R	1
Store instructions (ST, STB, STH, UNLOCK)	R	1	1	W	–
Multiply instructions (MUL)	R	1	3	–	1
Divide/remainder instructions (DIV, DIVU, REM, REMU)	R	1	37	–	1
Other instructions (including DSP function instructions)	R	1	1	–	1

The following shows the number of memory access cycles in the IF and MEM stages. Shown here are the minimum number of cycles required for memory access. Therefore, these values do not always reflect the number of cycles actually required for memory or bus access.

In write access, for example, although the CPU finishes the MEM stage by only writing to the write buffer, this operation actually is followed by a write to memory. Depending on the memory or bus state before or after the CPU requests a memory access, the instruction processing may take more time than the calculated value.

R (read cycle)

When existing in the instruction queue .....1 CPUCLK cycle
When reading the internal resource (ROM, RAM) .....1 CPUCLK cycle
When reading the internal resource (SFR) (byte or halfword) ..... 2 CPUCLK cycles
When reading the internal resource (SFR) (word) ..... 4 CPUCLK cycles
When reading external memory (byte or halfword) ..... 1 CPUCLK + 2 BCLK cycles (Note 1)
When reading external memory (word) ..... 1 CPUCLK + 4 BCLK cycles (Note 1)
When successively fetching instructions from external memory ..... 4 BCLK cycles (Note 1)

W (write cycle)

When writing to the internal resource (RAM) .....1 CPUCLK cycle
When writing to the internal resource (SFR) (byte or halfword) ..... 2 CPUCLK cycles
When writing to the internal resource (SFR) (word) ..... 4 CPUCLK cycles
When writing to external memory (byte or halfword) ..... 2 BCLK cycles (Note 1)
When writing to external memory (word) ..... 4 BCLK cycles (Note 1)

Note 1: This applies to the case where external access = one wait cycle. (When the M32R/ECU performs an external access, at least one wait cycle is inserted.)
Note: • BCLK and CPUCLK have the relationship 1 BCLK = 2 CPUCLK.

APPENDIX 3

PROCESSING OF UNUSED PINS

Appendix 3.1 Example Processing of Unused Pins

Appendix 3

Appendix 3.1 Example Processing of Unused Pins

An example of how to process the unused pins of the microcomputer is shown below.

(1) When operating in single-chip mode

Appendix Table 3.1.1 Example Processing of Unused Pins during Single-Chip Mode (Note 1)

Pin Name	Processing
Input/output ports (Note 2)	
P00–P07, P10–P17, P20–P27, P30–P37, P41–P47, P61–P63, P70–P77, P82–P87, P93–P97, P100–P107, P110–P117, P124–P127, P130–P137, P150, P153, P174, P175, P220, P221, P225	Set the port for input mode and pull each pin low to VSS or pull high to VCCE via a 1 kΩ–10 kΩ resistor. Or set the port for output mode and leave the pin open.
SBI# (Note 3)	Pull low to VSS via a 1 kΩ–10 kΩ resistor.
XOUT (Note 4)	Leave open
A/D converter	
AD0IN0–AD0IN15, AVREF0, AVSS0	Connect to VSS
AVCC0	Connect to VCCE
JTAG	
JTDO, JTMS, JTDI, JTCK	Pull high to VCCE or low to VSS via a 0–100 kΩ resistor
JTRST	Pull low to VSS via a 0–100 kΩ resistor

Note 1: Process the unused pins in the shortest wiring length possible (within 20 mm) from the microcomputer pins.
Note 2: If any port is set for output mode and left open, care should be taken because the port remains set for input before it is changed for output in a program after being reset. Therefore, the voltage level at the pin is instable, and the power supply current tends to increase while the port remains set for input. Because it is possible that the content of the port direction register will inadvertently be altered by noise or noise-induced runaway, higher reliability may be obtained by periodically setting the port direction register back again in a program. Note, however, that P221 is input-only port and does not work as an output port.
Note 3: Make sure that unintended falling edges due to noise, etc. will be not applied. (A falling edge at the SBI# input causes a system break interrupt to occur.)
Note 4: This is necessary when an external clock is connected to XIN.

(2) When operating in external extension/processor mode

Appendix Table 3.1.2 Example Processing of Unused Pins during External Extension/Processor Mode (Note 1)

Pin Name	Processing
Input/output ports (Note 2)	
P61–P63, P70–P77, P82–P87, P93–P97, P100–P107, P110–P117, P124–P127, P130–P137, P150, P153, P174, P175, P220, P221	Set the port for input mode and pull each pin low to VSS or pull high to VCCE via a 1 kΩ–10 kΩ resistor. Or set the port for output mode and leave the pin open.
A12–A30, DB0–DB15, BLW#/BLE#, BHW#/BHE#, RD#, CS#0, CS#1	Leave open
SBI# (Note 3)	Pull low to VSS via a 1 kΩ–10 kΩ resistor
XOUT (Note 4)	Leave open
A/D converter	
AD0IN0–AD0IN15, AVREF0, AVSS0	Connect to VSS
AVCC0	Connect to VCCE
JTAG	
JTDO, JTMS, JTDI, JTCK	Pull high to VCCE or low to VSS via a 0kΩ–100 kΩ resistor
JTRST	Pull low to VSS via a 0–100 kΩ resistor

Note 1: Process the unused pins in the shortest wiring length possible (within 20 mm) from the microcomputer pins.
Note 2: If any port is set for output mode and left open, care should be taken because the port remains set for input before it is changed for output in a program after being reset. Therefore, the voltage level at the pin is instable, and the power supply current tends to increase while the port remains set for input. Because it is possible that the content of the port direction register will inadvertently be altered by noise or noise-induced runaway, higher reliability may be obtained by periodically setting the port direction register back again in a program. Note, however, that P221 is input-only port and does not work as an output port.
Note 3: Make sure that unintended falling edges due to noise, etc. will be not applied. (A falling edge at the SBI# input causes a system break interrupt to occur.)
Note 4: This is necessary when an external clock is connected to XIN.

Appendix 3

This page is blank for reasons of layout.

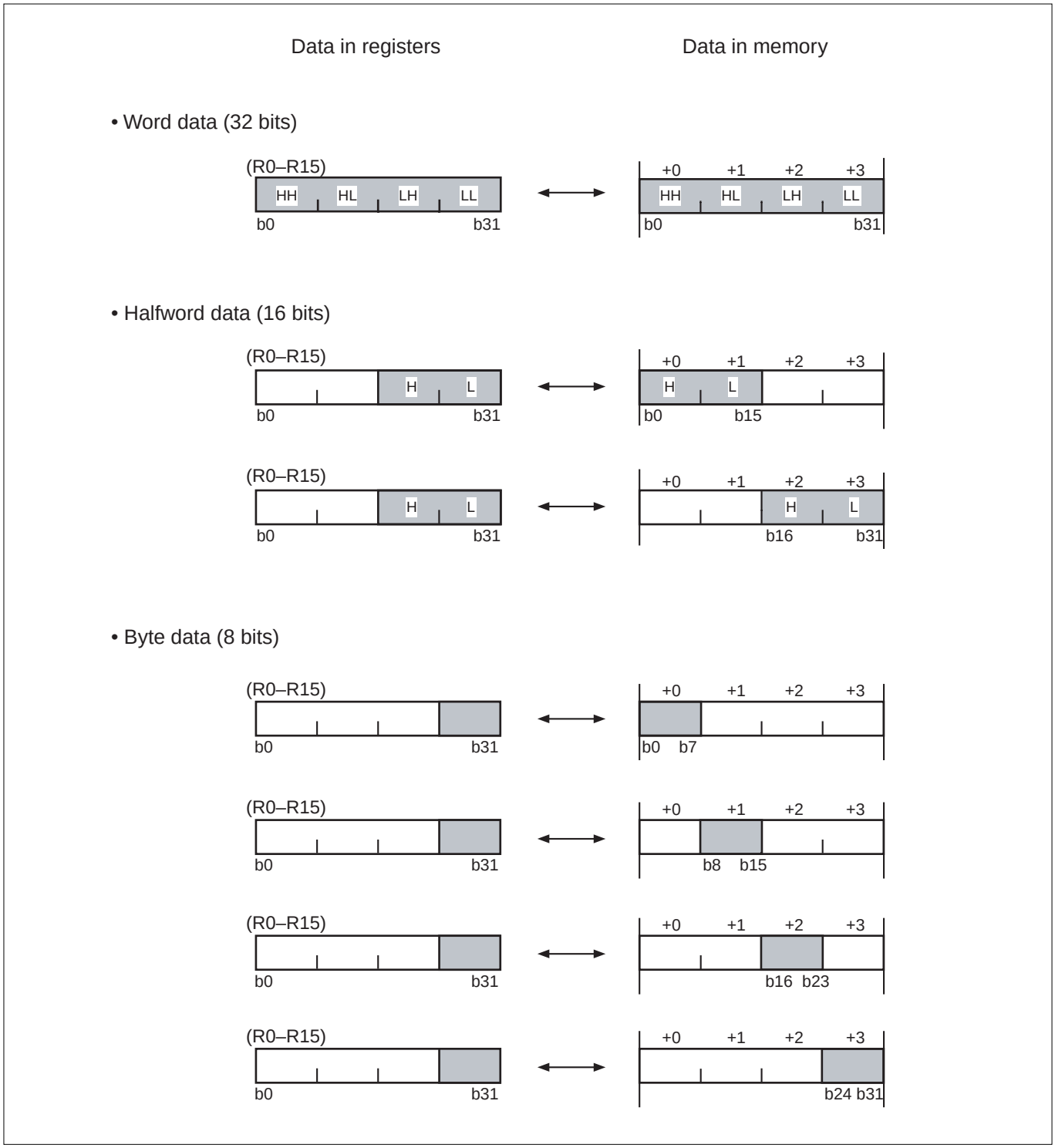
APPENDIX 4

SUMMARY OF PRECAUTIONS

Appendix 4.1 Notes on the CPU

• Precautions Regarding Data Transfer

When transferring data, be aware that data arrangements in registers and memory are different.



Appendix Figure 4.1.1 Differences in Data Arrangements

Appendix 4.2 Notes on the Address Space

- **Virtual flash emulation function**

The microcomputer has the function to map up to two 8-kbyte memory blocks of the internal RAM into areas of the internal flash memory (L banks) that are divided in 8-kbyte units, as well as to map up to two 4-kbyte memory blocks of the internal RAM into areas of the internal flash memory (S banks) that are divided in 4-kbyte units. This function is referred to as the virtual flash emulation function. For details about this function, refer to Section 6.6, “Virtual Flash Emulation Function.”

- **Dummy access areas**

Address H'0080 0600 - H'0080 0603 are dummy areas.
When there is access to these areas, writing value is disabled and reading value is undefined.
In addition, it does not effect on the other SFR area by writing and reading out operation to dummy access area.

Appendix 4.3 Notes on EIT

The Address Exception (AE) requires caution because if one of the instructions that use “register indirect + register update” addressing mode (following three) generates an address exception when it is executed, the values of the registers to be automatically updated (Rsrc and Rsrc2) become undefined.
Except that the values of Rsrc and Rsrc2 become undefined, these instructions behave the same way as when used in other addressing modes.

- **Applicable instructions**

```
LD   Rdest, @Rsrc+
ST   Rsrc1, @-Rsrc2
ST   Rsrc1, @+Rsrc2
```

If the above case applies, consider the fact that the register values become undefined when you design the processing to be performed after executing said instructions. (If an address exception occurs, it means that the system has some fatal fault already existing in it. Therefore, address exceptions must be used on condition that control will not be returned from the address exception handler to the program that was being executed when the exception occurred.)

Appendix 4.4 Notes on the Internal RAM

The following describes notes on the internal RAM

- When started by boot mode, internal RAM value is indefinite after started by boot mode in order to "Flash writing/erasing program" is transferred to internal RAM.

Appendix 4.5 Notes on Internal Flash Memory

The following describes precautions to be taken when programming/erasing the internal flash memory.

- When the internal flash memory is programmed or erased, a high voltage is generated internally. Because mode transitions during programming/erase operation may cause the chip to break down, make sure the mode setting pin/power supply voltages do not fluctuate to prevent unintended changes of modes.
- If the system uses any pins that are to be used by a general-purpose programming/erase tool, care must be taken to prevent adverse effects on the system when the tool is connected.

- If the internal flash memory needs to be protected while using a general-purpose programming/erase tool, set any ID in the flash memory protect ID verification area (H'0000 0084 to H'0000 0093).
- If the internal flash memory does not need to be protected while using a general-purpose programming/erase tool, fill the entire flash memory protect ID verification area (H'0000 0084 to H'0000 0093) with H'FF.
- If the Flash Status Register (FSTAT)'s each error status is to be cleared (initialized to H'80) by resetting the Flash Control Register 4 (FCNT4) FRESET bit, check to see that the Flash Status Register (FSTAT) FBUSY bit = "1" (ready) before clearing the error status.
- Before resetting the Flash Control Register 1 (FCNT1) FENTRY bit from "1" to "0", check to see that the Flash Status Register (FSTAT) FBUSY bit = "1" (ready).
- Do not clear the FENTRY bit if the Flash Control Register 1 (FCNT1) FENTRY bit = "1" and the Flash Status Register (FSTAT) FBUSY bit = "0" (being programmed or erased).
- When programming/erasing via JTAG, the flash memory can be programmed or erased regardless of the pin state because the FP pin is controlled internally within the chip.Appendix 4.5 Precautions To Be Observed after Reset

Appendix 4.6 Precautions To Be Observed after Exiting Reset

• Input/output ports

After exiting the reset state, the microcomputer's input/output ports are disabled against input in order to prevent current from flowing through the port. To use any ports in input mode, set the Port Input Special Function Control Register (PICNT) PIEN0 bit to enable them for input. For details, see Section 8.3, "Input/Output Port Related Registers."

Appendix 4.7 Notes on Input/Output Ports

• When using input/output ports in output mode

Because the value of the Port Data Register is undefined when exiting the reset state, the Port Data Register must have its initial value set in it before the Port Direction Register can be set for output. Conversely, if the Port Direction Register is set for output before setting data in the Port Data Register, the Port Data Register outputs an undefined value until any data is written into it.

• About the port input disable function

Because the input/output ports are disabled against input after reset, they must be enabled for input by setting the Port Input Enable (PIEN0) bit to "1" before their input functions can be used.
When disabled against input, the input/output ports are in a state equivalent to a situation where the pin has a low-level input applied. Consequently, if a peripheral input function is selected for any port while disabled against input by using the Port Operation Mode Register, the port may operate unexpectedly due to the low-level input on it.

• About the peripheral function input when it is set to the general purpose port

In the pin for both peripheral function input and general-purpose port, "H" level is entered to the peripheral function input when it is set to the general-purpose port in the operation mode register. Therefore, when "L" level is entered to the peripheral function input pin, edge signal is entered to the peripheral function input at manipulating operation mode register.

Appendix 4.8 Notes on the DMAC

• About writing to the DMAC related registers

Because DMA transfer involves exchanging data via the internal bus, the DMAC related registers basically can only be accessed for write immediately after reset or when transfer is disabled (transfer enable bit = "0"). When transfer is enabled, do not write to the DMAC related registers, except the DMA transfer enable bit, the transfer request flag and the DMA Transfer Count Register that is protected in hardware. This is a precaution necessary to ensure stable DMA operation.

The table below lists the registers that can or cannot be accessed for write.

Appendix Table 4.8.1 DMAC Related Registers That Can or Cannot Be Accessed for Write

Status	Transfer enable bit	Transfer request flag	DMA interrupt related registers	Other DMAC related registers
Transfer enabled	Can be accessed	Can be accessed	Can be accessed	Cannot be accessed
Transfer disabled	Can be accessed	Can be accessed	Can be accessed	Can be accessed

Even for registers that can exceptionally be written to while transfer is enabled, the following conditions must be observed:

(1) DMA Channel Control Register 0 transfer enable bit and transfer request flag

For all other bits in this register, be sure to write the same data that those bits had before the write. Note, however, that only writing "0" is effective for the transfer request flag.

(2) DMA Transfer Count Register

When transfer is enabled, this register is protected in hardware, so that any data rewritten to it is ignored.

(3) Rewriting the DMA source and DMA destination addresses on different channels by DMA transfer

Although this operation means accessing the DMAC related registers while DMA is enabled, there is no problem. Note, however, that no data can be transferred by DMA to the DMAC related registers on the currently active channel itself.

• Manipulating the DMAC related registers by DMA transfer

When manipulating the DMAC related registers by means of DMA transfer (e.g., reloading the DMAC related registers with the initial values by DMA transfer), do not write to the DMAC related registers on the currently active channel through that channel. (If this precaution is neglected, device operation cannot be guaranteed.) It is only the DMAC related registers on other channels that can be rewritten by means of DMA transfer. (For example, the DMAn Source Address and DMAn Destination Address Registers on channel 1 can be rewritten by DMA transfer through channel 0.)

• About the DMA Interrupt Request Status Register

When clearing the DMA Interrupt Request Status Register, be sure to write "1" to all bits, except those to be cleared. Writing "1" to any bits in this register has no effect, so that they retain the data they had before the write.

- **About the stable operation of DMA transfer**

To ensure the stable operation of DMA transfer, never rewrite the DMAC related registers, except the channel control register's transfer enable bit, unless transfer is disabled. One exception is that even when transfer is enabled, the DMA Source Address and DMA Destination Address Registers can be rewritten by DMA transfer from one channel to another.

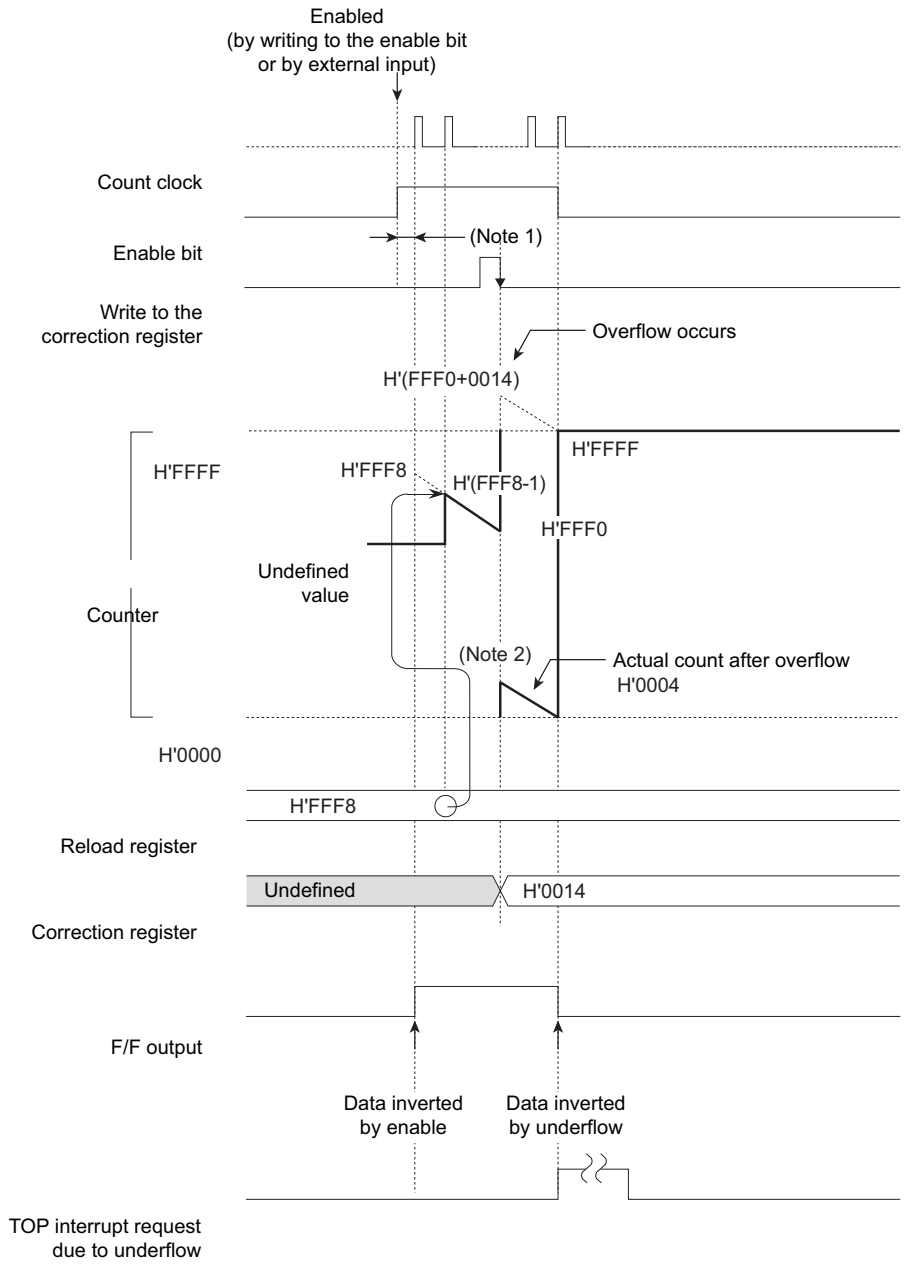
Appendix 4.9 Notes on the Multijunction Timers

Appendix 4.9.1 Notes on using TOP single-shot output mode

The following describes precautions to be observed when using TOP single-shot output mode.

- If the counter stops due to an underflow in the same clock period as the timer is enabled by external input, the former has priority so that the counter stops.
- If the counter stops due to an underflow in the same clock period as count is enabled by writing to the enable bit, the latter has priority so that count is enabled.
- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- Because the timer operates synchronously with the count clock, a count clock-dependent delay is included before starting F/F operation after the timer is enabled.
- When writing to the correction register, be careful not to cause the counter to overflow. Even if the counter overflows due to correction of counts, no interrupt requests are generated for reasons of an overflow. Therefore, if the counter underflows in the subsequent down-count after an overflow, a false interrupt request is generated for an underflow that includes the overflowed count.

In the example below, the reload register is initially set to H'FFF8. When the timer starts, the reload register value is loaded into the counter, letting it start counting down. In the diagram below, the value H'0014 is written to the correction register when the counter has counted down to H'FFF0. As a result of this correction, the count overflows to H'0004 and the counter fails to count correctly. Also, an interrupt request is generated for an erroneous overflowed count.



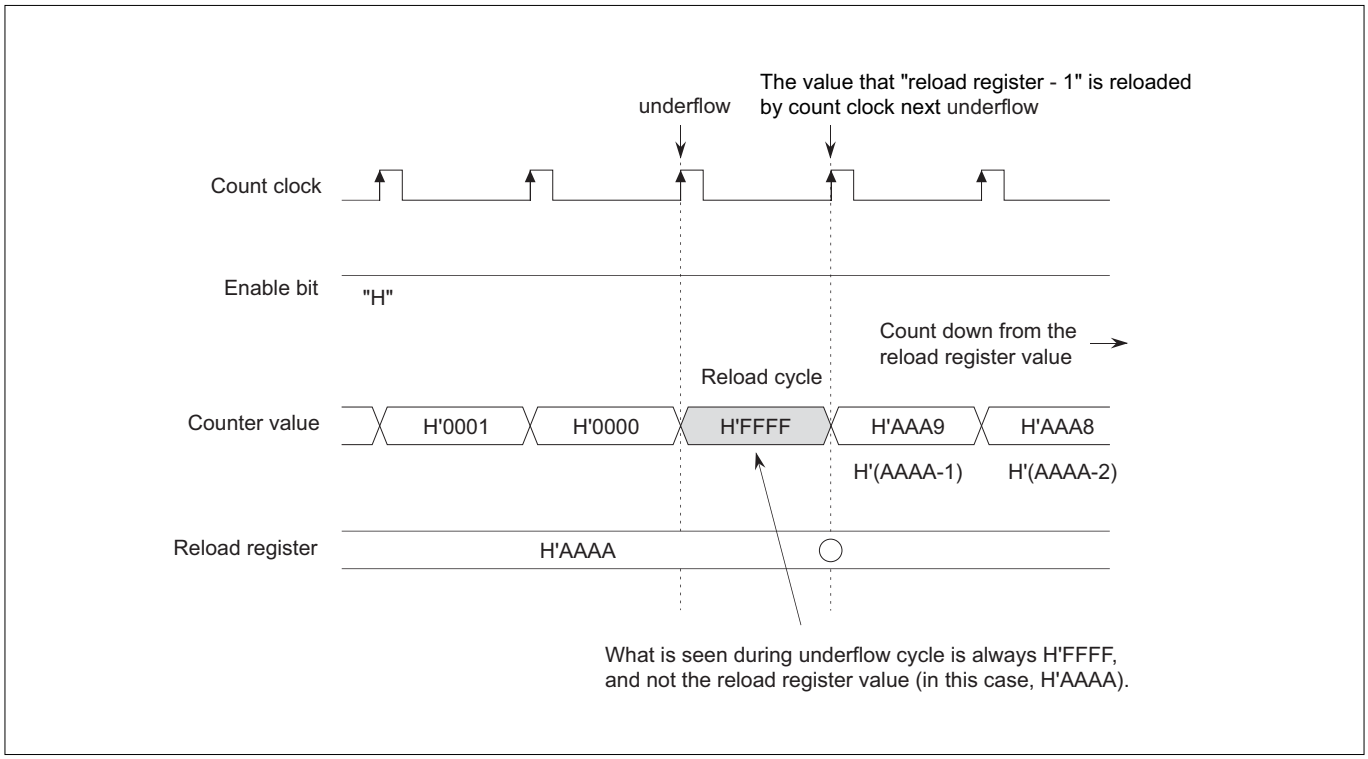
Note 1: A count clock dependent delay is included before F/F output changes state after the timer is enabled.
Note 2: The value that "reload register - 1" is reloaded.
Note: • This diagram does not show detailed timing information.

Appendix Figure 4.9.1 Example of an Operation in TOP Single-shot Output Mode Where Count Overflows Due to Correction

Appendix 4.9.2 Notes on using TOP delayed single-shot output mode

The following describes precautions to be observed when using TOP delayed single-shot output mode.

- If the counter stops due to an underflow in the same clock period as the timer is enabled by external input, the former has priority so that the counter stops.
- If the counter stops due to an underflow in the same clock period as count is enabled by writing to the enable bit, the latter has priority so that count is enabled.
- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- Even if the counter overflows due to correction of counts, no interrupt requests are generated for reasons of an overflow. Therefore, if the counter underflows in the subsequent down-count after an overflow, a false interrupt request is generated for an underflow that includes the overflowed count.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF. The reload reads the value that "the reload register -1" into the counter at the timing of the counter clock after the underflow.



Appendix Figure 4.9.2 Counter Value Immediately after Underflow

Appendix 4.9.3 Notes on using TOP continuous output mode

The following describes precautions to be observed when using TOP continuous output mode.

- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF. The reload reads the value that "the reload register -1" into the counter at the timing of the counter clock after the underflow.
- Because the timer operates synchronously with the count clock, a count clock-dependent delay is included before F/F output is inverted after the timer is enabled.

Appendix 4.9.4 Notes on using TIO measure free-run/clear input modes

The following describes precautions to be observed when using TIO measure free-run/clear input modes.

- If measure event input and write to the counter occur in the same clock period, the write value is set in the counter while at the same time latched into the measure register.

Appendix 4.9.5 Notes on using TIO PWM output mode

The following describes precautions to be observed when using TIO PWM output mode.

- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF. The reload reads the value that "the reload register -1" into the counter at the timing of the counter clock after the underflow.
- Because the timer operates synchronously with the count clock, a count clock-dependent delay is included before F/F output is inverted after the timer is enabled.

Appendix 4.9.6 Notes on using TIO single-shot output mode

The following describes precautions to be observed when using TIO single-shot output mode.

- If the counter stops due to an underflow in the same clock period as the timer is enabled by external input, the former has priority so that the counter stops.
- If the counter stops due to an underflow in the same clock period as count is enabled by writing to the enable bit, the latter has priority so that count is enabled.
- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- Because the timer operates synchronously with the count clock, a count clock-dependent delay is included before F/F output is inverted after the timer is enabled.

Appendix 4.9.7 Notes on using TIO delayed single-shot output mode

The following describes precautions to be observed when using TIO delayed single-shot output mode.

- If the counter stops due to an underflow in the same clock period as the timer is enabled by external input, the former has priority so that the counter stops.
- If the counter stops due to an underflow in the same clock period as count is enabled by writing to the enable bit, the latter has priority so that count is enabled.
- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF. The reload reads the value that "the reload register -1" into the counter at the timing of the counter clock after the underflow.

Appendix 4.9.8 Notes on using TIO continuous output mode

The following describes precautions to be observed when using TIO continuous output mode.

- If the timer is enabled by external input in the same clock period as count is disabled by writing to the enable bit, the latter has priority so that count is disabled.
- If the counter is accessed for read at the cycle of underflow, the counter value is read as H'FFFF. The reload reads the value that "the reload register -1" into the counter at the timing of the counter clock after the underflow.
- Because the timer operates synchronously with the count clock, a count clock-dependent delay is included before F/F output is inverted after the timer is enabled.

Appendix 4.9.9 Notes on using TMS measure input

The following describes precautions to be observed when using TMS measure input.

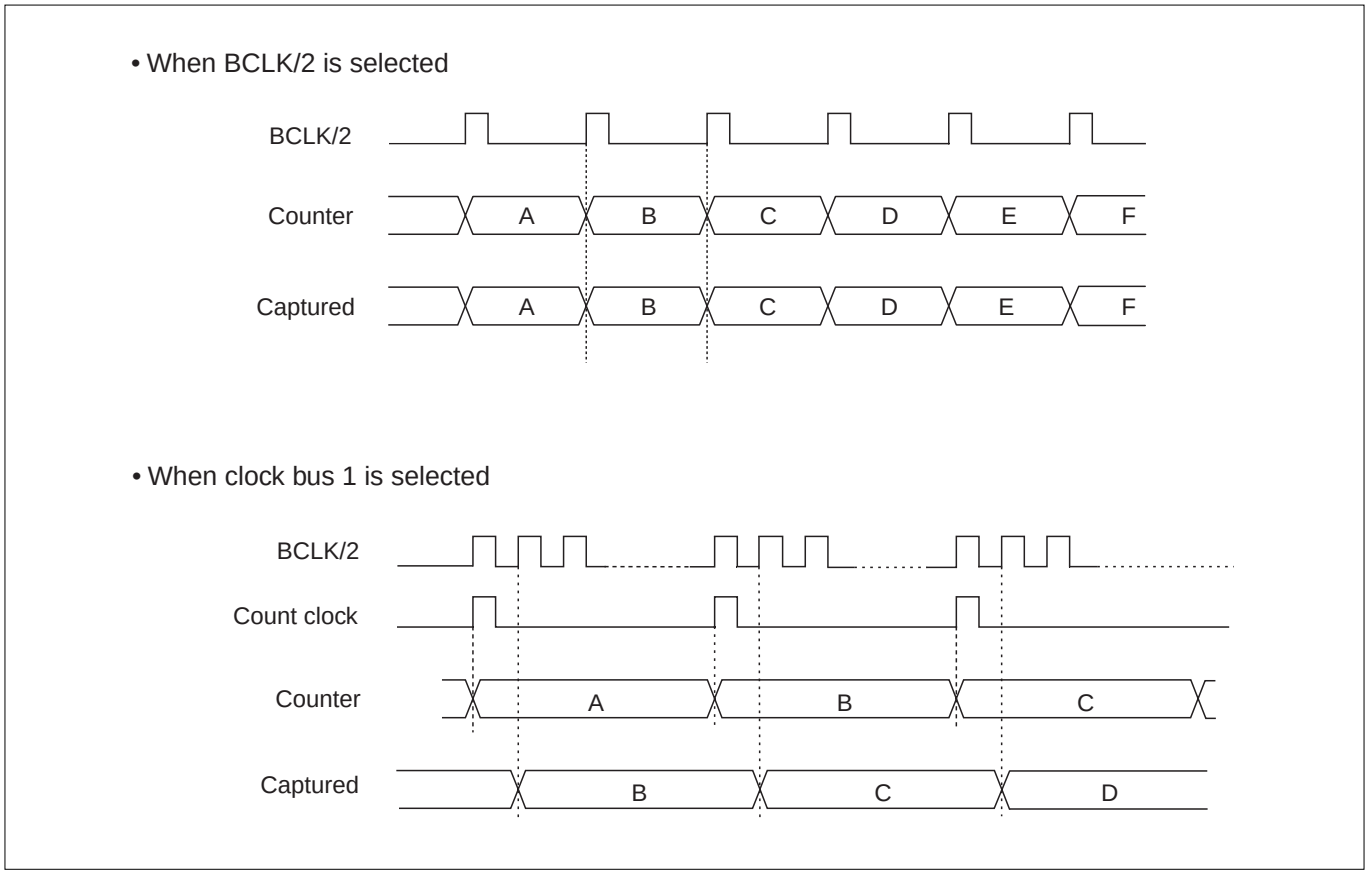
- If measure event input and write to the counter occur in the same clock period, the write value is set in the counter while at the same time latched into the measure register.

Appendix 4.9.10 Notes on using TML measure input

The following describes precautions to be observed when using TML measure input.

- If measure event input and write to the counter occur in the same clock period, the write value is set in the counter, whereas the up-count value (before being rewritten) is latched into the measure register.
- If clock bus 1 is selected and any clock other than BCLK/2 is used for the timer, the counter cannot be written normally. Therefore, when using any clock other than BCLK/2, do not write to the counter.
- If clock bus 1 is selected and any clock other than BCLK/2 is used for the timer, the value captured into the measure register is one count larger the counter value. During the count clock to BCLK/2 period interval, how ever, the captured value is exactly the counter value.

The diagram below shows the relationship between counter operation and the valid data that can be captured.



Appendix Figure 4.9.3 Mistimed Counter Value and the Captured Value

Appendix 4.10 Notes on the A/D Converter

• Forcible termination during scan operation

If A/D conversion is forcibly terminated by setting the A/D conversion stop bit (AD0CSTP) to "1" during scan mode operation and the A/D data register for the channel that was in the middle of conversion is accessed for read, the read value shows the last conversion result that had been transferred to the data register before the conversion was forcibly terminated.

• Modification of the A/D converter related registers

If the content of any register—A/D Conversion Interrupt Control Register, Single or Scan Mode Registers or A/D Successive Approximation Register, except the A/D conversion stop bit—is modified in the middle of A/D conversion, the conversion result cannot be guaranteed. Therefore, do not modify the contents of these registers while A/D conversion is in progress, or be sure to restart A/D conversion if register contents have been modified.

• Handling of analog input signals

When using the A/D Converter with its sample-and-hold function disabled, make sure the analog input level is fixed during A/D conversion.

• A/D conversion completed bit read timing

To read the A/D conversion completed bit (Single Mode Register 0 bit 5 or Scan Mode Register 0 bit 5) immediately after A/D conversion has started, be sure to adjust the timing 2 BCLK periods by, for example, inserting a NOP instruction before read.

• Regarding the analog input pins

Appendix Figure 4.10.1 shows the internal equivalent circuit of the A/D Converter's analog input part. To obtain accurate A/D conversion results, make sure the internal capacitor C2 of the A/D conversion circuit is charged up within a predetermined time (sampling time). To meet this sampling time requirement, it is recommended that a stabilizing capacitor C1 be connected external to the chip.
The method for determining the necessary value of this external stabilizing capacitor with respect to the output impedance of an analog output device is described below. Also, an explanation is made of the case where the output impedance of an analog output device is low and the external stabilizing capacitor C1 is unnecessary.

• Rated value of the absolute accuracy

The rated value of the absolute accuracy is the actual performance value of the microcomputer alone, with influences of the power supply wiring and noise on the board not taken into account. When designing the application system, use caution for the board layout by, for example, separating the analog circuit power supply and ground (AVCC0, AVSS0 and VREF0) from those of the digital circuit and incorporating measures to prevent the analog input pins from being affected by noise, etc. from other digital signals.

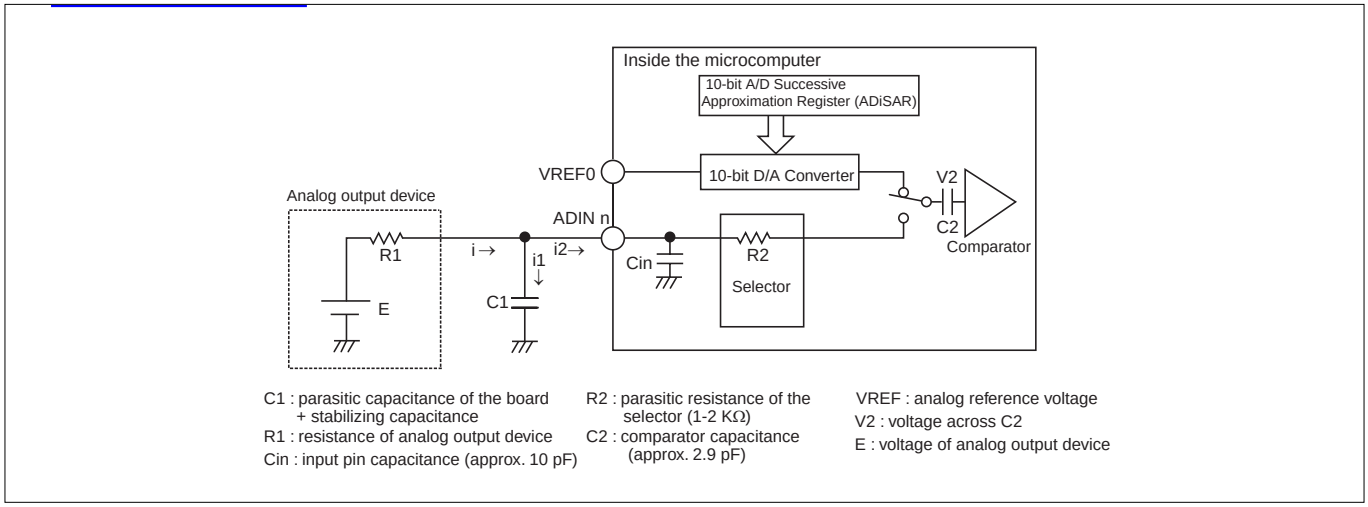
• Single and scan mode operation under sample-and-hold enabled mode

If either A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIM1) or A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as sample-and-hold enabled, both single and scan mode operate under sample-and-hold enabled mode.

If either A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIMI1) or A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as sample-and-hold enabled, and A/D sample-and-hold conversion speed select (ADSSHSPD) bit in A/D0 Single Mode Register 1 (AD0SM1) or A/D sample-and-hold conversion speed select (ADCSHSPD) bit in A/D0 Scan Mode Register 1 (AD0SCM1) is selected as fast sample-and-hold, both single and scan mode operate in fast sample-and-hold conversion spped.

Appendix 4

To use single or scan mode under sample-and-hold enabled mode, make sure that A/D conversion method select (ADSSHSL) bit in A/D0 Single Mode Register 1 (AD0SIM1), A/D conversion method select (ADCSHSL) bit in A/D0 Scan Mode Register 1 (AD0SCM1), A/D sample-and-hold conversion spped select (ADSSHSPD)bit in A/D0 Single Mode Register 1 (AD0SIM1) and A/D sample-and-hold conversion speed select (ADCSHSPD) bit in A/D0 Scan Mode Register 1 (AD0SCM1) are set in the same value.



Appendix Figure 4.10.1 Internal Equivalent Circuit of the Analog Input Part

(a) Example for calculating the external stabilizing capacitor C1 (addition of this capacitor is recommended)

Assuming the R1 in Appendix Figure 4.10.1 is infinitely large and that the current necessary to charge the internal capacitor C2 is supplied from C1, if the potential fluctuation, Vp, caused by capacitance division of C1 and C2 is to be within 0.1 LSB, then what amount of capacitance C1 should have. For a 10-bit A/D Converter where VREF is 5.12 V, 1 LSB determination voltage = 5.12 V / 1,024 = 5 mV. The potential fluctuation of 0.1 LSB means a 0.5 mV fluctuation.

The relationship between the capacitance division of C1 and C2 and the potential fluctuation, Vp, is obtained by the equation below:

$$Vp = \frac{C2}{C1 + C2} \times (E - V2) \quad \text{---------- Eq. A-1}$$

Vp is also obtained by the equation below:

$$Vp = Vp1 \times \sum_{i=0}^{x-1} \frac{1}{2^i} < \frac{VREF}{10 \times 2^x} \quad \text{---------- Eq. A-2}$$

where Vp1 = potential fluctuation in the first A/D conversion performed
and x = 10 for a 10-bit resolution A/D converter

When Eq. A-1 and Eq. A-2 are solved, the following results:

$$C1 = C2 \left\{ \frac{E - V2}{Vp1} - 1 \right\} \quad \text{---------- Eq. A-3}$$

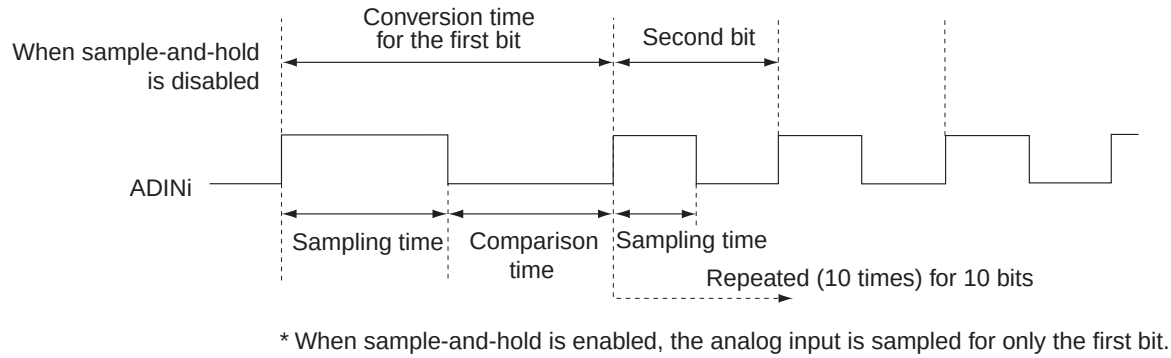
$$\therefore \quad C1 > C2 \left\{ 10 \times 2^x \times \sum_{i=0}^{x-1} \frac{1}{2^i} - 1 \right\} \quad \text{---------- Eq. A-4}$$

Thus, for a 10-bit resolution A-D Converter where C2 = 2.9 pF, C1 is 0.06 µF or more. Use this value for reference when setting up C1.

(b) Maximum value of the output impedance R1 when C1 is not added

If the external capacitor C1 in Appendix Figure 4.9.1 is not used, examination must be made to see if the analog output device can fully charge C2 within a predetermined time. First, the equation to find i2 when C1 in Appendix Figure 4.9.1 does not exist is shown below.

$$i2 = \frac{C2(E - V2)}{Cin \times R1 + C2(R1 + R2)} \times \exp \frac{\{ \quad - t \quad \}}{Cin \times R1 + C2(R1 + R2)} \quad \text{---------- Eq. B-1}$$



Appendix Figure 4.10.2 A/D Conversion Timing Diagram

Appendix Figure 4.10.2 shows an A/D conversion timing diagram. C2 must be charged up within the sampling time shown in this diagram. When the sample-and-hold function is disabled, the sampling time for the second and subsequent bits is about half that of the first bit.
The sampling times at the respective conversion speeds are listed in the Appendix Table 4.9.1. Note that when the sample-and-hold function is enabled, the analog input is sampled for only the first bit.

Appendix Table 4.10.1 Sampling Time (in Which C2 Needs to Be Charged)

Conversion start method	Conversion speed		Sampling time for the first bit	Sampling time for the second and subsequent bits
Single mode (when sample-and-hold disabled)	Slow mode	Normal speed	27.5BCLK	13.5BCLK
		Double speed	15.5BCLK	7.5BCLK
	Fast mode	Normal speed	11.5BCLK	5.5BCLK
		Double speed	7.5BCLK	3.5BCLK
Single mode (when sample-and-hold enabled)	Slow mode	Normal speed	27.5BCLK	–
		Double speed	15.5BCLK	–
	Fast mode	Normal speed	11.5BCLK	–
		Double speed	7.5BCLK	–
Comparator mode	Slow mode	Normal speed	27.5BCLK	–
		Double speed	15.5BCLK	–
	Fast mode	Normal speed	11.5BCLK	–
		Double speed	7.5BCLK	–

Therefore, the time in which C2 needs to be charged is found from Eq. B-1, as follows:

Sampling time (in which C2 needs to be charged) > Cin × R1 + C2(R1 + R2) ----Eq. B2

Thus, the maximum value of R1 can be obtained as a criterion from the equation below. Note, however, that for single mode (when sample-and-hold is disabled), the sampling time for the second and subsequent bits (C2 charging time) must be applied.

$$R1 < \frac{\text{C2 charging time - C2} \times \text{R2}}{\text{Cin + C2}}$$

Appendix 4.11 Notes on Serial Interface

Appendix 4.11.1 Notes on Using CSIO Mode

- **Settings of SIO Transmit/Receive Mode Register and SIO Baud Rate Register**

  The SIO Transmit/Receive Mode Register and SIO Baud Rate Register and the Transmit Control Register's BRG count source select bit must always be set when the serial interface is not operating. If a transmit or receive operation is in progress, wait until the transmit and receive operations are finished and then clear the transmit and receive enable bits before making changes.

- **Settings of BRG (Baud Rate Register)**

  If f(BCLK) is selected with the BRG clock source select bit, use caution when setting the BRG register so that the transfer rate will not exceed 2 Mbps.

- **About successive transmission**

  To transmit data successively, make sure the next transmit data is set in the SIO Transmit Buffer Register before the current data transmission finishes.

- **About reception**

  Because the receive shift clock in CSIO mode is derived by an operation of the transmit circuit, transmit operation must always be executed (by sending dummy data) even when the serial interface is used for only receiving data. In this case, be aware that if the port function is set for the TXD pin (by setting the operation mode register to "1"), dummy data may actually be output from the pin.

- **About successive reception**

  To receive data successively, make sure that data (dummy data) is set in the SIO Transmit Buffer Register before a transmit operation on the transmitter side starts.

- **Transmission/reception using DMA**

  To transmit/receive data in DMA request mode, enable the DMAC to accept transfer requests (by setting the DMA Mode Register) before serial communication starts.

- **About reception finished bit**

  If a receive error (overrun error) occurs, the reception finished bit can only be cleared by clearing the receive enable bit, and cannot be cleared by reading out the receive buffer register.

- **About overrun error**

  If all bits of the next received data have been set in the SIO Receive Shift Register before reading out the SIO Receive Buffer Register (i.e., an overrun error occurred), the received data is not stored in the receive buffer register, with the previous received data retained in it. Although a receive operation continues thereafter, the subsequent received data is not stored in the receive buffer register (receive status bit = "1").

  Before normal receive operation can be restarted, the receive enable bit must be temporarily cleared to "0". And this is the only way that the overrun error flag can be cleared.

- **About DMA transfer request generation during SIO transmission**

  If the transmit buffer register becomes empty (transmit buffer empty flag = "1") while the transmit enable bit remains set to "1" (transmission enabled), an SIO transmit buffer empty DMA transfer request is generated.

- **About DMA transfer request generation during SIO reception**

  If the reception finished bit is set to "1" (receive buffer register full), a reception finished DMA transfer request is generated. Be aware, however, that if an overrun error occurred during reception, this DMA transfer request is not generated.

- **Switching from general-purpose to serial interface pin**

  When switching general-purpose to serial interface pin, SCLKOn pin outputs "H" level (For the case of selecting internal clock and setting CKPOL bit to "0." When setting CKPOL bit to "1", it outputs "L" level.), and TXDn pin outputs undefined value. However, when switching general-purpose to serial interface pin with setting TEN bit of the SIOn transmit control register to "1" (transmit enable), TXDn pin outputs the last bit level of the previously output serial data.

Appendix 4.11.2 Notes on Using UART Mode

- **Settings of SIO Transmit/Receive Mode Register and SIO Baud Rate Register**

  The SIO Transmit/Receive Mode Register and SIO Baud Rate Register and the Transmit Control Register's BRG count source select bit must always be set when the serial interface is not operating. If a transmit or receive operation is in progress, wait until the transmit and receive operations are finished and then clear the transmit and receive enable bits before making changes.

- **Settings of BRG (Baud Rate Register)**

  Writes to the SIO Baud Rate Register take effect in the next cycle after the BRG counter has finished counting. However, if the register is accessed for write while transmission and reception are disabled, the written value takes effect at the same time it is written.

- **Transmission/reception using DMA**

  To transmit/receive data in DMA request mode, enable the DMAC to accept transfer requests (by setting the DMA Mode Register) before serial communication starts.

- **About overrun error**

  If all bits of the next received data have been set in the SIO Receive Shift Register before reading out the SIO Receive Buffer Register (i.e., an overrun error occurred), the received data is not stored in the receive buffer register, with the previous received data retained in it. Once an overrun error occurs, although a receive operation continues, the subsequent received data is not stored in the receive buffer register. Before normal receive operation can be restarted, the receive enable bit must be temporarily cleared. And this is the only way that the overrun error flag can be cleared.

- **Flags showing the status of UART receive operation**

  There are following flags that indicate the status of receive operation during UART mode:

  - SIO Receive Control Register receive status bit
  - SIO Receive Control Register reception finished bit
  - SIO Receive Control Register receive error sum bit
  - SIO Receive Control Register overrun error bit
  - SIO Receive Control Register parity error bit
  - SIO Receive Control Register framing error bit

  The manner in which the reception finished bit and various error flags are cleared differs depending on whether an overrun error occurred, as described below.

  [When an overrun error did not occur]
  Cleared by reading out the lower byte of the receive buffer register or by clearing the receive enable bit.

  [When an overrun error occurred]
  Cleared by only clearing the receive enable bit.

- **Switching from general-purpose to serial interface**

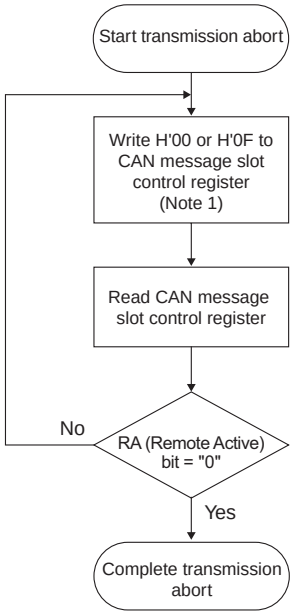
  When switching from general-purpose port to the serial interface pin by the port operation mode register, the terminal TXDn pin outputs "H" level.

Appendix 4.12 Notes on CAN Module

- **Note for cancelation of transmit and receive CAN remote frame**

When aborting remote frame transmission or canceling remote frame receiving, make sure that the RA (Remote Active) bit is cleared to "0" after writing "H'00" or "H'0F" to the CAN Message Slot Control Register.

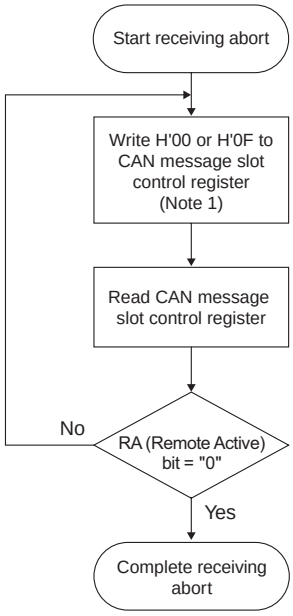
(1) When aborting remote frame transmission



Note 1: H'00 or H'0F can be used.

Appendix Figure 4.12.1 Opertion Flow when Aborting Remote Frame Transmission

(2) When canceling remote frame receiving



Note 1: H'00 or H'0F can be used.

Appendix Figure 4.12.2 Opertion Flow when Canceling Remote Frame Receiving

Appendix 4.13 Notes on RAM Backup Mode

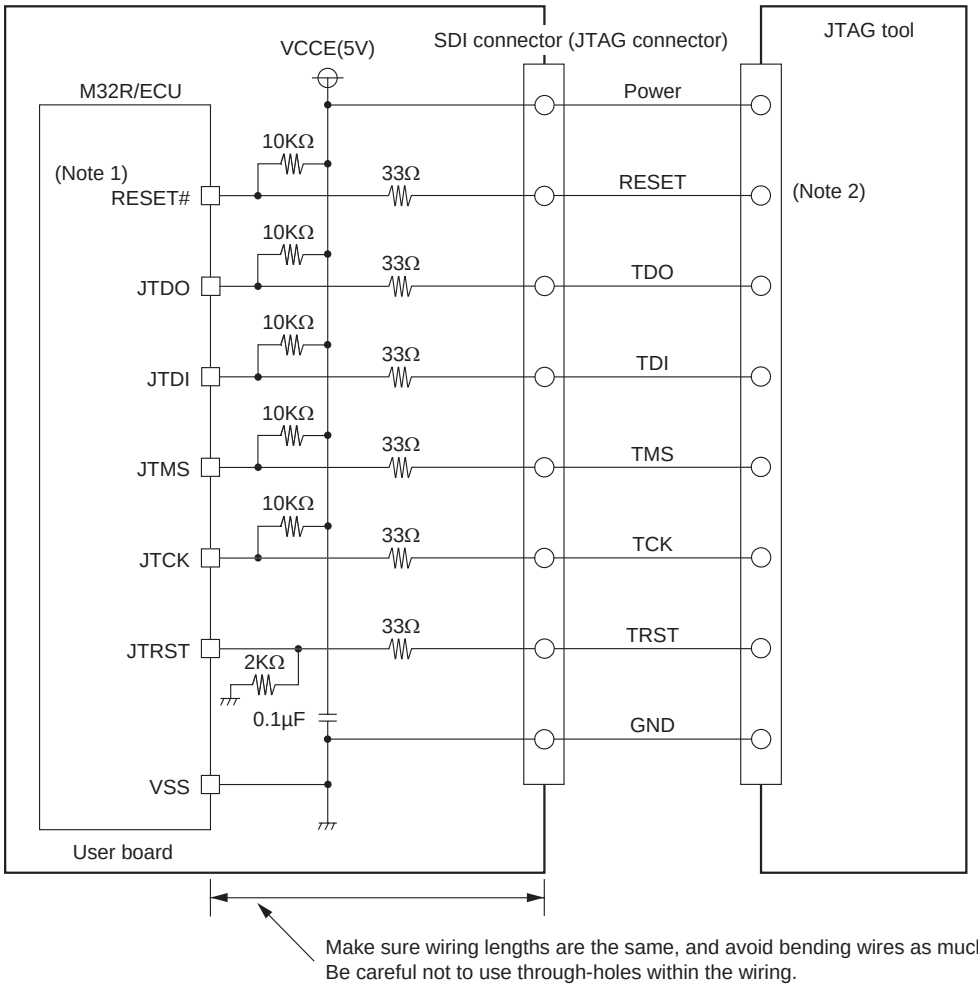
• Precautions to Be Observed at Power-On

When changing port n from input mode to output mode after power-on, pay attention to the following.
If port n is set for output mode while no data is set in the Port n Data Register, the port's initial output level is instable. Therefore, before changing port n for output mode, make sure the Port n Data Register is set to output a "H."
Unless this precaution is followed, port output may go "L" at the same time the port is set for output after the oscillation has stabilized, causing the microcomputer to enter RAM backup mode.

Appendix 4.14 Notes on JTAG

Appendix 4.14.1 Notes on Board Design when Connecting JTAG

To materialize fast and highly reliable communication with JTAG tools, make sure wiring lengths of JTAG pins are matched during board design.

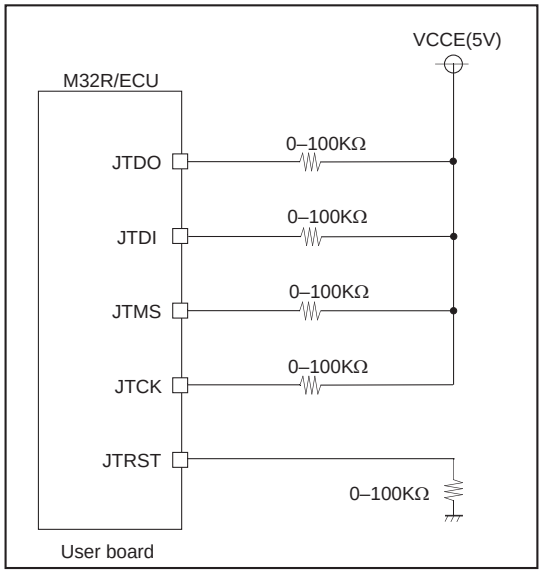


Note 1: The RESET# related circuit and resistance-capacitance values must be determined depending on the user board's system design conditions and the microcomputer's operating conditions.
Note 2: N-channel open-drain output is recommended for the RESET output of JTAG tools. For details, see JTAG tool specifications.
Notes: • Only if the JTRST pin is firmly tied to ground, the JTDO, JTDI, JTMS and JTCLK pins can be processed by either pullup or pulldown.
• Each of these pins must always be processed even when not using JTAG tools.
The same pullup/pulldown resistance values as when using JTAG tools may be used.

Appendix Figure 4.14.1 Notes on Board Design when Connecting JTAG Tools

Appendix 4.14.2 Processing Pins when Not Using JTAG

The following shows how the pins on the chip should be processed when not using JTAG tools.



Note: • Only if the JTRST pin is firmly tied to ground, the JTDO, JTDI, JTMS and JTCLK pins can be processed by either pullup or pulldown.

Appendix Figure 4.14.2 Processing Pins when Not Using JTAG

Appendix 4.15 Notes on Noise

The following describes precautions to be taken about noise and corrective measures against noise. The corrective measures described here are theoretically effective for noise, but require that the application system incorporating those measures be fully evaluated before it can actually be put to use.

Appendix 4.15.1 Reduction of Wiring Length

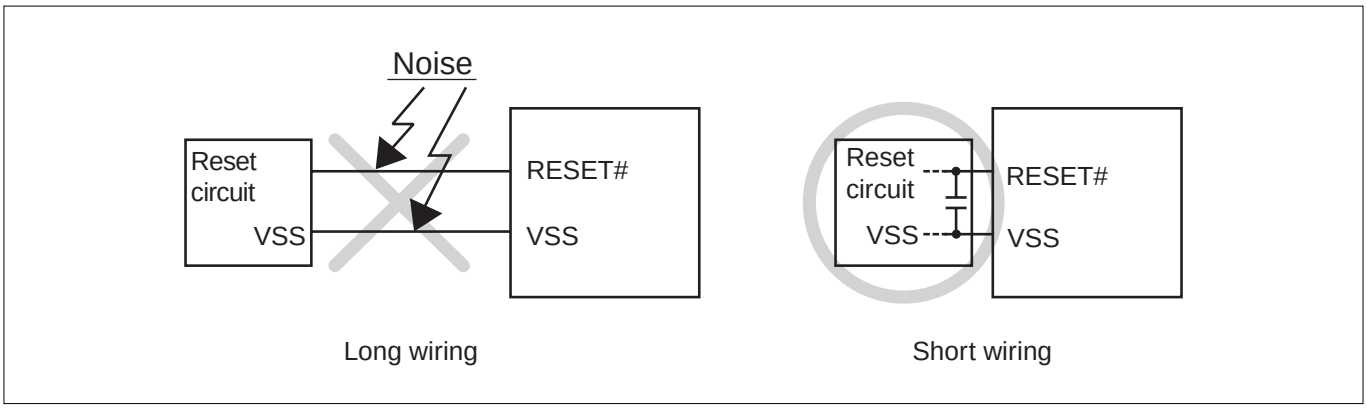
Wiring on the board may serve as an antenna to draw noise into the microcomputer. Shorter the total wiring length, the smaller the possibility of drawing noise into the microcomputer.

(1) Wiring of the RESET# pin

Reduce the length of wiring connecting to the RESET# pin. Especially when connecting a capacitor between the RESET# and VSS pins, make sure it is wired to each pin in the shortest distance possible (within 20 mm).

<Reasons>

Reset is a function to initialize the internal logic of the microcomputer. The width of a pulse applied to the RESET# pin is important and is therefore specified as part of timing requirements. If a pulse in width shorter than the specified duration (i.e., noise) is applied to the RESET# pin, the microcomputer will not be reset for a sufficient duration of time and come out of reset before its internal logic is fully initialized, causing the program to malfunction.



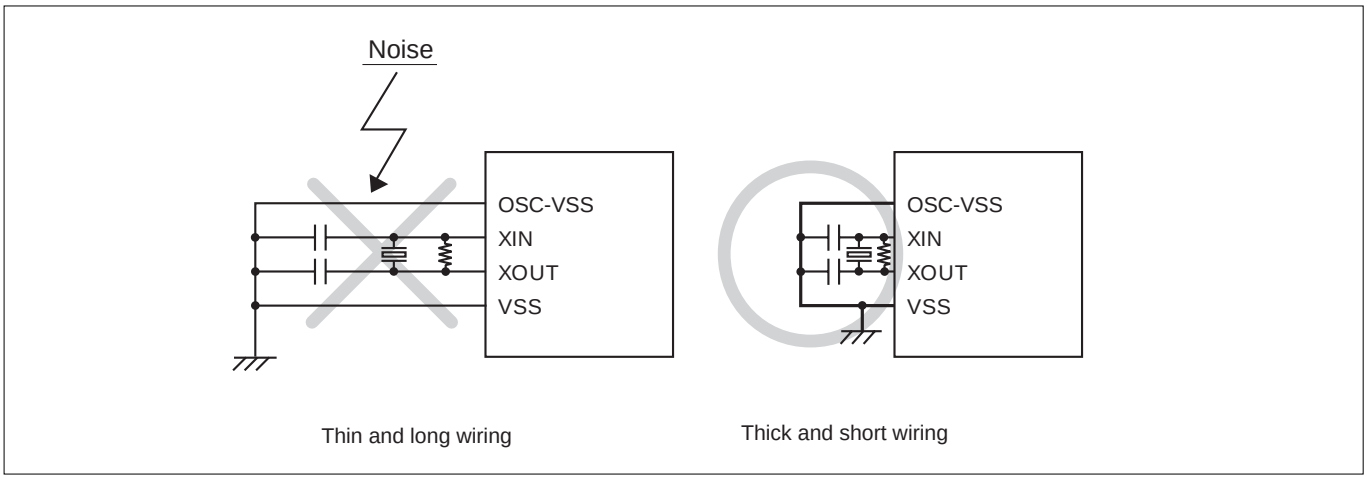
Appendix Figure 4.15.1 Example Wiring of the RESET# Pin

(2) Wiring of clock input/output pins

Use as much thick and short wiring as possible for connections to the clock input/output pins.
When connecting a capacitor to the oscillator, make sure its grounding lead wire and the OSC-VSS pin on the microcomputer are connected in the shortest distance possible (within 20 mm).
Also, make sure the VSS pattern used for clock oscillation is a large ground plane and is connected to GND.

<Reasons>

The microcomputer operates synchronously with the clock generated by an oscillator circuit. Inclusion of noise on the clock input/output pins causes the clock waveform to become distorted, which may result in the microcomputer operating erratically or getting out of control. Furthermore, if a noise-induced potential difference exists between the microcomputer's VSS level and that of the oscillator, the clock fed into the microcomputer may not be an exact clock.



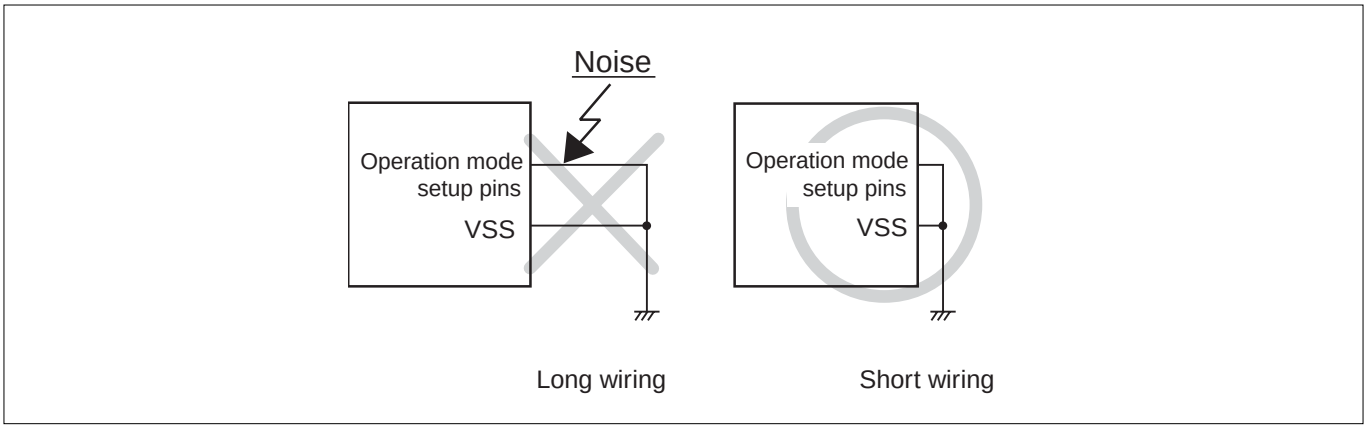
Appendix Figure 4.15.2 Example Wiring of Clock Input/Output Pins

(3) Wiring of the operation mode setup pins

When connecting the operation mode setup pins and the VCC or VSS pin, make sure they are wired in the shortest distance possible.

<Reasons>

The levels of the operation mode setup pins affect the microcomputer's operation mode. When connecting the operation mode setup pins and the VCC or VSS pin, be careful that no noise-induced potential difference will exist between the operation mode setup pins and the VCC or VSS pin. This is because the presence of such a potential difference makes operation mode instable, which may result in the microcomputer operating erratically or getting out of control.

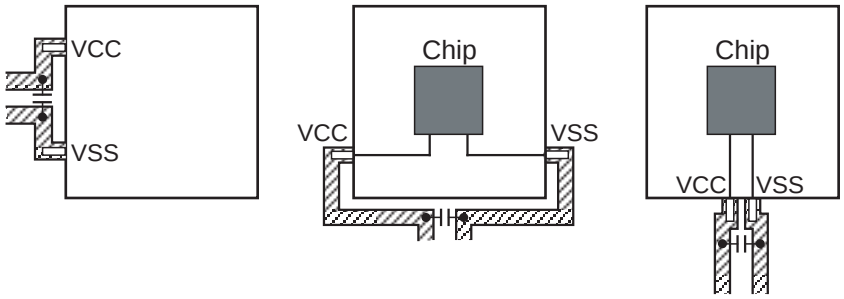


Appendix Figure 4.15.3 Example Wiring of the MOD0 and MOD1 Pins

Appendix 4.15.2 Inserting a Bypass Capacitor between VSS and VCC Lines

Insert a bypass capacitor of about 0.1 µF between the VSS and VCC lines. At this time, make sure the requirements described below are met.

- The wiring length between the VSS pin and bypass capacitor and that between the VCC pin and bypass capacitor are the same.
- The wiring length between the VSS pin and bypass capacitor and that between the VCC pin and bypass capacitor are the shortest distance possible.
- The VSS and VCC lines have a greater wiring width than that of all other signal lines.



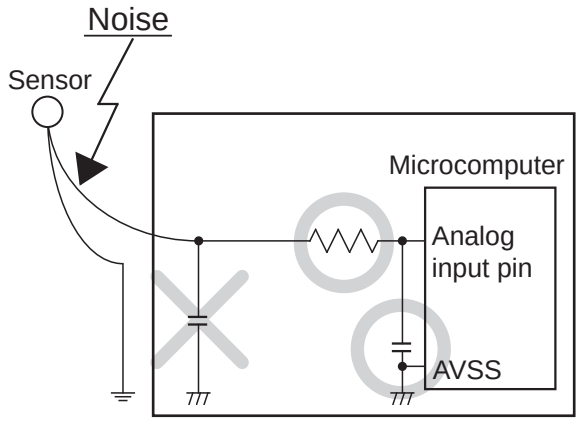
Appendix Figure 4.15.4 Example of a Bypass Capacitor Inserted between VSS and VCC Lines

Appendix 4.15.3 Processing Analog Input Pin Wiring

Insert a resistor of about 100 to 500Ω in series to the analog signal line connecting to the analog input pin at a position as close to the microcomputer as possible. Also, insert a capacitor of about 100 pF between the analog input pin and AVSS pin at a position as close to the AVSS pin as possible.

<Reasons>

The signal fed into the analog input pin (e.g., A/D converter input pin) normally is an output signal from a sensor. In many cases, a sensor to detect changes of event is located apart from the board on which the microcomputer is mounted, so that wiring to the analog input pin is inevitably long. Because a long wiring serves as an antenna which draws noise into the microcomputer, the signal fed into the analog input pin tends to be noise-ridden. Furthermore, if the capacitor connected between the analog input pin and AVSS pin is grounded at a position apart from the AVSS pin, noise riding on the ground line may penetrate into the microcomputer via the capacitor.



Appendix Figure 4.15.5 Example of a Resistor and Capacitor Inserted for the Analog Signal Line

Appendix 4.15.4 Consideration about the Oscillator

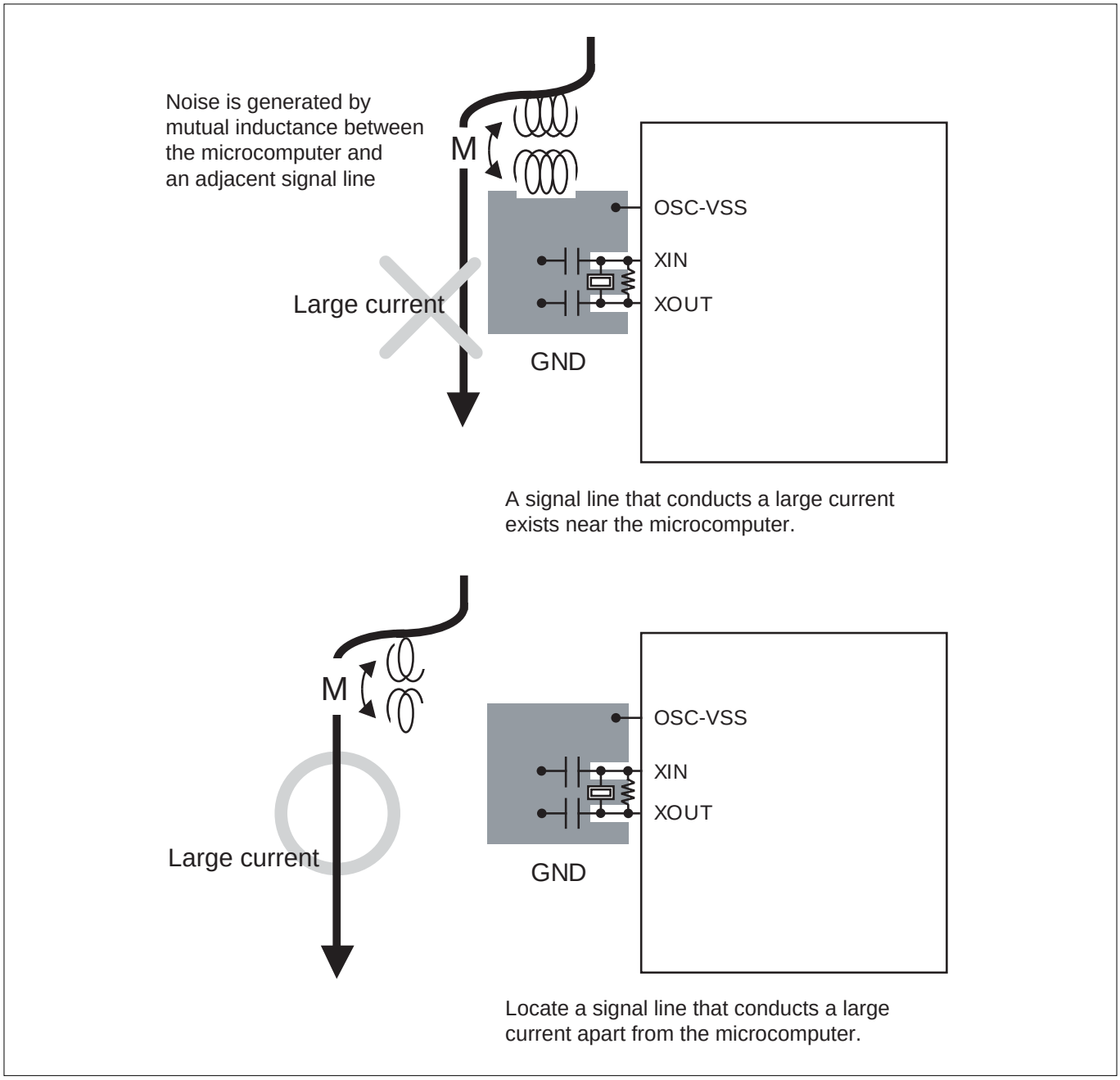
The oscillator that generates the fundamental clock for microcomputer operation requires consideration to make it unsusceptible to influences from other signals.

(1) Avoidance from large-current signal lines

Signal lines that conduct a large current exceeding the range of current values that the microcomputer can handle must be routed as far away from the microcomputer (especially the oscillator) as possible. Also, make sure the circuit is protected with a GND pattern.

<Reasons>

Systems using a microcomputer have signal lines to control a motor, LED or thermal head, for example. When a large current flows in these signal lines, it generates noise due to mutual inductance (M).



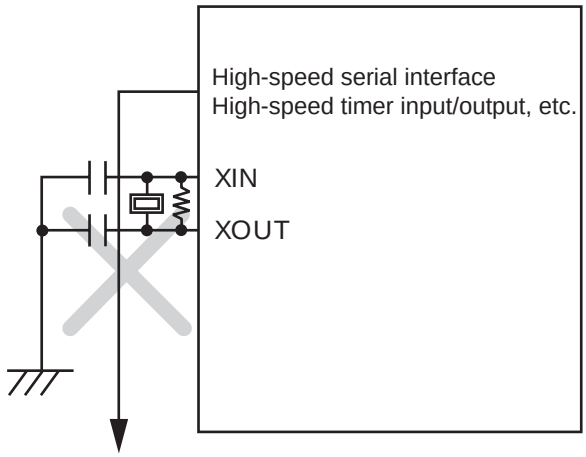
Appendix Figure 4.15.6 Example Wiring of a Large-Current Signal Line

(2) Avoiding effects of rapidly level-changing signal lines

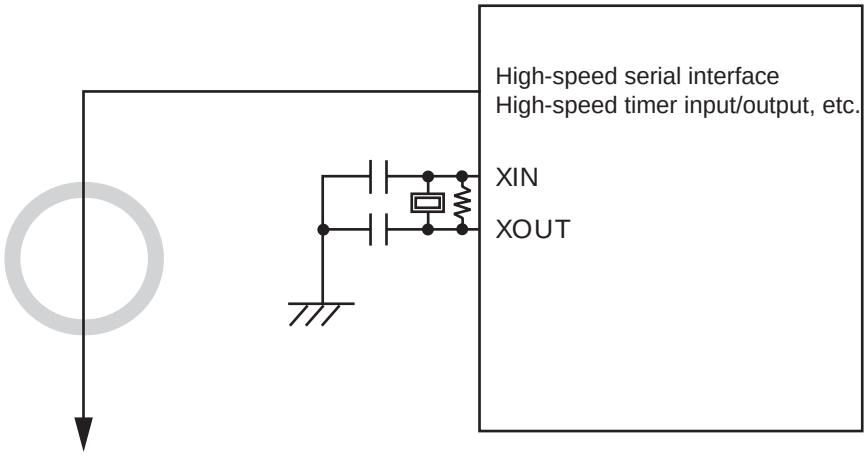
Locate signal lines whose levels change rapidly as far away from the oscillator as possible. Also, make sure the rapidly level-changing signal lines will not intersect the clock-related signal lines and other noise-sensitive signal lines.

<Reasons>

Rapidly level-changing signal lines tend to affect other signal lines as their voltage level frequently rises and falls. Especially if these signal lines intersect the clock-related signal lines, they will cause the clock waveform to become distorted, which may result in the microcomputer operating erratically or getting out of control.



Signal line intersecting the clock-related and other signal lines



Locate the signal line away from the clock-related and other signal lines

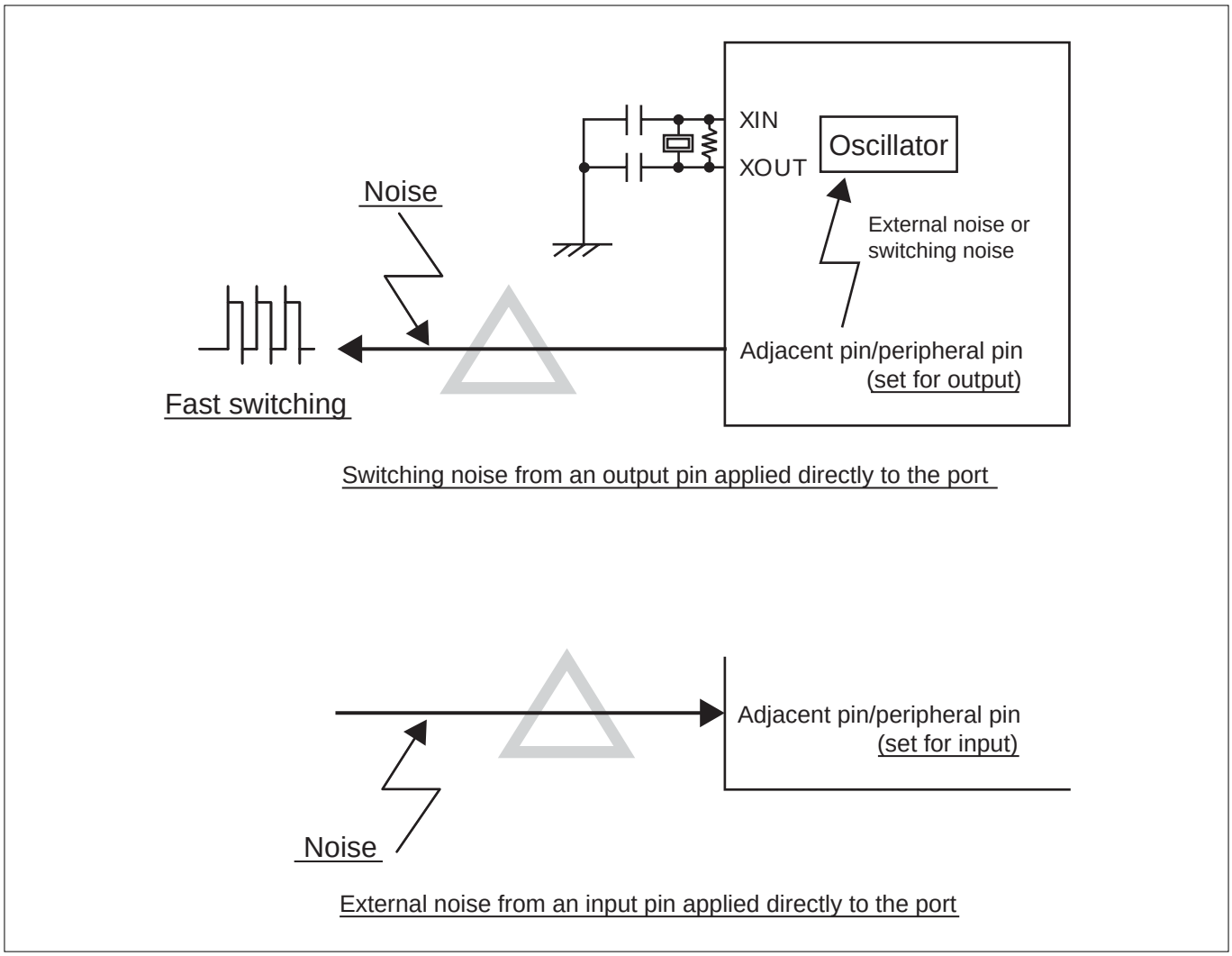
Appendix Figure 4.15.7 Example Wiring of a Rapidly Level-Changing Signal Line

(3) Protection against signal lines that are the source of strong noise

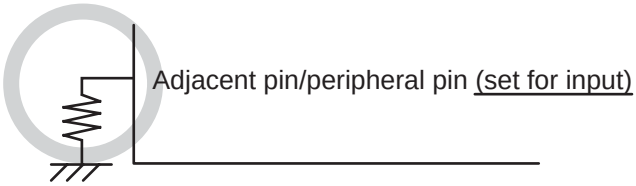
Do not use any pin that will probably be subject to strong noise for an adjacent port near the oscillator. If the pin can be left unused, set it for input and connect to GND via a resistor, or fix it to output and leave open. If the pin needs to be used, it is recommended that it be used for input-only.
For protection against a still stronger noise source, set the adjacent port for input and connect to GND via a resistor, and use those that belong to the same port group as much for input-only as possible. If greater stability is required, do not use those that belong to the same port group and set them for input and connect to GND via a resistor. If they need to be used, insert a limiting resistor for protection against noise.

<Reasons>

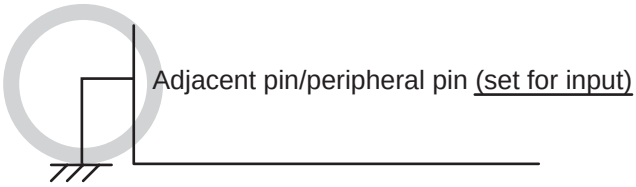
If the ports or pins adjacent to the oscillator operate at high speed or are exposed to strong noise from an external source, noise may affect the oscillator circuit, causing its oscillation to become instable.



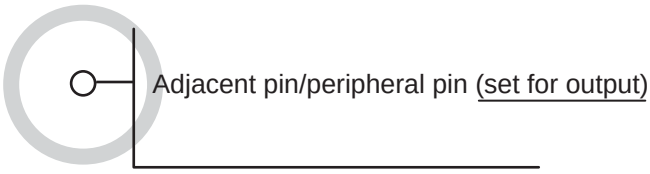
Appendix Figure 4.15.8 Example Processing of a Noise-Laden Pin



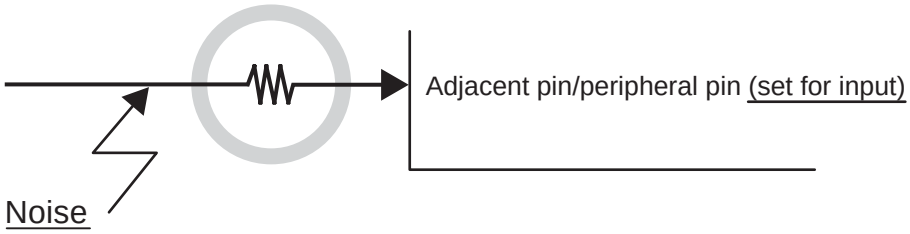
Method for limiting the effect of noise in input mode



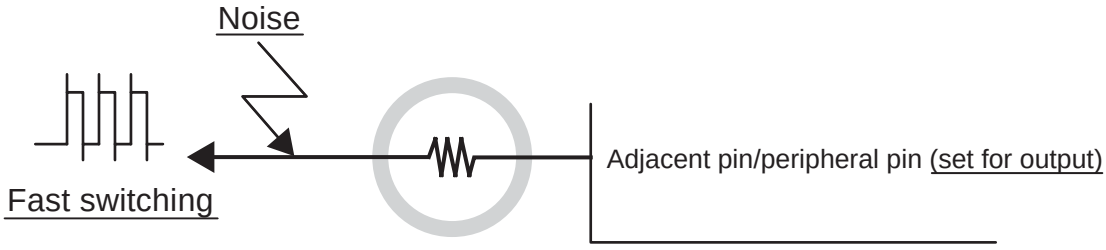
Method for limiting the effect of noise in input mode



Method for limiting the effect of noise in output mode



Method for limiting noise with a resistor



Method for limiting switching noise with a resistor

Appendix Figure 4.15.9 Example Processing of Pins Adjacent to the Oscillator

Appendix 4.15.5 Processing Input/Output Ports

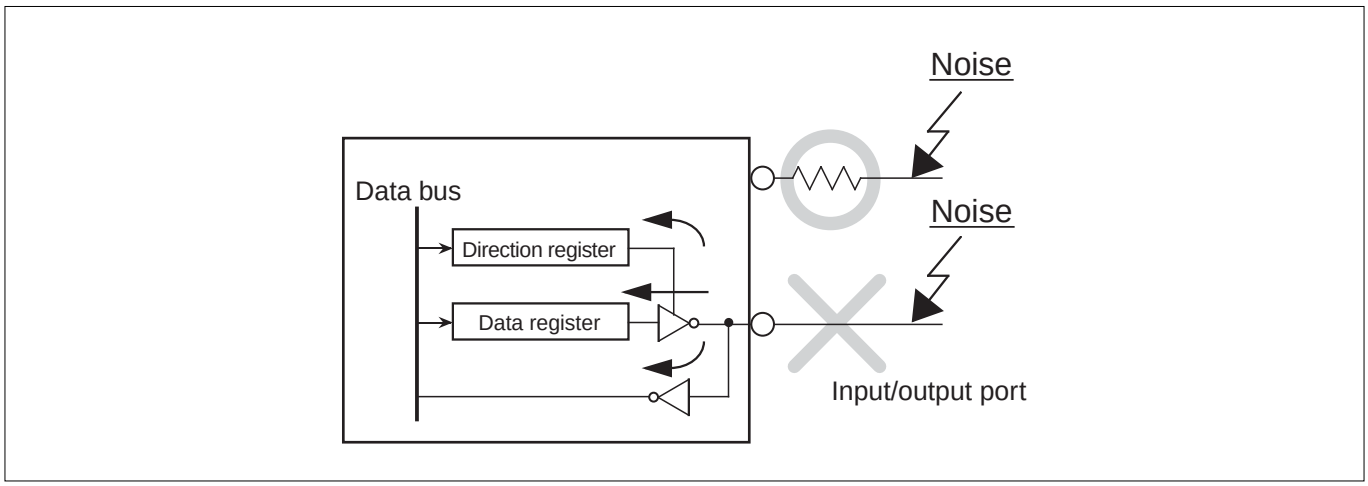
For input/output ports, take the appropriate measures in both hardware and software following the procedure described below.

Hardware measures

- Insert resistors of 100Ω or more in series to the input/output ports.

Software measures

- For input ports, read out data in a program two or more times to verify that the levels coincide.
- For output ports, rewrite the data register at certain intervals because there is a possibility of the output data being inverted by noise.
- Rewrite the direction register at certain intervals.



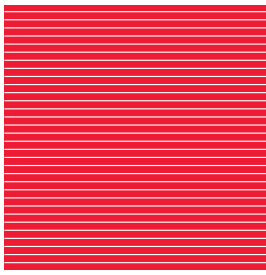
Appendix Figure 4.15.10 Example Processing of Input/Output Ports

32176 Group Hardware Manual

Publication Data : Rev. 1.01 Oct 31, 2003
Rev. 1.10 Jun 20, 2006

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

32176 Group
Hardware Manual



RENESAS

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan