To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

To all our customers

Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

User's Manual

32

32172
32173 Group

User's Manual

Renesas 32-bit RISC Single-chip Microcomputers
M32R Family M32R/ECU Series

Renesas Electronics
www.renesas.com

Rev.1.0 2001.10

Keep safety first in your circuit designs!

- Mitsubishi Electric Corporation 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

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor 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 Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation 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.
- 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 Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
  The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
  Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (http://www.mitsubishichips.com).
- 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. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation 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 Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor 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.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- 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.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

Rev.	Date	Description	
		Page	Summary
1.0	10/5/2001	-	First edition issued

How to read internal I/O register tables

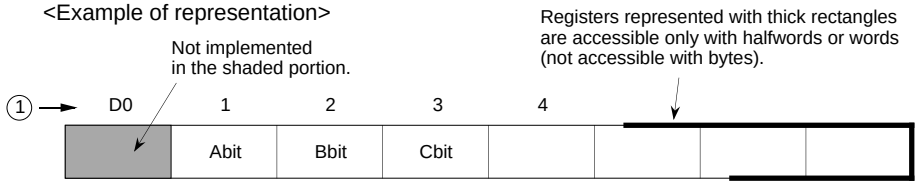
① Bit Numbers: Each register is connected with an internal bus of 16-bit wide, so the bit numbers of the registers located at even addresses are D0-D7, and those at odd addresses are D8-D15.

② State of Register at Reset: Represents the initial state of each register immediately after reset with hexadecimal numbers (undefined bits after reset are indicated each in column ③.)

③ At read:
- ○ ... read enabled
- ? ... read disabled (read value invalid)
- 0 ... Read always as 0
- 1 ... Read always as 1

④ At write:
- ○ : Write enabled
- Δ : Write enable conditionally (include some conditions at write)
- \- : Write disabled (Written value invalid)

<Example of representation>



② → <at reset: H'04>

D	Bit name	Function	R	W
0	Not assigned.		0	—
1	Abit	0: -----	○	○
	(.....)	1: -----		
2	Bbit	0: -----	○	○
	(.....)	1: -----		
3	Cbit	0: -----	○	○
	(.....)	1: -----		

③ ④

CONTENTS

CHAPTER 1 OVERVIEW

CHAPTER 2 CPU

CHAPTER 3 ADDRESS SPACE

CHAPTER 4 EIT

CHAPTER 5 INTERRUPT CONTROLLER (ICU)

CHAPTER 6 INTERNAL MEMORY

CHAPTER 7 RESET

CHAPTER 8 INPUT/OUTPUT PORTS AND PIN FUNCTIONS

CHAPTER 9 DMAC

CHAPTER 10 INPUT/OUTPUT TIMERS

CHAPTER 11 A-D CONVERTERS

CHAPTER 12 SERIAL I/O

CHAPTER 13 CAN MODULES

CHAPTER 14 REAL-TIME DEBUGGER (RTD)

CHAPTER 15 PD MODULE

CHAPTER 16 D-A CONVERTERS

CHAPTER 17 EXTERNAL BUS INTERFACE

CHAPTER 18 WAIT CONTROLLER

CHAPTER 19 RAM BACKUP MODE

CHAPTER 20 OSCILLATION CIRCUIT

CHAPTER 21 JTAG

CHAPTER 22 POWER-UP/POWER-SHUTDOWN SEQUENCE

CHAPTER 23 ELECTRICAL CHARACTERISTICS

CHAPTER 24 STANDARD CHARACTERISTICS

APPENDIX 1 MECHANICAL SPECIFICATIONS

APPENDIX 2 INSTRUCTION PROCESSING TIME

APPENDIX 3 PRECAUTIONS ABOUT NOISE

*** This is a blank page ***

CHAPTER 1
OVERVIEW

1.1 Overview

1.1.1 M32R Family CPU Core

(1) Uses the RISC architecture

- The 32172/32173 are 32-bit, RISC single-chip microcomputers built around the M32R family CPU core (hereafter referred to as the "M32R") and incorporating flash memory, RAM, and various other peripheral functions... all integrated into a single chip.
- The M32R uses the RISC architecture. Memory accesses are performed using Load and Store instructions, and various arithmetic operations are executed using register-to-register operating instructions. The M32R internally has sixteen 32-bit general-purpose registers and a total of 83 discrete instructions.
- In addition to Load and Store instructions, the M32R supports compound instructions such as Load & Address Update and Store & Address Update. These instructions are useful for speeding up data transfers.

(2) 5-stage pipelined processing

- The M32R uses 5-stage pipelined instruction processing consisting of Instruction Fetch, Decode, Execute, Memory Access, and Write Back. Not just load and store instructions or register-to-register operation instructions, compound instructions such as Load & Address Update and Store & Address Update also are executed in one cycle.
- Instructions are entered into the execution stage in the order they are fetched, but this does not always mean that the first instruction entered is executed first. If the execution of a load or store instruction entered earlier is delayed by one or more wait cycles inserted in memory access, a register-to-register operation instruction entered later may be executed before said load or store instruction. By using "out-of-order-completion" like this, the M32R controls instruction execution without wasting clock cycles.

(3) Compact instruction code

- Instructions of the M32R come in either a 16-bit instruction or a 32-bit instruction format. Use of the 16-bit instruction format especially helps to reduce the code size of a program.
- Some 32-bit instructions can branch directly to a location 32 Mbytes forward or backward from the currently executed instruction address. The availability of such instructions makes programming easier than for architectures with segmented address spaces.

1.1.2 Built-in Multiply-Accumulate Operation Function

(1) Built in high-speed multiplier

- The M32R incorporates a 32-bit x 16-bit high-speed multiplier/accumulator which allows the processor to execute a 32-bit x 32-bit integer multiplication instruction in three cycles (one cycle is 25 ns when CPU memory clock = 40 MHz).

(2) Supports Multiply-Accumulate operation instructions comparable to DSP

- The M32R supports the following four modes of Multiply-Accumulate operation instructions (or multiplication instructions) based on a 56-bit accumulator:
  - ① 16 high-order register bits x 16 high-order register bits
  - ② 16 low-order register bits x 16 low-order register bits
  - ③ All 32 register bits x 16 high-order register bits
  - ④ All 32 register bits x 16 low-order register bits

- The M32R has instructions to round off the value stored in the accumulator to 16 or 32 bits, as well as instructions to shift the accumulator value to adjust digits and store the digit-adjusted value in a register. These instructions also can be executed in one cycle, so that when combined with high-speed data transfer instructions such as Load & Address Update and Store & Address Update, they enable the M32R to exhibit high data processing capability comparable to that of DSP.

1.1.3 Built-in Flash Memory and RAM

- The 32172/32173 contains flash memory and RAM that can be accessed with no wait states, making it possible to build a high-speed embedded system.
- The internal flash memory allows for on-board programming (you can write to it while being mounted on the printed circuit board). Use of flash memory means the chip engineered at the development phase can be used directly in mass-production, so that you can smoothly migrate from prototype to mass-production without changing the printed circuit board.
- The internal flash memory can be rewritten 100 times.
- The internal flash memory has a virtual-flash emulation function, allowing the internal RAM tobe artificially mapped into part of the internal flash memory. This function, when combined with the internal Real-Time Debugger (RTD), facilitates data tuning on ROM tables.
- The internal RAM can be accessed for read or rewrite from an external device independently of the M32R by using RTD (real-time debugger). It is communicated with external devices by RTD's exclusive clock-synchronized serial I/O.

1.1.4 Built-in Clock Multiplier Circuit

- The 32172/32173 internally multiplies the frequency of the input clock signal by 4 (or by 2 for the internal peripheral clock). When the input clock frequency is 10.0 MHz, the CPU clock frequency is 40 MHz and that of the internal peripheral clock is 20 MHz.

1.1.5 Built-in Powerful Peripheral Functions

(1) Built-in input/output timers

- The timers used in the 32172/32173 consist of the following 26 channels of timers. (When not using the PDC module as a sensor interface circuit, eight more channels of input timers are available.)
  - ① 16-bit output related timers x 16 channels
  - ② 16-bit input related timers x 6 channels
  - ③ 32-bit input related timers x 4 channels

Each timer has multiple modes to choose from, depending on the purpose of use.

(2) Built-in 10-channel DMA

- The microcomputer contains 10 channels of DMA, allowing for data transfer between internal peripheral I/Os and between internal RAM and internal peripheral I/O. DMA transfer requests can be issued from the user-created software, as well as can be triggered by a signal generated by the internal peripheral I/O (A-D converter, input/output timer, or serial I/O).
- The microcomputer also supports cascaded operation between DMA channels (starting DMA transfer on a channel at the end of transfer on another channel). This makes advanced transfer processing possible without causing any additional CPU load.

(3) Built-in two blocks of A-D converters

- The microcomputer contains an 8-channel A-D converter and a 4-channel A-D converter, both capable of 10-bit resolution.
- In addition to ordinary A-D conversion, the converters support comparator mode in which a set value and the A-D converted value are compared to determine which is larger or smaller than the other.
- When A-D conversion is finished, the converters can generate a DMA transfer request, as well as an interrupt.

(4) High-speed serial I/O

- The microcomputer contains eight channels of serial I/Os which can be set for clock-synchronized serial I/O or UART.
- The transfer rate in clock-synchronized serial I/O mode is a high 2 Mbits per second, allowing for fast data transfer.
- The serial I/O has the function to generate a DMA transfer request when data reception is

completed or the transmit register becomes empty.

(5) Built-in Real-time Debugger (RTD)

- The Real-time Debugger (RTD) provides a function for accessing directly from the outside to the M32R/E's internal RAM. It uses a dedicated clock-synchronized serial I/O to communicate with external devices.
- Use of the RTD allows the contents of the internal RAM to be read out or its data to be rewritten from the outside, independently of the M32R.
- An RTD interrupt can be generated to indicate that RTD-based data transmission or reception is completed.

(6) 8-level interrupt controller

- The Interrupt Controller controls interrupt requests from internal peripheral I/Os by using eight priority levels (including interrupt-disabled state) which are assigned to each interrupt source. It also handles external interrupt requests generated upon detection of power outage or generated by the watchdog timer as System Break Interrupt (SBI).

(7) Three operation modes

- The M32R/E supports three operation modes: single-chip, external extended, and processor modes. The M32R/E's address space and external pin functions are switched over according to each mode. Modes are selected using the MOD0 and MOD1 pins.

(8) Wait controller

- The Wait Controller supports access to external devices. In other than single-chip mode, up to 4 Mbytes of space is available for an external extended area.

1.1.6 Built-in Full-CAN Function

- The microcomputer contains two CAN modules compliant with CAN Specification V2.0B active, each of which has 16-channel message slots.

1.1.7 Two Built-in D-A Converters

- The microcomputer contains two blocks of 8-bit resolution D-A converters.

- In addition to ordinary D-A conversion, these converters support the function to successively output any data. Also, the converters have a 256-byte output buffer (available for only the D-A0 converter).

1.1.8 Built-in Timer/Arithmetic Circuits for PD (Phase Digital) Sensors

- The microcomputer contains two blocks of timer/arithmetic circuits that operate along with PD (Phase Digital) sensors.
- With various arithmetic circuits needed for position predictive operations incorporated, and the timers interlocked with the D-A converters, fast data processing is possible.
- When not using the PD circuit, the PD sensor-handling timers can be used as ordinary input measurement timers or input event counters.

1.1.9 Built-in Debug Function

- The 32172/32173 supports the JTAG interface. Using this JTAG interface, the microcomputer can perform boundary scan test.

1.2 Block Diagram

Figure 1.2.1 shows a block diagram of the 32172/32173. The features of each block are outlined in Tables 1.2.1 to 1.2.3.

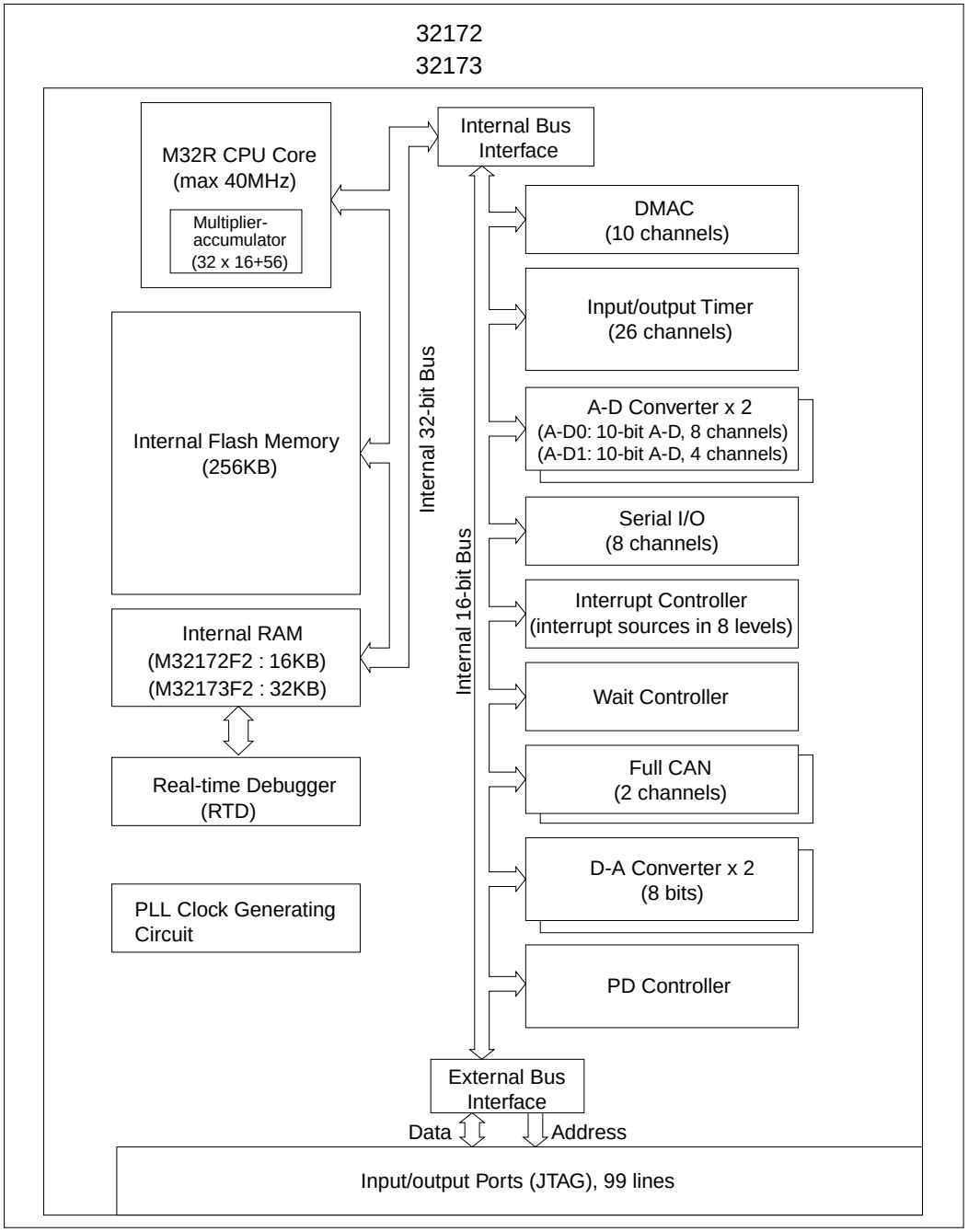


Figure 1.2.1 Block Diagram

Table 1.2.1 Features of the M32R Family CPU Core

Functional Block	Features
M32R family CPU core	• Bus specifications Basic bus cycle: 25 ns (when CPU clock = 40 MHz) Logical address space: 4 Gbytes linear External extended area: Maximum 4 Mbytes External data bus: 16 bits • Implementation: Five-stage pipelined processing • CPU core internally configured in 32 bits • Register configuration General-purpose register: 32 bits x 16 Control register: 32 bits x 5 • Instruction set 16-bit/32-bit instruction formats 83 discrete instructions/6 addressing modes • Built-in multiplier-accumulator (32 x 16 + 56)

Table 1.2.2 Features of the Internal Memory

Functional Block	Features
RAM	• Capacity M32172F2: 16 Kbytes M32173F2: 32 Kbytes • No-wait access • By using the RTD (Real-time Debugger), the internal RAM can be accessed for data read or rewrite from the outside, independently of the M32R.
Flash memory	• Capacity: 256 Kbytes • No-wait access • Durability: Can be rewritten 100 times

Table 1.2.3 Features of Internal Peripheral I/Os

Functional Block	Features
DMA	• 10-channel DMA • Supports data transfer between internal peripheral I/Os, between internal RAMs, and between internal peripheral I/O and internal RAM • Capable of fast DMA transfer when used in combination with internal peripheral I/O • Capable of cascaded operation between DMA channels (starting DMA transfer on a channel at the end of transfer on another)
Timer	• 26-channel multifunction timers • 16 channels of 16-bit output related timers, 6 channels of 16-bit input related timers, and 4 channels of 32-bit input related timers • Flexible timer configuration is possible by interconnecting each timer channel
A-D converter	• One 8-channel A-D converter and one 4-channel A-D converter, both capable of 10-bit resolution • Supports comparator mode • Can generate an interrupt or start DMA transfer at completion of A-D conversion • Can monitor pin levels on a total of 20 channels (with reduced accuracy, however)
Serial I/O	• 8-channel serial I/O • Can be set for clock-synchronized serial I/O or UART • Capable of fast data transfer at 2 Mbits/second during clock-synchronized mode or 156 Kbits/second during UART mode
Real-time debugger	• Can rewrite/monitor the internal RAM by command input from the outside, independently of the CPU • Comes with dedicated clock-synchronized serial port
Interrupt controller	• Controls interrupt requests from internal peripheral I/Os • Eight priority levels including interrupt-disabled state
Wait controller	• Controls wait state when accessing external extended area • Inserts 1 to 4 wait cycles by software setting + extends wait period by external $\overline{\text{WAIT}}$ signal input
Clock PLL	• Multiply-by-4 clock generating circuit • CPU clock of maximum 40 MHz (CPU, internal ROM, and internal RAM access) • Internal peripheral clock of maximum 20 MHz (peripheral module access) • Maximum external input clock frequency of 10.0 MHz
D-A converter	• Two channels of 8-bit resolution D-A converters D-A0 converter: D-A output, successive data output function, 256-byte output buffer available D-A1 converter: D-A output only
PD Controller	• Two blocks of PD sensor-accommodating timers and various arithmetic circuits for position predictive operation • 16-bit input measurement timer: 4 channels; 16-bit input related timer: 4 channels • When not using the PD circuit, the above timers can be used as input measurement timers or input event counters
CAN	• Two blocks of CAN modules, each with 16-channel message slots
JTAG	• Boundary scan function, Mitsubishi original SDI debug function included

1.3 Pin Functions

Figure 1.3.1 shows a pin function diagram of the 32172/32173. Table 1.3.1 provides a description of pin functions.

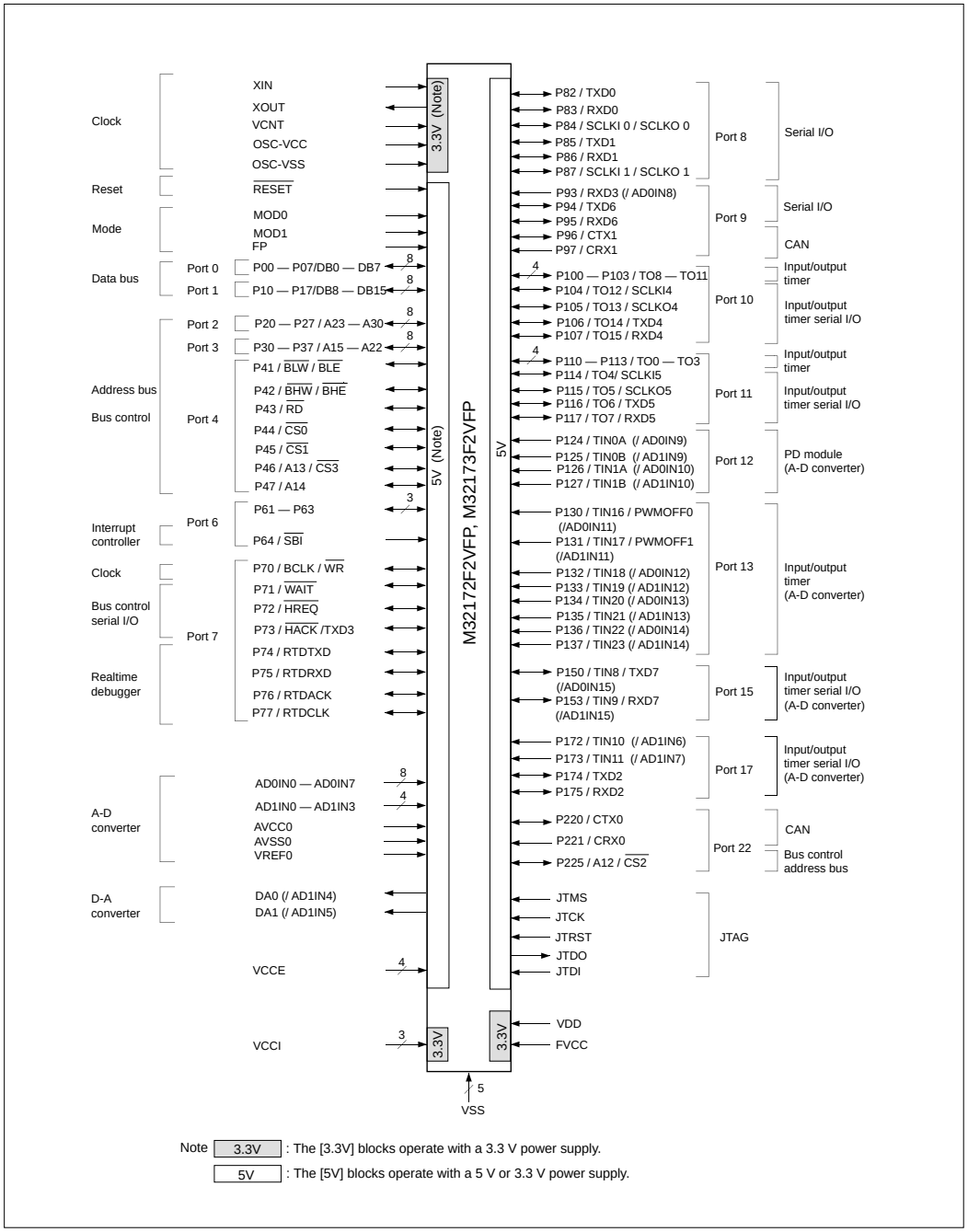


Figure 1.3.1 Pin Function Diagram of 144LQFP Package

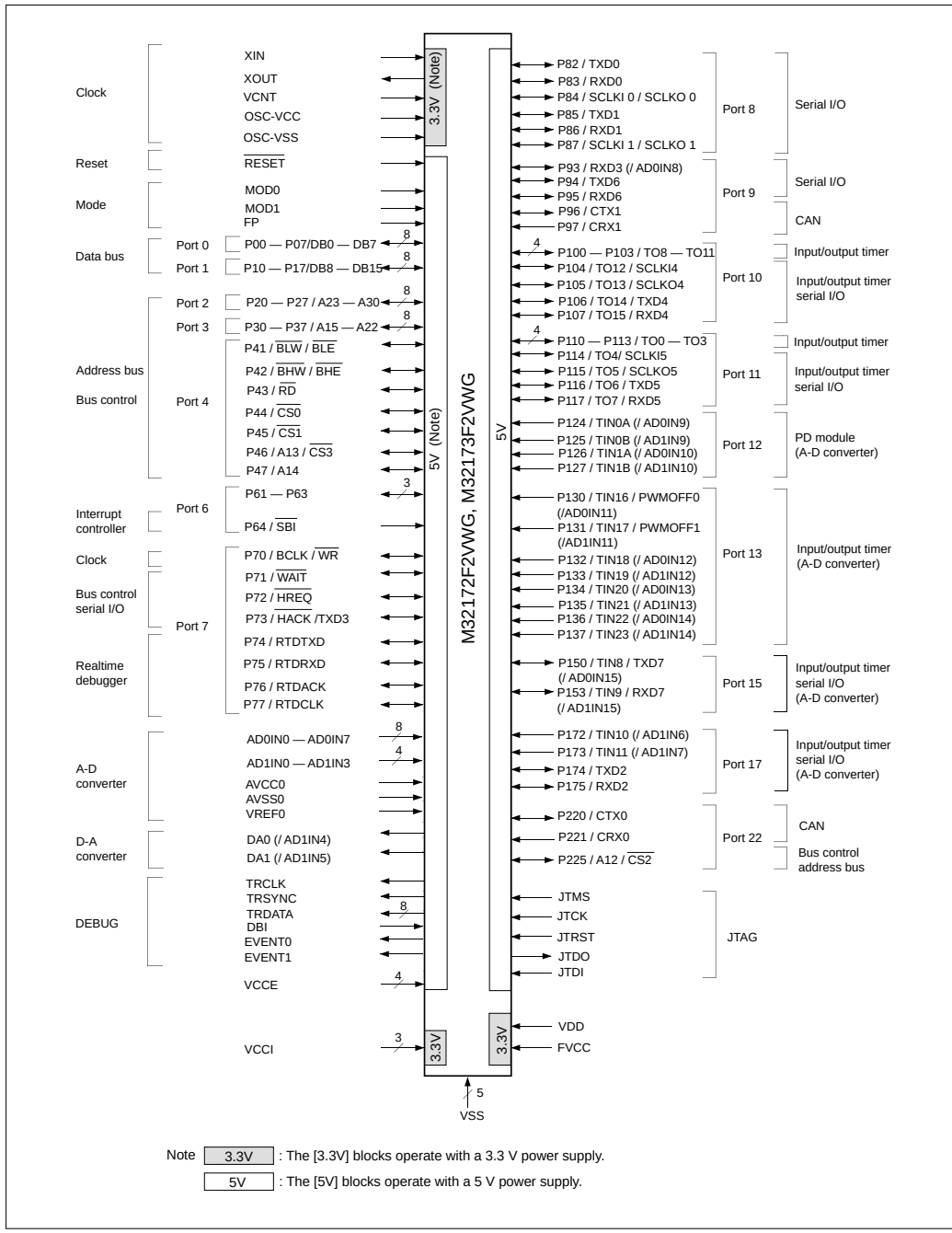


Figure 1.3.2 Pin Function Diagram of 175FBGA Package

Table 1.3.1 Description of Pin Functions (1/6)

Classification	Pin Name	Description	Type	Function
Power supply	VCCE	Power supply	—	Supplies power to external I/O ports (5 V).
	VCCI	Power supply	—	Supplies power to the internal logic (3.3 V).
	VDD	RAM power supply	—	Power supply pin for internal RAM backup (3.3 V).
	FVCC	Flash power supply	—	Power supply pin for the internal flash memory (3.3 V).
	VSS	Ground	—	Connect all VSS to ground (GND).
Clock	XIN, XOUT	Clock	Input Output	Clock input/output pin. With a PLL-based frequency multiplier circuit included, enter a clock with 1/4 the operating frequency (XIN input = 10.0 MHz for the CPU clock of 40 MHz). Use this clock for external synchronized design.
	BCLK/$\overline{WR}$	System clock	Output	When this signal is System Clock (BCLK), it outputs a clock whose is twice that of extemal inpout clock. (BCLK output = 20 MHz when CPU clock operates at 40 MHz). When this signal is Write ($\overline{WR}$), during extemal write access it indicates the valid data on the data bus to transfer. input clock (BCLK output = 20 MHz when the external input clock is 10 MHz).
	OSC-VCC	Power supply	—	Power supply for the PLL circuit. Connect OSC-VCC to the power supply (3.3 V).
	OSC-VSS	Ground	—	Connect OSC-VSS to ground.
	VCNT	PLL control	Input	PLL circuit control pin. Connect a resistor and capacitor to this pin. (For details about an external circuit, refer to Section 20.1.1, "Example of an Oscillator Circuit."
Reset	$\overline{RESET}$	Reset	Input	Resets the internal circuits.
Mode	MOD0 MOD1	Mode	Input	Sets operation mode.
Address bus	A12-A30	Address bus	Output	To allow four blocks of up to 1 MB memory space to be connected external to the chip, 19 address lines (A12-A30) are provided. A31 is not output to the outside.

FP	MOD0	MOD1	Mode
X	0	0	Single-chip mode
X	0	1	External extended mode
0	1	0	Processor mode
1	1	0	Boot mode
X	1	1	(Reserved)

Note: For details about boot mode, refer to Chapter 6, "Internal Memory."

Table 1.3.1 Description of Pin Functions (2/6)

Classification	Pin Name	Description	Type	Function
Data bus	DB0-DB15	Data bus	Input/Output	16-bit data bus for connecting external devices. In write cycle, the CPU outputs $\overline{BHW}$/$\overline{BHE}$ and $\overline{BLW}$/$\overline{BLE}$ indicating the valid byte position to write on the 16-bit data bus. In read cycle, the CPU always reads data from the full 16-bit data bus. However, data at only the valid byte position is transferred to the internal circuit of the M32R.
Bus control	$\overline{CS0}$, $\overline{CS1}$ $\overline{CS2}$, $\overline{CS3}$	Chip select	Output	Chip select signal for external devices. For details about areas for which the chip select signal is output, refer to Chapter 3, "Address Space."
	$\overline{RD}$	Read	Output	This signal is output when reading an external device.
	$\overline{BHW}$/$\overline{BHE}$	Byte high write/enable	Output	Indicates the byte position to which valid data will be transferred when writing to an external device.$\overline{BHW}$/ $\overline{BHE}$ is output for the upper address side (D0-D7 is valid), while $\overline{BLW}$/$\overline{BLE}$ is output for the lower address side (D8-D15 is valid).
	$\overline{BLW}$/$\overline{BLE}$	Byte low write/enable	Output	
	$\overline{WAIT}$	Wait	Input	If input on $\overline{WAIT}$ is low when the M32R accesses an external device, the wait cycle is extended.
	$\overline{HREQ}$	Hold request	Input	This input pin is provided for external devices to request control of the external bus. If input on the $\overline{HREQ}$ pin is pulled low, the M32R goes to a hold state.
	$\overline{HACK}$	Hold acknowledge	Output	This signal is used to indicate that the M32R has entered a hold state and relinquished control of the external bus.
Input/ output timer	TIN8-TIN11, TIN16-TIN23	Timer input	Input	Input/output timer input pin.
	TO0-TO15	Timer output	Output	Input/output timer output pin.
Interrupt controller	$\overline{SBI}$	System break interrupt	Input	System Break Interrupt (SBI) input pin for the interrupt controller.

Table 1.3.1 Description of Pin Functions (3/6)

Classification	Pin Name	Description	Type	Function
PD controller	TIN0A, TIN0B	Timer input	Input	PD0 sensor interface and timer input pin.
	TIN1A, TIN1B	Timer input	Input	PD1 sensor interface and timer input pin.
A-D converter	AVCC0	Analog power supply	—	AVCC0 is the power supply for the A-D and D-A converters. Connect AVCC0 to the power supply (5 V).
	AVSS0	Analog ground	—	AVSS0 is the analog ground for the A-D and D-A converters. Connect AVSS0 to ground.
	VREF0	Reference voltage input	Input	VREF0 is the reference voltage input pin for the A-D and D-A converters (5 V).
	AD0IN0 – AD0IN7	Analog input	Input	8-channel analog input pins for the A-D0 converter.
	AD1IN0 – AD1IN3	Analog input	Input	4-channel analog input pins for the A-D1 converter.
	(/AD0IN8) – (AD0IN15,) (/AD1IN4) – (/AD1IN15)	Analog input	Input	20-channel analog input pins used to monitor the pin levels.
D-A	DA0	Analog output	Output	Analog output pin for the D-A0 converter.
converter	DA1	Analog output	Output	Analog output pin for the D-A1 converter.
Serial I/O	SCLKI0/ SCLKO0, SCLKI1/ SCLKO1	UART transmit/ receive clock output or CSIO transmit/receive clock input/output	Input/ Output	For UART mode: These pins output a clock derived from BRG by dividing it by 2. For CSIO mode: These pins accept as input the transmit/receive clock when an external clock is selected or output the transmit/receive clock when an internal clock is selected

Table 1.3.1 Description of Pin Functions (4/6)

Classification	Pin Name	Description	Type	Function
Serial I/O	SCLKI4, SCLKI5	Clock output	Input	For UART mode: Use inhibited (in input state) For CSIO mode: Transmit/receive clock input when external clock is selected
	SCLKO4, SCLKO5	Clock output	Output	For UART mode: Clock output derived from BRG by dividing it by 2 For CSIO mode: Transmit/receive clock output
	TXD0-TXD7	Transmit data	Output	Serial I/O transmit data output pins.
	RXD0-RXD7	Received data	Input	Serial I/O received data input pins.

Table 1.3.1 Description of Pin Functions (5/6)

Classification	Pin Name	Description	Type	Function
Real-time	RTDTXD	Transmit data	Output	Serial data output pin for the real-time debugger.
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	Outputs a low-level pulse synchronously with the first clock cycle of the real-time debugger's serial data output word. The low-level pulse width indicates the type of command/data received by the real-time debugger.
Flash only	FP	Flash Protect	Input	This is a mode pin which has the function to protect the flash memory against E/W in hardware.
CAN	CTX0,CTX1	Data output	Output	These pins output the data from the CAN module.
	CRX0,CRX1	Data input	Input	These pins take in the data for 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.
	JTDI	Serial input	Input	This pin takes in the test instruction code or test data serially.
	JTDO	Serial output	Output	This pin outputs the test instruction code or test data serially.
Input/output	P00-P07	Input/output port 0	Input/Output	Programmable input/output port.
port	P10-P17	Input/output port 1	Input/Output	Programmable input/output port.
(Note)	P20-P27	Input/output port 2	Input/Output	Programmable input/output port.
	P30-P37	Input/output port 3	Input/Output	Programmable input/output port.

Table 1.3.1 Description of Pin Functions (6/6)

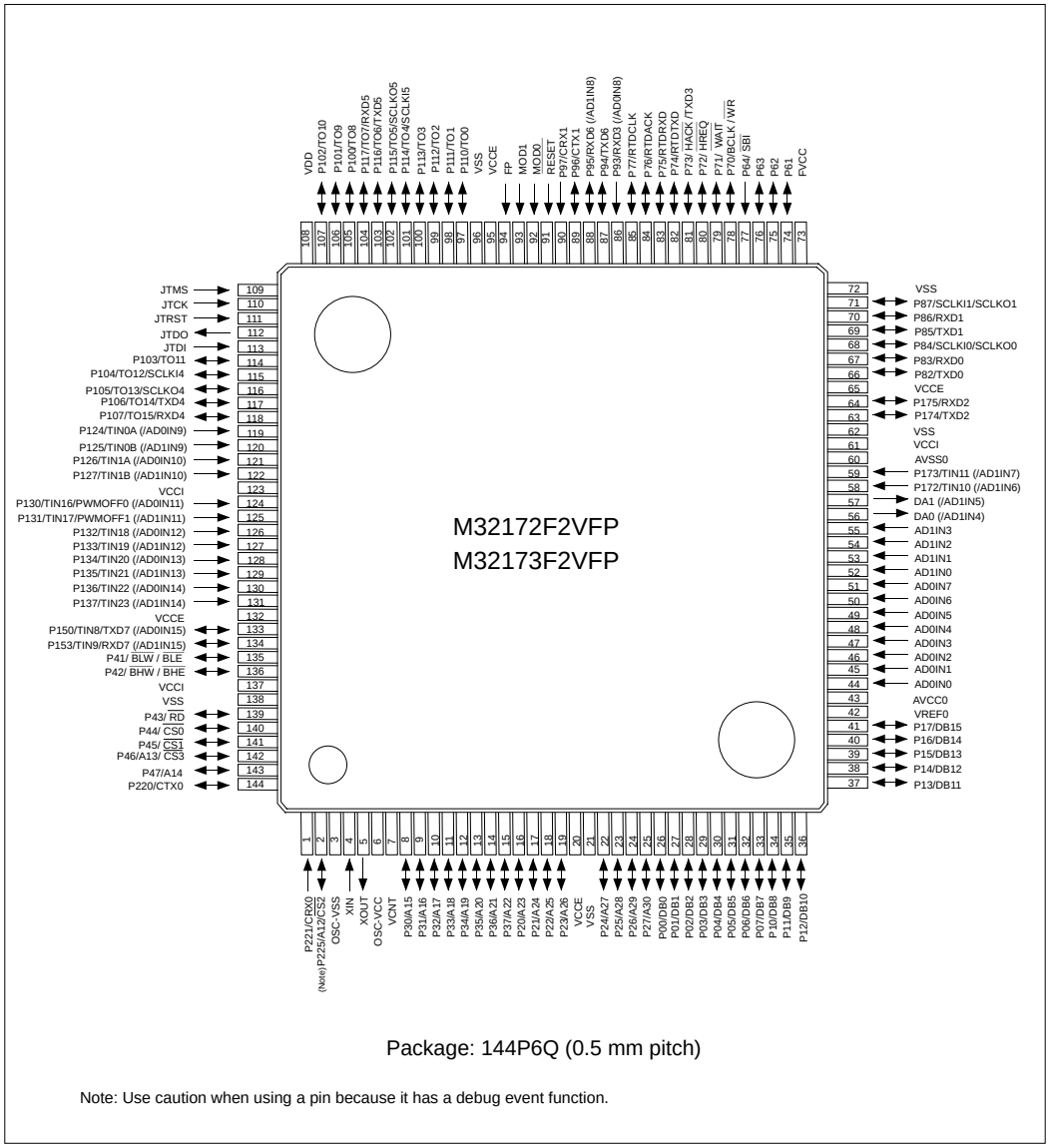
Classification	Pin Name	Description	Type	Function
Input/output port (Note 1)	P41-P47	Input/output port 4	Input/Output	Programmable input/output port.
	P61-P64	Input/output port 6	Input/Output	Programmable input/output port. (However, P64 is an input-only port.)
	P70-P77	Input/output port 7	Input/Output	Programmable input/output port.
	P82-P87	Input/output port 8	Input/Output	Programmable input/output port.
	P93-P97	Input/output port 9	Input/Output	Programmable input/output port. (However, P93 and P97 are input-only ports.)
	P100-P107	Input/output port 10	Input/Output	Programmable input/output port.
	P110-P117	Input/output port 11	Input/Output	Programmable input/output port.
	P124-P127	Input/output port 12	Input/Output	Input-only port.
	P130-P137	Input/output port 13	Input/Output	Input-only port.
	P150, P153	Input/output port 15	Input/Output	Programmable input/output port.
	P172-P175	Input/output port 17	Input/Output	Programmable input/output port. (However, P172 and P173 are input-only pins.)
	P220,P221	Input/output port 22	Input/Output	Programmable input/output port.
	P225 (Note 2)			(However, P22 is a CAN input-only pin.)

Note 1: Input/output port 5 is reserved for future use. Input/output ports 14, 16, 18, 19, 20, and 21 are nonexistent.

Note 2: Use of P225 requires caution because it has a debug event function.

1.4 Pin Layout

Figure 1.4.1 shows a pin layout diagram of the 32172/32173. Table 1.4.1 lists a pin arrangement of the 32172/32173.



Package: 144P6Q (0.5 mm pitch)

Note: Use caution when using a pin because it has a debug event function.

Figure 1.4.1 Pin Layout Diagram of the M32172F2VFP/M32173F2VFP (Top View)

Table 1.4.1 Pin Arrangement of the 144LQFP Package (1/2)

No.	Pin Name	No.	Pin Name	No.	Pin Name
1	P221/CRX0	31	P05/DB5	61	VCCI
2	P225/A12/$\overline{CS2}$	32	P06/DB6	62	VSS
3	OSC-VSS	33	P07/DB7	63	P174/TXD2
4	XIN	34	P10/DB8	64	P175/RXD2
5	XOUT	35	P11/DB9	65	VCCE
6	OSC-VCC	36	P12/DB10	66	P82/TXD0
7	VCNT	37	P13/DB11	67	P83/RXD0
8	P30/A15	38	P14/OB12	68	P84/SCLKI0/SCLKO0
9	P31/A16	39	P15/DB13	69	P85/TXD1
10	P32/A17	40	P16/DB14	70	P86/RXD1
11	P33/A18	41	P17/DB15	71	P87/SCLKI1/SCLKO1
12	P34/A19	42	VREF0	72	VSS
13	P35/A20	43	AVCC0	73	FVCC
14	P36/A21	44	AD0IN0	74	P61
15	P37/A22	45	AD0IN1	75	P62
16	P20/A23	46	AD0IN2	76	P63
17	P21/A24	47	AD0IN3	77	P64/$\overline{SBI}$
18	P22/A25	48	AD0IN4	78	P70/BCLK/$\overline{WR}$
19	P23/A26	49	AD0IN5	79	P71/$\overline{WAIT}$
20	VCCE	50	AD0IN6	80	P72/$\overline{HREQ}$
21	VSS	51	AD0IN7	81	P73/$\overline{HACK}$/TXD3
22	P24/A27	52	AD1IN0	82	P74/RTDTXD
23	P25/A28	53	AD1IN1	83	P75/RTDRXD
24	P26/A29	54	AD1IN2	84	P76/RTDACK
25	P27/A30	55	AD1IN3	85	P77/RTDCLK
26	P00/DB0	56	DA0 (/AD1IN4)	86	P93/RXD3 (/AD0IN8)
27	P01/DB1	57	DA1 (/AD1IN5)	87	P94/TXD6
28	P02/DB2	58	P172/TIN10 (/AD1IN6)	88	P95/RXD6 (/AD1IN8)
29	P03/DB3	59	P173/TIN11 (/AD1IN7)	89	P96/CTX1
30	P04/DB4	60	AVSS0	90	P97/CRX1

Table 1.4.2 Pin Arrangement of the 144LQFP Package (2/2)

No.	Pin Name	No.	Pin Name	No.	Pin Name
91	$\overline{\text{RESET}}$	111	JTRST	131	P137/TIN23 (/AD1IN14)
92	MOD0	112	JTDO	132	VCCE
93	MOD1	113	JTDI	133	P150/TIN8/TXD7 (/AD0IN15)
94	FP	114	P103/TO11	134	P153/TIN9/RXD7 (/AD1IN15)
95	VCCE	115	P104/TO12/SCLKI4	135	P41/$\overline{\text{BLW}}$/$\overline{\text{BLE}}$
96	VSS	116	P105/TO13/SCLKO4	136	P42/$\overline{\text{BHW}}$/$\overline{\text{BHE}}$
97	P110/TO0	117	P106/TO14/TXD4	137	VCCI
98	P111/TO1	118	P107/TO15/RXD4	138	VSS
99	P112/TO2	119	P124/TIN0A (/ADIN9)	139	P43/$\overline{\text{RD}}$
100	P113/TO3	120	P125/TIN0B (/AD1IN9)	140	P44/$\overline{\text{CS0}}$
101	P114/TO4/SCLKI5	121	P126/TIN1A (/AD0IN10)	141	P45/$\overline{\text{CS1}}$
102	P115/TO5/SCLKO5	122	P127/TIN1B (/AD1IN10)	142	P46/A13/$\overline{\text{CS3}}$
103	P116/TO6/TXD5	123	VCCI	143	P47/A14
104	P117/TO7/RXD5	124	P130/TIN16/PWMOFF0 (/AD0IN11)	144	P220/CTX0
105	P100/TO8	125	P131/TIN17/PWMOFF1 (/AD0IN11)		
106	P101/TO9	126	P132/TIN18 (/AD0IN12)		
107	P102/TO10	127	P133/TIN19 (/AD1IN12)		
108	VDD	128	P134/TIN20 (/AD0IN13)		
109	JTMS	129	P135/TIN21 (/AD1IN13)		
110	JTCK	130	P136/TIN22 (/AD0IN14)		

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R
15	JTRST	JTCK	P102/TO0	P117 /TO7 /RXD5	P114 /TO4 /SCLKI5	P110/TO0	MOD1	TRDATA6	TRDATA5	P93 /RXD3 (/AD0IN8)	P74 /RTDTXD	P71/$\overline{\text{WAIT}}$	P63	FVCC	N.C.
14	JTDO	JTMS	VDD	P100/TO8	P115/TO5 /SCLKO5	P111/TO1	FP	$\overline{\text{RESET}}$	P95 /RXD6 (/AD1IN8)	P77 /RTDCLK	P73/$\overline{\text{HACK}}$ /TXD3	P70/$\overline{\text{BCLK}}$ /$\overline{\text{WR}}$	P62	P61	VSS
13	N.C.	N.C.	N.C.	P101/TO9	P116/TO6 /TXD5	P112/TO2	VCCE	MOD0	P94/TXD6	P76 /RTDACK	P72/$\overline{\text{HREQ}}$	P64/$\overline{\text{SB1}}$	P87 /SCLKI1 /SCLKO1	N.C.	P86/RXD1
12	P103/TO11	JTDI	N.C.	N.C.	P113/TO3	VSS	TRDATA7	P97/CRX1	P96/CTX1	TRDATA4	P75/RTDRXD	N.C.	P85/TXD1	P84 /SCLKI0 /SCLKO0	P83/RXD0
11	P106/TO14 /TXD4	P105 /TO13 /SCLKO4	P104 /TO12 /SCLKI4	P107 /TO15 /RXD4								P174/TXD2	P82/TXD0	VCCE	P175/RXD2
10	DBI	P125 /TIN0B (/AD1IN9)	P124 /TIN0A (/AD0IN9)	P126 /TIN1A (/AD0IN10)								TRDATA0	TRDATA3	TRDATA2	TRDATA1
9	EVENT1	P127 /TIN1B (/AD1IN10)	EVENT0	VCCI								P173 /TIN11 (/AD1IN7)	VSS	VCCI	AVSS0
8	P131/TIN17 /PWMOFF1 (/AD1IN11)	P132 /TIN18 (/AD0IN12)	P133 /TIN19 (/AD1IN12)	P130/TIN16 /PWMOFF0 (/AD0IN11)								P172 /TIN10 (/AD1IN6)	AD1IN3	DA0 (/AD1IN14)	DA1 (/AD1IN15)
7	P135 /TIN21 (/AD1IN13)	P136 /TIN22 (/AD0IN14)	P137 /TIN23 (/AD1IN14)	P134 /TIN20 (/AD0IN13)								AD1IN2	AD0IN7	AD1IN0	AD1IN1
6	P150/TIN8 /TXD7 (/AD0IN15)	P153/TIN9 /RXD7 (/AD1IN15)	P41/$\overline{\text{BLW}}$ /$\overline{\text{BLE}}$	VCCE								AD0IN6	AD0IN3	AD0IN4	AD0IN5
5	VCCI	VSS	P43/$\overline{\text{RD}}$	P42/$\overline{\text{BHW}}$ /$\overline{\text{BHE}}$								AD0IN2	AVCC0	AD0IN0	AD0IN1
4	N.C.	N.C.	N.C.	N.C.	P30/A15	P34/A19	P37/A22	P23/A26	VCCE	P25/A28	P01/DB1	N.C.	N.C.	P17/DB15	VREF0
3	N.C.	N.C.	P44/$\overline{\text{CS0}}$	P225/A12 /$\overline{\text{CS2}}$	XOUT	P31/A16	P35/A20	P20/A23	P24/A27	P00/DB0	P04/DB4	P07/DB7	P16/DB14	N.C.	N.C.
2	P45/$\overline{\text{CS1}}$	P47/A14	P220/CTX0	OSC-VSS	OSC-VCC	P32/A17	TRCLK	P21/A24	TRSYNC	P27/A30	P03/DB3	P06/DB6	P11/DB9	P12/DB10	P15/DB13
1		P46/A13 /$\overline{\text{CS3}}$	P221/CRX0	XIN	VCNT	P33/A18	P36/A21	P22/A25	VSS	P26/A29	P02/DB2	P05/DB5	P10/DB8	P13/DB11	P14/DB12

M32172F2VWG
M32173F2VWG

Figure 1.4.2 Pin Layout Diagram (Top View)

Table 1.4.3 Pin Arrangement of the 175FBGA Package (1/2)

No.	Pin Name	No.	Pin Name	No.	Pin Name
A1	—	D1	XIN	G1	P36/A21
A2	P45/$\overline{CS1}$	D2	OSC-VSS	G2	TRCLK
A3	N.C.	D3	P225/A12/$\overline{CS2}$	G3	P35/A20
A4	N.C.	D4	N.C.	G4	P37/A22
A5	VCCI	D5	P42/$\overline{BHW}$/$\overline{BHE}$	G5	—
A6	P150/TIN8/TXD7(/AD0IN15)	D6	VCCE	G6	—
A7	P135/TIN21(/AD1IN13)	D7	P134/TIN20(/AD0IN13)	G7	—
A8	P131/TIN17/PMWOFF1(/AD1IN11)	D8	P130/TIN16/PWMOFF0(/AD0IN11)	G8	—
A9	EVENT1	D9	VCCI	G9	—
A10	DBI	D10	P126/TIN1A(/AD0IN10)	G10	—
A11	P106/TO14/TXD4	D11	P107/TO15/RXD4	G11	—
A12	P103/TO11	D12	N.C.	G12	TRDATA7
A13	N.C.	D13	P101/TO9	G13	VCCE
A14	JTDO	D14	P100/TO8	G14	FP
A15	JTRST	D15	P117/TO7/RXD5	G15	MOD1
B1	P46/A13/$\overline{CS3}$	E1	VCNT	H1	P22/A25
B2	P47/A14	E2	OSC-VCC	H2	P21/A24
B3	N.C.	E3	XOUT	H3	P20/A23
B4	N.C.	E4	P30/A15	H4	P23/A26
B5	VSS	E5	—	H5	—
B6	P153/TIN9/RXD7(/AD1IN15)	E6	—	H6	—
B7	P136/TIN22(/AD0IN14)	E7	—	H7	—
B8	P132/TIN18(/AD01N12)	E8	—	H8	—
B9	P127/TIN1B(/AD1IN10)	E9	—	H9	—
B10	P125/TIN0B(/AD1IN9)	E10	—	H10	—
B11	P105/TO13/SCLKO4	E11	—	H11	—
B12	JTDI	E12	P113/TO3	H12	P97/CRX1
B13	N.C.	E13	P116/TO6/TXD5	H13	MOD0
B14	JTMS	E14	P115/TO5/SCLKO5	H14	$\overline{RESET}$
B15	JTCK	E15	P114/TO4/SCLKI5	H15	TRDATA6
C1	P221/CRX0	F1	P33/A18	J1	VSS
C2	P210/CTX0	F2	P32/A17	J2	TRSYNC
C3	P44/$\overline{CS0}$	F3	P31/A16	J3	P24/A27
C4	N.C.	F4	P34/A19	J4	VCCE
C5	P43/$\overline{RD}$	F5	—	J5	—
C6	P41/$\overline{BLW}$/$\overline{BLE}$	F6	—	J6	—
C7	P137/TIN23(/AD1IN14)	F7	—	J7	—
C8	P133/TIN19(/AD1IN12)	F8	—	J8	—
C9	EVENT0	F9	—	J9	—
C10	P124/TIN0A(/AD0IN9)	F10	—	J10	—
C11	P104/TO12/SCLK14	F11	—	J11	—
C12	N.C.	F12	VSS	J12	P96/CTX1
C13	N.C.	F13	P112/TO2	J13	P94/TXD6
C14	VDD	F14	P111/TO1	J14	P95/RXD6(/AD1IN8)
C15	P102/TO10	F15	P110/TO0	J15	TRDATA5

Table 1.4.4 Pin Arrangement of the 175FBGA Package (2/2)

No.	Pin Name	No.	Pin Name	No.	Pin Name
K1	P26/A29	M1	P05/DB5	P1	P13/DB11
K2	P27/A30	M2	P06/DB6	P2	P12/DB10
K3	P00/DB0	M3	P07/DB7	P3	N.C.
K4	P25/A28	M4	N.C.	P4	P17/DB15
K5	—	M5	AD0IN2	P5	AD0IN0
K6	—	M6	AD0IN6	P6	AD0IN4
K7	—	M7	AD1IN2	P7	AD1IN0
K8	—	M8	P172/TIN10(/AD1IN6)	P8	DA0(/AD1IN14)
K9	—	M9	P173/TIN11(/AD1IN7)	P9	VCCI
K10	—	M10	TRDATA0	P10	TRDATA2
K11	—	M11	P174/TXD2	P11	VCCE
K12	TRDATA4	M12	N.C.	P12	P84/SCLKI0/SCLKO0
K13	P76/RTDACK	M13	P64/$\overline{SB1}$	P13	N.C.
K14	P77/RTDCLK	M14	P70/BCLK/$\overline{WR}$	P14	P61
K15	P93/RXD3(/AD0IN8)	M15	P71/$\overline{WAIT}$	P15	FVCC
L1	P02/DB2	N1	P10/DB8	R1	P14/DB12
L2	P03/DB3	N2	P11/DB9	R2	P15/DB13
L3	P04/DB4	N3	P16/DB14	R3	N.C.
L4	P01/DB1	N4	N.C.	R4	VREF0
L5	—	N5	AVCC0	R5	AD0IN1
L6	—	N6	AD0IN3	R6	AD0IN5
L7	—	N7	AD0IN7	R7	AD1IN1
L8	—	N8	AD1IN3	R8	DA1(/AD1IN15)
L9	—	N9	VSS	R9	AVSS0
L10	—	N10	TRDATA3	R10	TRDATA1
L11	—	N11	P82/TXD0	R11	P175/RXD2
L12	P75/RTDRXD	N12	P85/TXD1	R12	P83/RXD0
L13	P72/$\overline{HREQ}$	N13	P87/SCLKI1/SCLKO1	R13	P86/RXD1
L14	P73/$\overline{HACK}$/TXD3	N14	P62	R14	VSS
L15	P74/RTDTXD	N15	P63	R15	N.C.

*** This is a blank page ***

CHAPTER 2
CPU

2.1 CPU Registers

The M32R has sixteen general-purpose registers, five control registers, an accumulator, and a program counter. The accumulator is a 56-bit configuration, and all other registers are a 32-bit configuration.

2.2 General-purpose Registers

General-purpose registers are 32 bits in width and there are sixteen of them (R0 to R15), which are used to hold data and base addresses. Especially, R14 is used as a link register, and R15 is used as a stack pointer. The link register is used to store the return address when executing a subroutine call instruction. The stack pointer is switched between an interrupt stack pointer (SPI) and a user stack pointer (SPU) depending on the value of the Processor Status Word register (PSW)'s stack mode (SM) bit.

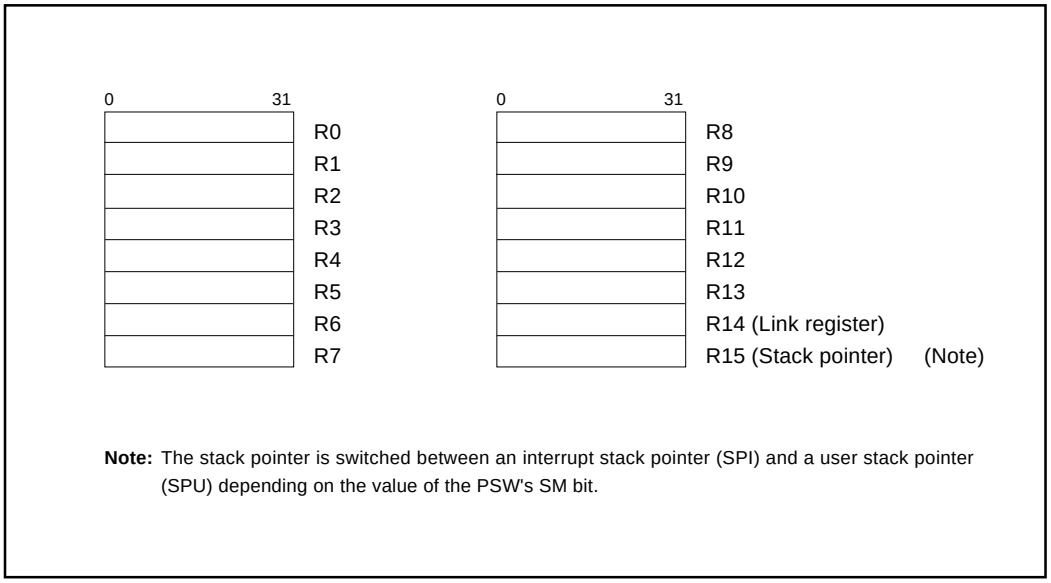
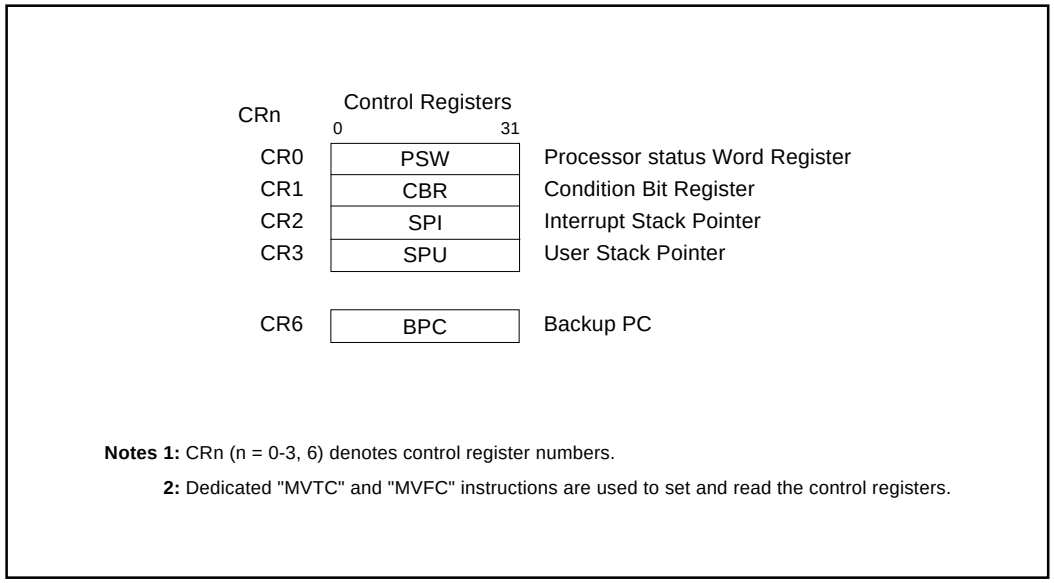


Figure 2.2.1 General-purpose Registers

2.3 Control Registers

There are five control registers-Processor Status Word Register (PSW), Condition Bit Register (CBR), Interrupt Stack Pointer (SPI), User Stack Pointer (SPU), and Backup PC (BPC). Dedicated "MVTC" and "MVFC" instructions are used to set and read these control registers.



Notes 1: CRn (n = 0-3, 6) denotes control register numbers.

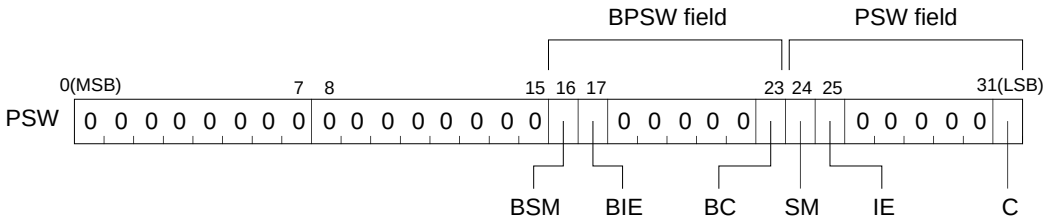
2: Dedicated "MVTC" and "MVFC" instructions are used to set and read the control registers.

Figure 2.3.1 Control Registers

2.3.1 Processor Status Word Register: PSW (CR0)

The Processor Status Word Register (PSW) is used to indicate the status of the M32R. It consists of a regularly used PSW field and a special BPSW field which is used to save the PSW field when an EIT occurs.

The PSW field consists of several bits labeled Stack Mode (SM), Interrupt Enable (IE), and Condition bit (C). The BPSW field consists of backup bits of the foregoing, i.e., Backup SM bit (BSM), Backup IE bit (BIE), and Backup C bit (BC).



(Note 1)

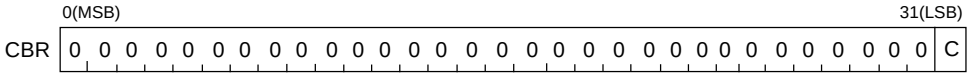
D	Bit Name	Function	Initial	R	W
16	BSM (Backup SM)	Holds the value of SM bit when EIT is accepted.	Indeterminate	○	○
17	BIE (Backup IE)	Holds the value of IE bit when EIT is accepted.	Indeterminate	○	○
23	BC (Backup C)	Holds the value of C bit when EIT is accepted.	Indeterminate	○	○
24	SM (Stack Mode)	0: Interrupt stack pointer is used. 1: User stack pointer is used.	0	○	○
25	IE (Interrupt Enable)	0: No interrupt is accepted. 1: Interrupt is accepted.	0	○	○
31	C (Condition bit)	Depending on instruction execution, it indicates whether operation resulted in a carry, borrow, or overflow.	0	○	○

Note 1: "Initial" shows the state immediately after reset, R = O means the register is readable, W = O means the register is writable.

Note 2: For changes of the state of each bit when an EIT event occurs, refer to Chapter 4, "EIT."

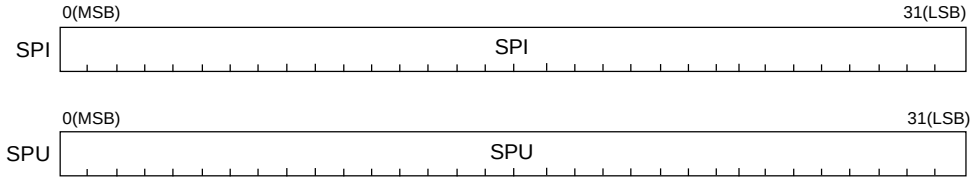
2.3.2 Condition Bit Register: CBR (CR1)

The Condition Bit Register (CBR) is created as a separate register from the PSW by extracting the Condition bit (C) from it. The value written to the PSW C bit is reflected in this register. This register is a read-only register (writes to this register by "MVTC" instruction are ignored).



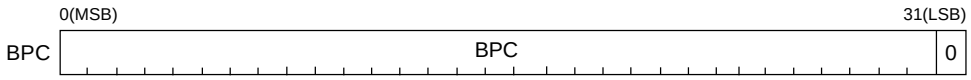
2.3.3 Interrupt Stack Pointer: SPI (CR2)
User Stack Pointer: SPU (CR3)

The Interrupt Stack Pointer (SPI) and User Stack Pointer (SPU) hold the current address of the stack pointer. These registers can be accessed as general-purpose register R15. In this case, whether R15 is used as SPI or as SPU depends on the PSW's Stack Mode (SM) bit.



2.3.4 Backup PC: BPC (CR6)

The Backup PC (BPC) is a register 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 value held in the PC immediately before the EIT occurred or the value of the next instruction is set in this register. When the "RTE" instruction is executed, the saved value is returned from the BPC to the PC. However, the two low-order bits of the PC when thus returned are always fixed to "00" (control always returns to word boundaries.)

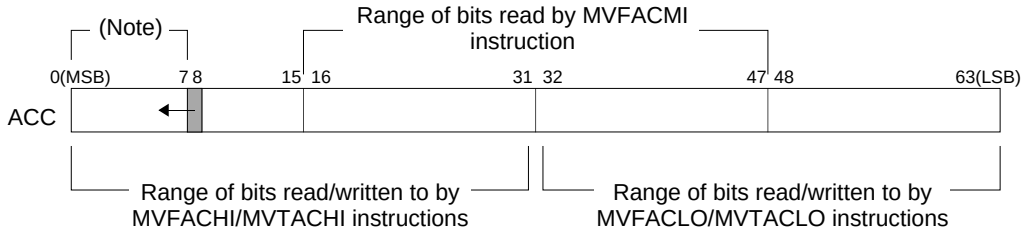


2.4 Accumulator

The accumulator (ACC) is a 56-bit register used by DSP function instructions. When read out or written to, it is handled as a 64-bit register. When reading, the value of bit 8 is sign-extended. When writing, bits 0-7 are ignored. Also, the accumulator is used by the multiplication instruction "MUL." Note that when executing this instruction, the value of the accumulator is destroyed.

The "MVTACHI" and "MVTACLO" instructions are used to write to the accumulator. The "MVTACHI" instruction writes data to the 32 high-order bits (bits 0-31), and the "MVTACLO" instruction writes data to the 32 low-order bits (bits 32-63).

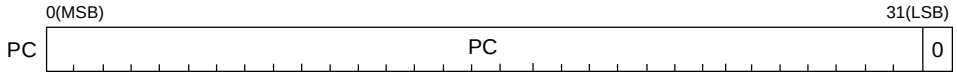
The "MVFACHI," "MVFACLO," and "MVFACMI" instructions are used to read data from the accumulator. The "MVFACHI" instruction reads data from the 32 high-order bits (bits 0-31), the "MVFACLO" instruction reads data from the 32 low-order bits (bits 32-63), and the "MVFACHI" instruction reads data from the 32 middle bits (bits 16-47).



Note: Bits 0-7 always show the sign-extended value of bit 8. Writes to this bit field are ignored.

2.5 Program Counter

The Program Counter (PC) is a 32-bit counter used to hold the address of the currently executed instruction. Because M32R instructions each start from an even address, the LSB (bit 31) is always 0.



2.6 Data Formats

2.6.1 Data Types

There are several data types that can be handled by the M32R's instruction set. These include signed and unsigned 8, 16, and 32-bit integers. Values of signed integers are represented by 2's complements.

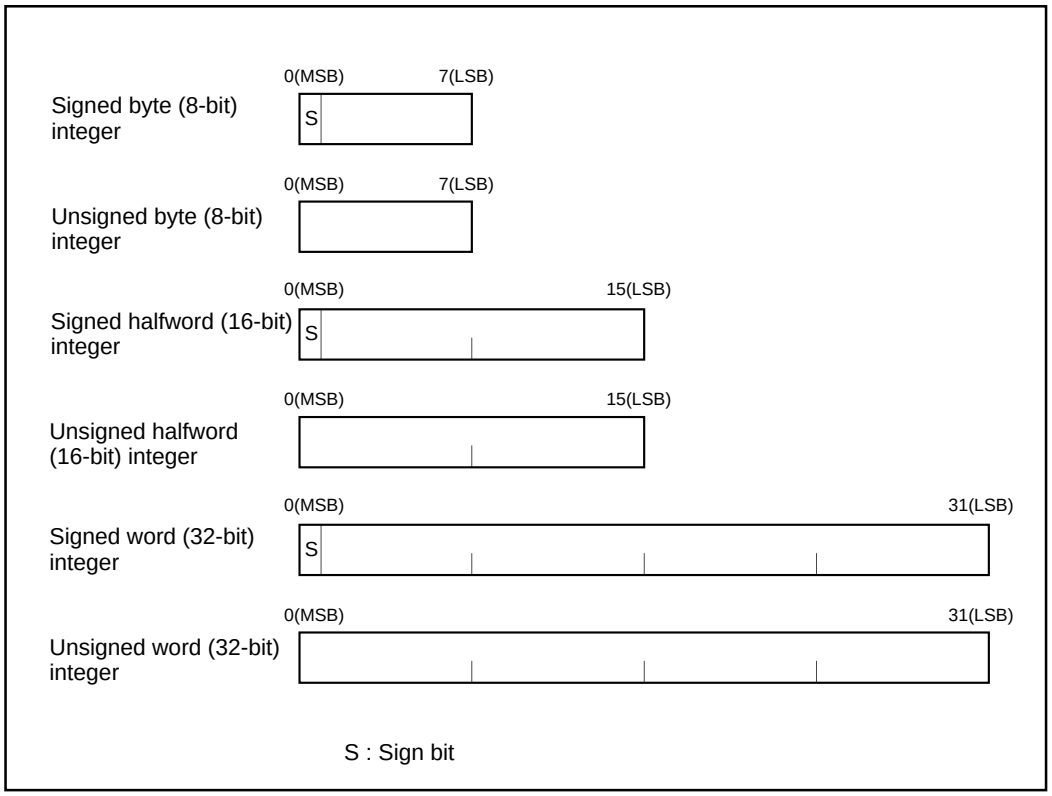


Figure 2.6.1 Data Types

2.6.2 Data Formats

(1) Data formats in register

Data sizes in 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) into word (32-bit) data before being stored in the register. When storing data from M32R register into memory, the register data is stored in memory in different sizes depending on the instructions used. The ST instruction stores the entire 32-bit data of the register, the STH instruction stores the least significant 16-bit data, and the STB instruction stores the least significant 8-bit data.

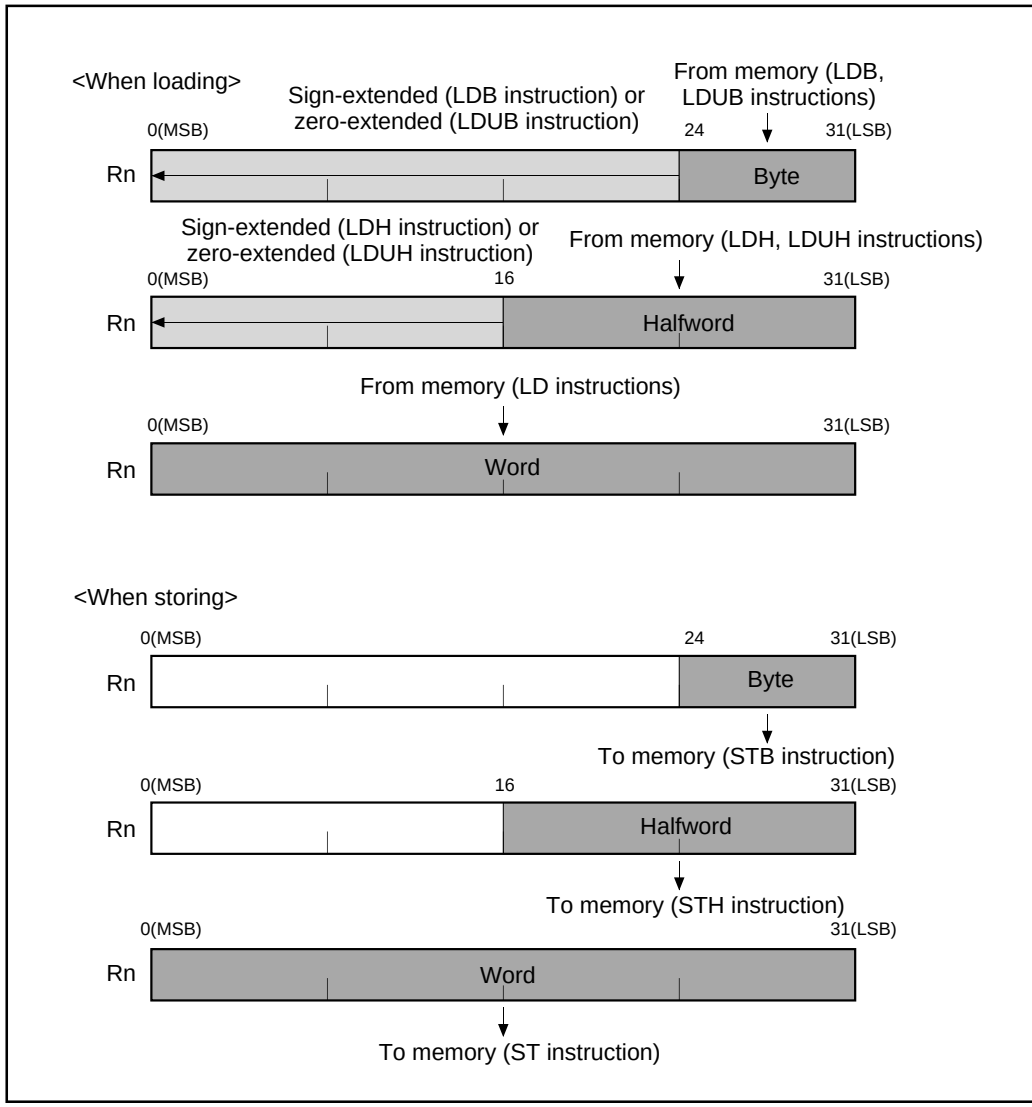


Figure 2.6.2 Data Formats in Register

(2) Data formats in memory

Data sizes in memory are either byte (8 bits), halfword (16 bits), or word (32 bits). Byte data can be located at any address. However, halfword data must be located at halfword boundaries (where the LSB address bit = "0"), and word data must be located at word boundaries (where two LSB address bits = "00"). If an attempt is made to access memory data across these halfword or word boundaries, an address exception is generated.

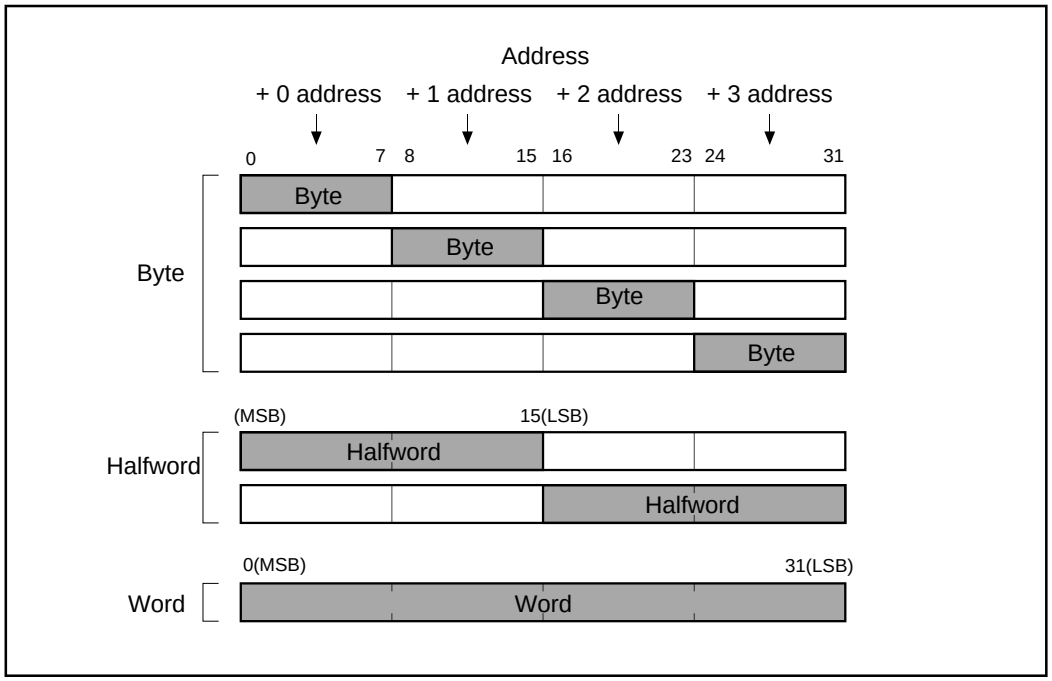


Figure 2.6.3 Data Formats in Memory

(3) Endian

The following shows the generally used endian methods and the M32R family endian.

	Bit endian (H'01)	Byte endian (H'01234567)
Big endian	MSB LSB B'0000001 D0 D7	MSB LSB H'01 H'23 H'45 H'67 HH HL LH LL
Little endian	MSB LSB B'0000001 D7 D0	MSB LSB H'67 H'45 H'23 H'01 LL LH HL HH

Note: Even for bit big endian, H'01 is not B'10000000.

Figure 2.6.4 Endian Methods

MPU name	7700 family M16C family	Competition	M32R family M16 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	MSB LSB LL LH HL HH	MSB LSB HH HL LH LL	MSB LSB HH HL LH LL
Bit number	31-24 23-16 15-8 7-0	31-24 23-16 15-8 7-0	0-7 8-15 16-23 24-31
Ex:0x01234567	.byte 67,45,23,01	.byte 01,23,45,67	.byte 01,23,45,67

Note: The M32R's endian method is big endian for both bit and byte.

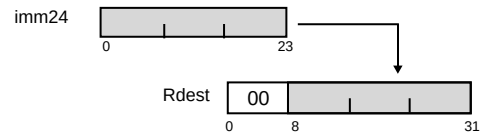
Figure 2.6.5 M32R Family Endian

(4) Transfer instructions

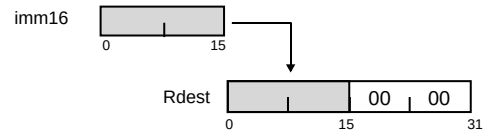
• Constant transfer

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

LD24 Rdest, #imm24



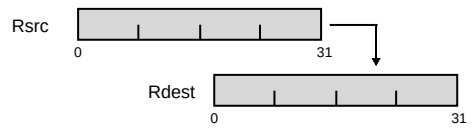
SETH Rdest, #imm16



• Register to register transfer

MV Rdest, Rsrc

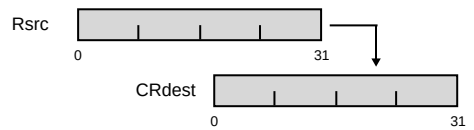
MV Rdest, Rsrc



• Control register transfer

MVFC Rdest, CRsrc
MVTC Rsrc, CRdest

MVTC Rsrc, CRdest



Note: For the MVTC instruction, the condition bit C does not change unless CRdest is CR0 (PSW).

Figure 2.6.6 Transfer instructions

(5) Memory (signed) to register transfer

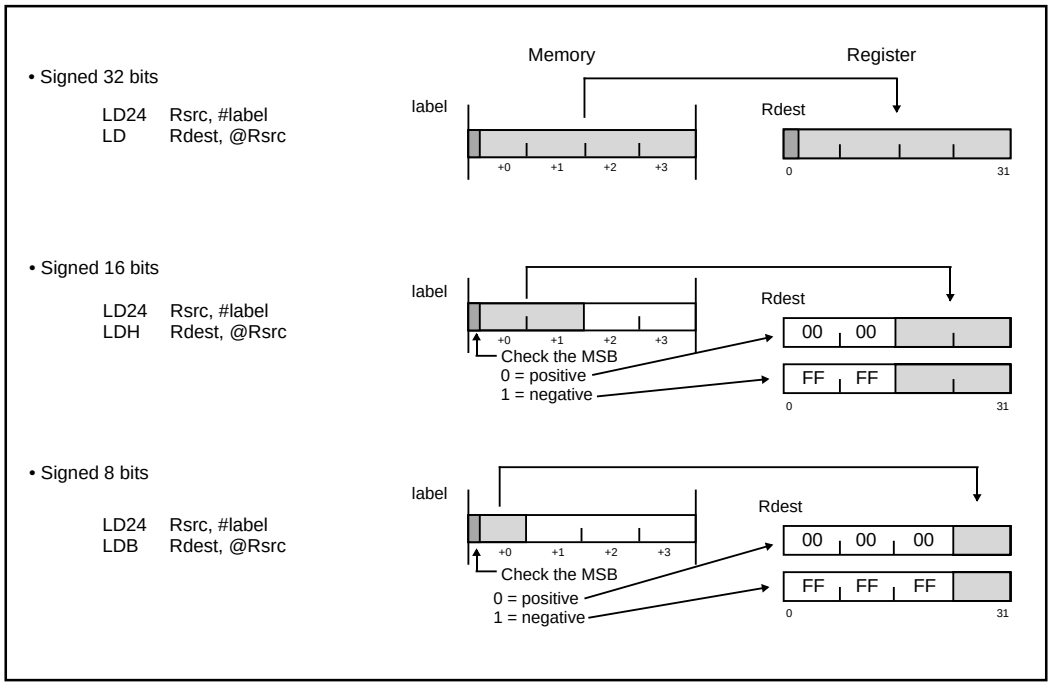


Figure 2.6.7 Memory (signed) to register transfer

(6) Memory (unsigned) to register transfer

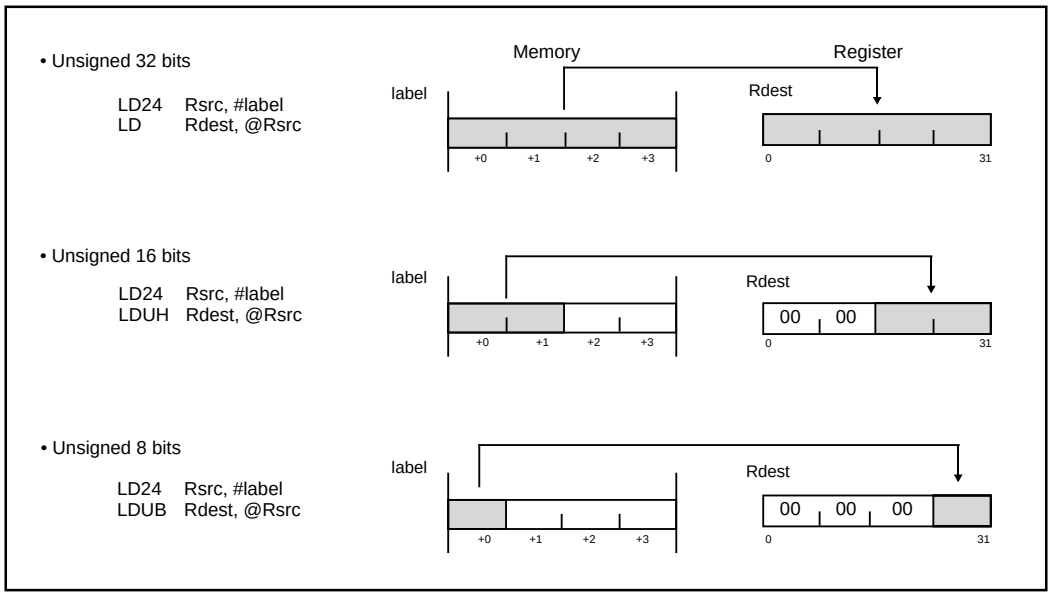


Figure 2.6.8 Memory (unsigned) to register transfer

(7) Things to be noted for data transfer

Note that in data transfer, data arrangements in registers and those in memory are different.

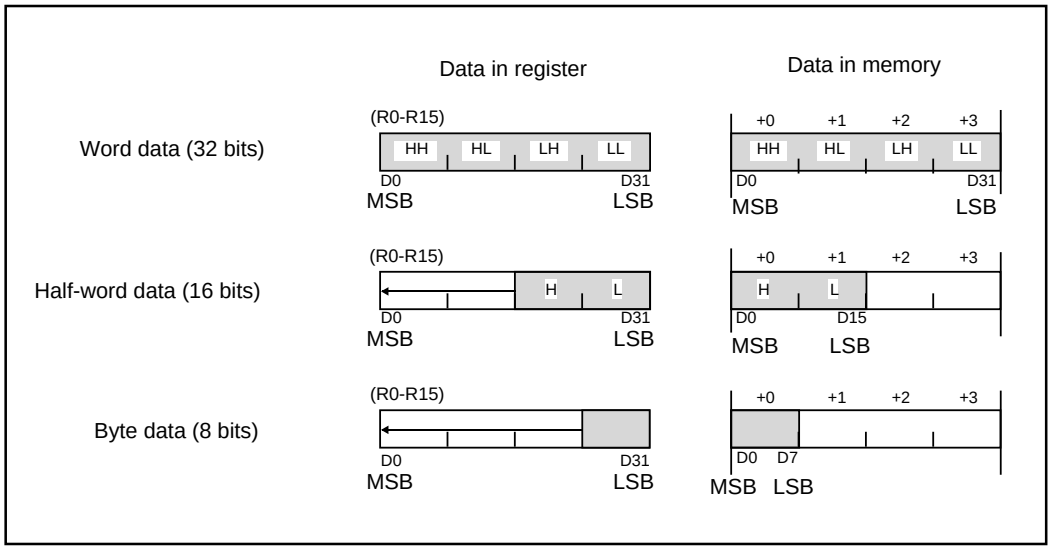


Figure 2.6.9 Difference in Data Arrangements

❊ This is a blank page. ❊

CHAPTER 3

ADDRESS SPACE

3.1 Outline of the Address Space

The logical addresses of the M32R are always handled in 32-bit width, providing a 4-Gbyte linear address space. The address space of the M32R consists of the following:

(1) User space
- Internal ROM area
- External extended area
- Internal RAM area
- SFR (Special Function Register) area

(2) Boot program space

(3) System space (not open to the user)

(1) User space

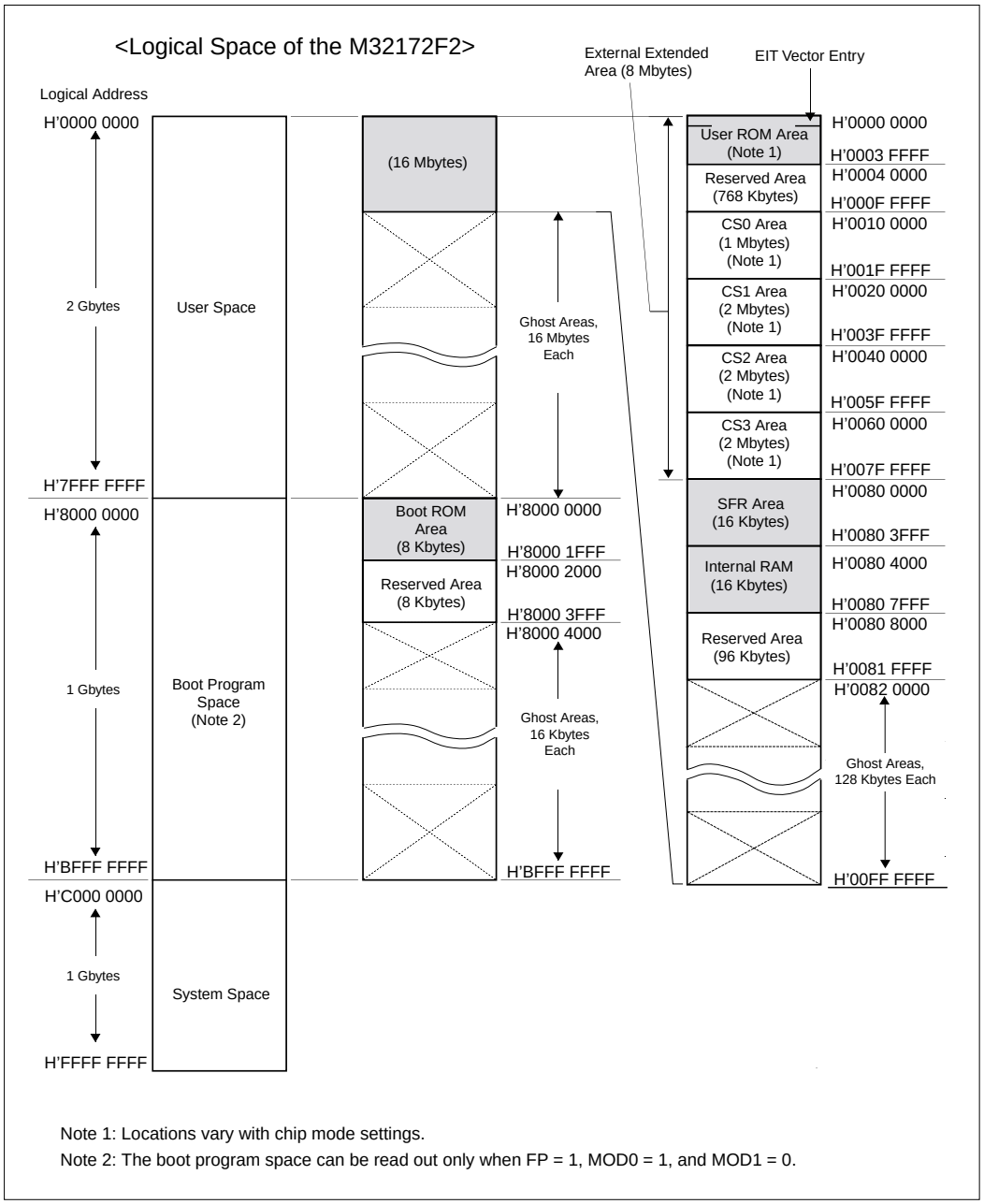
A 2 Gbytes of space in addresses from H'0000 0000 to H'7FFF FFFF is the user space. Located in this space are the internal ROM, external extended, and internal RAM areas and the SFR (Special Function Register) area (i.e., internal peripheral I/O registers). Of these, the internal ROM and external extended areas are allocated to different addresses depending on mode settings which are described later.

(2) Boot program space

A 1 Gbytes of space in addresses from H'8000 0000 to H'BFFF FFFF is the boot program space. This space stores a program (boot program) which enables on-board programming when the internal flash area is blank.

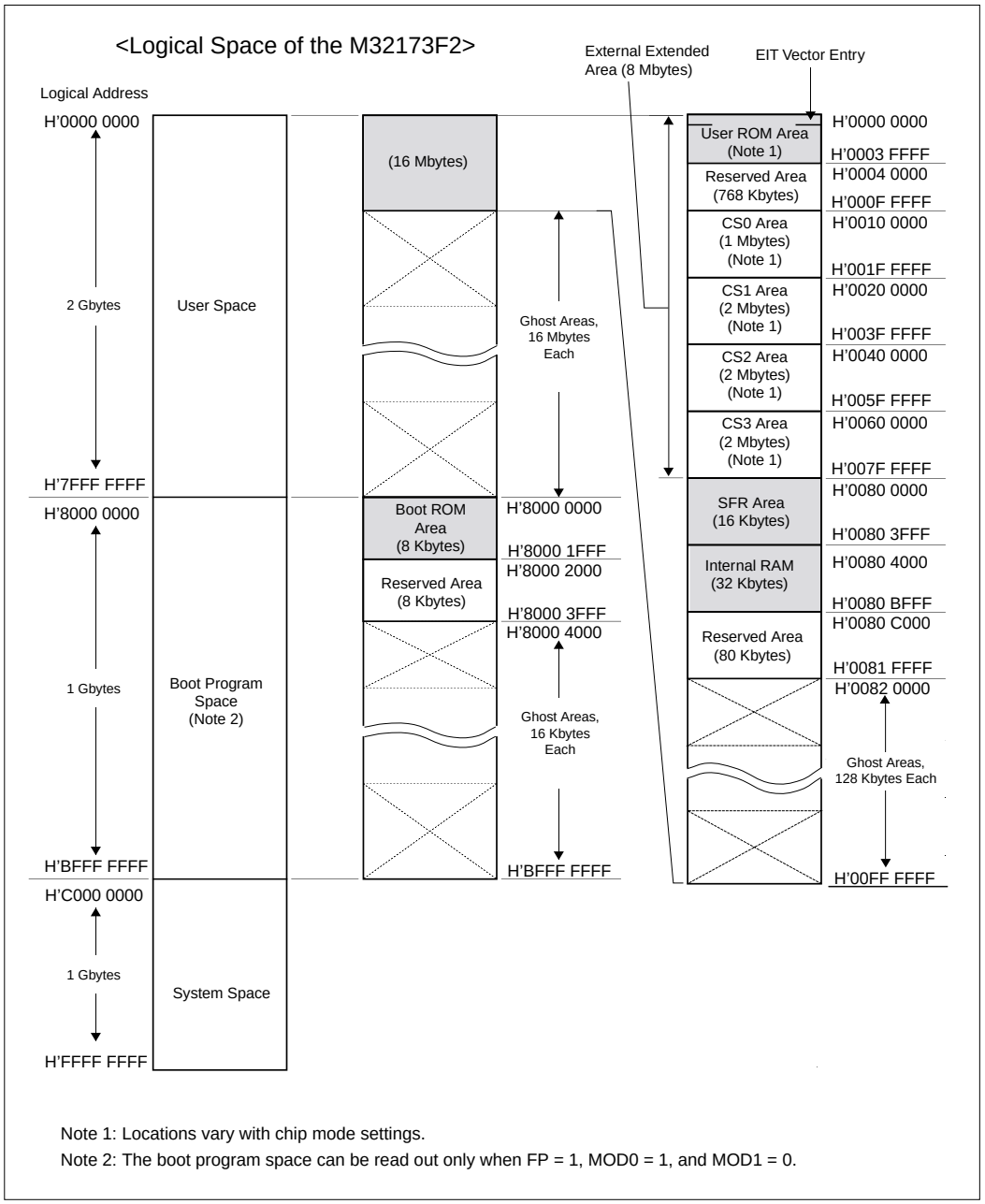
(3) System space

A 1 Gbytes of space in addresses from H'C000 0000 to H'FFFF FFFF is the system space. This space is reserved for use by development tools such as an in-circuit emulator or debug monitor, and cannot be used by the user.



Note 1: Locations vary with chip mode settings.
Note 2: The boot program space can be read out only when FP = 1, MOD0 = 1, and MOD1 = 0.

Figure 3.1.1 Address Space of the M32172F2



Note 1: Locations vary with chip mode settings.
Note 2: The boot program space can be read out only when FP = 1, MOD0 = 1, and MOD1 = 0.

Figure 3.1.2 Address Space of the M32173F2

3.2 Operation Modes

The 32172/32173 is placed in one of the following modes by settings of chip operation mode (setting the MOD0 and MOD1 pins) . For details about internal flash memory rewrite mode, refer to Section 6.5, "Programming the Internal Flash Memory."

Table 3.2.1 Setting Operation Modes

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

Note 1: VCC and VSS are connected to +5 V and GND, respectively.

Note 2: For internal flash memory rewrite mode (when FP=VCC) not listed in the above table, refer to Section 6.5, "Programming the Internal Flash Memory."

The locations of the internal ROM and external extended areas in the address space of the 32172/32173 vary depending on its operation mode. (All other areas in address space located the same way.) Also, during external extended mode, the available size of the external extended area varies with pin functions of $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, and $\overline{CS3}$. Figure 3.2.1 shows an address map of internal ROM and external extended areas in each mode. Figure 3.2.2 shows an address map of internal ROM and external extended areas varying with pin functions of $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, and $\overline{CS3}$ during external extended mode. (For details about internal flash memory rewrite mode, refer to Section 6.5, "Programming the Internal Flash Memory.")

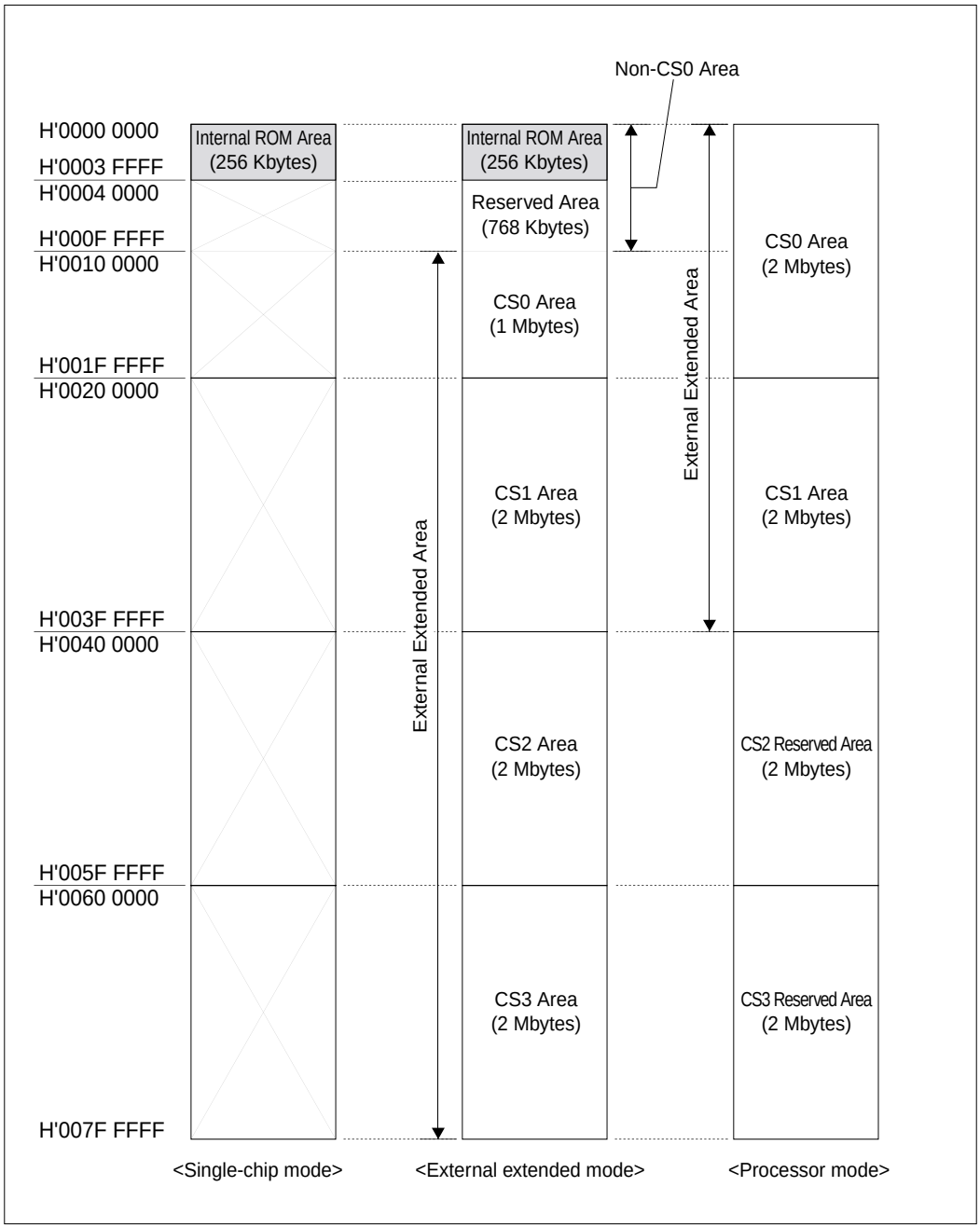


Figure 3.2.1 Internal ROM and External Extended Areas in Each Operation Mode of the M32172F2/M32173F2

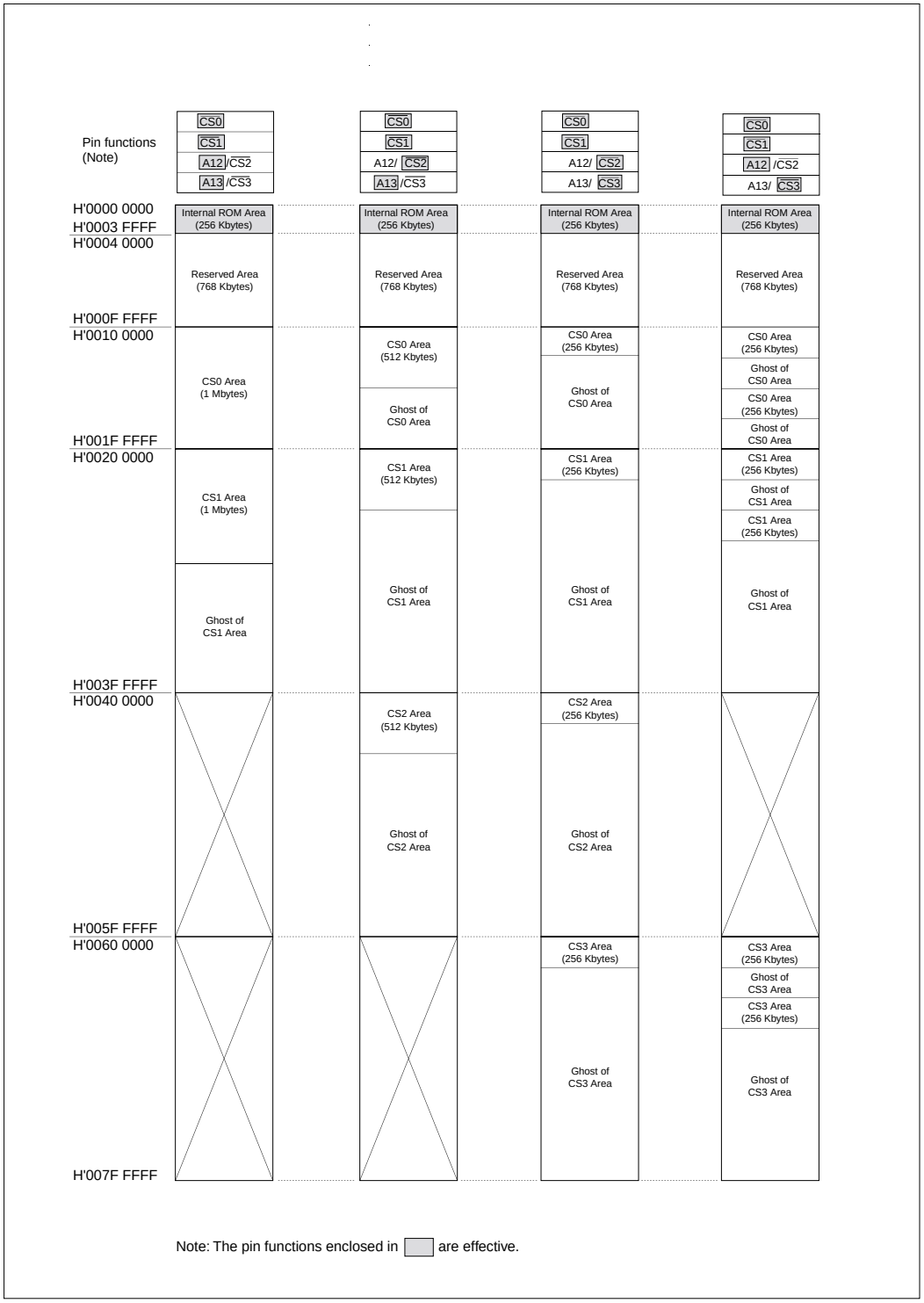


Figure 3.2.2 Internal ROM and External Extended Areas Varying with Pin Functions of the M32172F2/M32173F2

3.3 Internal ROM and External Extended Areas

The 8-Mbyte area in user space addresses from H'0000 0000 to H'007F FFFF is used for the internal ROM and external extended areas.
For details on how the locations of the internal ROM and external extended areas vary depending on 32172/32173 operation mode settings, refer to Section 3.2, "Operation Modes."

3.3.1 Internal ROM Area

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

Table 3.3.1 Internal ROM Area

Type Name	Size	Location Address
M32172F2 , M32173F2	256 Kbytes	H'0000 0000-H'0003 FFFF

3.3.2 External Extended Area

The external extended area is available only when external extended or processor mode is selected for the chip operation mode. For access to the external extended area, the 32172/32173 outputs the control signals that are required for accessing an external device.

The 32172/32173's $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, and $\overline{CS3}$ signals are output according to the address into which the external extended area is mapped. Namely, the $\overline{CS0}$ signal is output for the CS0 area, the $\overline{CS1}$ signal is output for the CS1 area, the $\overline{CS2}$ sognal is output for the CS2 area, and the $\overline{CS3}$ signal is output for the CS3 area.

Table 3.3.2 Address Mapping of the External Extended Area in Each Operation Mode of the 32172/32173

Operation Mode	Address Mapping of External Extended Area
Single-chip mode	None
External extended mode	Addresses H'0010 0000 to H'001F FFFF (CS0 area: 1 Mbytes) Addresses H'0020 0000 to H'003F FFFF (CS1 area: 2 Mbytes) Addresses H'0040 0000 to H'005F FFFF (CS2 area: 2 Mbytes) Addresses H'0060 0000 to H'007F FFFF (CS3 area: 2 Mbytes)
Processor mode	Addresses H'0000 0000 to H'001F FFFF (CS0 area: 2 Mbytes) Addresses H'0020 0000 to H'003F FFFF (CS1 area: 2 Mbytes)

3.4 Internal RAM and SFR Areas

The 8-Mbyte area in user space addresses from H'0080 0000 to H'00FF FFFF is used for the internal RAM area and the SFR (Special Function Register) area. Of these, the space that the user can actually use is a 128-Kbyte area from H'0080 0000 to H'0081 FFFF, and the other addresses comprise ghost areas in units of 128 Kbytes. (When programming, do not use the ghost area unless absolutely necessary.)

3.4.1 Internal RAM Area

For the M32172F2, the internal RAM is allocated to addresses H'0080 4000 through H'0080 7FFF (16 Kbytes). For the M32173F2, the internal RAM is allocated to addresses H'0080 4000 through H'0080 BFFF (32 Kbytes).

3.4.2 SFR (Special Function Register) Area

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

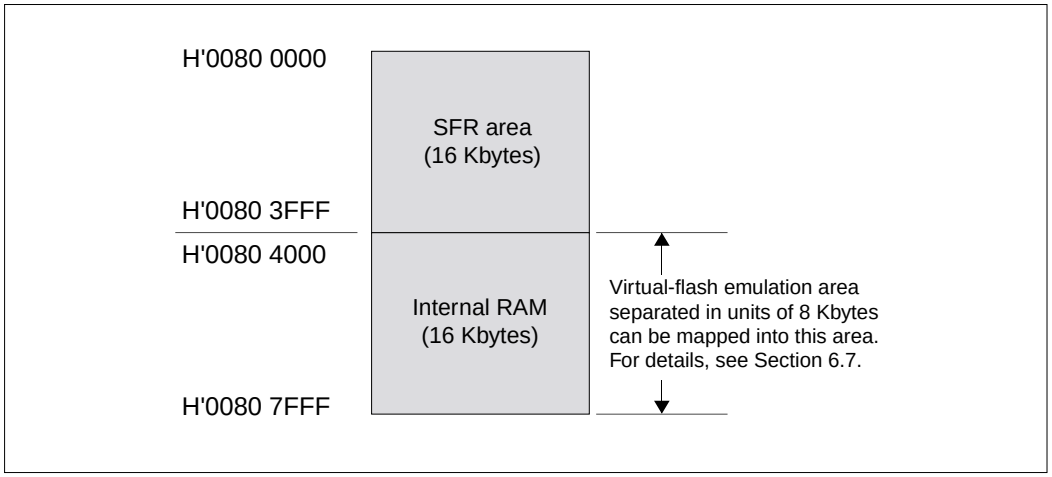


Figure 3.4.1 Internal RAM Area and SFR (Special Function Register) Area of the M32172F2

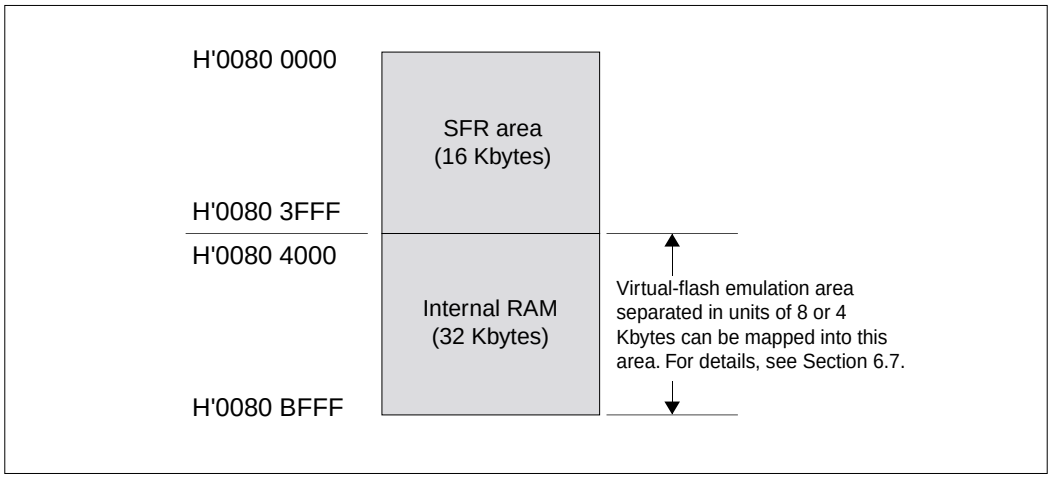
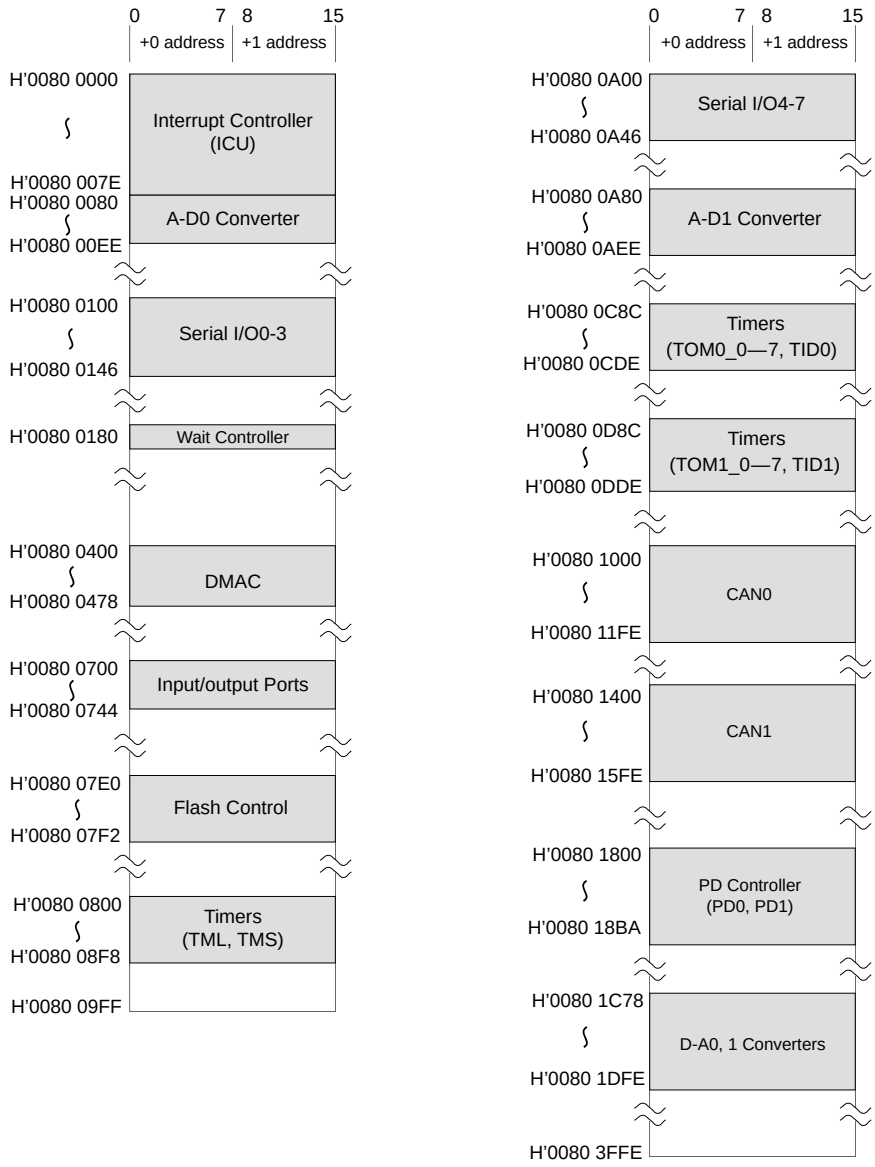


Figure 3.4.2 Internal RAM Area and SFR (Special Function Register) Area of the M32173F2



Note: The Realtime Debugger (RTD) is an independent module operated on from the outside, and is designed not to be transparent to the CPU.

Figure 3.4.3 Outline of the SFR Area Mapping

Address	+0 address D0	+1 address D8 D15
H'0080 0000	Interrupt Vector Register (IVECT)	
H'0080 0002		
H'0080 0004	Interrupt Mask Register (IMASK)	
H'0080 0006	SBI Control Register (SBICR)	
≈		≈
H'0080 0060	CAN1 Transmit/Receive & Error Interrupt Control Register (ICAN1CR)	CAN0 Transmit/Receive & Error Interrupt Control Register (ICAN0CR)
H'0080 0062	PDC Compare Match & Error Interrupt Control Register (IPDCOPCR)	RTD Interrupt Control Register (IRTDCR)
H'0080 0064	SIO6,7 Transmit/Receive Interrupt Control Register (ISIO67CR)	SIO5 Transmit Interrupt Control Register (ISIO5TXCR)
H'0080 0066	SIO5 Receive Interrupt Control Register (ISIO5RXCR)	SIO4 Transmit Interrupt Control Register (ISIO4TXCR)
H'0080 0068	SIO4 Receive Interrupt Control Register (ISIO4RXCR)	SIO2,3 Transmit/Receive Interrupt Control Register (ISIO23CR)
H'0080 006A	DMA5-9 Interrupt Control Register (IDMA59CR)	A-D1 Conversion Interrupt Control Register (IAD1CCR)
H'0080 006C	SIO1 Transmit Interrupt Control Register (ISIO1TXCR)	SIO1 Receive Interrupt Control Register (ISIO1RXCR)
H'0080 006E	SIO0 Transmit Interrupt Control Register (ISIO0TXCR)	SIO0 Receive Interrupt Control Register (ISIO0RXCR)
H'0080 0070	A-D0 Conversion Interrupt Control Register (IAD0CCR)	DMA0-4 Interrupt Control Register (IDMA04CR)
H'0080 0072	TID1 Output Interrupt Control Register (ITID1CR)	TID0 Output Interrupt Control Register (ITID0CR)
H'0080 0074	TMS0 Output Interrupt Control Register (ITMS0CR)	TOM1 Output Interrupt Control Register (ITOM1CR)
H'0080 0076	TOM0 Output Interrupt Control Register (ITOM0CR)	Timer Input Interrupt Control Register 0 (IMJTOCR0)
H'0080 0078	Timer Input Interrupt Control Register 1 (IMJTOCR1)	Timer Input Interrupt Control Register 2 (IMJTOCR2)
H'0080 007A	Timer Input Interrupt Control Register 3 (IMJTICR3)	Timer Input Interrupt Control Register 4 (IMJTICR4)
H'0080 007C	Timer Input Interrupt Control Register 5 (IMJTICR5)	PWM Off Input Interrupt Control Register (IPWMOFFCR)
H'0080 007E	PDC Input & Error Interrupt Control Register (IPDCCR)	
H'0080 0080	A-D0 Single Mode Register 0 (AD0SIM0)	A-D0 Single Mode Register 1 (AD0SIM1)
H'0080 0082		
H'0080 0084	A-D0 Scan Mode Register 0 (AD0SCM0)	A-D0 Scan Mode Register 1 (AD0SCM1)
H'0080 0086		A-D0 Conversion Rate Control Register (AD0CVSCR)
H'0080 0088	A-D0 Successive Approximation Register (AD0SAR)	
H'0080 008A		
H'0080 008C	A-D0 Comparate Data Register (AD0CMP)	
H'0080 008E	A-D0 Digital Input Control Register (AD0CHCON)	
H'0080 0090	10-bit A-D0 Data Register 0 (AD0DT0)	
H'0080 0092	10-bit A-D0 Data Register 1 (AD0DT1)	
H'0080 0094	10-bit A-D0 Data Register 2 (AD0DT2)	
H'0080 0096	10-bit A-D0 Data Register 3 (AD0DT3)	
H'0080 0098	10-bit A-D0 Data Register 4 (AD0DT4)	
H'0080 009A	10-bit A-D0 Data Register 5 (AD0DT5)	
H'0080 009C	10-bit A-D0 Data Register 6 (AD0DT6)	
H'0080 009E	10-bit A-D0 Data Register 7 (AD0DT7)	
H'0080 00A0	10-bit A-D0 Data Register 8 (AD0DT8)	
H'0080 00A2	10-bit A-D0 Data Register 9 (AD0DT9)	
H'0080 00A4	10-bit A-D0 Data Register 10 (AD0DT10)	
H'0080 00A6	10-bit A-D0 Data Register 11 (AD0DT11)	
H'0080 00A8	10-bit A-D0 Data Register 12 (AD0DT12)	
H'0080 00AA	10-bit A-D0 Data Register 13 (AD0DT13)	
H'0080 00AC	10-bit A-D0 Data Register 14 (AD0DT14)	
H'0080 00AE	10-bit A-D0 Data Register 15 (AD0DT15)	
≈		≈
H'0080 00D0		8-bit A-D0 Data Register 0 (AD08DT0)

Blank areas are reserved for future use.

Figure 3.4.4 Register Mapping of the SFR Area (1)

Address	+0 address D0	D7	+1 address D8	D15
H'0080 00D2			8-bit A-D0 Data Register 1 (AD08DT1)	
H'0080 00D4			8-bit A-D0 Data Register 2 (AD08DT2)	
H'0080 00D6			8-bit A-D0 Data Register 3 (AD08DT3)	
H'0080 00D8			8-bit A-D0 Data Register 4 (AD08DT4)	
H'0080 00DA			8-bit A-D0 Data Register 5 (AD08DT5)	
H'0080 00DC			8-bit A-D0 Data Register 6 (AD08DT6)	
H'0080 00DE			8-bit A-D0 Data Register 7 (AD08DT7)	
H'0080 00E0			8-bit A-D0 Data Register 8 (AD08DT8)	
H'0080 00E2			8-bit A-D0 Data Register 9 (AD08DT9)	
H'0080 00E4			8-bit A-D0 Data Register 10 (AD08DT10)	
H'0080 00E6			8-bit A-D0 Data Register 11 (AD08DT11)	
H'0080 00E8			8-bit A-D0 Data Register 12 (AD08DT12)	
H'0080 00EA			8-bit A-D0 Data Register 13 (AD08DT13)	
H'0080 00EC			8-bit A-D0 Data Register 14 (AD08DT14)	
H'0080 00EE			8-bit A-D0 Data Register 15 (AD08DT15)	
H'0080 0100	SIO23 Interrupt Status Register (SI23STAT)		SIO03 Interrupt Mask Register (SI03MASK)	
H'0080 0102	SIO03 Receive Interrupt Cause Select Register (SI03SEL)			
H'0080 0110	SIO0 Transmit Control Register (S0TCNT)		SIO0 Transmit/Receive Mode Register (S0MOD)	
H'0080 0112	SIO0 Transmit Buffer Register (S0TXB)			
H'0080 0114	SIO0 Receive Buffer Register (S0RXB)			
H'0080 0116	SIO0 Receive Control Register (S0RCNT)		SIO0 Baud Rate Register (S0BAUR)	
H'0080 0120	SIO1 Transmit Control Register (S1TCNT)		SIO1 Transmit/Receive Mode Register (S1MOD)	
H'0080 0122	SIO1 Transmit Buffer Register (S1TXB)			
H'0080 0124	SIO1 Receive Buffer Register (S1RXB)			
H'0080 0126	SIO1 Receive Control Register (S1RCNT)		SIO1 Baud Rate Register (S1BAUR)	
H'0080 0130	SIO2 Transmit Control Register (S2TCNT)		SIO2 Transmit/Receive Mode Register (S2MOD)	
H'0080 0132	SIO2 Transmit Buffer Register (S2TXB)			
H'0080 0134	SIO2 Receive Buffer Register (S2RXB)			
H'0080 0136	SIO2 Receive Control Register (S2RCNT)		SIO2 Baud Rate Register (S2BAUR)	
H'0080 0140	SIO3 Transmit Control Register (S3TCNT)		SIO3 Transmit/Receive Mode Register (S3MOD)	
H'0080 0142	SIO3 Transmit Buffer Register (S3TXB)			
H'0080 0144	SIO3 Receive Buffer Register (S3RXB)			
H'0080 0146	SIO3 Receive Control Register (S3RCNT)		SIO3 Baud Rate Register (S3BAUR)	
H'0080 0180	Wait Cycles Control Register (WTCCR)			

Blank areas are reserved for future use.

Figure 3.4.5 Register Mapping of the SFR Area (2)

Address	+0 address D0	+0 address D7	+1 address D8	+1 address D15
H'0080 0400	DMA0-4 Interrupt Request Status Register (DM04ITST)		DMA0-4 Interrupt Mask Register (DM04ITMK)	
≈				≈
H'0080 0408	DMA5-9 Interrupt Request Status Register (DM59ITST)		DMA5-9 Interrupt Mask Register (DM59ITMK)	
≈				≈
H'0080 0410	DMA0 Channel Control Register (DM0CNT)		DMA0 Transfer Count Register (DM0TCT)	
H'0080 0412	DMA0 Source Address Register (DM0SA)			
H'0080 0414	DMA0 Destination Address Register (DM0DA)			
H'0080 0416	DMA0 Request Cause Extension Register (DM0REQ)			
H'0080 0418	DMA5 Channel Control Register (DM5CNT)		DMA5 Transfer Count Register (DM5TCT)	
H'0080 041A	DMA5 Source Address Register (DM5SA)			
H'0080 041C	DMA5 Destination Address Register (DM5DA)			
H'0080 041E	DMA5 Request Cause Extension Register (DM5REQ)			
H'0080 0420	DMA1 Channel Control Register (DM1CNT)		DMA1 Transfer Count Register (DM1TCT)	
H'0080 0422	DMA1 Source Address Register (DM1SA)			
H'0080 0424	DMA1 Destination Address Register (DM1DA)			
H'0080 0426	DMA1 Request Cause Extension Register (DM1REQ)			
H'0080 0428	DMA6 Channel Control Register (DM6CNT)		DMA6 Transfer Count Register (DM6TCT)	
H'0080 042A	DMA6 Source Address Register (DM6SA)			
H'0080 042C	DMA6 Destination Address Register (DM6DA)			
H'0080 042E	DMA6 Request Cause Extension Register (DM6REQ)			

Blank areas are reserved for future use.

Figure 3.4.6 Register Mapping of the SFR Area (3)

Address	D0 +0 address	D7, D8 +1 address D15
H'0080 0430	DMA2 Channel Control Register (DM2CNT)	DMA2 Transfer Count Register (DM2TCT)
H'0080 0432	DMA2 Source Address Register (DM2SA)	
H'0080 0434	DMA2 Destination Address Register (DM2DA)	
H'0080 0436	DMA2 Request Cause Extension Register (DM2REQ)	
H'0080 0438	DMA7 Channel Control Register (DM7CNT)	DMA7 Transfer Count Register (DM7TCT)
H'0080 043A	DMA7 Source Address Register (DM7SA)	
H'0080 043C	DMA7 Destination Address Register (DM7DA)	
H'0080 043E	DMA7 Request Cause Extension Register (DM7REQ)	
H'0080 0440	DMA3 Channel Control Register (DM3CNT)	DMA3 Transfer Count Register (DM3TCT)
H'0080 0442	DMA3 Source Address Register (DM3SA)	
H'0080 0444	DMA3 Destination Address Register (DM3DA)	
H'0080 0446	DMA3 Request Cause Extension Register (DM3REQ)	
H'0080 0448	DMA8 Channel Control Register (DM8CNT)	DMA8 Transfer Count Register (DM8TCT)
H'0080 044A	DMA8 Source Address Register (DM8SA)	
H'0080 044C	DMA8 Destination Address Register (DM8DA)	
H'0080 044E	DMA8 Request Cause Extension Register (DM8REQ)	
H'0080 0450	DMA4 Channel Control Register (DM4CNT)	DMA4 Transfer Count Register (DM4TCT)
H'0080 0452	DMA4 Source Address Register (DM4SA)	
H'0080 0454	DMA4 Destination Address Register (DM4DA)	
H'0080 0456	DMA4 Request Cause Extension Register (DM4REQ)	
H'0080 0458	DMA9 Channel Control Register (DM9CNT)	DMA9 Transfer Count Register (DM9TCT)
H'0080 045A	DMA9 Source Address Register (DM9SA)	
H'0080 045C	DMA9 Destination Address Register (DM9DA)	
H'0080 045E	DMA9 Request Cause Extension Register (DM9REQ)	
H'0080 0460	DMA0 Software Request Generation Register (DM0SRI)	
H'0080 0462	DMA1 Software Request Generation Register (DM1SRI)	
H'0080 0464	DMA2 Software Request Generation Register (DM2SRI)	
H'0080 0466	DMA3 Software Request Generation Register (DM3SRI)	
H'0080 0468	DMA4 Software Request Generation Register (DM4SRI)	
≈		≈
H'0080 0470	DMA5 Software Request Generation Register (DM5SRI)	
H'0080 0472	DMA6 Software Request Generation Register (DM6SRI)	
H'0080 0474	DMA7 Software Request Generation Register (DM7SRI)	
H'0080 0476	DMA8 Software Request Generation Register (DM8SRI)	
H'0080 0478	DMA9 Software Request Generation Register (DM9SRI)	
≈		≈
H'0080 0700	P0 Data Register (P0DATA)	P1 Data Register (P1DATA)
H'0080 0702	P2 Data Register (P2DATA)	P3 Data Register (P3DATA)
H'0080 0704	P4 Data Register (P4DATA)	
H'0080 0706	P6 Data Register (P6DATA)	P7 Data Register (P7DATA)
H'0080 0708	P8 Data Register (P8DATA)	P9 Data Register (P9DATA)
H'0080 070A	P10 Data Register (P10DATA)	P11 Data Register (P11DATA)
H'0080 070C	P12 Data Register (P12DATA)	P13 Data Register (P13DATA)
H'0080 070E		P15 Data Register (P15DATA)
H'0080 0710		P17 Data Register (P17DATA)
H'0080 0712		
H'0080 0714		

Blank areas are reserved for future use.

Figure 3.4.7 Register Mapping of the SFR Area (4)

Address	D0 +0 address	D7 D8 +1 address D15
H'0080 0716	P22 Data Register (P22DATA)	
H'0080 0720	P0 Direction Register (P0DIR)	P1 Direction Register (P1DIR)
H'0080 0722	P2 Direction Register (P2DIR)	P3 Direction Register (P3DIR)
H'0080 0724	P4 Direction Register (P4DIR)	
H'0080 0726	P6 Direction Register (P6DIR)	P7 Direction Register (P7DIR)
H'0080 0728	P8 Direction Register (P8DIR)	P9 Direction Register (P9DIR)
H'0080 072A	P10 Direction Register (P10DIR)	P11 Direction Register (P11DIR)
H'0080 072C		
H'0080 072E		P15 Direction Register (P15DIR)
H'0080 0730		P17 Direction Register (P17DIR)
H'0080 0736	P22 Direction Register (P22DIR)	
H'0080 0740	P0 Operation Mode Register (P0MOD)	P1 Operation Mode Register (P1MOD)
H'0080 0742	P2 Operation Mode Register (P2MOD)	P3 Operation Mode Register (P3MOD)
H'0080 0744	P4 Operation Mode Register (P4MOD)	Port Input Function Enable Register (PIEN)
H'0080 0746		P7 Operation Mode Register (P7MOD)
H'0080 0748	P8 Operation Mode Register (P8MOD)	P9 Operation Mode Register (P9MOD)
H'0080 074A	P10 Operation Mode Register (P10MOD)	P11 Operation Mode Register (P11MOD)
H'0080 074C	P12 Operation Mode Register (P12MOD)	P13 Operation Mode Register (P13MOD)
H'0080 074E		P15 Operation Mode Register (P15MOD)
H'0080 0750		P17 Operation Mode Register (P17MOD)
H'0080 0752		
H'0080 0754		
H'0080 0756	P22 Operation Mode Register (P22MOD)	
H'0080 0764	P4 Peripheral Output Select Register (P4SMOD)	
H'0080 0766		P7 Peripheral Output Select Register (P7SMOD)
H'0080 0768		
H'0080 076A	P10-P11 Peripheral Output Select Register (P1011SMOD)	
H'0080 076C		
H'0080 076E		P15 Peripheral Output Select Register (P15SMOD)
H'0080 0776	P22 Peripheral Output Select Register (P22SMOD)	
H'0080 077E		Bus Mode Control Register (BUSMODC)
H'0080 07A0	PWM Output Disable Register 1 (PWMOFF1)	PWM Output Disable Register 0 (PWMOFF0)
H'0080 07A2		
H'0080 07A4	PWM Output Disable Control Register 1 (PLVCNT1)	PWM Output Disable Control Register 0 (PLVCNT0)
H'0080 07E0	Flash Mode Register (FMOD)	Flash Status Register 1 (FSTAT1)
H'0080 07E2	Flash Control Register 1 (FCNT1)	Flash Control Register 2 (FCNT2)
H'0080 07E4	Flash Control Register 3 (FCNT3)	Flash Control Register 4 (FCNT4)
H'0080 07E6		
H'0080 07E8	Virtual-flash L Bank Register 0 (FELBANK0)	
H'0080 07EA	Virtual-flash L Bank Register 1 (FELBANK1)	
H'0080 07EC	Virtual-flash L Bank Register 2 (FELBANK2)	
H'0080 07EE		
H'0080 07F0	Virtual-flash S Bank Register 0 (FESBANK0)	
H'0080 07F2	Virtual-flash S Bank Register 1 (FESBANK1)	

Blank areas are reserved for future use.

Figure 3.4.8 Register Mapping of the SFR Area (5)

Address	D0 +0 address D7	D8 +1 address D15
H'0080 0800	Input Processing Control Register 0 (TINCR0)	Input Processing Control Register 1 (TINCR1)
H'0080 0802	Input Processing Control Register 2 (TINCR2)	Input Processing Control Register 3 (TINCR3)
H'0080 0804	Input Processing Control Register 4 (TINCR4)	Input Processing Control Register 5 (TINCR5)
H'0080 0840	TIN Interrupt Status Register 0 (TINIST0)	TIN Interrupt Mask Register 0 (TINIMA0)
H'0080 0842	TIN Interrupt Status Register 1 (TINIST1)	TIN Interrupt Mask Register 1 (TINIMA1)
H'0080 0844	TIN Interrupt Status Register 2 (TINIST2)	TIN Interrupt Mask Register 2 (TINIMA2)
H'0080 0846	TIN Interrupt Status Register 3 (TINIST3)	TIN Interrupt Mask Register 3 (TINIMA3)
H'0080 0848	TIN Interrupt Status Register 4 (TINIST4)	TIN Interrupt Mask Register 4 (TINIMA4)
H'0080 084A	TIN Interrupt Status Register 5 (TINIST5)	TIN Interrupt Mask Register 5 (TINIMA5)
H'0080 0850	TIN Interrupt Status Register 8 (TINIST8)	TIN Interrupt Mask Register 8 (TINIMA8)
H'0080 0880	TML0 Counter H (TML0CTH)	
H'0080 0882	TML0 Counter L (TML0CTL)	
H'0080 088A	TML0 Control Register 0 (TML0CR)	Prescaler 1 (PRS1)
H'0080 0890	TML0 Measurement 3 Register H (TML0MR3H)	
H'0080 0892	TML0 Measurement 3 Register L (TML0MR3L)	
H'0080 0894	TML0 Measurement 2 Register H (TML0MR2H)	
H'0080 0896	TML0 Measurement 2 Register L (TML0MR2L)	
H'0080 0898	TML0 Measurement 1 Register H (TML0MR1H)	
H'0080 089A	TML0 Measurement 1 Register L (TML0MR1L)	
H'0080 089C	TML0 Measurement 0 Register H (TML0MR0H)	
H'0080 089E	TML0 Measurement 0 Register L (TML0MR0L)	
H'0080 08A0	TML0 Old Measurement 3 Register H (TML0OLDMR3H)	
H'0080 08A2	TML0 Old Measurement 3 Register L (TML0OLDMR3L)	
H'0080 08A4	TML0 Old Measurement 2 Register H (TML0OLDMR2H)	
H'0080 08A6	TML0 Old Measurement 2 Register L (TML0OLDMR2L)	
H'0080 00A8	TML0 Old Measurement 1 Register H (TML0OLDMR1H)	
H'0080 08AA	TML0 Old Measurement 1 Register L (TML0OLDMR1L)	
H'0080 08AC	TML0 Old Measurement 0 Register H (TML0OLDMR0H)	
H'0080 08AE	TML0 Old Measurement 0 Register L (TML0OLDMR0L)	
H'0080 08E0	TMS0 Counter (TMS0CT)	
H'0080 08E2	TMS0 Measurement 3 Register (TMS0MR3)	
H'0080 08E4	TMS0 Measurement 2 Register (TMS0MR2)	
H'0080 08E6	TMS0 Measurement 1 Register (TMS0MR1)	
H'0080 08E8	TMS0 Measurement 0 Register (TMS0MR0)	
H'0080 08EA	TMS0 Control Register (TMS0CR)	Prescaler (PRS0)
H'0080 08F2	TMS0 Old Measurement 3 Register (TMS0OLDMR3)	
H'0080 08F4	TMS0 Old Measurement 2 Register (TMS0OLDMR2)	
H'0080 08F6	TMS0 Old Measurement 1 Register (TMS0OLDMR1)	
H'0080 08F8	TMS0 Old Measurement 0 Register (TMS0OLDMR0)	

Blank areas are reserved for future use.

Figure 3.4.9 Register Mapping of the SFR Area (6)

Address	D0 +0 address D7	D8 +1 address D15
H'0080 0A00	SIO67 Interrupt Status Register (SI67STAT)	SIO47 Interrupt Mask Register (SI47MASK)
H'0080 0A02	SIO47 Receive Interrupt Cause Select Register (SI47SEL)	
H'0080 0A10	SIO4 Transmit Control Register (S4TCNT)	SIO4 Transmit/Receive Mode Register (S4MOD)
H'0080 0A12	SIO4 Transmit Buffer Register (S4TXB)	
H'0080 0A14	SIO4 Receive Buffer Register (S4RXB)	
H'0080 0A16	SIO4 Receive Control Register (S4RCNT)	SIO4 Baud Rate Register (S4BAUR)
H'0080 0A20	SIO5 Transmit Control Register (S5TCNT)	SIO5 Transmit/Receive Mode Register (S5MOD)
H'0080 0A22	SIO5 Transmit Buffer Register (S5TXB)	
H'0080 0A24	SIO5 Receive Buffer Register (S5RXB)	
H'0080 0A26	SIO5 Receive Control Register (S5RCNT)	SIO5 Baud Rate Register (S5BAUR)
H'0080 0A30	SIO6 Transmit Control Register (S6TCNT)	SIO6 Transmit/Receive Mode Register (S6MOD)
H'0080 0A32	SIO6 Transmit Buffer Register (S6TXB)	
H'0080 0A34	SIO6 Receive Buffer Register (S6RXB)	
H'0080 0A36	SIO6 Receive Control Register (S6RCNT)	SIO6 Baud Rate Register (S6BAUR)
H'0080 0A40	SIO7 Transmit Control Register (S7TCNT)	SIO7 Transmit/Receive Mode Register (S7MOD)
H'0080 0A42	SIO7 Transmit Buffer Register (S7TXB)	
H'0080 0A44	SIO7 Receive Buffer Register (S7RXB)	
H'0080 0A46	SIO7 Receive Control Register (S7RCNT)	SIO7 Baud Rate Register (S7BAUR)
H'0080 0A80	A-D1 Single Mode Register 0 (AD1SIM0)	A-D1 Single Mode Register 1 (AD1SIM1)
H'0080 0A82		
H'0080 0A84	A-D1 Scan Mode Register 0 (AD1SCM0)	A-D1 Scan Mode Register 1(AD1SCM1)
H'0080 0A86		A-D1 Conversion Rate Control Register (AD1CVCR)
H'0080 0A88	A-D1 Successive Approximation Register (AD1SAR)	
H'0080 0A8A		
H'0080 0A8C	A-D1 Comparate Data Register (AD1CMP)	
H'0080 0A8E	A-D1 Digital Input Control Register (AD1CHCON)	
H'0080 0A90	10-bit A-D1 Data Register 0 (AD1DT0)	
H'0080 0A92	10-bit A-D1 Data Register 1 (AD1DT1)	
H'0080 0A94	10-bit A-D1 Data Register 2 (AD1DT2)	
H'0080 0A96	10-bit A-D1 Data Register 3 (AD1DT3)	
H'0080 0A98	10-bit A-D1 Data Register 4 (AD1DT4)	
H'0080 0A9A	10-bit A-D1 Data Register 5 (AD1DT5)	
H'0080 0A9C	10-bit A-D1 Data Register 6 (AD1DT6)	
H'0080 0A9E	10-bit A-D1 Data Register 7 (AD1DT7)	
H'0080 0AA0	10-bit A-D1 Data Register 8 (AD1DT8)	
H'0080 0AA2	10-bit A-D1 Data Register 9 (AD1DT9)	
H'0080 0AA4	10-bit A-D1 Data Register 10 (AD1DT10)	
H'0080 0AA6	10-bit A-D1 Data Register 11 (AD1DT11)	
H'0080 0AA8	10-bit A-D1 Data Register 12 (AD1DT12)	
H'0080 0AAA	10-bit A-D1 Data Register 13 (AD1DT13)	
H'0080 0AAC	10-bit A-D1 Data Register 14 (AD1DT14)	
H'0080 0AAE	10-bit A-D1 Data Register 15 (AD1DT15)	

Blank areas are reserved for future use.

Figure 3.4.10 Register Mapping of the SFR Area (7)

Address	D0 +0 address D7	D8 +1 address D15
H'0080 0AD0		8-bit A-D1 Data Register 0 (AD18DT0)
H'0080 0AD2		8-bit A-D1 Data Register 1 (AD18DT1)
H'0080 0AD4		8-bit A-D1 Data Register 2 (AD18DT2)
H'0080 0AD6		8-bit A-D1 Data Register 3 (AD18DT3)
H'0080 0AD8		8-bit A-D1 Data Register 4 (AD18DT4)
H'0080 0ADA		8-bit A-D1 Data Register 5 (AD18DT5)
H'0080 0ADC		8-bit A-D1 Data Register 6 (AD18DT6)
H'0080 0ADE		8-bit A-D1 Data Register 7 (AD18DT7)
H'0080 0AE0		8-bit A-D1 Data Register 8 (AD18DT8)
H'0080 0AE2		8-bit A-D1 Data Register 9 (AD18DT9)
H'0080 0AE4		8-bit A-D1 Data Register 10 (AD18DT10)
H'0080 0AE6		8-bit A-D1 Data Register 11 (AD18DT11)
H'0080 0AE8		8-bit A-D1 Data Register 12 (AD18DT12)
H'0080 0AEA		8-bit A-D1 Data Register 13 (AD18DT13)
H'0080 0AEC		8-bit A-D1 Data Register 14 (AD18DT14)
H'0080 0AEE		8-bit A-D1 Data Register 15 (AD18DT15)

Blank areas are reserved for future use.

Figure 3.4.11 Register Mapping of the SFR Area (8)

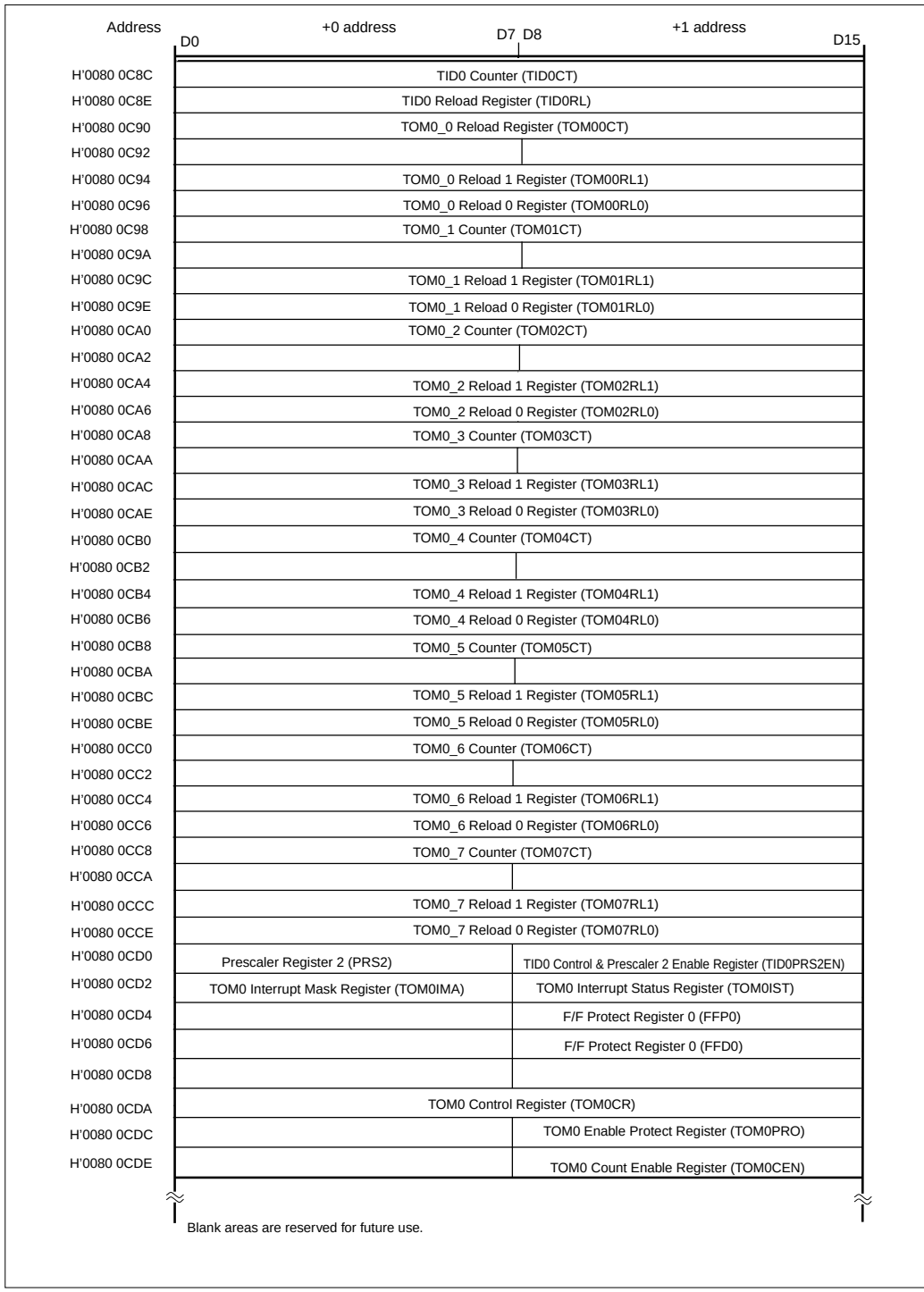


Figure 3.4.12 Register Mapping of the SFR Area (9)

Address	+0 address D0	D7 D8	+1 address D15
H'0080 0D8C	TID1 Counter (TID1CT)		
H'0080 0D8E	TID1 Reload Register (TID1RL)		
H'0080 0D90	TOM1_0 Counter Register (TOM10CT)		
H'0080 0D92			
H'0080 0D94	TOM1_0 Reload 1 Register (TOM10RL1)		
H'0080 0D96	TOM1_0 Reload 0 Register (TOM10RL0)		
H'0080 0D98	TOM1_1 Counter (TOM11CT)		
H'0080 0D9A			
H'0080 0D9C	TOM1_1 Reload 1 Register (TOM11RL1)		
H'0080 0D9E	TOM1_1 Reload 0 Register (TOM11RL0)		
H'0080 0DA0	TOM1_2 Counter (TOM12CT)		
H'0080 0DA2			
H'0080 0DA4	TOM1_2 Reload 1 Register (TOM12RL1)		
H'0080 0DA6	TOM1_2 Reload 0 Register (TOM12RL0)		
H'0080 0DA8	TOM1_3 Counter (TOM13CT)		
H'0080 0DAA			
H'0080 0CAC	TOM1_3 Reload 1 Register (TOM13RL1)		
H'0080 0DAE	TOM1_3 Reload 0 Register (TOM13RL0)		
H'0080 0DB0	TOM1_4 Counter (TOM14CT)		
H'0080 0DB2			
H'0080 0DB4	TOM1_4 Reload 1 Register (TOM14RL1)		
H'0080 0DB6	TOM1_4 Reload 0 Register (TOM14RL0)		
H'0080 0DB8	TOM1_5 Counter (TOM15CT)		
H'0080 0DBA			
H'0080 0DBC	TOM1_5 Reload 1 Register (TOM15RL1)		
H'0080 0DBE	TOM1_5 Reload 0 Register (TOM15RL0)		
H'0080 0DC0	TOM1_6 Counter (TOM16CT)		
H'0080 0DC2			
H'0080 0DC4	TOM1_6 Reload 1 Register (TOM16RL1)		
H'0080 0DC6	TOM1_6 Reload 0 Register (TOM16RL0)		
H'0080 0DC8	TOM1_7 Counter (TOM17CT)		
H'0080 0DCA			
H'0080 0DCC	TOM1_7 Reload 1 Register (TOM17RL1)		
H'0080 0DCE	TOM1_7 Reload 0 Register (TOM17RL0)		
H'0080 0DD0	Prescaler 3 Register (PRS3)		TID1 Control & Prescaler 3 Enable Register (TID1PRS3EN)
H'0080 0DD2	TOM1 Interrupt Mask Register (TOM1IMA)		TOM1 Interrupt Status Register (TOM1IST)
H'0080 0DD4			F/F Protect Register 1 (FFP1)
H'0080 0DD6			F/F Data Register 1 (FFD1)
H'0080 0DD8			
H'0080 0DDA	TOM1 Control Register (TOM1CR)		
H'0080 0DDC			TOM1 Enable Protect Register (TOM1PRO)
H'0080 0DDE			TOM1 Count Enable Register (TOM1CEN)

Blank areas are reserved for future use.

Figure 3.4.13 Register Mapping of the SFR Area (10)

Address	+0 address D0	+1 address D7, D8 D15
H'0080 1000	CAN0 Control Register (CAN0CNT)	
H'0080 1002	CAN0 Status Register (CAN0STAT)	
H'0080 1004	CAN0 Extended ID Register (CAN0EXTID)	
H'0080 1006	CAN0 Configuration Register (CAN0CONF)	
H'0080 1008	CAN0 Timestamp Count Register (CAN0TSTMP)	
H'0080 100A	CAN0 Receive Error Count Register (CAN0REC)	CAN0 Transmit Error Count Register (CAN0TEC)
H'0080 100C	CAN0 Slot Interrupt Status Register (CAN0SLIST)	
H'0080 100E		
H'0080 1010	CAN0 Slot Interrupt Mask Register (CAN0SLIMK)	
H'0080 1012		
H'0080 1014	CAN0 Error Interrupt Status Register (CAN0ERIST)	CAN0 Error Interrupt Mask Register (CAN0ERIMK)
H'0080 1016	CAN0 Baud Rate Prescaler (CAN0BRP)	
≈		≈
H'0080 1028	CAN0 Global Mask Register Standard ID0 (C0GMSKS0)	CAN0 Global Mask Register Standard ID1 (C0GMSKS1)
H'0080 102A	CAN0 Global Mask Register Extended ID0 (C0GMSKE0)	CAN0 Global Mask Register Extended ID1 (C0GMSKE1)
H'0080 102C	CAN0 Global Mask Register Extended ID2 (C0GMSKE2)	
H'0080 102E		
H'0080 1030	CAN0 Local Mask Register A Standard ID0 (C0LMSKAS0)	CAN0 Local Mask Register A Standard ID1 (C0LMSKAS1)
H'0080 1032	CAN0 Local Mask Register A Extended ID0 (C0LMSKAE0)	CAN0 Local Mask Register A Extended ID1 (C0LMSKAE1)
H'0080 1034	CAN0 Local Mask Register A Extended ID2 (C0LMSKAE2)	
H'0080 1036		
H'0080 1038	CAN0 Local Mask Register B Standard ID0 (C0LMSKBS0)	CAN0 Local Mask Register B Standard ID1 (C0LMSKBS1)
H'0080 103A	CAN0 Local Mask Register B Extended ID0 (C0LMSKBE0)	CAN0 Local Mask Register B Extended ID1 (C0LMSKBE1)
H'0080 103C	CAN0 Local Mask Register B Extended ID2 (C0LMSKBE2)	
≈		≈
H'0080 1050	CAN0 Message Slot 0 Control Register (C0MSL0CNT)	CAN0 Message Slot 1 Control Register (C0MSL1CNT)
H'0080 1052	CAN0 Message Slot 2 Control Register (C0MSL2CNT)	CAN0 Message Slot 3 Control Register (C0MSL3CNT)
H'0080 1054	CAN0 Message Slot 4 Control Register (C0MSL4CNT)	CAN0 Message Slot 5 Control Register (C0MSL5CNT)
H'0080 1056	CAN0 Message Slot 6 Control Register (C0MSL6CNT)	CAN0 Message Slot 7 Control Register (C0MSL7CNT)
H'0080 1058	CAN0 Message Slot 8 Control Register (C0MSL8CNT)	CAN0 Message Slot 9 Control Register (C0MSL9CNT)
H'0080 105A	CAN0 Message Slot 10 Control Register (C0MSL10CNT)	CAN0 Message Slot 11 Control Register (C0MSL11CNT)
H'0080 105C	CAN0 Message Slot 12 Control Register (C0MSL12CNT)	CAN0 Message Slot 13 Control Register (C0MSL13CNT)
H'0080 105E	CAN0 Message Slot 14 Control Register (C0MSL14CNT)	CAN0 Message Slot 15 Control Register (C0MSL15CNT)
≈		≈

Blank areas are reserved for future use.

Figure 3.4.14 Register Mapping of the SFR Area (11)

Address	+0 address D0	D7 D8	+1 address D15
H'0080 1100	CAN0 Message Slot 0 Standard ID0 (C0MSL0SID0)		CAN0 Message Slot 0 Standard ID1(C0MSL0SID1)
H'0080 1102	CAN0 Message Slot 0 Extended ID0 (C0MSL0EID0)		CAN0 Message Slot 0 Extended ID1(C0MSL0EID1)
H'0080 1104	CAN0 Message Slot 0 Extended ID2(C0MSL0EID2)		CAN0 Message Slot 0 Data Length Register (C0MSL0DLC)
H'0080 1106	CAN0 Message Slot 0 Data 0 (C0MSL0DT0)		CAN0 Message Slot 0 Data 1(C0MSL0DT1)
H'0080 1108	CAN0 Message Slot 0 Data 2(C0MSL0DT2)		CAN0 Message Slot 0 Data 3(C0MSL0DT3)
H'0080 110A	CAN0 Message Slot 0 Data 4(C0MSL0DT4)		CAN0 Message Slot 0 Data 5(C0MSL0DT5)
H'0080 110C	CAN0 Message Slot 0 Data 6(C0MSL0DT6)		CAN0 Message Slot 0 Data 7(C0MSL0DT7)
H'0080 110E	CAN0 Message Slot 0 Timestamp (C0MSL0TSP)		
H'0080 1110	CAN0 Message Slot 1 Standard ID0(C0MSL1SID0)		CAN0 Message Slot 1 Standard ID1(C0MSL1SID1)
H'0080 1112	CAN0 Message Slot 1 Extended ID0(C0MSL1EID0)		CAN0 Message Slot 1 Extended ID1(C0MSL1EID1)
H'0080 1114	CAN0 Message Slot 1 Extended ID2(C0MSL1EID2)		CAN0 Message Slot 1 Data Length Register(C0MSL1DLC)
H'0080 1116	CAN0 Message Slot 1 Data 0(C0MSL1DT0)		CAN0 Message Slot 1 Data 1(C0MSL1DT1)
H'0080 1118	CAN0 Message Slot 1 Data 2(C0MSL1DT2)		CAN0 Message Slot 1 Data 3(C0MSL1DT3)
H'0080 111A	CAN0 Message Slot 1 Data 4(C0MSL1DT4)		CAN0 Message Slot 1 Data 5(C0MSL1DT5)
H'0080 111C	CAN0 Message Slot 1 Data 6(C0MSL1DT6)		CAN0 Message Slot 1 Data 7(C0MSL1DT7)
H'0080 111E	CAN0 Message Slot 1 Timestamp(C0MSL1TSP)		
H'0080 1120	CAN0 Message Slot 2 Standard ID0(C0MSL2SID0)		CAN0 Message Slot 2 Standard ID1(C0MSL2SID1)
H'0080 1122	CAN0 Message Slot 2 Extended ID0(C0MSL2EID0)		CAN0 Message Slot 2 Extended ID1(C0MSL2EID1)
H'0080 1124	CAN0 Message Slot 2 Extended ID2(C0MSL2EID2)		CAN0 Message Slot 2 Data Length Register(C0MSL2DLC)
H'0080 1126	CAN0 Message Slot 2 Data 0(C0MSL2DT0)		CAN0 Message Slot 2 Data 1(C0MSL2DT1)
H'0080 1128	CAN0 Message Slot 2 Data 2(C0MSL2DT2)		CAN0 Message Slot 2 Data 3(C0MSL2DT3)
H'0080 112A	CAN0 Message Slot 2 Data 4(C0MSL2DT4)		CAN0 Message Slot 2 Data 5(C0MSL2DT5)
H'0080 112C	CAN0 Message Slot 2 Data 6(C0MSL2DT6)		CAN0 Message Slot 2 Data 7(C0MSL2DT7)
H'0080 112E	CAN0 Message Slot 2 Timestamp(C0MSL2TSP)		
H'0080 1130	CAN0 Message Slot 3 Standard ID0(C0MSL3SID0)		CAN0 Message Slot 3 Standard ID1(C0MSL3SID1)
H'0080 1132	CAN0 Message Slot 3 Extended ID0(C0MSL3EID0)		CAN0 Message Slot 3 Extended ID1(C0MSL3EID1)
H'0080 1134	CAN0 Message Slot 3 Extended ID2(C0MSL3EID2)		CAN0 Message Slot 3 Data Length Register(C0MSL3DLC)
H'0080 1136	CAN0 Message Slot 3 Data 0(C0MSL3DT0)		CAN0 Message Slot 3 Data 1(C0MSL3DT1)
H'0080 1138	CAN0 Message Slot 3 Data 2(C0MSL3DT2)		CAN0 Message Slot 3 Data 3(C0MSL3DT3)
H'0080 113A	CAN0 Message Slot 3 Data 4(C0MSL3DT4)		CAN0 Message Slot 3 Data 5(C0MSL3DT5)
H'0080 113C	CAN0 Message Slot 3 Data 6(C0MSL3DT6)		CAN0 Message Slot 3 Data 7(C0MSL3DT7)
H'0080 113E	CAN0 Message Slot 3 Timestamp(C0MSL3TSP)		
H'0080 1140	CAN0 Message Slot 4 Standard ID0(C0MSL4SID0)		CAN0 Message Slot 4 Standard ID1(C0MSL4SID1)
H'0080 1142	CAN0 Message Slot 4 Extended ID0(C0MSL4EID0)		CAN0 Message Slot 4 Extended ID1(C0MSL4EID1)
H'0080 1144	CAN0 Message Slot 4 Extended ID2(C0MSL4EID2)		CAN0 Message Slot 4 Data Length Register(C0MSL4DLC)
H'0080 1146	CAN0 Message Slot 4 Data 0(C0MSL4DT0)		CAN0 Message Slot 4 Data 1(C0MSL4DT1)
H'0080 1148	CAN0 Message Slot 4 Data 2(C0MSL4DT2)		CAN0 Message Slot 4 Data 3(C0MSL4DT3)
H'0080 114A	CAN0 Message Slot 4 Data 4(C0MSL4DT4)		CAN0 Message Slot 4 Data 5(C0MSL4DT5)
H'0080 114C	CAN0 Message Slot 4 Data 6(C0MSL4DT6)		CAN0 Message Slot 4 Data 7(C0MSL4DT7)
H'0080 114E	CAN0 Message Slot 4 Timestamp(C0MSL4TSP)		
H'0080 1150	CAN0 Message Slot 5 Standard ID0(C0MSL5SID0)		CAN0 Message Slot 5 Standard ID1(C0MSL5SID1)
H'0080 1152	CAN0 Message Slot 5 Extended ID0(C0MSL5EID0)		CAN0 Message Slot 5 Extended ID1(C0MSL5EID1)

Blank areas are reserved for future use.

Figure 3.4.15 Register Mapping of the SFR Area (12)

Address	D0 +0 address D7	D8 +1 address D15
H'0080 1154	CAN0 Message Slot 5 Extended ID2 (C0MSL5EID2)	CAN0 Message Slot 5 Data Length Register (C0MSL5DLC)
H'0080 1156	CAN0 Message Slot 5 Data 0 (C0MSL5DT0)	CAN0 Message Slot 5 Data 1(C0MSL5DT1)
H'0080 1158	CAN0 Message Slot 5 Data 2(C0MSL5DT2)	CAN0 Message Slot 5 Data 3(C0MSL5DT3)
H'0080 115A	CAN0 Message Slot 5 Data 4(C0MSL5DT4)	CAN0 Message Slot 5 Data 5(C0MSL5DT5)
H'0080 115C	CAN0 Message Slot 5 Data 6(C0MSL5DT6)	CAN0 Message Slot 5 Data 7(C0MSL5DT7)
H'0080 115E	CAN0 Message Slot 5 Timestamp (C0MSL5TSP)	
H'0080 1160	CAN0 Message Slot 6 Standard ID0 (C0MSL6SID0)	CAN0 Message Slot 6 Standard ID1(C0MSL6SID1)
H'0080 1162	CAN0 Message Slot 6 Extended ID0 (C0MSL6EID0)	CAN0 Message Slot 6 Extended ID1(C0MSL6EID1)
H'0080 1164	CAN0 Message Slot 6 Extended ID2(C0MSL6EID2)	CAN0 Message Slot 6 Data Length Register (C0MSL6DLC)
H'0080 1166	CAN0 Message Slot 6 Data 0(C0MSL6DT0)	CAN0 Message Slot 6 Data 1(C0MSL6DT1)
H'0080 1168	CAN0 Message Slot 6 Data 2(C0MSL6DT2)	CAN0 Message Slot 6 Data 3(C0MSL6DT3)
H'0080 116A	CAN0 Message Slot 6 Data 4(C0MSL6DT4)	CAN0 Message Slot 6 Data 5(C0MSL6DT5)
H'0080 116C	CAN0 Message Slot 6 Data 6(C0MSL6DT6)	CAN0 Message Slot 6 Data 7(C0MSL6DT7)
H'0080 116E	CAN0 Message Slot 6 Timestamp (C0MSL6TSP)	
H'0080 1170	CAN0 Message Slot 7 Standard ID0(C0MSL7SID0)	CAN0 Message Slot 7 Standard ID1(C0MSL7SID1)
H'0080 1172	CAN0 Message Slot 7 Extended ID0(C0MSL7EID0)	CAN0 Message Slot 7 Extended ID1(C0MSL7EID1)
H'0080 1174	CAN0 Message Slot 7 Extended ID2(C0MSL7EID2)	CAN0 Message Slot 7 Data Length Register (C0MSL7DLC)
H'0080 1176	CAN0 Message Slot 7 Data 0(C0MSL7DT0)	CAN0 Message Slot 7 Data 1(C0MSL7DT1)
H'0080 1178	CAN0 Message Slot 7 Data 2(C0MSL7DT2)	CAN0 Message Slot 7 Data 3(C0MSL7DT3)
H'0080 117A	CAN0 Message Slot 7 Data 4(C0MSL7DT4)	CAN0 Message Slot 7 Data 5(C0MSL7DT5)
H'0080 117C	CAN0 Message Slot 7 Data 6(C0MSL7DT6)	CAN0 Message Slot 7 Data 7(C0MSL7DT7)
H'0080 117E	CAN0 Message Slot 7 Timestamp (C0MSL7TSP)	
H'0080 1180	CAN0 Message Slot 8 Standard ID0(C0MSL8SID0)	CAN0 Message Slot 8 Standard ID1(C0MSL8SID1)
H'0080 1182	CAN0 Message Slot 8 Extended ID0(C0MSL8EID0)	CAN0 Message Slot 8 Extended ID1(C0MSL8EID1)
H'0080 1184	CAN0 Message Slot 8 Extended ID2(C0MSL8EID2)	CAN0 Message Slot 8 Data Length Register (C0MSL8DLC)
H'0080 1186	CAN0 Message Slot 8 Data 0(C0MSL8DT0)	CAN0 Message Slot 8 Data 1(C0MSL8DT1)
H'0080 1188	CAN0 Message Slot 8 Data 2(C0MSL8DT2)	CAN0 Message Slot 8 Data 3(C0MSL8DT3)
H'0080 118A	CAN0 Message Slot 8 Data 4(C0MSL8DT4)	CAN0 Message Slot 8 Data 5(C0MSL8DT5)
H'0080 118C	CAN0 Message Slot 8 Data 6(C0MSL8DT6)	CAN0 Message Slot 8 Data 7(C0MSL8DT7)
H'0080 118E	CAN0 Message Slot 8 Timestamp (C0MSL8TSP)	
H'0080 1190	CAN0 Message Slot 9 Standard ID0(C0MSL9SID0)	CAN0 Message Slot 9 Standard ID1(C0MSL9SID1)
H'0080 1192	CAN0 Message Slot 9 Extended ID0(C0MSL9EID0)	CAN0 Message Slot 9 Extended ID1(C0MSL9EID1)
H'0080 1194	CAN0 Message Slot 9 Extended ID2(C0MSL9EID2)	CAN0 Message Slot 9 Data Length Register (C0MSL9DLC)
H'0080 1196	CAN0 Message Slot 9 Data 0(C0MSL9DT0)	CAN0 Message Slot 9 Data 1(C0MSL9DT1)
H'0080 1198	CAN0 Message Slot 9 Data 2(C0MSL9DT2)	CAN0 Message Slot 9 Data 3(C0MSL9DT3)
H'0080 119A	CAN0 Message Slot 9 Data 4(C0MSL9DT4)	CAN0 Message Slot 9 Data 5(C0MSL9DT5)
H'0080 119C	CAN0 Message Slot 9 Data 6(C0MSL9DT6)	CAN0 Message Slot 9 Data 7(C0MSL9DT7)
H'0080 119E	CAN0 Message Slot 9 Timestamp (C0MSL9TSP)	
H'0080 11A0	CAN0 Message Slot 10 Standard ID0(C0MSL10SID0)	CAN0 Message Slot 10 Standard ID1(C0MSL10SID1)
H'0080 11A2	CAN0 Message Slot 10 Extended ID0(C0MSL10EID0)	CAN0 Message Slot 10 Extended ID1(C0MSL10EID1)
H'0080 11A4	CAN0 Message Slot 10 Extended ID2(C0MSL10EID2)	CAN0 Message Slot 10 Data Length Register (C0MSL10DLC)
H'0080 11A6	CAN0 Message Slot 10 Data 0(C0MSL10DT0)	CAN0 Message Slot 10 Data 1(C0MSL10DT1)

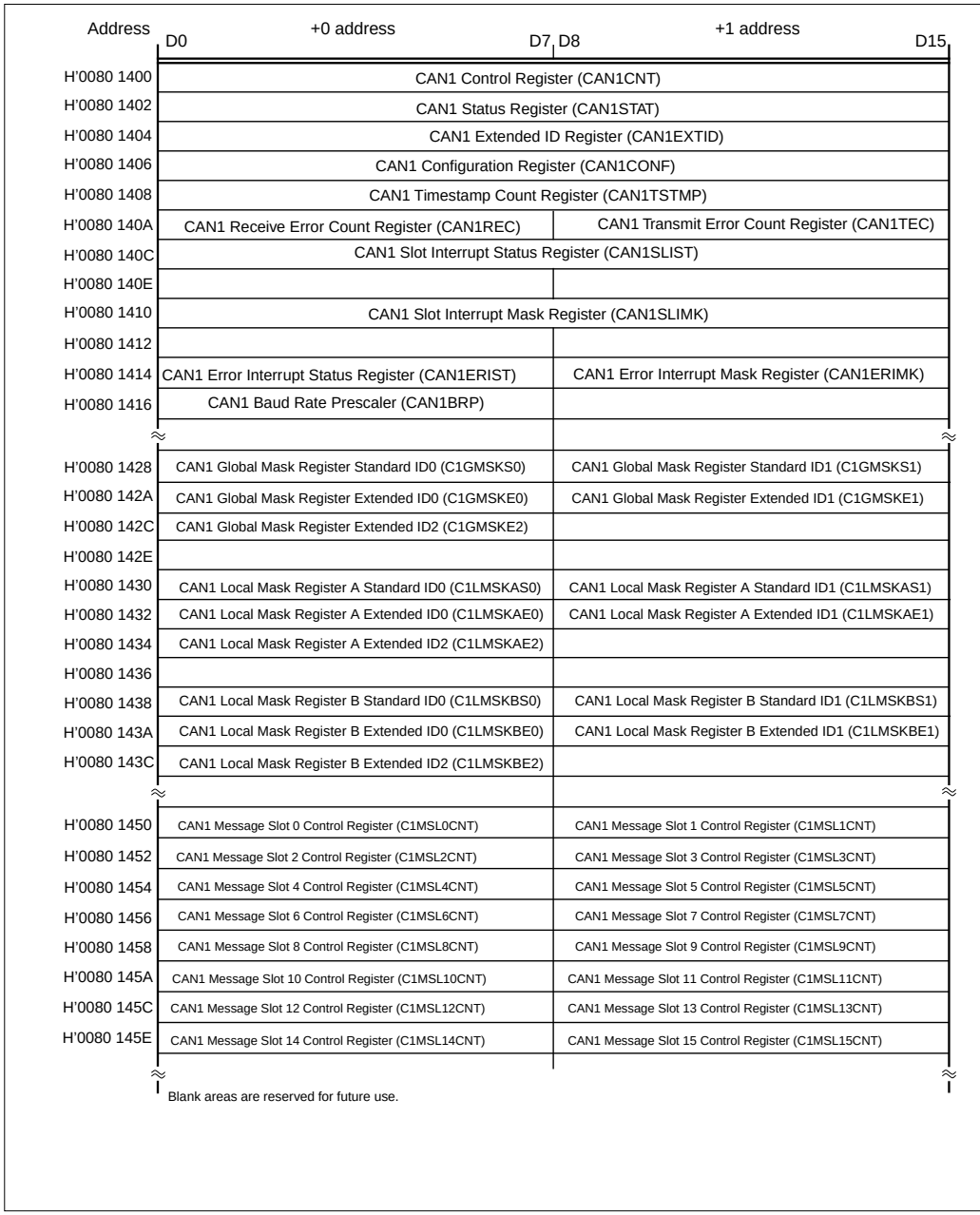
Blank areas are reserved for future use.

Figure 3.4.16 Register Mapping of the SFR Area (13)

Address	D0 +0 address D7	D8 +1 address D15
H'0080 11A8	CAN0 Message Slot 10 Data 2(C0MSL10DT2)	CAN0 Message Slot 10 Data 3(C0MSL10DT3)
H'0080 11AA	CAN0 Message Slot 10 Data 4(C0MSL10DT4)	CAN0 Message Slot 10 Data 5(C0MSL10DT5)
H'0080 11AC	CAN0 Message Slot 10 Data 6(C0MSL10DT6)	CAN0 Message Slot 10 Data 7(C0MSL10DT7)
H'0080 11AE	CAN0 Message Slot 10 Timestamp (C0MSL10TSP)	
H'0080 11B0	CAN0 Message Slot 11 Standard ID0(C0MSL11SID0)	CAN0 Message Slot 11 Standard ID1(C0MSL11SID1)
H'0080 11B2	CAN0 Message Slot 11 Extended ID0(C0MSL11EID0)	CAN0 Message Slot 11 Extended ID1(C0MSL11EID1)
H'0080 11B4	CAN0 Message Slot 11 Extended ID2(C0MSL11EID2)	CAN0 Message Slot 11 Data Length Register(C0MSL11DLC)
H'0080 11B6	CAN0 Message Slot 11 Data 0(C0MSL11DT0)	CAN0 Message Slot 11 Data 1(C0MSL11DT1)
H'0080 11B8	CAN0 Message Slot 11 Data 2(C0MSL11DT2)	CAN0 Message Slot 11 Data 3(C0MSL11DT3)
H'0080 11BA	CAN0 Message Slot 11 Data 4(C0MSL11DT4)	CAN0 Message Slot 11 Data 5(C0MSL11DT5)
H'0080 11BC	CAN0 Message Slot 11 Data 6(C0MSL11DT6)	CAN0 Message Slot 11 Data 7(C0MSL11DT7)
H'0080 11BE	CAN0 Message Slot 11 Timestamp (C0MSL11TSP)	
H'0080 11C0	CAN0 Message Slot 12 Standard ID0(C0MSL12SID0)	CAN0 Message Slot 12 Standard ID1(C0MSL12SID1)
H'0080 11C2	CAN0 Message Slot 12 Extended ID0(C0MSL12EID0)	CAN0 Message Slot 12 Extended ID1(C0MSL12EID1)
H'0080 11C4	CAN0 Message Slot 12 Extended ID2(C0MSL12EID2)	CAN0 Message Slot 12 Data Length Register(C0MSL12DLC)
H'0080 11C6	CAN0 Message Slot 12 Data 0(C0MSL12DT0)	CAN0 Message Slot 12 Data 1(C0MSL12DT1)
H'0080 11C8	CAN0 Message Slot 12 Data 2(C0MSL12DT2)	CAN0 Message Slot 12 Data 3(C0MSL12DT3)
H'0080 11CA	CAN0 Message Slot 12 Data 4(C0MSL12DT4)	CAN0 Message Slot 12 Data 5(C0MSL12DT5)
H'0080 11CC	CAN0 Message Slot 12 Data 6(C0MSL12DT6)	CAN0 Message Slot 12 Data 7(C0MSL12DT7)
H'0080 11CE	CAN0 Message Slot 12 Timestamp (C0MSL12TSP)	
H'0080 11D0	CAN0 Message Slot 13 Standard ID0(C0MSL13SID0)	CAN0 Message Slot 13 Standard ID1(C0MSL13SID1)
H'0080 11D2	CAN0 Message Slot 13 Extended ID0(C0MSL13EID0)	CAN0 Message Slot 13 Extended ID1(C0MSL13EID1)
H'0080 11D4	CAN0 Message Slot 13 Extended ID2(C0MSL13EID2)	CAN0 Message Slot 13 Data Length Register(C0MSL13DLC)
H'0080 11D6	CAN0 Message Slot 13 Data 0(C0MSL13DT0)	CAN0 Message Slot 13 Data 1(C0MSL13DT1)
H'0080 11D8	CAN0 Message Slot 13 Data 2(C0MSL13DT2)	CAN0 Message Slot 13 Data 3(C0MSL13DT3)
H'0080 11DA	CAN0 Message Slot 13 Data 4(C0MSL13DT4)	CAN0 Message Slot 13 Data 5(C0MSL13DT5)
H'0080 11DC	CAN0 Message Slot 13 Data 6(C0MSL13DT6)	CAN0 Message Slot 13 Data 7(C0MSL13DT7)
H'0080 11DE	CAN0 Message Slot 13 Timestamp (C0MSL13TSP)	
H'0080 11E0	CAN0 Message Slot 14 Standard ID0(C0MSL14SID0)	CAN0 Message Slot 14 Standard ID1(C0MSL14SID1)
H'0080 11E2	CAN0 Message Slot 14 Extended ID0(C0MSL14EID0)	CAN0 Message Slot 14 Extended ID1(C0MSL14EID1)
H'0080 11E4	CAN0 Message Slot 14 Extended ID2(C0MSL14EID2)	CAN0 Message Slot 14 Data Length Register(C0MSL14DLC)
H'0080 11E6	CAN0 Message Slot 14 Data 0(C0MSL14DT0)	CAN0 Message Slot 14 Data 1(C0MSL14DT1)
H'0080 11E8	CAN0 Message Slot 14 Data 2(C0MSL14DT2)	CAN0 Message Slot 14 Data 3(C0MSL14DT3)
H'0080 11EA	CAN0 Message Slot 14 Data 4(C0MSL14DT4)	CAN0 Message Slot 14 Data 5(C0MSL14DT5)
H'0080 11EC	CAN0 Message Slot 14 Data 6(C0MSL14DT6)	CAN0 Message Slot 14 Data 7(C0MSL14DT7)
H'0080 11EE	CAN0 Message Slot 14 Timestamp (C0MSL14TSP)	
H'0080 11F0	CAN0 Message Slot 15 Standard ID0(C0MSL15SID0)	CAN0 Message Slot 15 Standard ID1(C0MSL15SID1)
H'0080 11F2	CAN0 Message Slot 15 Extended ID0(C0MSL15EID0)	CAN0 Message Slot 15 Extended ID1(C0MSL15EID1)
H'0080 11F4	CAN0 Message Slot 15 Extended ID2(C0MSL15EID2)	CAN0 Message Slot 15 Data Length Register(C0MSL15DLC)
H'0080 11F6	CAN0 Message Slot 15 Data 0(C0MSL15DT0)	CAN0 Message Slot 15 Data 1(C0MSL15DT1)
H'0080 11F8	CAN0 Message Slot 15 Data 2(C0MSL15DT2)	CAN0 Message Slot 15 Data 3(C0MSL15DT3)
H'0080 11FA	CAN0 Message Slot 15 Data 4(C0MSL15DT4)	CAN0 Message Slot 15 Data 5(C0MSL15DT5)
H'0080 11FC	CAN0 Message Slot 15 Data 6(C0MSL15DT6)	CAN0 Message Slot 15 Data 7(C0MSL15DT7)
H'0080 11FE	CAN0 Message Slot 15 Timestamp (C0MSL15TSP)	

Blank areas are reserved for future use.

Figure 3.4.17 Register Mapping of the SFR Area (14)



Address	D0 +0 address	D7 D8 +1 address D15
H'0080 1400	CAN1 Control Register (CAN1CNT)	
H'0080 1402	CAN1 Status Register (CAN1STAT)	
H'0080 1404	CAN1 Extended ID Register (CAN1EXTID)	
H'0080 1406	CAN1 Configuration Register (CAN1CONF)	
H'0080 1408	CAN1 Timestamp Count Register (CAN1TSTMP)	
H'0080 140A	CAN1 Receive Error Count Register (CAN1REC)	CAN1 Transmit Error Count Register (CAN1TEC)
H'0080 140C	CAN1 Slot Interrupt Status Register (CAN1SLIST)	
H'0080 140E		
H'0080 1410	CAN1 Slot Interrupt Mask Register (CAN1SLIMK)	
H'0080 1412		
H'0080 1414	CAN1 Error Interrupt Status Register (CAN1ERIST)	CAN1 Error Interrupt Mask Register (CAN1ERIMK)
H'0080 1416	CAN1 Baud Rate Prescaler (CAN1BRP)	
≈		
H'0080 1428	CAN1 Global Mask Register Standard ID0 (C1GMSKS0)	CAN1 Global Mask Register Standard ID1 (C1GMSKS1)
H'0080 142A	CAN1 Global Mask Register Extended ID0 (C1GMSKE0)	CAN1 Global Mask Register Extended ID1 (C1GMSKE1)
H'0080 142C	CAN1 Global Mask Register Extended ID2 (C1GMSKE2)	
H'0080 142E		
H'0080 1430	CAN1 Local Mask Register A Standard ID0 (C1LMSKAS0)	CAN1 Local Mask Register A Standard ID1 (C1LMSKAS1)
H'0080 1432	CAN1 Local Mask Register A Extended ID0 (C1LMSKAE0)	CAN1 Local Mask Register A Extended ID1 (C1LMSKAE1)
H'0080 1434	CAN1 Local Mask Register A Extended ID2 (C1LMSKAE2)	
H'0080 1436		
H'0080 1438	CAN1 Local Mask Register B Standard ID0 (C1LMSKBS0)	CAN1 Local Mask Register B Standard ID1 (C1LMSKBS1)
H'0080 143A	CAN1 Local Mask Register B Extended ID0 (C1LMSKBE0)	CAN1 Local Mask Register B Extended ID1 (C1LMSKBE1)
H'0080 143C	CAN1 Local Mask Register B Extended ID2 (C1LMSKBE2)	
≈		
H'0080 1450	CAN1 Message Slot 0 Control Register (C1MSL0CNT)	CAN1 Message Slot 1 Control Register (C1MSL1CNT)
H'0080 1452	CAN1 Message Slot 2 Control Register (C1MSL2CNT)	CAN1 Message Slot 3 Control Register (C1MSL3CNT)
H'0080 1454	CAN1 Message Slot 4 Control Register (C1MSL4CNT)	CAN1 Message Slot 5 Control Register (C1MSL5CNT)
H'0080 1456	CAN1 Message Slot 6 Control Register (C1MSL6CNT)	CAN1 Message Slot 7 Control Register (C1MSL7CNT)
H'0080 1458	CAN1 Message Slot 8 Control Register (C1MSL8CNT)	CAN1 Message Slot 9 Control Register (C1MSL9CNT)
H'0080 145A	CAN1 Message Slot 10 Control Register (C1MSL10CNT)	CAN1 Message Slot 11 Control Register (C1MSL11CNT)
H'0080 145C	CAN1 Message Slot 12 Control Register (C1MSL12CNT)	CAN1 Message Slot 13 Control Register (C1MSL13CNT)
H'0080 145E	CAN1 Message Slot 14 Control Register (C1MSL14CNT)	CAN1 Message Slot 15 Control Register (C1MSL15CNT)
≈		

Blank areas are reserved for future use.

Figure 3.4.18 Register Mapping of the SFR Area (15)

Address	+0 address D0	+0 address D7	+1 address D8	+1 address D15
H'0080 1500	CAN1 Message Slot 0 Standard ID0(C1MSL0SID0)		CAN1 Message Slot 0 Standard ID1(C1MSL0SID1)	
H'0080 1502	CAN1 Message Slot 0 Extended ID0(C1MSL0EID0)		CAN1 Message Slot 0 Extended ID1(C1MSL0EID1)	
H'0080 1504	CAN1 Message Slot 0 Extended ID2(C1MSL0EID2)		CAN1 Message Slot 0 Data Length Register (C1MSL0DLC)	
H'0080 1506	CAN1 Message Slot 0 Data 0(C1MSL0DT0)		CAN1 Message Slot 0 Data 1(C1MSL0DT1)	
H'0080 1508	CAN1 Message Slot 0 Data 2(C1MSL0DT2)		CAN1 Message Slot 0 Data 3(C1MSL0DT3)	
H'0080 150A	CAN1 Message Slot 0 Data 4(C1MSL0DT4)		CAN1 Message Slot 0 Data 5(C1MSL0DT5)	
H'0080 150C	CAN1 Message Slot 0 Data 6(C1MSL0DT6)		CAN1 Message Slot 0 Data 7(C1MSL0DT7)	
H'0080 150E	CAN1 Message Slot 0 Timestamp (C1MSL0TSP)			
H'0080 1510	CAN1 Message Slot 1 Standard ID0(C1MSL1SID0)		CAN1 Message Slot 1 Standard ID1(C1MSL1SID1)	
H'0080 1512	CAN1 Message Slot 1 Extended ID0(C1MSL1EID0)		CAN1 Message Slot 1 Extended ID1(C1MSL1EID1)	
H'0080 1514	CAN1 Message Slot 1 Extended ID2(C1MSL1EID2)		CAN1 Message Slot 1 Data Length Register (C1MSL1DLC)	
H'0080 1516	CAN1 Message Slot 1 Data 0(C1MSL1DT0)		CAN1 Message Slot 1 Data 1(C1MSL1DT1)	
H'0080 1518	CAN1 Message Slot 1 Data 2(C1MSL1DT2)		CAN1 Message Slot 1 Data 3(C1MSL1DT3)	
H'0080 151A	CAN1 Message Slot 1 Data 4(C1MSL1DT4)		CAN1 Message Slot 1 Data 5(C1MSL1DT5)	
H'0080 151C	CAN1 Message Slot 1 Data 6(C1MSL1DT6)		CAN1 Message Slot 1 Data 7(C1MSL1DT7)	
H'0080 151E	CAN1 Message Slot 1 Timestamp (C1MSL1TSP)			
H'0080 1520	CAN1 Message Slot 2 Standard ID0(C1MSL2SID0)		CAN1 Message Slot 2 Standard ID1(C1MSL2SID1)	
H'0080 1522	CAN1 Message Slot 2 Extended ID0(C1MSL2EID0)		CAN1 Message Slot 2 Extended ID1(C1MSL2EID1)	
H'0080 1524	CAN1 Message Slot 2 Extended ID2(C1MSL2EID2)		CAN1 Message Slot 2 Data Length Register(C1MSL2DLC)	
H'0080 1526	CAN1 Message Slot 2 Data 0(C1MSL2DT0)		CAN1 Message Slot 2 Data 1(C1MSL2DT1)	
H'0080 1528	CAN1 Message Slot 2 Data 2(C1MSL2DT2)		CAN1 Message Slot 2 Data 3(C1MSL2DT3)	
H'0080 152A	CAN1 Message Slot 2 Data 4(C1MSL2DT4)		CAN1 Message Slot 2 Data 5(C1MSL2DT5)	
H'0080 152C	CAN1 Message Slot 2 Data 6(C1MSL2DT6)		CAN1 Message Slot 2 Data 7(C1MSL2DT7)	
H'0080 152E	CAN1 Message Slot 2 Timestamp (C1MSL2TSP)			
H'0080 1530	CAN1 Message Slot 3 Standard ID0(C1MSL3SID0)		CAN1 Message Slot 3 Standard ID1(C1MSL3SID1)	
H'0080 1532	CAN1 Message Slot 3 Extended ID0(C1MSL3EID0)		CAN1 Message Slot 3 Extended ID1(C1MSL3EID1)	
H'0080 1534	CAN1 Message Slot 3 Extended ID2(C1MSL3EID2)		CAN1 Message Slot 3 Data Length Register(C1MSL3DLC)	
H'0080 1536	CAN1 Message Slot 3 Data 0(C1MSL3DT0)		CAN1 Message Slot 3 Data 1(C1MSL3DT1)	
H'0080 1538	CAN1 Message Slot 3 Data 2(C1MSL3DT2)		CAN1 Message Slot 3 Data 3(C1MSL3DT3)	
H'0080 153A	CAN1 Message Slot 3 Data 4(C1MSL3DT4)		CAN1 Message Slot 3 Data 5(C1MSL3DT5)	
H'0080 153C	CAN1 Message Slot 3 Data 6(C1MSL3DT6)		CAN1 Message Slot 3 Data 7(C1MSL3DT7)	
H'0080 153E	CAN1 Message Slot 3 Timestamp (C1MSL3TSP)			
H'0080 1540	CAN1 Message Slot 4 Standard ID0(C1MSL4SID0)		CAN1 Message Slot 4 Standard ID1(C1MSL4SID1)	
H'0080 1542	CAN1 Message Slot 4 Extended ID0(C1MSL4EID0)		CAN1 Message Slot 4 Extended ID1(C1MSL4EID1)	
H'0080 1544	CAN1 Message Slot 4 Extended ID2(C1MSL4EID2)		CAN1 Message Slot 4 Data Length Register(C1MSL4DLC)	
H'0080 1546	CAN1 Message Slot 4 Data 0(C1MSL4DT0)		CAN1 Message Slot 4 Data 1(C1MSL4DT1)	
H'0080 1548	CAN1 Message Slot 4 Data 2(C1MSL4DT2)		CAN1 Message Slot 4 Data 3(C1MSL4DT3)	
H'0080 154A	CAN1 Message Slot 4 Data 4(C1MSL4DT4)		CAN1 Message Slot 4 Data 5(C1MSL4DT5)	
H'0080 154C	CAN1 Message Slot 4 Data 6(C1MSL4DT6)		CAN1 Message Slot 4 Data 7(C1MSL4DT7)	
H'0080 154E	CAN1 Message Slot 4 Timestamp (C1MSL4TSP)			
H'0080 1550	CAN1 Message Slot 5 Standard ID0(C1MSL5SID0)		CAN1 Message Slot 5 Standard ID1(C1MSL5SID1)	
H'0080 1552	CAN1 Message Slot 5 Extended ID0(C1MSL5EID0)		CAN1 Message Slot 5 Extended ID1(C1MSL5EID1)	

Blank areas are reserved for future use.

Figure 3.4.19 Register Mapping of the SFR Area (16)

Address	+0 address D0	D7 D8	+1 address D15
H'0080 1554	CAN1 Message Slot 5 Extended ID2(C1MSL5EID2)		CAN1 Message Slot 5 Data Length Register (C1MSL5DLC)
H'0080 1556	CAN1 Message Slot 5 Data 0(C1MSL5DT0)		CAN1 Message Slot 5 Data 1(C1MSL5DT1)
H'0080 1558	CAN1 Message Slot 5 Data 2(C1MSL5DT2)		CAN1 Message Slot 5 Data 3(C1MSL5DT3)
H'0080 155A	CAN1 Message Slot 5 Data 4(C1MSL5DT4)		CAN1 Message Slot 5 Data 5(C1MSL5DT5)
H'0080 155C	CAN1 Message Slot 5 Data 6(C1MSL5DT6)		CAN1 Message Slot 5 Data 7(C1MSL5DT7)
H'0080 155E	CAN1 Message Slot 5 Timestamp (C1MSL5TSP)		
H'0080 1560	CAN1 Message Slot 6 Standard ID0(C1MSL6SID0)		CAN1 Message Slot 6 Standard ID1(C1MSL6SID1)
H'0080 1562	CAN1 Message Slot 6 Extended ID0(C1MSL6EID0)		CAN1 Message Slot 6 Extended ID1(C1MSL6EID1)
H'0080 1564	CAN1 Message Slot 6 Extended ID2(C1MSL6EID2)		CAN1 Message Slot 6 Data Length Register (C1MSL6DLC)
H'0080 1566	CAN1 Message Slot 6 Data 0(C1MSL6DT0)		CAN1 Message Slot 6 Data 1(C1MSL6DT1)
H'0080 1568	CAN1 Message Slot 6 Data 2(C1MSL6DT2)		CAN1 Message Slot 6 Data 3(C1MSL6DT3)
H'0080 156A	CAN1 Message Slot 6 Data 4(C1MSL6DT4)		CAN1 Message Slot 6 Data 5(C1MSL6DT5)
H'0080 156C	CAN1 Message Slot 6 Data 6(C1MSL6DT6)		CAN1 Message Slot 6 Data 7(C1MSL6DT7)
H'0080 156E	CAN1 Message Slot 6 Timestamp (C1MSL6TSP)		
H'0080 1570	CAN1 Message Slot 7 Standard ID0(C1MSL7SID0)		CAN1 Message Slot 7 Standard ID1(C1MSL7SID1)
H'0080 1572	CAN1 Message Slot 7 Extended ID0(C1MSL7EID0)		CAN1 Message Slot 7 Extended ID1(C1MSL7EID1)
H'0080 1574	CAN1 Message Slot 7 Extended ID2(C1MSL7EID2)		CAN1 Message Slot 7 Data Length Register (C1MSL7DLC)
H'0080 1576	CAN1 Message Slot 7 Data 0(C1MSL7DT0)		CAN1 Message Slot 7 Data 1(C1MSL7DT1)
H'0080 1578	CAN1 Message Slot 7 Data 2(C1MSL7DT2)		CAN1 Message Slot 7 Data 3(C1MSL7DT3)
H'0080 157A	CAN1 Message Slot 7 Data 4(C1MSL7DT4)		CAN1 Message Slot 7 Data 5(C1MSL7DT5)
H'0080 157C	CAN1 Message Slot 7 Data 6(C1MSL7DT6)		CAN1 Message Slot 7 Data 7(C1MSL7DT7)
H'0080 157E	CAN1 Message Slot 7 Timestamp (C1MSL7TSP)		
H'0080 1580	CAN1 Message Slot 8 Standard ID0(C1MSL8SID0)		CAN1 Message Slot 8 Standard ID1(C1MSL8SID1)
H'0080 1582	CAN1 Message Slot 8 Extended ID0(C1MSL8EID0)		CAN1 Message Slot 8 Extended ID1(C1MSL8EID1)
H'0080 1584	CAN1 Message Slot 8 Extended ID2(C1MSL8EID2)		CAN1 Message Slot 8 Data Length Register (C1MSL8DLC)
H'0080 1586	CAN1 Message Slot 8 Data 0(C1MSL8DT0)		CAN1 Message Slot 8 Data 1(C1MSL8DT1)
H'0080 1588	CAN1 Message Slot 8 Data 2(C1MSL8DT2)		CAN1 Message Slot 8 Data 3(C1MSL8DT3)
H'0080 158A	CAN1 Message Slot 8 Data 4(C1MSL8DT4)		CAN1 Message Slot 8 Data 5(C1MSL8DT5)
H'0080 158C	CAN1 Message Slot 8 Data 6(C1MSL8DT6)		CAN1 Message Slot 8 Data 7(C1MSL8DT7)
H'0080 158E	CAN1 Message Slot 8 Timestamp (C1MSL8TSP)		
H'0080 1590	CAN1 Message Slot 9 Standard ID0(C1MSL9SID0)		CAN1 Message Slot 9 Standard ID1(C1MSL9SID1)
H'0080 1592	CAN1 Message Slot 9 Extended ID0(C1MSL9EID0)		CAN1 Message Slot 9 Extended ID1(C1MSL9EID1)
H'0080 1594	CAN1 Message Slot 9 Extended ID2(C1MSL9EID2)		CAN1 Message Slot 9 Data Length Register (C1MSL9DLC)
H'0080 1596	CAN1 Message Slot 9 Data 0(C1MSL9DT0)		CAN1 Message Slot 9 Data 1(C1MSL9DT1)
H'0080 1598	CAN1 Message Slot 9 Data 2(C1MSL9DT2)		CAN1 Message Slot 9 Data 3(C1MSL9DT3)
H'0080 159A	CAN1 Message Slot 9 Data 4(C1MSL9DT4)		CAN1 Message Slot 9 Data 5(C1MSL9DT5)
H'0080 159C	CAN1 Message Slot 9 Data 6(C1MSL9DT6)		CAN1 Message Slot 9 Data 7(C1MSL9DT7)
H'0080 159E	CAN1 Message Slot 9 Timestamp (C1MSL9TSP)		
H'0080 15A0	CAN1 Message Slot 10 Standard ID0(C1MSL10SID0)		CAN1 Message Slot 10 Standard ID1(C1MSL10SID1)
H'0080 15A2	CAN1 Message Slot 10 Extended ID0(C1MSL10EID0)		CAN1 Message Slot 10 Extended ID1(C1MSL10EID1)
H'0080 15A4	CAN1 Message Slot 10 Extended ID2(C1MSL10EID2)		CAN1 Message Slot 10 Data Length Register (C1MSL10DLC)
H'0080 15A6	CAN1 Message Slot 10 Data 0(C1MSL10DT0)		CAN1 Message Slot 10 Data 1(C1MSL10DT1)

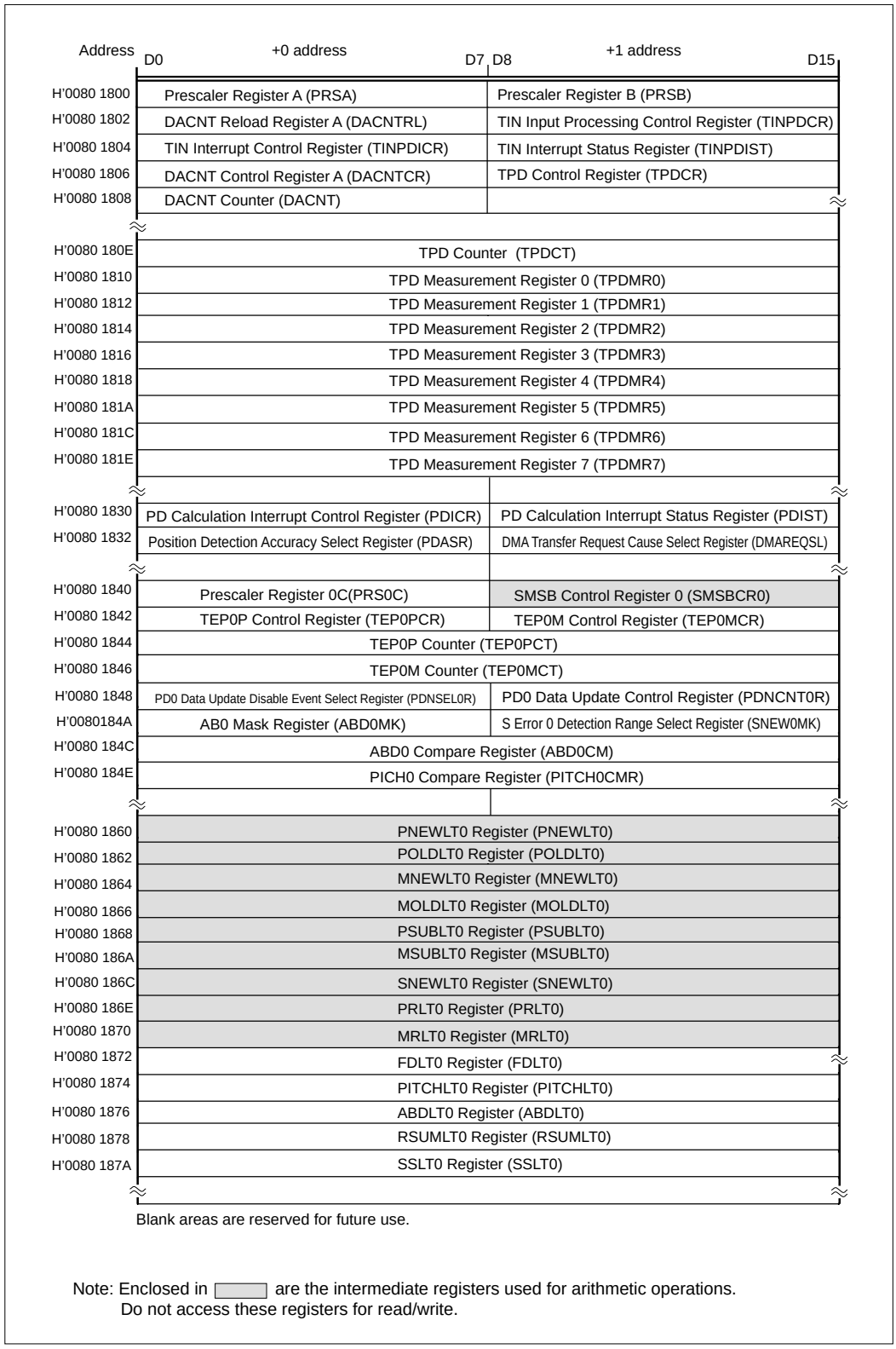
Blank areas are reserved for future use.

Figure 3.4.20 Register Mapping of the SFR Area (17)

Address	+0 address (D0 – D7)	+1 address (D8 – D15)
H'0080 15A8	CAN1 Message Slot 10 Data 2(C1MSL10DT2)	CAN1 Message Slot 10 Data 3(C1MSL10DT3)
H'0080 15AA	CAN1 Message Slot 10 Data 4(C1MSL10DT4)	CAN1 Message Slot 10 Data 5(C1MSL10DT5)
H'0080 15AC	CAN1 Message Slot 10 Data 6(C1MSL10DT6)	CAN1 Message Slot 10 Data 7(C1MSL10DT7)
H'0080 15AE	CAN1 Message Slot 10 Timestamp (C1MSL10TSP)	
H'0080 15B0	CAN1 Message Slot 11 Standard ID0(C1MSL11SID0)	CAN1 Message Slot 11 Standard ID1(C1MSL11SID1)
H'0080 15B2	CAN1 Message Slot 11 Extended ID0(C1MSL11EID0)	CAN1 Message Slot 11 Extended ID1(C1MSL11EID1)
H'0080 15B4	CAN1 Message Slot 11 Extended ID2(C1MSL11EID2)	CAN1 Message Slot 11 Data Length Register (C1MSL11DLC)
H'0080 15B6	CAN1 Message Slot 11 Data 0(C1MSL11DT0)	CAN1 Message Slot 11 Data 1(C1MSL11DT1)
H'0080 15B8	CAN1 Message Slot 11 Data 2(C1MSL11DT2)	CAN1 Message Slot 11 Data 3(C1MSL11DT3)
H'0080 15BA	CAN1 Message Slot 11 Data 4(C1MSL11DT4)	CAN1 Message Slot 11 Data 5(C1MSL11DT5)
H'0080 15BC	CAN1 Message Slot 11 Data 6(C1MSL11DT6)	CAN1 Message Slot 11 Data 7(C1MSL11DT7)
H'0080 15BE	CAN1 Message Slot 11 Timestamp (C1MSL11TSP)	
H'0080 15C0	CAN1 Message Slot 12 Standard ID0(C1MSL12SID0)	CAN1 Message Slot 12 Standard ID1(C1MSL12SID1)
H'0080 15C2	CAN1 Message Slot 12 Extended ID0(C1MSL12EID0)	CAN1 Message Slot 12 Extended ID1(C1MSL12EID1)
H'0080 15C4	CAN1 Message Slot 12 Extended ID2(C1MSL12EID2)	CAN1 Message Slot 12 Data Length Register (C1MSL12DLC)
H'0080 15C6	CAN1 Message Slot 12 Data 0(C1MSL12DT0)	CAN1 Message Slot 12 Data 1(C1MSL12DT1)
H'0080 15C8	CAN1 Message Slot 12 Data 2(C1MSL12DT2)	CAN1 Message Slot 12 Data 3(C1MSL12DT3)
H'0080 15CA	CAN1 Message Slot 12 Data 4(C1MSL12DT4)	CAN1 Message Slot 12 Data 5(C1MSL12DT5)
H'0080 15CC	CAN1 Message Slot 12 Data 6(C1MSL12DT6)	CAN1 Message Slot 12 Data 7(C1MSL12DT7)
H'0080 15CE	CAN1 Message Slot 12 Timestamp (C1MSL12TSP)	
H'0080 15D0	CAN1 Message Slot 13 Standard ID0(C1MSL13SID0)	CAN1 Message Slot 13 Standard ID1(C1MSL13SID1)
H'0080 15D2	CAN1 Message Slot 13 Extended ID0(C1MSL13EID0)	CAN1 Message Slot 13 Extended ID1(C1MSL13EID1)
H'0080 15D4	CAN1 Message Slot 13 Extended ID2(C1MSL13EID2)	CAN1 Message Slot 13 Data Length Register (C1MSL13DLC)
H'0080 15D6	CAN1 Message Slot 13 Data 0(C1MSL13DT0)	CAN1 Message Slot 13 Data 1(C1MSL13DT1)
H'0080 15D8	CAN1 Message Slot 13 Data 2(C1MSL13DT2)	CAN1 Message Slot 13 Data 3(C1MSL13DT3)
H'0080 15DA	CAN1 Message Slot 13 Data 4(C1MSL13DT4)	CAN1 Message Slot 13 Data 5(C1MSL13DT5)
H'0080 15DC	CAN1 Message Slot 13 Data 6(C1MSL13DT6)	CAN1 Message Slot 13 Data 7(C1MSL13DT7)
H'0080 15DE	CAN1 Message Slot 13 Timestamp (C1MSL13TSP)	
H'0080 15E0	CAN1 Message Slot 14 Standard ID0(C1MSL14SID0)	CAN1 Message Slot 14 Standard ID1(C1MSL14SID1)
H'0080 15E2	CAN1 Message Slot 14 Extended ID0(C1MSL14EID0)	CAN1 Message Slot 14 Extended ID1(C1MSL14EID1)
H'0080 15E4	CAN1 Message Slot 14 Extended ID2(C1MSL14EID2)	CAN1 Message Slot 14 Data Length Register (C1MSL14DLC)
H'0080 15E6	CAN1 Message Slot 14 Data 0(C1MSL14DT0)	CAN1 Message Slot 14 Data 1(C1MSL14DT1)
H'0080 15E8	CAN1 Message Slot 14 Data 2(C1MSL14DT2)	CAN1 Message Slot 14 Data 3(C1MSL14DT3)
H'0080 15EA	CAN1 Message Slot 14 Data 4(C1MSL14DT4)	CAN1 Message Slot 14 Data 5(C1MSL14DT5)
H'0080 15EC	CAN1 Message Slot 14 Data 6(C1MSL14DT6)	CAN1 Message Slot 14 Data 7(C1MSL14DT7)
H'0080 15EE	CAN1 Message Slot 14 Timestamp (C1MSL14TSP)	
H'0080 15F0	CAN1 Message Slot 15 Standard ID0(C1MSL15SID0)	CAN1 Message Slot 15 Standard ID1(C1MSL15SID1)
H'0080 15F2	CAN1 Message Slot 15 Extended ID0(C1MSL15EID0)	CAN1 Message Slot 15 Extended ID1(C1MSL15EID1)
H'0080 15F4	CAN1 Message Slot 15 Extended ID2(C1MSL15EID2)	CAN1 Message Slot 15 Data Length Register (C1MSL15DLC)
H'0080 15F6	CAN1 Message Slot 15 Data 0(C1MSL15DT0)	CAN1 Message Slot 15 Data 1(C1MSL15DT1)
H'0080 15F8	CAN1 Message Slot 15 Data 2(C1MSL15DT2)	CAN1 Message Slot 15 Data 3(C1MSL15DT3)
H'0080 15FA	CAN1 Message Slot 15 Data 4(C1MSL15DT4)	CAN1 Message Slot 15 Data 5(C1MSL15DT5)
H'0080 15FC	CAN1 Message Slot 15 Data 6(C1MSL15DT6)	CAN1 Message Slot 15 Data 7(C1MSL15DT7)
H'0080 15FE	CAN1 Message Slot 15 Timestamp (C1MSL15TSP)	
≈		≈

Blank areas are reserved for future use.

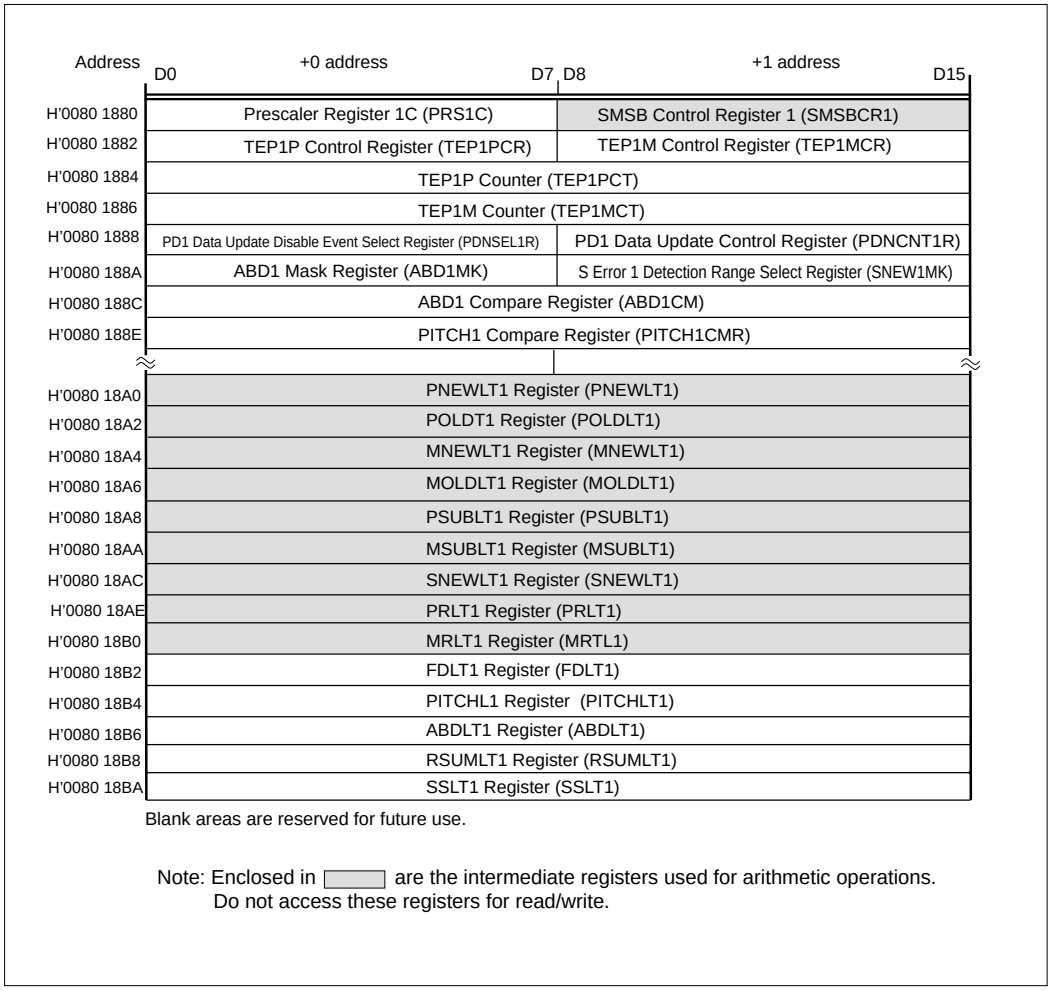
Figure 3.4.21 Register Mapping of the SFR Area (18)



Blank areas are reserved for future use.

Note: Enclosed in ▭ are the intermediate registers used for arithmetic operations.
Do not access these registers for read/write.

Figure 3.4.22 Register Mapping of the SFR Area (19)



Blank areas are reserved for future use.

Note: Enclosed in ▭ are the intermediate registers used for arithmetic operations.
Do not access these registers for read/write.

Figure 3.4.23 Register Mapping of the SFR Area (20)

Address	D0 +0 address D7	D8 +1 address D15
H'0080 1C78	D-A0 Conversion Register (DA0CNV)	
H'0080 1C7A	D-A1 Conversion Register (DA1CNV)	
H'0080 1C7C	D-A Conversion Register (DACR)	
H'0080 1D00	D-A0 Data Register 0 (DA0DT0)	D-A0 Data Register 1 (DA0DT1)
H'0080 1D02	D-A0 Data Register 2 (DA0DT2)	D-A0 Data Register 3 (DA0DT3)
H'0080 1D04	D-A0 Data Register 4 (DA0DT4)	D-A0 Data Register 5 (DA0DT5)
H'0080 1D06	D-A0 Data Register 6 (DA0DT6)	D-A0 Data Register 7 (DA0DT7)
H'0080 1D08	D-A0 Data Register 8 (DA0DT8)	D-A0 Data Register 9 (DA0DT9)
H'0080 1D0A	D-A0 Data Register 10 (DA0DT10)	D-A0 Data Register 11 (DA0DT11)
H'0080 1D0C	D-A0 Data Register 12 (DA0DT12)	D-A0 Data Register 13 (DA0DT13)
H'0080 1D0E	D-A0 Data Register 14 (DA0DT14)	D-A0 Data Register 15 (DA0DT15)
H'0080 1D10	D-A0 Data Register 16 (DA0DT16)	D-A0 Data Register 17 (DA0DT17)
H'0080 1D12	D-A0 Data Register 18 (DA0DT18)	D-A0 Data Register 19 (DA0DT19)
H'0080 1D14	D-A0 Data Register 20 (DA0DT20)	D-A0 Data Register 21 (DA0DT21)
H'0080 1D16	D-A0 Data Register 22 (DA0DT22)	D-A0 Data Register 23 (DA0DT23)
H'0080 1D18	D-A0 Data Register 24 (DA0DT24)	D-A0 Data Register 25 (DA0DT25)
H'0080 1D1A	D-A0 Data Register 26 (DA0DT26)	D-A0 Data Register 27 (DA0DT27)
H'0080 1D1C	D-A0 Data Register 28 (DA0DT28)	D-A0 Data Register 29 (DA0DT29)
H'0080 1D1E	D-A0 Data Register 30 (DA0DT30)	D-A0 Data Register 31 (DA0DT31)
H'0080 1D20	D-A0 Data Register 32 (DA0DT32)	D-A0 Data Register 33 (DA0DT33)
H'0080 1D22	D-A0 Data Register 34 (DA0DT34)	D-A0 Data Register 35 (DA0DT35)
H'0080 1D24	D-A0 Data Register 36 (DA0DT36)	D-A0 Data Register 37 (DA0DT37)
H'0080 1D26	D-A0 Data Register 38 (DA0DT38)	D-A0 Data Register 39 (DA0DT39)
H'0080 1D28	D-A0 Data Register 40 (DA0DT40)	D-A0 Data Register 41 (DA0DT41)
H'0080 1D2A	D-A0 Data Register 42 (DA0DT42)	D-A0 Data Register 43 (DA0DT43)
H'0080 1D2C	D-A0 Data Register 44 (DA0DT44)	D-A0 Data Register 45 (DA0DT45)
H'0080 1D2E	D-A0 Data Register 46 (DA0DT46)	D-A0 Data Register 47 (DA0DT47)
H'0080 1D30	D-A0 Data Register 48 (DA0DT48)	D-A0 Data Register 49 (DA0DT49)
H'0080 1D32	D-A0 Data Register 50 (DA0DT50)	D-A0 Data Register 51 (DA0DT51)
H'0080 1D34	D-A0 Data Register 52 (DA0DT52)	D-A0 Data Register 53 (DA0DT53)
H'0080 1D36	D-A0 Data Register 54 (DA0DT54)	D-A0 Data Register 55 (DA0DT55)
H'0080 1D38	D-A0 Data Register 56 (DA0DT56)	D-A0 Data Register 57 (DA0DT57)
H'0080 1D3A	D-A0 Data Register 58 (DA0DT58)	D-A0 Data Register 59 (DA0DT59)
H'0080 1D3C	D-A0 Data Register 60 (DA0DT60)	D-A0 Data Register 61 (DA0DT61)
H'0080 1D3E	D-A0 Data Register 62 (DA0DT62)	D-A0 Data Register 63 (DA0DT63)
H'0080 1D40	D-A0 Data Register 64 (DA0DT64)	D-A0 Data Register 65 (DA0DT65)
H'0080 1D42	D-A0 Data Register 66 (DA0DT66)	D-A0 Data Register 67 (DA0DT67)
H'0080 1D44	D-A0 Data Register 68 (DA0DT68)	D-A0 Data Register 69 (DA0DT69)
H'0080 1D46	D-A0 Data Register 70 (DA0DT70)	D-A0 Data Register 71 (DA0DT71)
H'0080 1D48	D-A0 Data Register 72 (DA0DT72)	D-A0 Data Register 73 (DA0DT73)
H'0080 1D4A	D-A0 Data Register 74 (DA0DT74)	D-A0 Data Register 75 (DA0DT75)
H'0080 1D4C	D-A0 Data Register 76 (DA0DT76)	D-A0 Data Register 77 (DA0DT77)
H'0080 1D4E	D-A0 Data Register 78 (DA0DT78)	D-A0 Data Register 79 (DA0DT79)

Blank areas are reserved for future use.

Figure 3.4.24 Register Mapping of the SFR Area (21)

Address	+0 address D0 – D7	+1 address D8 – D15
H'0080 1D50	D-A0 Data Register 80 (DA0DT80)	D-A0 Data Register 81 (DA0DT81)
H'0080 1D52	D-A0 Data Register 82 (DA0DT82)	D-A0 Data Register 83 (DA0DT83)
H'0080 1D54	D-A0 Data Register 84 (DA0DT84)	D-A0 Data Register 85 (DA0DT85)
H'0080 1D56	D-A0 Data Register 86 (DA0DT86)	D-A0 Data Register 87 (DA0DT87)
H'0080 1D58	D-A0 Data Register 88 (DA0DT88)	D-A0 Data Register 89 (DA0DT89)
H'0080 1D5A	D-A0 Data Register 90 (DA0DT90)	D-A0 Data Register 91 (DA0DT91)
H'0080 1D5C	D-A0 Data Register 92 (DA0DT92)	D-A0 Data Register 93 (DA0DT93)
H'0080 1D5E	D-A0 Data Register 94 (DA0DT94)	D-A0 Data Register 95 (DA0DT95)
H'0080 1D60	D-A0 Data Register 96 (DA0DT96)	D-A0 Data Register 97 (DA0DT97)
H'0080 1D62	D-A0 Data Register 98 (DA0DT98)	D-A0 Data Register 99 (DA0DT99)
H'0080 1D64	D-A0 Data Register 100 (DA0DT100)	D-A0 Data Register 101 (DA0DT101)
H'0080 1D66	D-A0 Data Register 102 (DA0DT102)	D-A0 Data Register 103 (DA0DT103)
H'0080 1D68	D-A0 Data Register 104 (DA0DT104)	D-A0 Data Register 105 (DA0DT105)
H'0080 1D6A	D-A0 Data Register 106 (DA0DT106)	D-A0 Data Register 107 (DA0DT107)
H'0080 1D6C	D-A0 Data Register 108 (DA0DT108)	D-A0 Data Register 109 (DA0DT109)
H'0080 1D6E	D-A0 Data Register 110 (DA0DT110)	D-A0 Data Register 111 (DA0DT111)
H'0080 1D70	D-A0 Data Register 112 (DA0DT112)	D-A0 Data Register 113 (DA0DT113)
H'0080 1D72	D-A0 Data Register 114 (DA0DT114)	D-A0 Data Register 115 (DA0DT115)
H'0080 1D74	D-A0 Data Register 116 (DA0DT116)	D-A0 Data Register 117 (DA0DT117)
H'0080 1D76	D-A0 Data Register 118 (DA0DT118)	D-A0 Data Register 119 (DA0DT119)
H'0080 1D78	D-A0 Data Register 120 (DA0DT120)	D-A0 Data Register 121 (DA0DT121)
H'0080 1D7A	D-A0 Data Register 122 (DA0DT122)	D-A0 Data Register 123 (DA0DT123)
H'0080 1D7C	D-A0 Data Register 124 (DA0DT124)	D-A0 Data Register 125 (DA0DT125)
H'0080 1D7E	D-A0 Data Register 126 (DA0DT126)	D-A0 Data Register 127 (DA0DT127)
H'0080 1D80	D-A0 Data Register 128 (DA0DT128)	D-A0 Data Register 129 (DA0DT129)
H'0080 1D82	D-A0 Data Register 130 (DA0DT130)	D-A0 Data Register 131 (DA0DT131)
H'0080 1D84	D-A0 Data Register 132 (DA0DT132)	D-A0 Data Register 133 (DA0DT133)
H'0080 1D86	D-A0 Data Register 134 (DA0DT134)	D-A0 Data Register 135 (DA0DT135)
H'0080 1D88	D-A0 Data Register 136 (DA0DT136)	D-A0 Data Register 137 (DA0DT137)
H'0080 1D8A	D-A0 Data Register 138 (DA0DT138)	D-A0 Data Register 139 (DA0DT139)
H'0080 1D8C	D-A0 Data Register 140 (DA0DT140)	D-A0 Data Register 141 (DA0DT141)
H'0080 1D8E	D-A0 Data Register 142 (DA0DT142)	D-A0 Data Register 143 (DA0DT143)
H'0080 1D90	D-A0 Data Register 144 (DA0DT144)	D-A0 Data Register 145 (DA0DT145)
H'0080 1D92	D-A0 Data Register 146 (DA0DT146)	D-A0 Data Register 147 (DA0DT147)
H'0080 1D94	D-A0 Data Register 148 (DA0DT148)	D-A0 Data Register 149 (DA0DT149)
H'0080 1D96	D-A0 Data Register 150 (DA0DT150)	D-A0 Data Register 151 (DA0DT151)
H'0080 1D98	D-A0 Data Register 152 (DA0DT152)	D-A0 Data Register 153 (DA0DT153)
H'0080 1D9A	D-A0 Data Register 154 (DA0DT154)	D-A0 Data Register 155 (DA0DT155)
H'0080 1D9C	D-A0 Data Register 156 (DA0DT156)	D-A0 Data Register 157 (DA0DT157)
H'0080 1D9E	D-A0 Data Register 158 (DA0DT158)	D-A0 Data Register 159 (DA0DT159)
H'0080 1DA0	D-A0 Data Register 160 (DA0DT160)	D-A0 Data Register 161 (DA0DT161)
H'0080 1DA2	D-A0 Data Register 162 (DA0DT162)	D-A0 Data Register 163 (DA0DT163)
H'0080 1DA4	D-A0 Data Register 164 (DA0DT164)	D-A0 Data Register 165 (DA0DT165)
H'0080 1DA6	D-A0 Data Register 166 (DA0DT166)	D-A0 Data Register 167 (DA0DT167)
H'0080 1DA8	D-A0 Data Register 168 (DA0DT168)	D-A0 Data Register 169 (DA0DT169)

Blank areas are reserved for future use.

Figure 3.4.25 Register Mapping of the SFR Area (22)

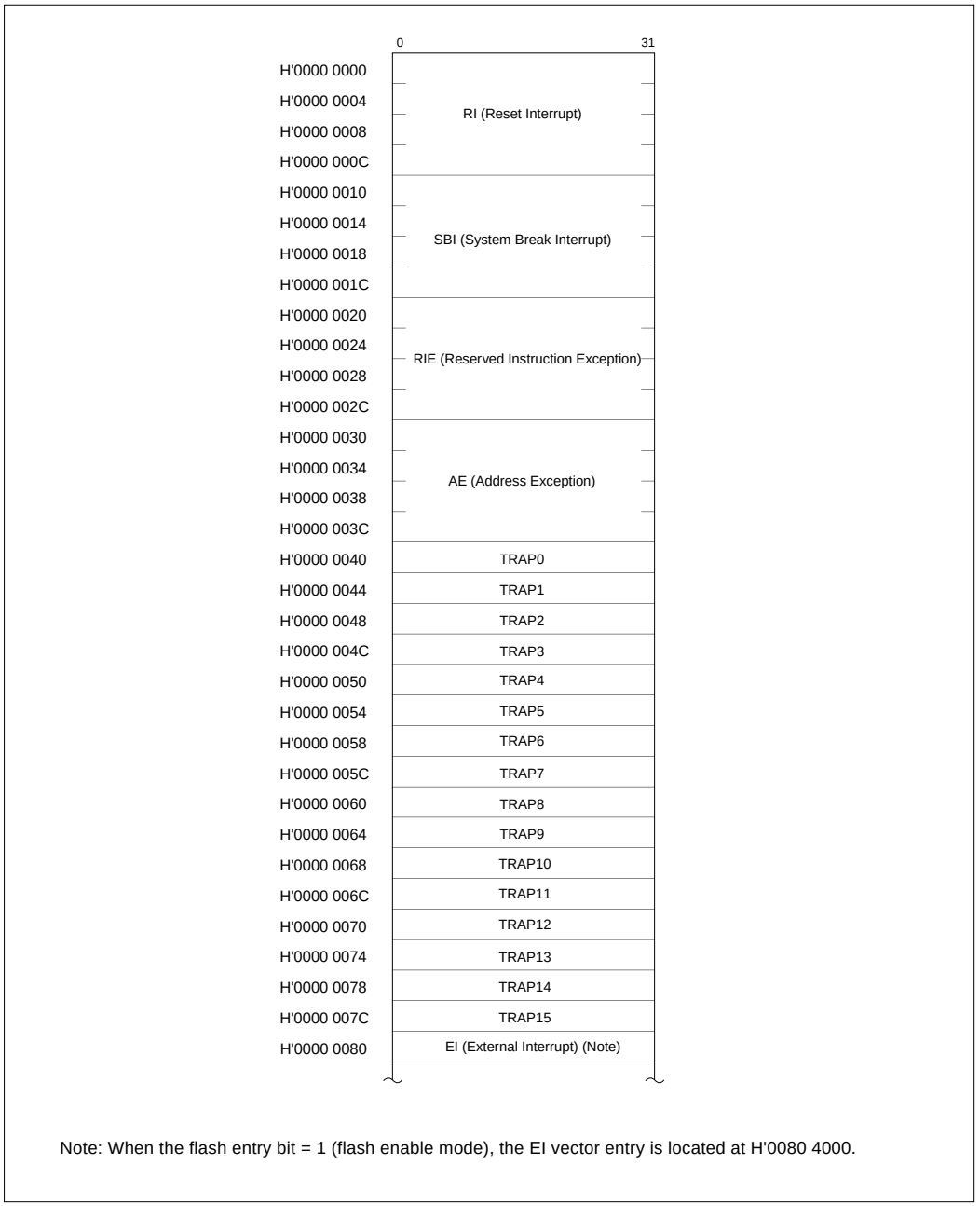
Address	+0 address (D0 – D7)	+1 address (D8 – D15)
H'0080 1DAA	D-A0 Data Register 170 (DA0DT170)	D-A0 Data Register 171 (DA0DT171)
H'0080 1DAC	D-A0 Data Register 172 (DA0DT172)	D-A0 Data Register 173 (DA0DT173)
H'0080 1DAE	D-A0 Data Register 174 (DA0DT174)	D-A0 Data Register 175 (DA0DT175)
H'0080 1DB0	D-A0 Data Register 176 (DA0DT176)	D-A0 Data Register 177 (DA0DT177)
H'0080 1DB2	D-A0 Data Register 178 (DA0DT178)	D-A0 Data Register 179 (DA0DT179)
H'0080 1DB4	D-A0 Data Register 180 (DA0DT180)	D-A0 Data Register 181 (DA0DT181)
H'0080 1DB6	D-A0 Data Register 182 (DA0DT182)	D-A0 Data Register 183 (DA0DT183)
H'0080 1DB8	D-A0 Data Register 184 (DA0DT184)	D-A0 Data Register 185 (DA0DT185)
H'0080 1DBA	D-A0 Data Register 186 (DA0DT186)	D-A0 Data Register 187 (DA0DT187)
H'0080 1DBC	D-A0 Data Register 188 (DA0DT188)	D-A0 Data Register 189 (DA0DT189)
H'0080 1DBE	D-A0 Data Register 190 (DA0DT190)	D-A0 Data Register 191 (DA0DT191)
H'0080 1DC0	D-A0 Data Register 192 (DA0DT192)	D-A0 Data Register 193 (DA0DT193)
H'0080 1DC2	D-A0 Data Register 194 (DA0DT194)	D-A0 Data Register 195 (DA0DT195)
H'0080 1DC4	D-A0 Data Register 196 (DA0DT196)	D-A0 Data Register 197 (DA0DT197)
H'0080 1DC6	D-A0 Data Register 198 (DA0DT198)	D-A0 Data Register 199 (DA0DT199)
H'0080 1DC8	D-A0 Data Register 200 (DA0DT200)	D-A0 Data Register 201 (DA0DT201)
H'0080 1DCA	D-A0 Data Register 202 (DA0DT202)	D-A0 Data Register 203 (DA0DT203)
H'0080 1DCC	D-A0 Data Register 204 (DA0DT204)	D-A0 Data Register 205 (DA0DT205)
H'0080 1DCE	D-A0 Data Register 206 (DA0DT206)	D-A0 Data Register 207 (DA0DT207)
H'0080 1DD0	D-A0 Data Register 208 (DA0DT208)	D-A0 Data Register 209 (DA0DT209)
H'0080 1DD2	D-A0 Data Register 210 (DA0DT210)	D-A0 Data Register 211 (DA0DT211)
H'0080 1DD4	D-A0 Data Register 212 (DA0DT212)	D-A0 Data Register 213 (DA0DT213)
H'0080 1DD6	D-A0 Data Register 214 (DA0DT214)	D-A0 Data Register 215 (DA0DT215)
H'0080 1DD8	D-A0 Data Register 216 (DA0DT216)	D-A0 Data Register 217 (DA0DT217)
H'0080 1DDA	D-A0 Data Register 218 (DA0DT218)	D-A0 Data Register 219 (DA0DT219)
H'0080 1DDC	D-A0 Data Register 220 (DA0DT220)	D-A0 Data Register 221 (DA0DT221)
H'0080 1DDE	D-A0 Data Register 222 (DA0DT222)	D-A0 Data Register 223 (DA0DT223)
H'0080 1DE0	D-A0 Data Register 224 (DA0DT224)	D-A0 Data Register 225 (DA0DT225)
H'0080 1DE2	D-A0 Data Register 226 (DA0DT226)	D-A0 Data Register 227 (DA0DT227)
H'0080 1DE4	D-A0 Data Register 228 (DA0DT228)	D-A0 Data Register 229 (DA0DT229)
H'0080 1DE6	D-A0 Data Register 230 (DA0DT230)	D-A0 Data Register 231 (DA0DT231)
H'0080 1DE8	D-A0 Data Register 232 (DA0DT232)	D-A0 Data Register 233 (DA0DT233)
H'0080 1DEA	D-A0 Data Register 234 (DA0DT234)	D-A0 Data Register 235 (DA0DT235)
H'0080 1DEC	D-A0 Data Register 236 (DA0DT236)	D-A0 Data Register 237 (DA0DT237)
H'0080 1DEE	D-A0 Data Register 238 (DA0DT238)	D-A0 Data Register 239 (DA0DT239)
H'0080 1DF0	D-A0 Data Register 240 (DA0DT240)	D-A0 Data Register 241 (DA0DT241)
H'0080 1DF2	D-A0 Data Register 242 (DA0DT242)	D-A0 Data Register 243 (DA0DT243)
H'0080 1DF4	D-A0 Data Register 244 (DA0DT244)	D-A0 Data Register 245 (DA0DT245)
H'0080 1DF6	D-A0 Data Register 246 (DA0DT246)	D-A0 Data Register 247 (DA0DT247)
H'0080 1DF8	D-A0 Data Register 248 (DA0DT248)	D-A0 Data Register 249 (DA0DT249)
H'0080 1DFA	D-A0 Data Register 250 (DA0DT250)	D-A0 Data Register 251 (DA0DT251)
H'0080 1DFC	D-A0 Data Register 252 (DA0DT252)	D-A0 Data Register 253 (DA0DT253)
H'0080 1DFE	D-A0 Data Register 254 (DA0DT254)	D-A0 Data Register 255 (DA0DT255)

Blank areas are reserved for future use.

Figure 3.4.26 Register Mapping of the SFR Area (23)

3.5 EIT Vector Entry

The EIT vector entry is located at the beginning of the internal ROM/extended external areas. Instructions for branching to the start addresses of respective EIT event handlers are written here. Note that it is branch instructions and not the jump addresses that are written here. For details, refer to Chapter 4, "EIT."



Note: When the flash entry bit = 1 (flash 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. The start addresses of interrupt handlers for interrupt requests from internal peripheral I/Os are set at the corresponding addresses of this table, as shown below. For details, refer to Chapter 5, "Interrupt Controller."

Figures 3.6.1 and 3.6.2 show a configuration of the ICU vector table.

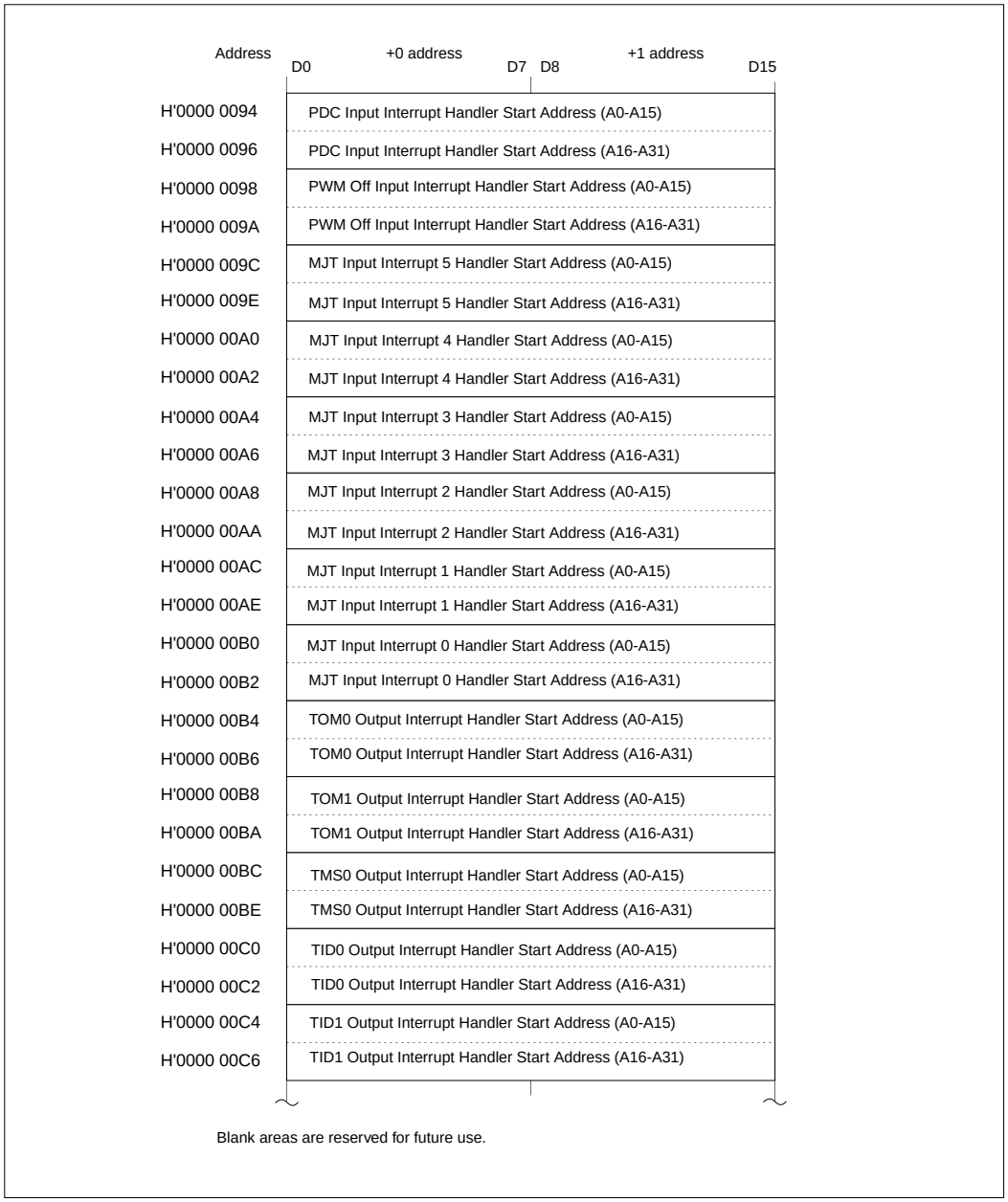


Figure 3.6.1 Configuration of the ICU Vector Table (1/2)

Address	+0 address D0 to D7	+1 address D8 to D15
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	A-D0 Conversion Interrupt Handler Start Address (A0-A15)	
H'0000 00CE	A-D0 Conversion Interrupt Handler Start Address (A16-A31)	
H'0000 00D0	SIO0 Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00D2	SIO0 Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00D4	SIO0 Transmit Interrupt Handler Start Address (A0-A15)	
H'0000 00D6	SIO0 Transmit Interrupt Handler Start Address (A16-A31)	
H'0000 00D8	SIO1 Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00DA	SIO1 Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00DC	SIO1 Transmit Interrupt Handler Start Address (A0-A15)	
H'0000 00DE	SIO1 Transmit Interrupt Handler Start Address (A16-A31)	
H'0000 00E0	A-D1 Conversion Interrupt Handler Start Address (A0-A15)	
H'0000 00E2	A-D1 Conversion Interrupt Handler Start Address (A16-A31)	
H'0000 00E4	DMA5-9 Interrupt Handler Start Address (A0-A15)	
H'0000 00E6	DMA5-9 Interrupt Handler Start Address (A16-A31)	
H'0000 00E8	SIO2,3 Transmit/Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00EA	SIO2,3 Transmit/Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00EC	SIO4 Transmit/Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00EE	SIO4 Transmit/Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00F0	SIO4 Transmit Interrupt Handler Start Address (A0-A15)	
H'0000 00F2	SIO4 Transmit Interrupt Handler Start Address (A16-A31)	
H'0000 00F4	SIO5 Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00F6	SIO5 Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00F8	SIO5 Transmit Interrupt Handler Start Address (A0-A15)	
H'0000 00FA	SIO5 Transmit Interrupt Handler Start Address (A16-A31)	
H'0000 00FC	SIO6,7 Transmit/Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00FE	SIO6,7 Transmit/Receive Interrupt Handler Start Address (A16-A31)	
H'0000 0100	RTD Interrupt Handler Start Address (A0-A15)	
H'0000 0102	RTD Interrupt Handler Start Address (A16-A31)	
H'0000 0104	PDC Compare Match & Error Interrupt Handler Start Address (A0-A15)	
H'0000 0106	PDC Compare Match & Error Interrupt Handler Start Address (A16-A31)	
H'0000 0108	CAN0 Transmit/Receive & Error Interrupt Handler Start Address (A0-A15)	
H'0000 010A	CAN0 Transmit/Receive & Error Interrupt Handler Start Address (A16-A31)	
H'0000 010C	CAN1 Transmit/Receive & Error Interrupt Handler Start Address (A0-A15)	
H'0000 010E	CAN1 Transmit/Receive & Error Interrupt Handler Start Address (A16-A31)	

Blank areas are reserved for future use.

Figure 3.6.2 Configuration of the ICU Vector Table (2/2)

3.7 Precautions on Address Space

- **Virtual-flash emulation function**

The 32172 has a function for mapping up to two 8-Kbyte blocks of the internal RAM beginning with the first address into the internal flash memory areas divided in units of 8 Kbytes (L banks). Similarly, the 32173 has a function for mapping up to three 8-Kbyte blocks of the internal RAM beginning with the first address into the internal flash memory areas divided in units of 8 Kbytes (L banks), as well as mapping up to two 4-Kbyte blocks of the internal RAM beginning with the H'0080 A000 area into the internal flash memory areas divided in units of 4 Kbytes (S banks) (the latter available for only the 32173). This is referred to as the virtual-flash emulation function. For details about this function, refer to Section 6.7, "Virtual-flash Emulation Function."

CHAPTER 4
EIT

4.1 Outline of EIT

If an event occurs while the CPU is executing an ordinary program, the CPU may have to suspend execution of the program and execute another program. Such an event is referred to by the generic name "EIT (Exception, Interrupt, Trap)."

(1) Exception

This event relates to the context being executed, and is generated by an error or a violation of rules in instruction execution. In the M32R/E, Address Exception (AE) and Reserved Instruction Exception (RIE) fall under the category of this type of event.

(2) Interrupt

This event occurs independently of the context being executed. It is generated by a signal sent by means of hardware from the outside. In the M32R/E, External Interrupt (EI), System Break Interrupt (SBI), and Reset Interrupt (RI) fall under the category of this type of event.

(3) Trap

This refers to a software interrupt, which is issued by executing the TRAP instruction. As in the case of system calls of the OS, this type of event is generated intentionally in a program by the programmer.

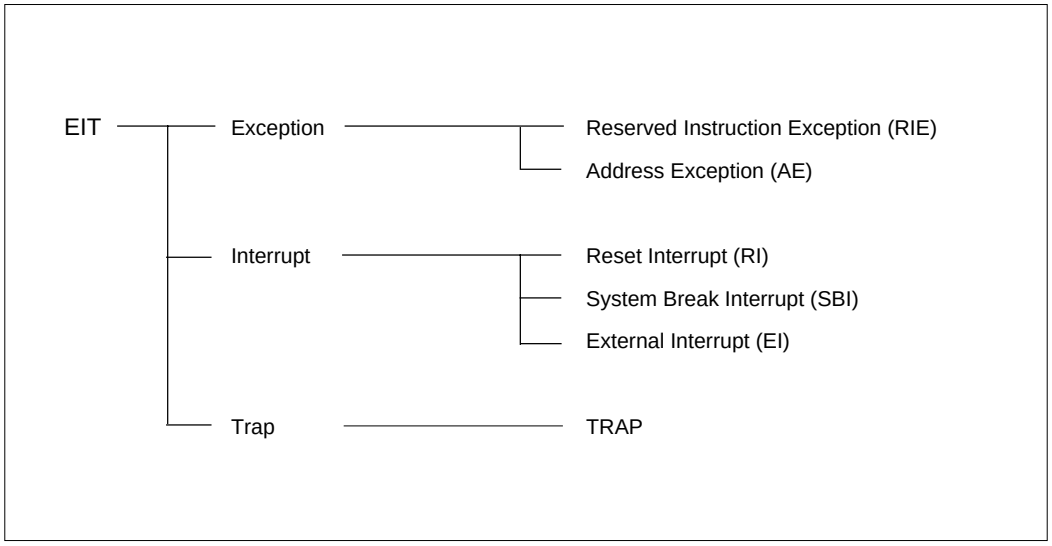


Figure 4.1.1 Classification of EIT

4.2 EIT Events

4.2.1 Exceptions

(1) Reserved Instruction Exception (RIE)

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

(2) Address Exception (AE)

An Address Exception (AE) occurs when access to an unaligned address is attempted in a Load or Store instruction.

4.2.2 Interrupts

(1) Reset Interrupt (RI)

A Reset Interrupt (RI) is accepted by asserting a $\overline{\text{RESET}}$ signal to the CPU. The Reset Interrupt has the highest priority.

(2) System Break Interrupt (SBI)

The System Break Interrupt (SBI) is an emergency interrupt which is issued when power outage is detected or a fault condition is notified from an external watchdog timer. This interrupt can be used only when after interrupt processing, the CPU does not as a rule return to the program it was executing when the interrupt occurred.

(3) External Interrupt (EI)

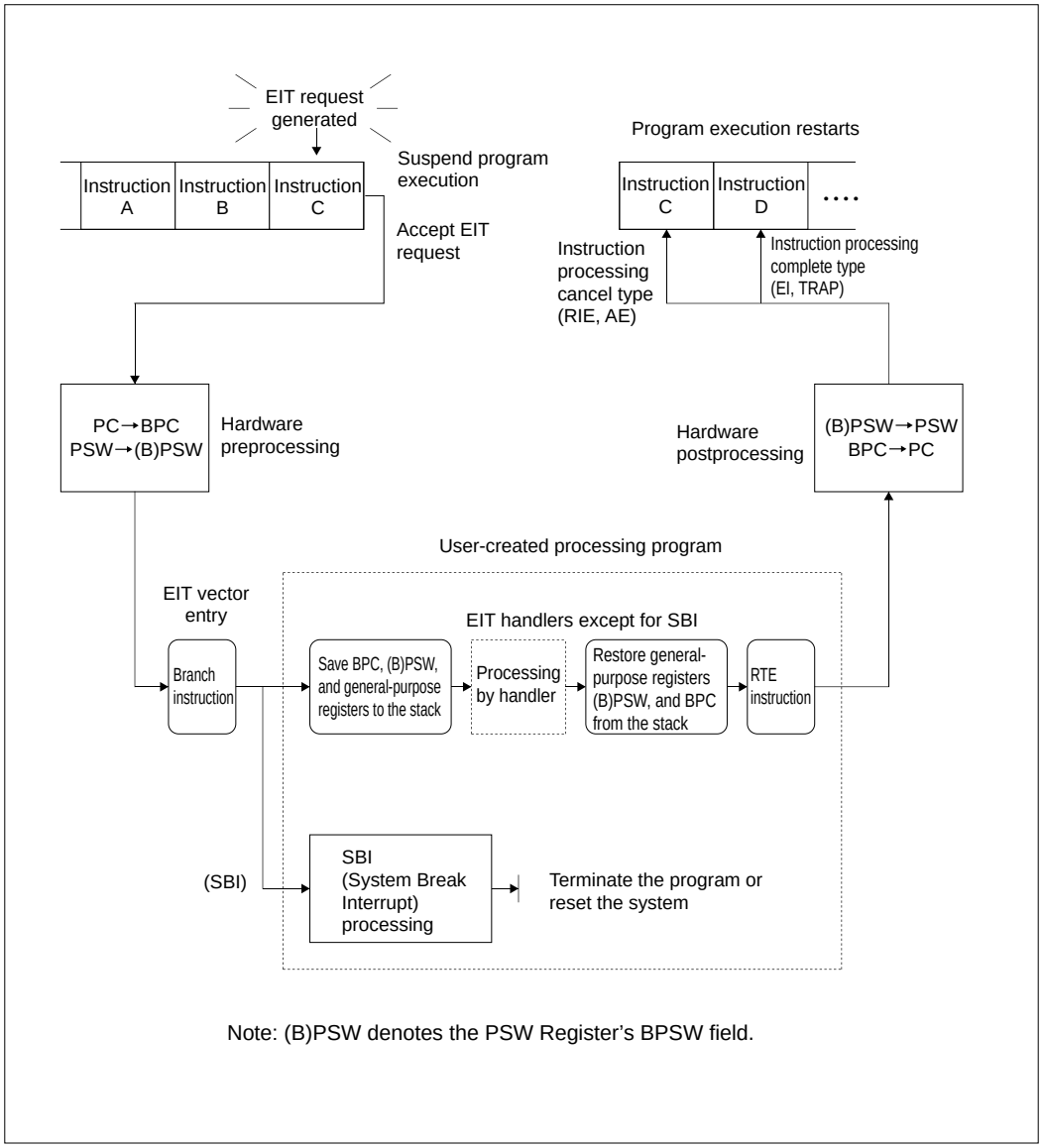
The External Interrupt (EI) is an interrupt request from one of the internal peripheral I/Os managed by the Interrupt Controller. The M32R's internal Interrupt Controller controls these interrupts by means of eight interrupt priority levels (including an interrupt-disabled state).

4.2.3 Trap

The Trap (TRAP) is a software interrupt, which is generated by executing the TRAP instruction. Sixteen vector addresses are provided, corresponding to operands 0-15 of the TRAP instruction.

4.3 EIT Processing Procedure

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



Note: (B)PSW denotes the PSW Register's BPSW field.

Figure 4.3.1 Outline of EIT Processing Procedure

When an EIT is accepted, the M32R/E saves the PC and PSW to the stack (described later) and branches to the EIT vector. The EIT vector has entry addresses assigned for each EIT. This is where the BRA (branch) instruction (note that these are not branch addresses) for the EIT handler is written.

In hardware preprocessing by the M32R/E, the contents of the PC and PSW Register are transferred to the backup registers (BPC Register and the PSW Register BPSW field). This is the only operation performed here.
Therefore, it is necessary to save to the stack in a user-created EIT handler the BPC Register, the PSW Register (including the BPSW field), and the general-purpose registers to be used in the EIT handler. (Remember that these registers must be saved to the stack in a program by the user.)

When processing by the EIT handler is completed, restore the registers from the stack to which they have been saved and then execute the RTE instruction to return from EIT processing to the original program (except for System Break Interrupt).
In hardware postprocessing by the M32R/E, the contents of the backup registers (BPC Register and the PSW Register BPSW field) are restored into the PC and PSW Registers.

4.4 EIT Processing Mechanism

The EIT processing mechanism of the M32R/E consists of the M32R CPU Core unit and the internal peripheral I/O Interrupt Controller. It also has backup registers for the PC and PSW (BPC Register and the BPSW fild of the PSW register). The M32R/E's internal EIT processing mechanism is shown below.

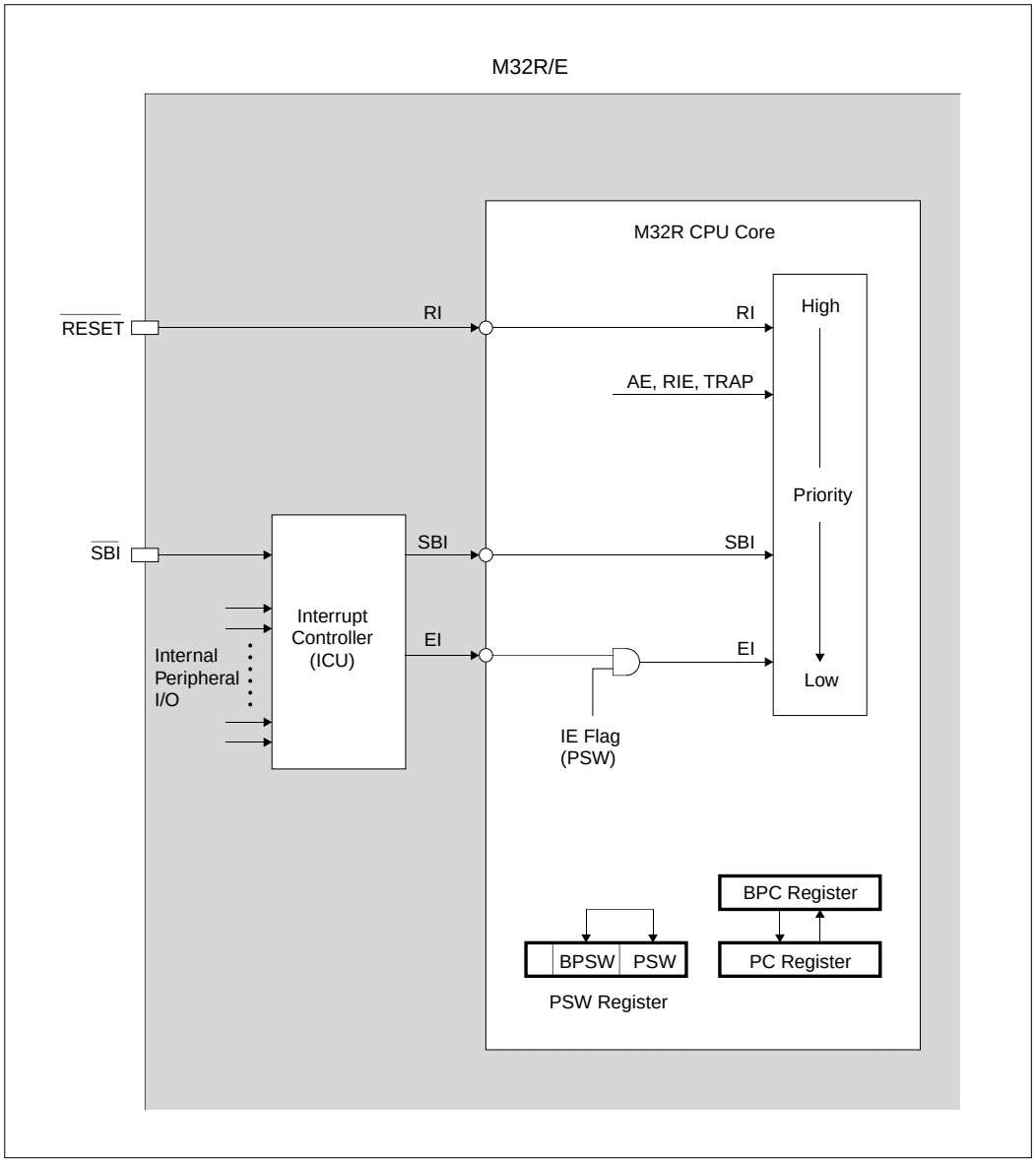


Figure 4.4.1 EIT Processing Mechanism of the M32R/E

4.5 Accepting EIT Events

When an EIT event occurs, the M32R/E suspends execution of the program being executed and branches to EIT handler processing. The table below shows occurrence conditions of each EIT event and the timing at which they are accepted.

Table 4.5.1 Accepting EIT Events

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

4.6 Saving and Restoring PC and PSW

The following describes the operations performed by the M32R when accepting an EIT and when executing the RTE instruction.

(1) Hardware preprocessing when accepting an EIT

(a) Save the PSW Register SM, IE, and C bits

BSM ← SM
BIE ← IE
BC ← C

(b) Update the PSW Register SM, IE, and C bits

SM ← Unchanged (RIE, AE, TRAP) or set to 0 (SBI, EI, RI)
IE ← Set to 0
C ← Set to 0

(c) Save the PC Register

BPC ← PC

(d) Set the vector address in the PC Register

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

(2) Hardware postprocessing when executing the RTE instruction

(e) Restore the PSW Register BSM, BIE, and BC bits

SM ← BSM
IE ← BIE
C ← BC

(f) Restore the BPC Register value to the PC Register

PC ← BPC

Note: After executing the RTE instruction, the values of the BPC Register and the PSW Register BSM, BIE, and BC bits are indeterminate.

(a) Save the SM, IE, and C bits

BSM ← SM
BIE ← IE
BC ← C

(b) Update the SM, IE, and C bits

SM ← Unchanged/0
IE ← 0
C ← 0

(c) Save the PC

BPC ← PC

(d) Set the vector address in the PC

PC ← Vector address

(e) Restore the BSM, BIE, and BC bits

SM ← BSM
IE ← BIE
C ← BC

After executing the RTE instruction, the values of the BSM, BIE, and BC bits are indeterminate.

(f) Restore the BPC value into the PC

After executing the RTE instruction, the value of the BPC is indeterminate.

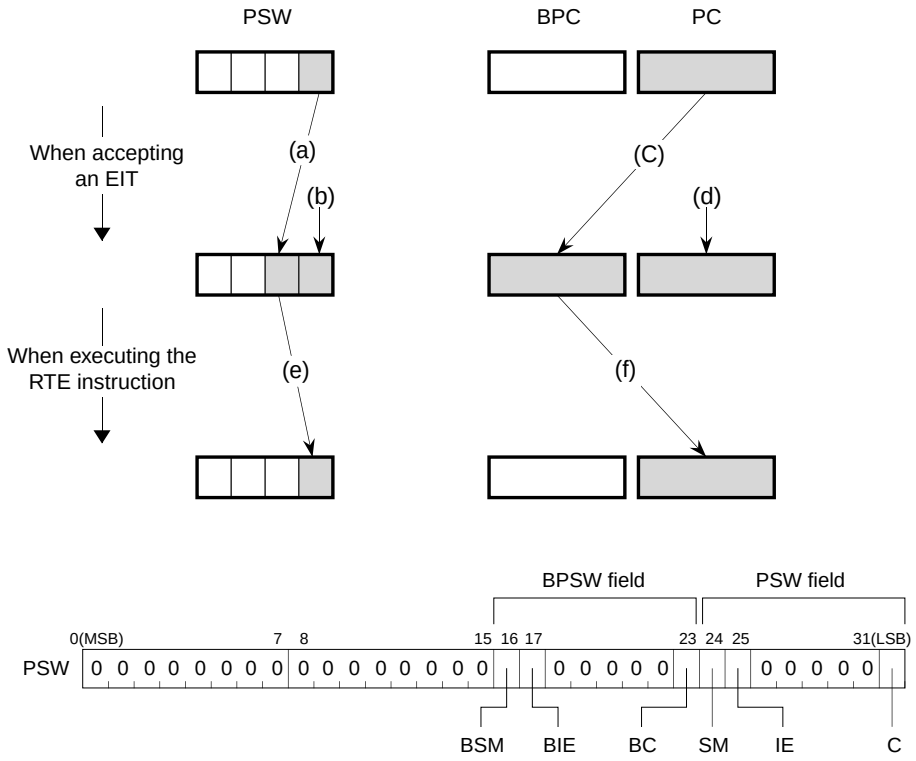


Figure 4.6.1 Saving and Restoring the PC and PSW

4.7 EIT Vector Entry

The EIT vector entry is placed in the user space beginning with address H'0000 0000. The EIT vector entry is listed below.

Table 4.7.1 EIT Vector Entry

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

Note 1: During boot mode, control jumps to the beginning of the boot ROM (address H'8000 0000). For details, refer to Section 6.5 "Programming the Internal Flash Memory."

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

4.8 Exception Handling

4.8.1 Reserved Instruction Exception (RIE)

[Occurrence condition]

A Reserved Instruction Exception (RIE) occurs when execution of a reserved instruction (an 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 caused the exception is not executed. Even if an external interrupt is requested when a Reserved Instruction Exception is detected, the Reserved Instruction Exception is accepted.

[EIT processing]

(1) Saving the SM, IE, and C bits

The PSW Register SM, IE, and C bits are saved to the backup bits-BSM, BIE, and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating the SM, IE, and C bits

The PSW Registmer SM, IE, and C bits are updated in the manner shown below.

SM ← Unchanged
IE ← 0
C ← 0

(3) Saving the PC

The PC value of the instruction that caused the Reserved Instruction Exception is set in the BPC Register. For example, if the instruction that caused the Reserved Instruction Exception is at address 4, then the value 4 is set in the BPC Register; if at address 6, then 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 caused the Reserved Instruction Exception exists on the word boundary (BPC[30] = 0), or not (BPC[30] = 1).
However, in either case of the above, the location to which the RTE instruction returns after the end of the EIT handler is always address 4. (This is because the two low-order bits are cleared to '00' when returning to the PC.)

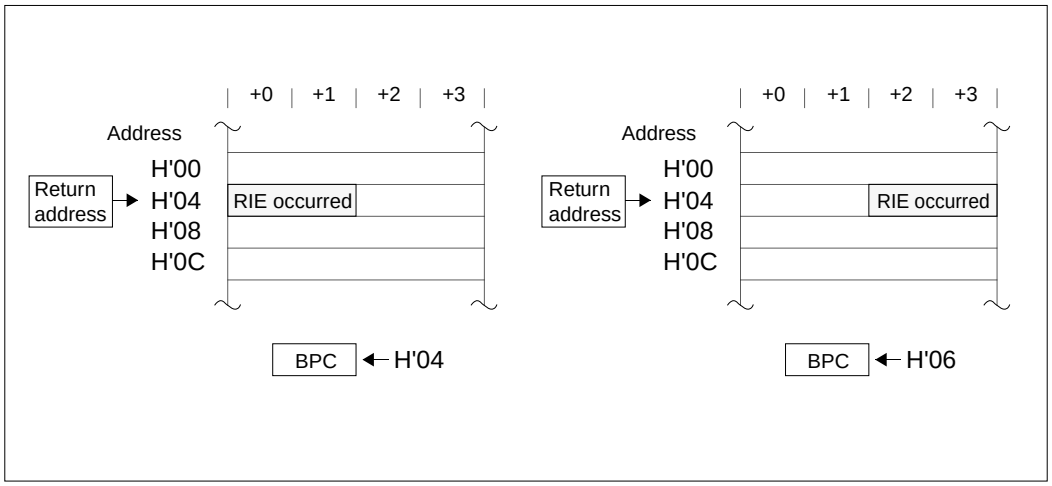


Figure 4.8.1 Example of a Return Address for Reserved Instruction Exception (RIE)

(4) Branching to the EIT vector entry

Control branches to a user space address H'0000 0020. This is the last operation in hardware preprocessing performed by the M32R/E.

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

The M32R/E executes the BRA instruction written at the EIT vector entry address H'0000 0020 by the user, thereby branching to the start address of the user-created handler. At the beginning of the user-created EIT handler, the BPC and PSW Registers and the necessary general-purpose registers must first be saved to the stack.

(6) Returning from the EIT handler

At the end of the EIT handler, restore the general-purpose registers and the BPC and PSW Registers from the stack and execute the RTE instruction. Then hardware postprocessing will be performed automatically. In this case, operation restarts from a word-boundary instruction including one that caused the RIE exception (see Figure 4.8.1).

4.8.2 Address Exception (AE)

[Occurrence condition]

An Address Exception (AE) occurs when access to an unaligned address is attempted in a Load or Store instruction. The following shows combinatorial instruction and address conditions under which an Address Exception is invoked.

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

When an Address Exception occurs, memory access by the instruction that caused the exception is not performed. Even if an external interrupt is requested when an Address Exception is detected, the Address Exception is accepted.

[EIT processing]

(1) Saving the SM, IE, and C bits

The PSW Register SM, IE, and C bits are saved to the backup bits-BSM, BIE, and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating the SM, IE, and C bits

The PSW Register SM, IE, and C bits are updated in the manner shown below.

SM ← Unchanged
IE ← 0
C ← 0

(3) Saving the PC

The PC value of the instruction that caused the Address Exception is set in the BPC Register. For example, if the instruction that caused the Address Exception is at address 4, then the value 4 is set in the BPC Register; if at address 6, then 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 caused the Address Exception exists on the word boundary (BPC[30] = 0), or not (BPC[30] = 1).
However, in either case of the above, the location to which the RTE instruction returns after the end of the EIT handler is always address 4. (This is because the two low-order bits are cleared to '00' when returning to the PC.)

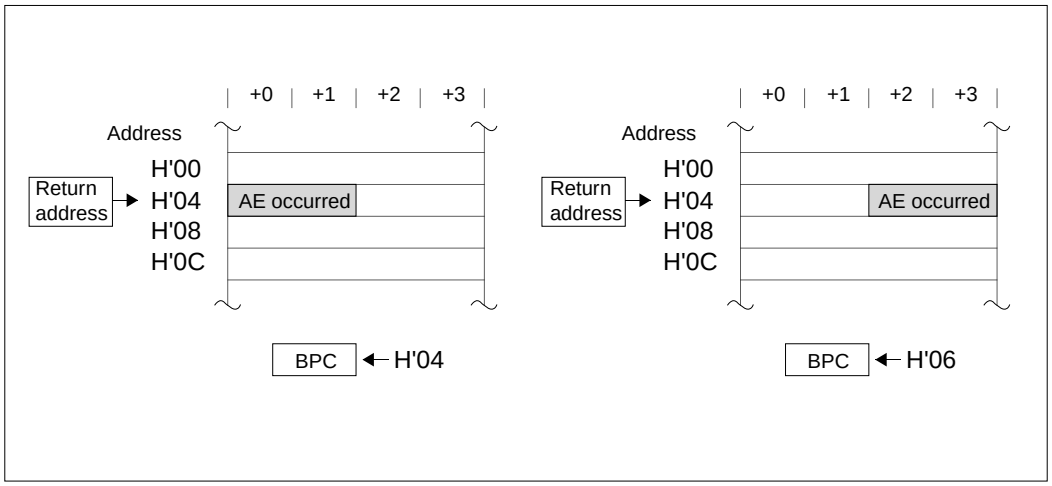


Figure 4.8.2 Example of a Return Address for Address Exception (AE)

(4) Branching to the EIT vector entry

Control branches to a user space address H'0000 0030. This is the last operation in hardware preprocessing performed by the M32R/E.

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

The M32R/E executes the BRA instruction written at the EIT vector entry address H'0000 0030 by the user, thereby branching to the start address of the user-created handler. At the beginning of the user-created EIT handler, the BPC and PSW Registers and the necessary general-purpose registers must first be saved to the stack.

(6) Returning from the EIT handler

At the end of the EIT handler, restore the general-purpose registers and the BPC and PSW Registers from the stack and execute the RTE instruction. Then hardware postprocessing will be performed automatically. In this case, operation restarts from a word-boundary instruction including one that caused the AE exception (see Figure 4.8.2).

4.9 Interrupt Handling

4.9.1 Reset Interrupt (RI)

[Occurrence condition]

When input on $\overline{\text{RESET}}$ pin is pulled low, a Reset Interrupt (RI) is accepted unconditionally in each machine cycle. The Reset Interrupt has the highest priority of all EIT.

[EIT processing]

(1) Initializing the SM, IE, and C bits

The PSW Register SM, IE, and C bits are initialized in the manner shown below.

SM ← 0
IE ← 0
C ← 0

For Reset Interrupt, the values of the BSM, BIE, and BC bits become indeterminate.

(2) Branching to the EIT vector entry

Control branches to a user space address H'0000 0000. During boot mode, however, control goes to the beginning of the boot ROM (address H'8000 0000). (For details, refer to Section 6.5, "Programming the Internal Flash Memory.")

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

The M32R/E executes the instruction written at the EIT vector entry address H'0000 0000 by the user. In the reset vector entry, initialize the PSW and SPI Registers and then branch to the start address of the user program.

4.9.2 System Break Interrupt (SBI)

The System Break Interrupt (SBI) is an emergency interrupt which is issued when power outage is detected or a fault condition is notified from an external watchdog timer. The System Break Interrupt cannot be masked with the PSW Register IE bit.
Therefore, the System Break Interrupt can only be used when some fatal event has already occurred in the system when the interrupt is detected. Also, this interrupt must be used on condition that after processing with the SBI handler, the control will not return to the program it was executing when the system break interrupt occurred.

[Occurrence condition]

A System Break Interrupt is accepted by a falling edge at the input of the $\overline{\text{SBI}}$ pin. (The System Break Interrupt cannot be masked with the PSW Register IE bit.)

In no case will a System Break Interrupt be activated immediately after executing any 16-bit instruction that begins from the word boundary. (However, this does not apply to 16-bit branch instructions, in which case the interrupt is accepted immediately after branching.)

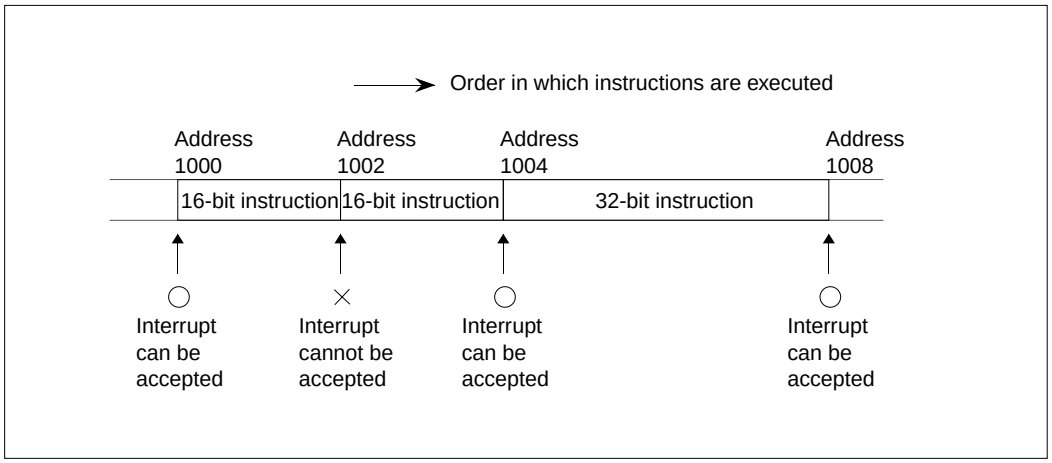


Figure 4.9.1 Timing at Which System Break Interrupt (SBI) is Accepted

[EIT processing]

(1) Saving the SM, IE, and C bits

The PSW Register SM, IE, and C bits are saved to the backup bits-BSM, BIE, and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating the SM, IE, and C bits

The PSW Register SM, IE, and C bits are updated in the manner shown below.

SM ← 0
IE ← 0
C ← 0

(3) Saving the PC

The contents of the PC Register (always on word boundary) are saved to the BPC Register.

(4) Branching to the EIT vector entry

Control branches to a user space address H'0000 0010. This is the last operation in hardware preprocessing performed by the M32R/E.

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

The M32R/E executes the BRA instruction written at the EIT vector entry address H'0000 0010 by the user, thereby branching to the start address of the user-created handler.

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

4.9.3 External Interrupt (EI)

An External Interrupt (EI) is generated based on interrupt requests output by the internal Interrupt Controller. The Interrupt Controller manages interrupt requests by means of 7-level interrupt priority. For details about the Interrupt Controller, refer to Chapter 5, "Interrupt Controller." For details about the causes of interrupts, refer to each relevant chapter where the internal peripheral I/O in interest is described.

[Occurrence condition]

External Interrupts are managed by the Interrupt Controller based on interrupt requests from each internal peripheral I/O, and are notified to the M32R CPU by the Interrupt Controller. The M32R/E checks for this request at instruction breaks on word boundaries. When an interrupt request from the Interrupt Controller is detected and the PSW Register IE flag = 1, an External Interrupt is accepted.
In no case will an External Interrupt be activated immediately after executing any 16-bit instruction that begins from the word boundary. (However, this does not apply to 16-bit branch instructions, in which case the interrupt is accepted immediately after branching.)

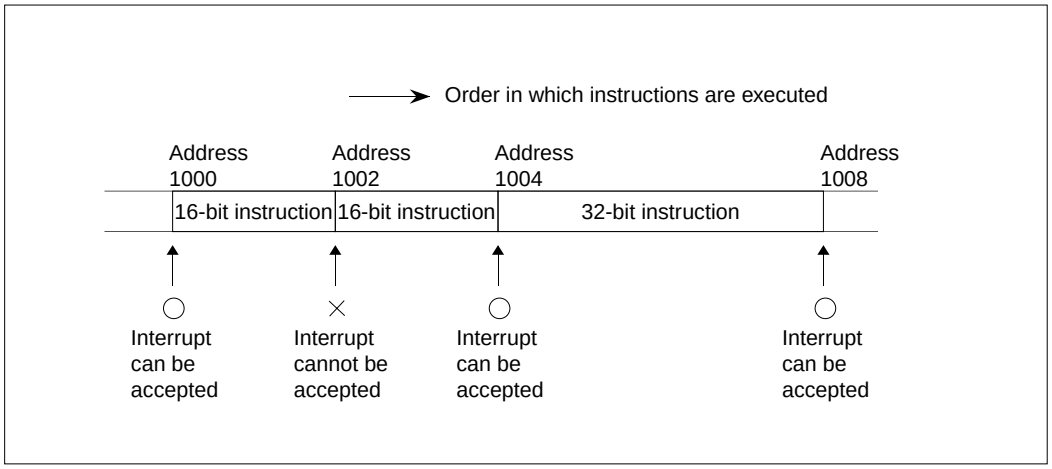


Figure 4.9.2 Timing at Which External Interrupt (EI) is Accepted

[EIT Processing]

(1) Saving the SM, IE, and C bits

The PSW Register SM, IE, and C bits are saved to the backup bits-BSM, BIE, and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating the SM, IE, and C bits

The PSW Register SM, IE, and C bits are updated in the manner shown below.

SM ← 0
IE ← 0
C ← 0

(3) Saving the PC

The contents of the PC Register (always on word boundary) are saved to the BPC Register.

(4) Branching to the EIT vector entry

Control branches to a user space address H'0000 0080. However, when in flash E/W enable mode, control jumps to the beginning of the internal RAM (address H'0080 4000). (For details, refer to Section 6.5, "Programming the Internal Flash Memory.") This is the last operation in hardware preprocessing performed by the M32R/E.

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

The M32R/E executes the BRA instruction written at the EIT vector entry address H'0000 0080 by the user, thereby branching to the start address of the user-created handler. At the beginning of the user-created EIT handler, the BPC and PSW Registers and the necessary general-purpose registers must first be saved to the stack.

(6) Returning from the EIT handler

At the end of the EIT handler, restore the general-purpose registers and the BPC and PSW Registers from the stack and execute the RTE instruction. Then hardware postprocessing will be performed automatically.

4.10 Trap Handling

4.10.1 Trap (TRAP)

[Occurrence condition]

The Trap (TRAP) is a software interrupt, which is generated by executing the TRAP instruction. Sixteen types of Traps are generated corresponding to operands 0-15 of the TRAP instruction. Accordingly, there are sixteen vector entry addresses, one for each type of Trap.

[EIT processing]

(1) Saving the SM, IE, and C bits

The PSW Register SM, IE, and C bits are saved to the backup bits-BSM, BIE, and BC.

BSM ← SM
BIE ← IE
BC ← C

(2) Updating the SM, IE, and C bits

The PSW Register SM, IE, and C bits are updated in the manner 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, then the value H'08 is set in the BPC Register; if 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 exists on the word boundary (BPC[30] = 0), or not (BPC[30] = 1).
However, in either case of the above, the location to which the RTE instruction returns after the end of the EIT handler is always address 8. (This is because the two low-order bits are cleared to '00' when returning to the PC.)

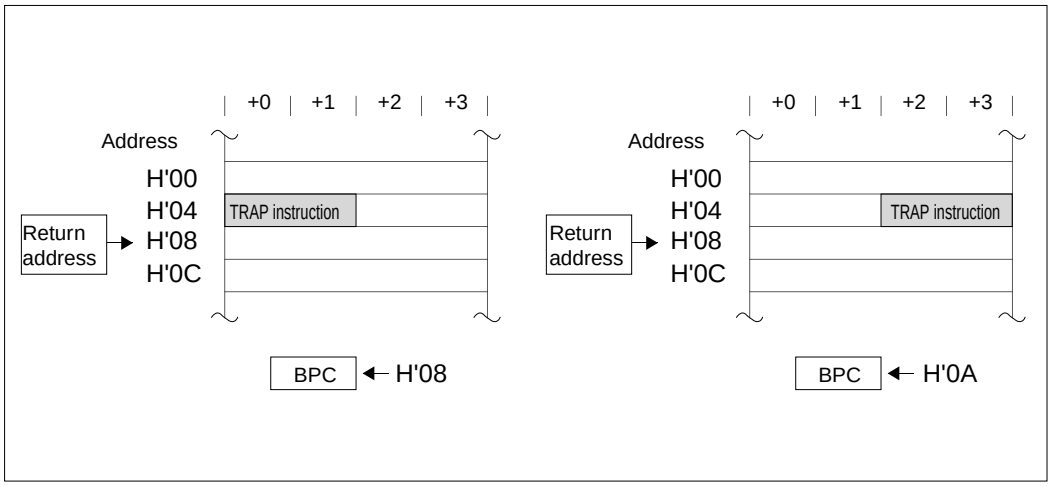


Figure 4.10.1 Example of a Return Address for Trap (TRAP)

(4) Branching to the EIT vector entry

Control branches to user space addresses H'0000 0040 through H'0000 007C. This is the last operation in hardware preprocessing performed by the M32R/E.

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

The M32R/E executes the BRA instruction written at the EIT vector entry addresses H'0000 0040 through H'0000 007C by the user, thereby branching to the start address of the user-created handler. At the beginning of the user-created EIT handler, the BPC and PSW Registers and the necessary general-purpose registers must first be saved to the stack.

(6) Returning from the EIT handler

At the end of the EIT handler, restore the general-purpose registers and the BPC and PSW Registers from the stack and execute the RTE instruction. After execution of the RTE instruction, hardware postprocessing is performed automatically.

4.11 EIT Priority

The priority of EIT events is shown below. If two or more EIT events occur at the same time, the EIT with the highest priority of all is accepted first.

Table 4.11.1 Priority of EIT Events and Manner of Return

Priority	EIT Event	Processing Type	Values Set in BPC Register
1 (highest priority)	Reset Interrupt (RI)	Instruction processing abandon type	Indeterminate
2	Address Exception (AE)	Instruction processing cancel type	PC of the instruction that generated EIT
	Reserved Instruction Exception (RIE)	Instruction processing cancel type	PC of the instruction that generated EIT
	Trap (TRAP)	Instruction processing complete type	TRAP instruction + 4
3	System Break Interrupt (SBI)	Instruction processing complete type	PC of the next instruction
4	External Interrupt (EI)	Instruction processing complete type	PC of the next instruction

For External Interrupt (EI), the priority of each interrupt request from peripheral I/O is set by the M32R's internal Interrupt Controller. For details, refer to Chapter 5, "Interrupt Controller."

4.12 Example of EIT Processing

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

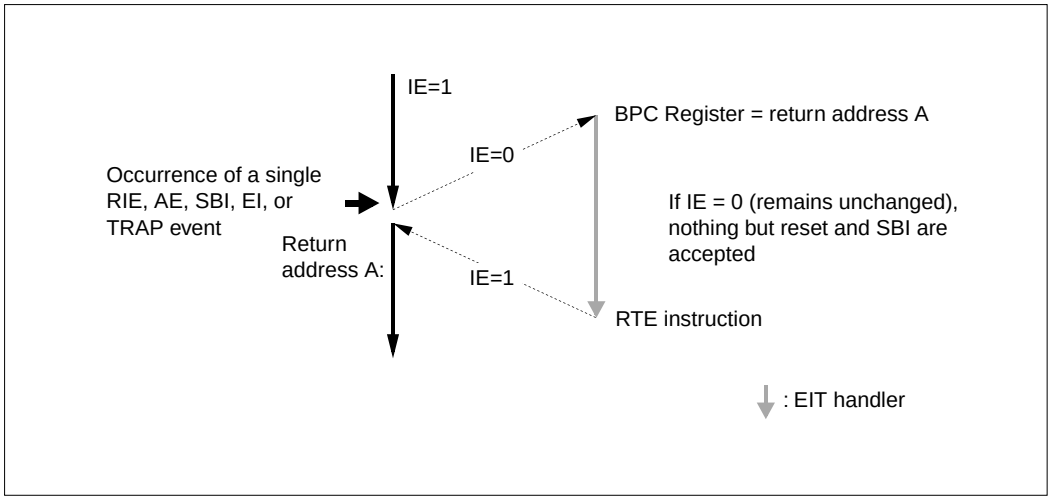


Figure 4.12.1 Processing of a Single Event of RIE, AE, SBI, EI, or TRAP

(2) When an RIE, AE, or TRAP occurs simultaneously with EI

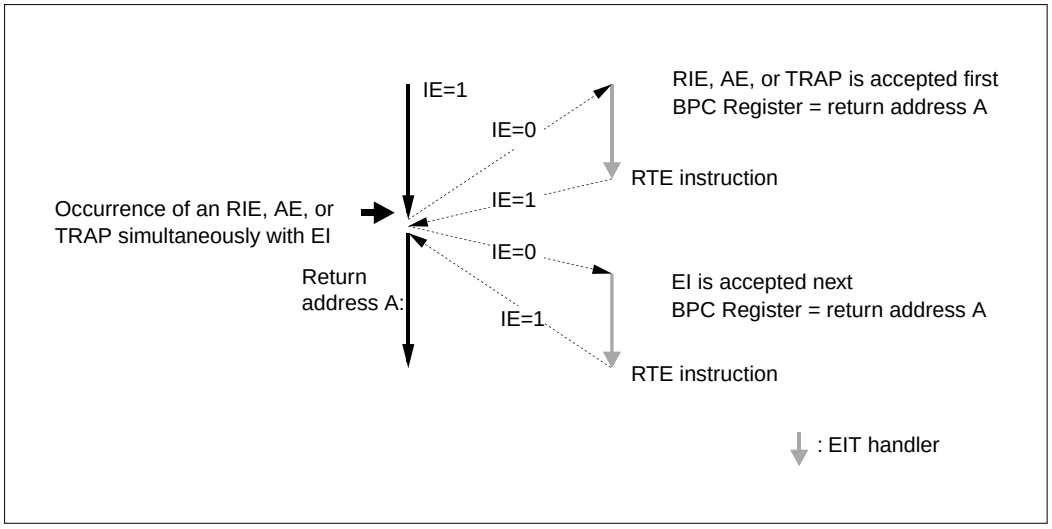


Figure 4.12.2 Processing of RIE, AE, or TRAP and EI when They Occurred Simultaneously

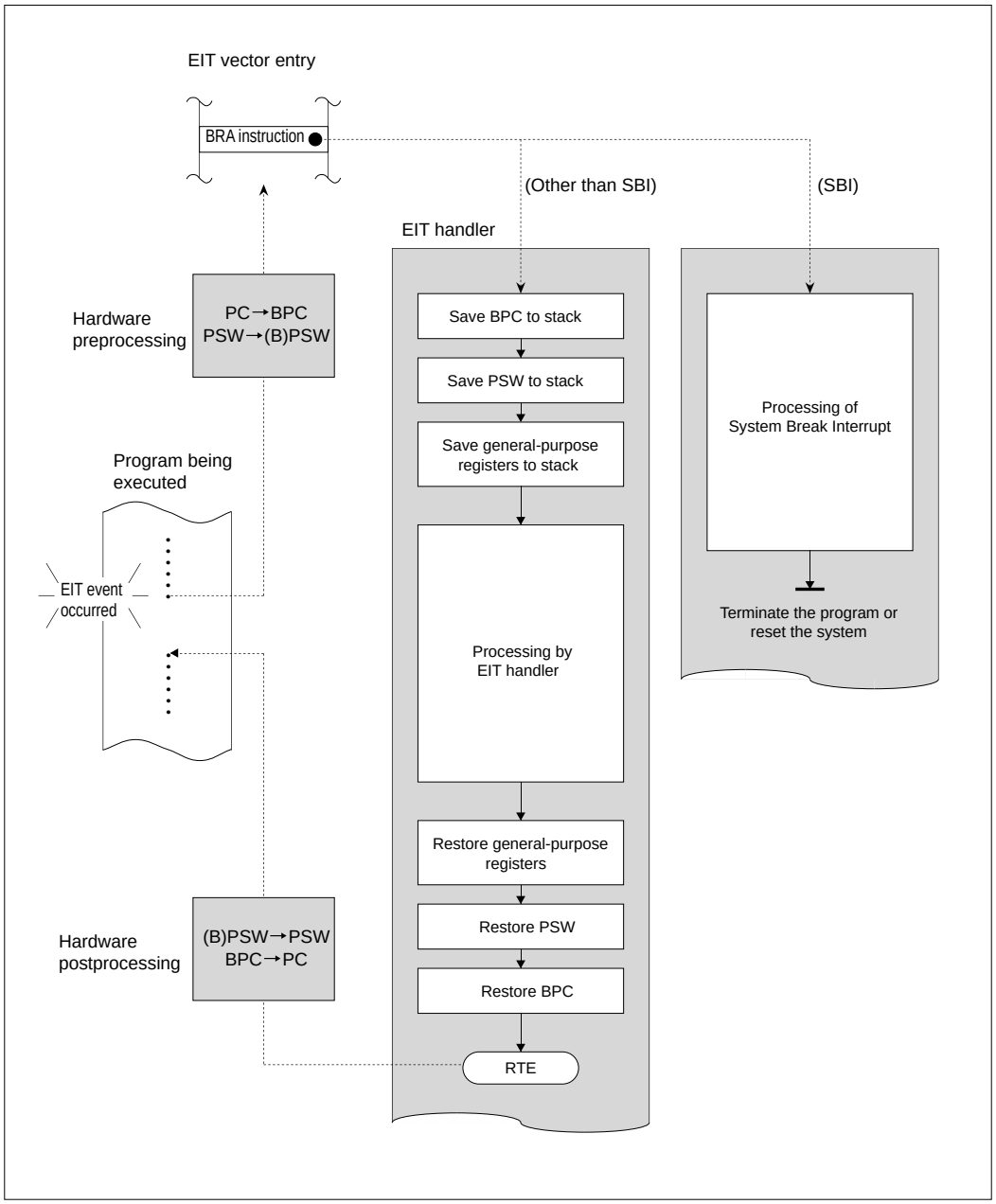


Figure 4.12.3 Example of EIT Processing

4.13 Precautions on EIT

The Address Exception (AE) requires caution because when an Address Exception occurs pursuant to execution of an instruction (one of the following three) that uses the "register indirect + register update" addressing mode, the value of the automatically updated register (Rsrc or Rsrc2) becomes indeterminate.
Except that the values of Rsrc and Rsrc2 are indeterminate, the behavior is the same as when using other addressing modes.

- **Applicable instructions**

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

If the above applies, take into account the fact that the register value becomes indeterminate when programming the system processing to be performed after occurrence of the exception. (If an Address Exception occurs, it means that some fatal fault already occurred in the system at that point in time. Therefore, use the Address Exception on condition that after processing by the Address Exception handler, the CPU will not return to the program it was executing when the exception occurred.)

*** This is a blank page ***

CHAPTER 5

INTERRUPT CONTROLLER (ICU)

5.1 Outline of the Interrupt Controller (ICU)

The Interrupt Controller (ICU) controls maskable interrupts from internal peripheral I/Os and System Break Interrupt (SBI). The maskable interrupts from internal peripheral I/Os are notified to the M32R CPU as External Interrupts (EI).

There is a total of 31 sources for the maskable interrupts from internal peripheral I/Os, which are controlled by assigning them one of eight priority levels including an interrupt-disabled state. If multiple interrupt requests with the same priority level occur simultaneously, they are prioritized according to the fixed hardware priority. Which sources of internal peripheral I/Os generated the interrupt requests are identified by reading the Interrupt Status Register of the internal peripheral I/O.

On the other hand, the System Break Interrupt (SBI) is an interrupt generated by a falling edge of the $\overline{\text{SBI}}$ input signal. This interrupt is always accepted regardless of the status of the PSW Register IE bit, and is used as an emergency interrupt which is issued when power failure is detected or a fault condition is notified from an external watchdog timer. After processing for the SBI is finished, terminate or reset the system without returning to the program that was executing when the interrupt occurred.

The Interrupt Controller is outlined below.

Table 5.1.1 Outline of the Interrupt Controller (ICU)

Item	Specification	
Interrupt source	Maskable interrupts from internal peripheral I/Os	: 31 sources
	System Break Interrupt	: 1 source (entered from the $\overline{\text{SBI}}$ pin)
Priority level	8 levels including interrupt-disabled state	
	(However, interrupts with the same priority level are prioritized by the fixed hardware priority.)	

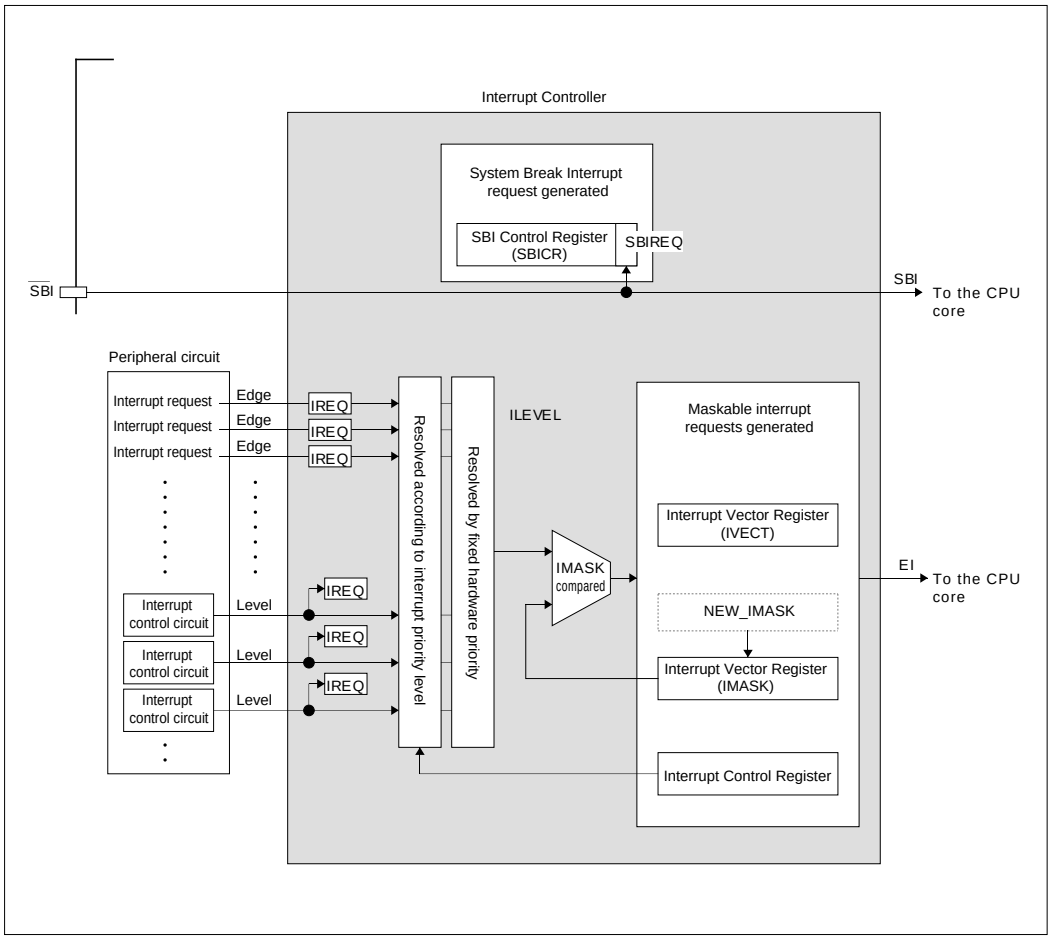


Figure 5.1.1 Block Diagram of the Interrupt Controller

5.2 Interrupt Sources of Internal Peripheral I/Os

The Interrupt Controller accepts as its input the interrupt requests from the timers, DMA, serial I/O, A-D converter, RTD, CAN, and PD controller. For details about these interrupts, refer to each relevant chapter where the internal peripheral I/O in interest is described.

Table 5.2.1 Interrupt Sources of Internal Peripheral I/Os (1/2)

Interrupt Source	Content	No. of Input Sources	Type of Input Source (Note)
PDC input & error detection interrupt	PDC input 0-3 (TINA0, TINA1, TINB0, or TINB1) detected, PDC input 0-3 error detected	8	Level
PWM-off input interrupt	PWM off input 0,1 interrupt	2	Level
A-D0 converter interrupt	A-D0 converter scan mode single shot, single mode, or comparator mode finished	1	Edge
A-D1 converter interrupt	A-D1 converter scan mode single shot, single mode, or comparator mode finished	1	Edge
SIO0 transmit interrupt	SIO0 transmit buffer empty interrupt	1	Edge
SIO0 receive interrupt	SIO0 reception finished or receive error interrupt	1	Edge
SIO1 transmit interrupt	SIO1 transmit buffer empty interrupt	1	Edge
SIO1 receive interrupt	SIO1 reception finished or receive error interrupt	1	Edge
SIO2,3 transmit/receive interrupt	SIO2,3 reception finished or receive error interrupt, transmit buffer empty interrupt	2	Level
SIO4 transmit interrupt	SIO4 transmit buffer empty interrupt	1	Edge
SIO4 receive interrupt	SIO4 reception finished or receive error interrupt	1	Edge
SIO5 transmit interrupt	SIO5 transmit buffer empty interrupt	1	Edge
SIO5 receive interrupt	SIO5 reception finished or receive error interrupt	1	Edge
SIO6,7 transmit/receive interrupt	SIO6,7 reception finished or receive error interrupt, transmit buffer empty interrupt	2	Level
RTD interrupt	RTD interrupt generating command	1	Edge
DMA transfer interrupt 0	DMA0-4 transfer finished	5	Level
DMA transfer interrupt 1	DMA5-9 transfer finished	5	Level
PDC compare match & error interrupt	PDC compare match, S error interrupt	8	Level
CAN0 transmit/receive & error interrupt	CAN0 transmission finished, CAN0 reception finished, CAN0 error passive, CAN0 error bus off, CAN0 bus error	19	Level
CAN1 transmit/receive & error interrupt	CAN1 transmission finished, CAN1 reception finished, CAN1 error passive, CAN1 error bus off, CAN1 bus error	19	Level

Note: Type of input source

- Edge: An interrupt request is generated by a falling edge on the interrupt signal fed into the ICU.
- Level: An interrupt request is generated while the interrupt signal fed into the ICU is held low. Level-type interrupt requests cannot be set or cleared by writing to the ICU's Interrupt Control Register IRQ bit in software.

Table 5.2.2 Interrupt Sources of Internal Peripheral I/Os (2/2)

Interrupt Source	Content	No. of Input Sources	Type of Input Source (Note)
TOM0 output interrupt	TOM0_0 to TOM0_7 output	8	Level
TOM1 output interrupt	TOM1_0 to TOM1_7 output	8	Level
TMS0 output interrupt	TMS0 output	1	Edge
TID0 output interrupt	TID0 output	1	Edge
TID1 output interrupt	TID1 output	1	Edge
Timer input interrupt 5	Timer input interrupt group 5 (TIN8, TIN9 input)	2	Level
Timer input interrupt 4	Timer input interrupt group 4 (TIN10, TIN11 input)	2	Level
Timer input interrupt 3	Timer input interrupt group 3 (TIN20, TIN21 input)	3	Level
Timer input interrupt 2	Timer input interrupt group 2 (TIN22, TIN23 input)	3	Level
Timer input interrupt 1	Timer input interrupt group 1 (TIN16, TIN17 input)	3	Level
Timer input interrupt 0	Timer input interrupt group 0 (TIN18, TIN19 input)	3	Level

Note: Type of input source

- Edge: An interrupt request is generated by a falling edge on the interrupt signal fed into the ICU.
- Level: An interrupt request is generated while the interrupt signal fed into the ICU is held low. Level-type interrupt requests cannot be set or cleared by writing to the ICU's Interrupt Control Register IRQ bit in software.

5.3 ICU Related Registers

A register map associated with the Interrupt Controller (ICU) is shown below.

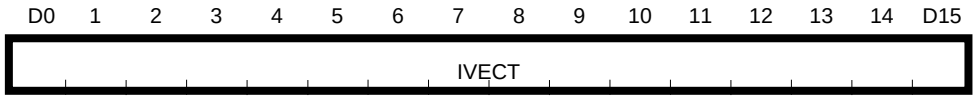
Address	+0 address D0 – D7	+1 address D8 – D15
H'0080 0000	Interrupt Vector Register (IVECT)	
H'0080 0002		
H'0080 0004	Interrupt Mask Register (IMASK)	
H'0080 0006	SBI Control Register (SBICR)	
H'0080 0060	CAN1 Transmit/Receive & Error Interrupt Control Register (ICAN1CR)	CAN0 Transmit/Receive & Error Interrupt Control Register (ICAN0CR)
H'0080 0062	PDC Compare Match & Error Interrupt Control Register (IPDCOPCR)	RTD Interrupt Control Register (IRTDCR)
H'0080 0064	SIO6,7 Transmit/Receive Interrupt Control Register (ISIO67CR)	SIO5 Transmit Interrupt Control Register (ISIO5TXCR)
H'0080 0066	SIO5 Receive Interrupt Control Register (ISIO5RXCR)	SIO4 Transmit Interrupt Control Register (ISIO4TXCR)
H'0080 0068	SIO4 Receive Interrupt Control Register (ISIO4RXCR)	SIO2,3 Transmit/Receive Interrupt Control Register (ISIO23CR)
H'0080 006A	DMA5-9 Interrupt Control Register (IDMA59CR)	A-D1 Conversion Interrupt Control Register (IAD1CCR)
H'0080 006C	SIO1 Transmit Interrupt Control Register (ISIO1TXCR)	SIO1 Receive Interrupt Control Register (ISIO1RXCR)
H'0080 006E	SIO0 Transmit Interrupt Control Register (ISIO0TXCR)	SIO0 Receive Interrupt Control Register (ISIO0RXCR)
H'0080 0070	A-D0 Conversion Interrupt Control Register (IAD0CCR)	DMA0-4 Interrupt Control Register (IDMA04CR)
H'0080 0072	TID1 Output Interrupt Control Register (ITID1CR)	TID0 Output Interrupt Control Register (ITID0CR)
H'0080 0074	TMS0 Output Interrupt Control Register (ITMS0CR)	TOM1 Output Interrupt Control Register (ITOM1CR)
H'0080 0076	TOM0 Output Interrupt Control Register (ITOM0CR)	Timer Input Interrupt Control Register 0 (IMJTICR0)
H'0080 0078	Timer Input Interrupt Control Register 1 (IMJTICR1)	Timer Input Interrupt Control Register 2 (IMJTICR2)
H'0080 007A	Timer Input Interrupt Control Register 3 (IMJTICR3)	Timer Input Interrupt Control Register 4 (IMJTICR4)
H'0080 007C	Timer Input Interrupt Control Register 5 (IMJTICR5)	PWM Off Input Interrupt Control Register (IPWMOFFCR)
H'0080 007E	PDC Input & Error Detection Interrupt Control Register (IPDCCR)	

Blank areas are reserved for future use.

Figure 5.3.1 Interrupt Controller (ICU) Related Register Map

5.3.1 Interrupt Vector Register

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



<When reset: Indeterminate>

D	Bit Name	Function	R	W
0-15	IVECT (ICU vector table address, 16 low-order bits)	When an interrupt is accepted, this register stores the 16 low-order bits of ICU vector table address for the accepted interrupt source.	○	–

Note: This register must always be accessed in halfwords.

The Interrupt Vector Register (IVECT) is a register which when an interrupt is accepted, stores the 16 low-order bits of ICU vector table address for the accepted interrupt source.

The ICU vector table (addresses H'0000 0094 to H'0000 010F) contains the start address of the interrupt handler for each internal peripheral I/O which have been set in advance. When an interrupt is accepted, the 16 low-order bits of ICU vector table address for the accepted interrupt source are set in the IVECT Register. The EIT handler reads the content of the IVECT Register using the LDH instruction to get the ICU vector table address needed.

When the IVECT Register is read out, operations (1) through (4) below are automatically performed in hardware.

(1) Set the accepted new IMASK value (NEW_IMASK) in the IMASK Register
(2) Clear the accepted interrupt request (not cleared for level interrupt sources)
(3) Interrupt request (EI) to the CPU core is cleared.
(4) Activate the ICU's internal sequencer to start internal processing (interrupt priority resolution)

CAUTION

Do not read the Interrupt Vector Register (IVECT) in other than the EIT handler (unless the PSW Register IE bit is disabled). In the EIT handler, be sure to read the Interrupt Mask Register (IMASK) before reading the IVECT Register.

5.3.2 Interrupt Mask Register

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

<When reset: H'07>

D	Bit Name	Function	R	W
0-4	No functions assigned		0	–
5-7	IMASK (interrupt mask)	000: Disables maskable interrupt	○	○
		001: Enables level 0 interrupt to be accepted		
		010: Enables level 0-1 interrupts to be accepted		
		011: Enables level 0-2 interrupts to be accepted		
		100: Enables level 0-3 interrupts to be accepted		
		101: Enables level 0-4 interrupts to be accepted		
		110: Enables level 0-5 interrupts to be accepted		
		111: Enables level 0-6 interrupts to be accepted		

The Interrupt Mask Register (IMASK) is used to set an interrupt mask to be compared with the priority level that has been set for each interrupt source (i.e., the Interrupt Control Register ILEVEL bit) to determine whether or not to accept the interrupt request.
When the Interrupt Vector Register (IVECT) described above is read out, a new mask value (NEW_IMASK) is set in this IMASK Register.

Upon writing to the IMASK Register, operations (1) to (2) below are automatically performed in hardware.

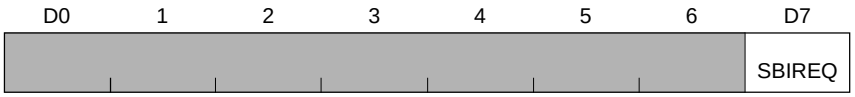
(1) Negate the interrupt request (EI) sent to the CPU core
(2) Activate the ICU's internal sequencer to start internal processing (interrupt priority resolution)

CAUTION

Do not write to the Interrupt Mask Register (IMASK) in other than the EIT handler (unless the PSW Register IE bit is disabled).

5.3.3 SBI (System Break Interrupt) Control Register

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



<When reset: H'00>

D	Bit Name	Function	R	W
0-6	No functions assigned		0	–
7	SBIREQ (SBI request)	0: SBI not requested 1: SBI requested	○	△

W = △ : Writable for only clearing (see the explanation below)

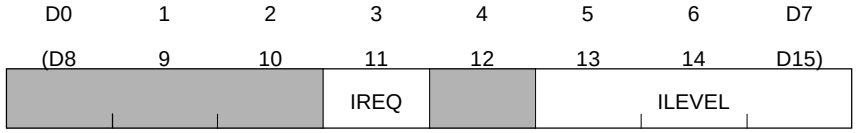
The SBI (System Break Interrupt) is an interrupt generated by a falling edge of the $\overline{\text{SBI}}$ input signal.

When an SBI interrupt occurs, the SBI Control Register SBIREQ (SBI request) bit is set to 1. The SBIREQ bit cannot be set in software. To clear the SBIREQ bit after being set, follow the procedure described below. (However, do not perform this clearing operation unless an SBI request has been generated.)

- Write a 1 and then a 0 to SBIREQ.

5.3.4 Interrupt Control Registers

- CAN1 Transmit/Receive & Error Interrupt Control Register (ICAN1CR) <Address: H'0080 0060>
- CAN0 Transmit/Receive & Error Interrupt Control Register (ICAN0CR) <Address: H'0080 0061>
- PDC Compare Match & Error Interrupt Control Register (IPDCOPCR) <Address: H'0080 0062>
- RTD Interrupt Control Register (IRTDCR) <Address: H'0080 0063>
- SIO6,7 Transmit/Receive Interrupt Control Register (ISIO67CR) <Address: H'0080 0064>
- SIO5 Transmit Interrupt Control Register (ISIO5TXCR) <Address: H'0080 0065>
- SIO5 Receive Interrupt Control Register (ISIO5RXCR) <Address: H'0080 0066>
- SIO4 Transmit/Receive Interrupt Control Register (ISIO4TXCR) <Address: H'0080 0067>
- SIO4 Receive Interrupt Control Register (ISIO4RXCR) <Address: H'0080 0068>
- SIO2,3 Transmit/Receive Interrupt Control Register (ISIO23CR) <Address: H'0080 0069>
- DMA5-9 Interrupt Control Register (IDMA59CR) <Address: H'0080 006A>
- A-D1 Conversion Interrupt Control Register (IAD1CCR) <Address: H'0080 006B>
- SIO1 Transmit Interrupt Control Register (ISIO1TXCR) <Address: H'0080 006C>
- SIO1 Receive Interrupt Control Register (ISIO1RXCR) <Address: H'0080 006D>
- SIO0 Transmit Interrupt Control Register (ISIO0TXCR) <Address: H'0080 006E>
- SIO0 Receive Interrupt Control Register (ISIO0RXCR) <Address: H'0080 006F>
- A-D0 Conversion Interrupt Control Register (IAD0CCR) <Address: H'0080 0070>
- DMA0-4 Interrupt Control Register (IDMA04CR) <Address: H'0080 0071>
- TID1 Output Interrupt Control Register (ITID1CR) <Address: H'0080 0072>
- TID0 Output Interrupt Control Register (ITID0CR) <Address: H'0080 0073>
- TMS0 Output Interrupt Control Register (ITMS0CR) <Address: H'0080 0074>
- TOM1 Output Interrupt Control Register (ITOM1CR) <Address: H'0080 0075>
- TOM0 Output Interrupt Control Register (ITOM0CR) <Address: H'0080 0076>
- Timer Input Interrupt Control Register 0 (IMJTICR0) <Address: H'0080 0077>
- Timer Input Interrupt Control Register 1 (IMJTICR1) <Address: H'0080 0078>
- Timer Input Interrupt Control Register 2 (IMJTICR2) <Address: H'0080 0079>
- Timer Input Interrupt Control Register 3 (IMJTICR3) <Address: H'0080 007A>
- Timer Input Interrupt Control Register 4 (IMJTICR4) <Address: H'0080 007B>
- Timer Input Interrupt Control Register 5 (IMJTICR5) <Address: H'0080 007C>
- PWM Off Input Interrupt Control Register (IPWMOFFCR) <Address: H'0080 007D>
- PDC Input & Error Detection Interrupt Control Register 4 (IPDCCR) <Address: H'0080 007E>



<When reset: H'07>

D	Bit Name	Function	R	W
0-2 (8-10)	No functions assigned		0	–
3 (11)	IREQ (interrupt request)	0 : Interrupt not requested 1 : Interrupt requested	○	△
4 (12)	No functions assigned		0	–
5-7 (13-15)	ILEVEL (interrupt priority level)	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 state)	○	○

W = △ : Can only be set/cleared when the "type of input source" is the edge type (and one input source).

(1) IREQ (interrupt request) bit (D3 or D11)

When an interrupt request from internal peripheral I/O occurs, the IREQ (interrupt request) bit is set to 1.
This bit can be set and cleared in software for only the edge input type of interrupt sources (and cannot for the level type). The IREQ bit when set for an interrupt request generated by an edge input type of interrupt source is automatically cleared to 0 upon reading the Interrupt Vector Register (IVECT). (However, this bit is not automatically cleared for the level type of interrupt source.)

If this bit is cleared in software at the same time it is set by an interrupt request generated, the former has priority (i.e., the bit is cleared). Also, if this bit is cleared by reading the IVECT Register at the same time it is set by an interrupt request generated, the former has priority (i.e., the bit is cleared).

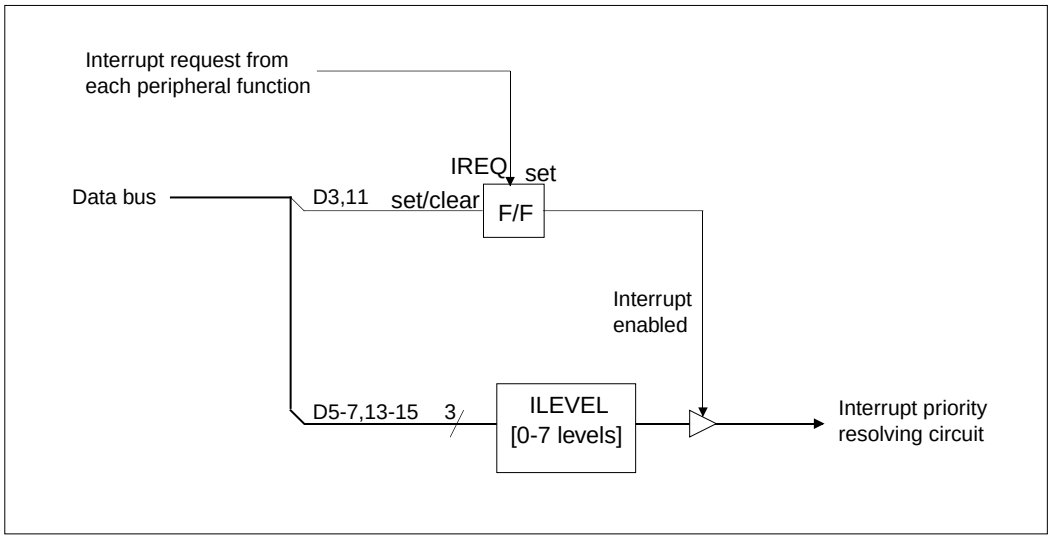


Figure 5.3.2 Configuration of the Interrupt Control Register (Edge Type)

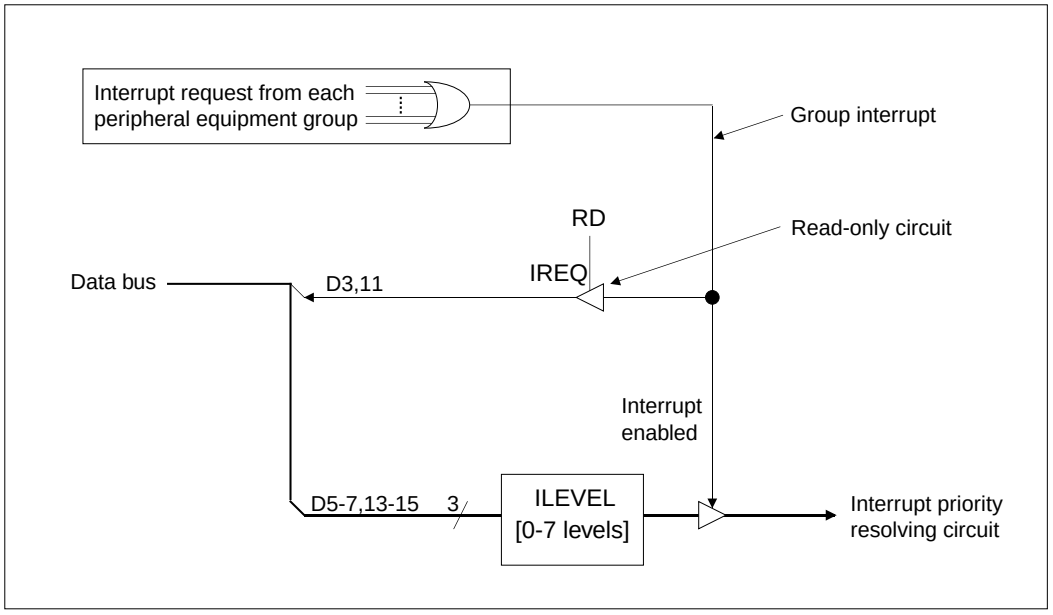


Figure 5.3.3 Configuration of the Interrupt Control Register (Level Type)

(2) ILEVEL (interrupt priority level) (D5-D7 or D13-D15)

These bits set the priority level of an interrupt request from each internal peripheral I/O. Set the priority level to 7 to disable the interrupt from internal peripheral I/O or 0-6 to use the interrupt. When an interrupt occurs, the interrupt controller resolves priority between this interrupt and other interrupt sources based on ILEVEL settings and finally compares its priority with the IMASK value to determine whether to forward an EI request to the CPU or keep it pending.

The relationship between ILEVEL values and the IMASK values at which interrupts are accepted is shown below.

Table 5.3.1 ILEVEL Settings and the Accepted IMASK Values

LEVEL set value		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 (interrupt disabled)

5.4 ICU Vector Table

The ICU vector table is used to set the start address of the interrupt handler for each internal peripheral I/O. The 31 interrupt sources are assigned the following vector table addresses.

Table 5.4.1 ICU Vector Table Addresses

Interrupt Source	ICU Vector Table Address
PDC input & error detection interrupt	H'0000 009 – H'0000 0097
PWM off input interrupt	H'0000 0098 – H'0000 009B
Timer input interrupt 5	H'0000 009C – H'0000 009F
Timer input interrupt 4	H'0000 00A0 – H'0000 00A3
Timer input interrupt 3	H'0000 00A4 – H'0000 00A7
Timer input interrupt 2	H'0000 00A8 – H'0000 00AB
Timer input interrupt 1	H'0000 00AC – H'0000 00AF
Timer input interrupt 0	H'0000 00B0 – H'0000 00B3
TOM0 output interrupt	H'0000 00B4 – H'0000 00B7
TOM1 output interrupt	H'0000 00B8 – H'0000 00BB
TMS0 output interrupt	H'0000 00BC – H'0000 00BF
TID0 output interrupt	H'0000 00C0 – H'0000 00C3
TID1 output interrupt	H'0000 00C4 – H'0000 00C7
DMA0-4 interrupt	H'0000 00C8 – H'0000 00CB
A-D0 converter interrupt	H'0000 00CC – H'0000 00CF
SIO0 receive interrupt	H'0000 00D0 – H'0000 00D3
SIO0 transmit interrupt	H'0000 00D4 – H'0000 00D7
SIO1 receive interrupt	H'0000 00D8 – H'0000 00DB
SIO1 transmit interrupt	H'0000 00DC – H'0000 00DF
A-D1 converter interrupt	H'0000 00E0 – H'0000 00E3
DMA5-9 interrupt	H'0000 00E4 – H'0000 00E7
SIO2,3 transmit/receive interrupt	H'0000 00E8 – H'0000 00EB
SIO4 receive interrupt	H'0000 00EC – H'0000 00EF
SIO4 transmit interrupt	H'0000 00F0 – H'0000 00F3
SIO5 receive interrupt	H'0000 00F4 – H'0000 00F7
SIO5 transmit interrupt	H'0000 00F8 – H'0000 00FB
SIO6,7 transmit/receive interrupt	H'0000 00FC – H'0000 00FE
RTD interrupt	H'0000 0100 – H'0000 0103
PDC compare match & error interrupt	H'0000 0104 – H'0000 0107
CAN0 transmit/receive & error interrupt	H'0000 0108 – H'0000 010B
CAN1 transmit/receive & error interrupt	H'0000 010C – H'0000 010F

Address	+0 address D0 – D7	+1 address D8 – D15
H'0000 0094	PDC Input & Error Detection Interrupt Handler Start Address (A0-A15)	
H'0000 0096	PDC Input & Error Detection Interrupt Handler Start Address(A16-A31)	
H'0000 0098	PWM Off Input Interrupt Handler Start Address (A0-A15)	
H'0000 009A	PWM Off Input Interrupt Handler Start Address (A16-A31)	
H'0000 009C	Timer Input Interrupt 5 Handler Start Address (A0-A15)	
H'0000 009E	Timer Input Interrupt 5 Handler Start Address (A16-A31)	
H'0000 00A0	Timer Input Interrupt 4 Handler Start Address (A0-A15)	
H'0000 00A2	Timer Input Interrupt 4 Handler Start Address (A16-A31)	
H'0000 00A4	Timer Input Interrupt 3 Handler Start Address (A0-A15)	
H'0000 00A6	Timer Input Interrupt 3 Handler Start Address (A16-A31)	
H'0000 00A8	Timer Input Interrupt 2 Handler Start Address (A0-A15)	
H'0000 00AA	Timer Input Interrupt 2 Handler Start Address (A16-A31)	
H'0000 00AC	Timer Input Interrupt 1 Handler Start Address (A0-A15)	
H'0000 00AE	Timer Input Interrupt 1 Handler Start Address (A16-A31)	
H'0000 00B0	Timer Input Interrupt 0 Handler Start Address (A0-A15)	
H'0000 00B2	Timer Input Interrupt 0 Handler Start Address (A16-A31)	
H'0000 00B4	TOM0 Output Interrupt Handler Start Address (A0-A15)	
H'0000 00B6	TOM0 Output Interrupt Handler Start Address (A16-A31)	
H'0000 00B8	TOM1 Output Interrupt Handler Start Address (A0-A15)	
H'0000 00BA	TOM1 Output Interrupt Handler Start Address (A16-A31)	
H'0000 00BC	TMS0 Output Interrupt Handler Start Address (A0-A15)	
H'0000 00BE	TMS0 Output Interrupt Handler Start Address (A16-A31)	
H'0000 00C0	TID0 Output Interrupt Handler Start Address (A0-A15)	
H'0000 00C2	TID0 Output Interrupt Handler Start Address (A16-A31)	
H'0000 00C4	TID1 Output Interrupt Handler Start Address (A0-A15)	
H'0000 00C6	TID1 Output Interrupt Handler Start Address (A16-A31)	

Blank areas are reserved for future use.

Figure 5.4.1 Memory Map of ICU Vector Table (1/2)

Address	+0 address D0 – D7	+1 address D8 – D15
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	A-D0 Conversion Interrupt Handler Start Address (A0-A15)	
H'0000 00CE	A-D0 Conversion Interrupt Handler Start Address (A16-A31)	
H'0000 00D0	SIO0 Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00D2	SIO0 Receive Interrupt Handler Start Address(A16-A31)	
H'0000 00D4	SIO0 Transmit Interrupt Handler Start Address (A0-A15)	
H'0000 00D6	SIO0 Transmit Interrupt Handler Start Address(A16-A31)	
H'0000 00D8	SIO1 Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00DA	SIO1 Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00DC	SIO1 Transmit Interrupt Handler Start Address (A0-A15)	
H'0000 00DE	SIO1 Transmit Interrupt Handler Start Address (A16-A31)	
H'0000 00E0	A-D1 Conversion Interrupt Handler Start Address (A0-A15)	
H'0000 00E2	A-D1 Conversion Interrupt Handler Start Address (A16-A31)	
H'0000 00E4	DMA5-9 Interrupt Handler Start Address (A0-A15)	
H'0000 00E6	DMA5-9 Interrupt Handler Start Address (A16-A31)	
H'0000 00E8	SIO2,3 Transmit/Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00EA	SIO2,3 Transmit/Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00EC	SIO4 Transmit/Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00EE	SIO4 Transmit/Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00F0	SIO4 Transmit Interrupt Handler Start Address (A0-A15)	
H'0000 00F2	SIO4 Transmit Interrupt Handler Start Address (A16-A31)	
H'0000 00F4	SIO5 Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00F6	SIO5 Receive Interrupt Handler Start Address (A16-A31)	
H'0000 00F8	SIO5 Transmit Interrupt Handler Start Address (A0-A15)	
H'0000 00FA	SIO5 Transmit Interrupt Handler Start Address (A16-A31)	
H'0000 00FC	SIO6,7 Transmit/Receive Interrupt Handler Start Address (A0-A15)	
H'0000 00FE	SIO6,7 Transmit/Receive Interrupt Handler Start Address (A16-A31)	
H'0000 0100	RTD Interrupt Handler Start Address (A0-A15)	
H'0000 0102	RTD Interrupt Handler Start Address (A16-A31)	
H'0000 0104	PDC Compare Match & Error Interrupt Handler Start Address (A0-A15)	
H'0000 0106	PDC Compare Match & Error Interrupt Handler Start Address (A16-A31)	
H'0000 0108	CAN0 Transmit/Receive & Error Interrupt Handler Start Address (A0-A15)	
H'0000 010A	CAN0 Transmit/Receive & Error Interrupt Handler Start Address (A16-A31)	
H'0000 010C	CAN1 Transmit/Receive & Error Interrupt Handler Start Address (A0-A15)	
H'0000 010E	CAN1 Transmit/Receive & Error Interrupt Handler Start Address (A16-A31)	

Blank areas are reserved for future use.

Figure 5.4.2 Memory Map of ICU Vector Table (2/2)

5.5 Description of Interrupt Operation

5.5.1 Accepting Interrupts from Internal Peripheral I/O

An interrupt from any internal peripheral I/O is accepted when its priority level is found to be higher than the IMASK value by comparing its ILEVEL set value in the Interrupt Control Register and the Interrupt Mask Register's IMASK value. However, if multiple interrupt requests occur at the same time, their priority is resolved following the procedure described below to determine which interrupt request to accept.

① Compare the ILEVEL values set with the Interrupt Control Registers for the respective internal peripheral I/Os
② If the ILEVEL values are the same, use the predetermined hardware priority to arbitrate
③ Compare the ILEVEL and the IMASK values

When multiple interrupt requests occur at the same time, their priority levels set with ILEVEL of the respective Interrupt Control Registers are compared to select the interrupt request with the highest priority. If the selected interrupt requests have the same ILEVEL value, they are arbitrated according to the fixed hardware priority. The ILEVEL of the finally selected interrupt request is compared with the IMASK value, and if its priority is higher than the IMASK value, an EI request for it is sent to the CPU.

Interrupt requests may be masked setting the Interrupt Mask Register and the Interrupt Control Register ILEVEL bits (level 7 disables interrupt) provided for each internal peripheral I/O, as well as setting the PSW Register IE bit.

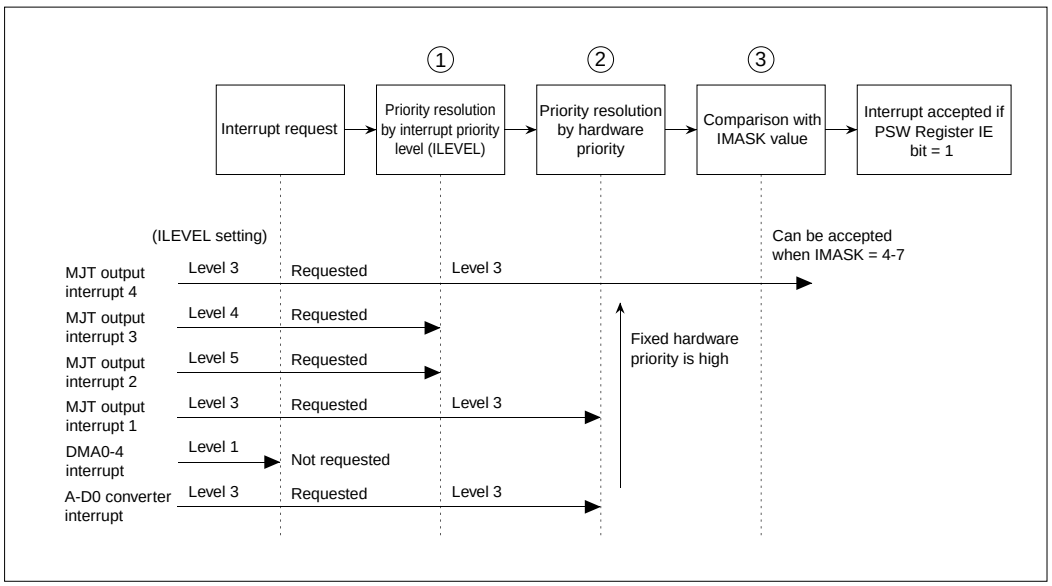


Figure 5.5.1 Example of Priority Resolution when Accepting an Interrupt Request

Table 5.5.1 Fixed Hardware Priority

Priority	Interrupt Source	ICU Vector Table Address	Type of Input Source
High	PDC input & error detection interrupt	H'0000 0094 – H'0000 0097	Level
↓	PWM off input interrupt	H'0000 0098 – H'0000 009B	Level
	Timer input interrupt 5 (IRQ28)	H'0000 009C – H'0000 009F	Level
	Timer input interrupt 4 (IRQ27)	H'0000 00A0 – H'0000 00A3	Level
	Timer input interrupt 3 (IRQ25)	H'0000 00A4 – H'0000 00A7	Level
	Timer input interrupt 2 (IRQ24)	H'0000 00A8 – H'0000 00AB	Level
	Timer input interrupt 1 (IRQ23)	H'0000 00AC – H'0000 00AF	Level
	Timer input interrupt 0 (IRQ22)	H'0000 00B0 – H'0000 00B3	Level
	TOM0 output interrupt (IRQ21)	H'0000 00B4 – H'0000 00B7	Level
	TOM1 output interrupt (IRQ20)	H'0000 00B8 – H'0000 00BB	Level
	TMS0 output interrupt (IRQ19)	H'0000 00BC – H'0000 00BF	Level
	TID0 output interrupt (IRQ18)	H'0000 00C0 – H'0000 00C3	Level
	TID1 output interrupt (IRQ17)	H'0000 00C4 – H'0000 00C7	Level
	DMA0-4 interrupt	H'0000 00C8 – H'0000 00CB	Level
	A-D0 converter interrupt	H'0000 00CC – H'0000 00CF	Edge
	SIO0 receive interrupt	H'0000 00D0 – H'0000 00D3	Edge
	SIO0 transmit interrupt	H'0000 00D4 – H'0000 00D7	Edge
	SIO1 receive interrupt	H'0000 00D8 – H'0000 00DB	Edge
	SIO1 transmit interrupt	H'0000 00DC – H'0000 00DF	Edge
	A-D1 converter interrupt	H'0000 00E0 – H'0000 00E3	Edge
	DMA5-9 interrupt	H'0000 00E4 – H'0000 00E7	Level
	SIO2,3 transmit/receive interrupt	H'0000 00E8 – H'0000 00EB	Level
	SIO4 receive interrupt	H'0000 00EC – H'0000 00EF	Edge
	SIO4 transmit interrupt	H'0000 00F0 – H'0000 00F3	Edge
	SIO5 receive interrupt	H'0000 00F4 – H'0000 00F7	Edge
	SIO5 transmit interrupt	H'0000 00F8 – H'0000 00FB	Edge
	SIO6,7 transmit/receive interrupt	H'0000 00FC – H'0000 00FF	Level
	RTD interrupt	H'0000 0100 – H'0000 0103	Edge
	PDC compare match & error interrupt	H'0000 0100 – H'0000 0103	Level
	CAN0 transmit/receive & error interrupt	H'0000 0108 – H'0000 010B	Level
Low	CAN1 transmit/receive & error interrupt	H'0000 010C – H'0000 010F	Level

Table 5.5.2 ILEVEL Settings and the Accepted IMASK Values

ILEVEL set value	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 (interrupt disabled)

5.5.2 Processing of Internal Peripheral I/O Interrupts by Handler

(1) Branching to the interrupt handler

When the CPU accepts an interrupt, it branches to the EIT vector entry after performing hardware preprocessing as described in Section 4.3, "EIT Processing Procedure." The EIT vector entry assigned to the External Interrupt (EI) resides at address H'0000 0080, at which the instruction for branching to the beginning of the interrupt handler routine for the External Interrupt is written by the user. Note that this is the branch instruction, and not the jump address.

(2) Processing in the interrupt handler

In the External Interrupt (EI) handler, first save the BPC Register, PSW Register, and general-purpose registers to the stack.

Next, read the Interrupt Mask Register (IMASK) and save it to the stack. Then read the Interrupt Vector Register (IVECT). Always be sure to read the IMASK before reading the IVECT. Reading the IMASK and the IVECT each activates operations necessary to clear the interrupt request to the CPU and accept the next interrupt. On top of that, reading the IVECT causes NEW_IMASK to be set in the IMASK Register and the accepted interrupt request to be cleared (not cleared in the case of level-recognized interrupt sources, however).

The IVECT Register has set in it the 16 low-order bits of the ICU vector table address for the accepted interrupt source. Read the IVECT Register using a signed halfword load instruction (LDH instruction) to get the vector table address and then the content of the ICU interrupt vector table at that address. The ICU vector table contains the start address of the interrupt handler for each internal peripheral I/O, so branch to the address thus read out and execute the interrupt handler.

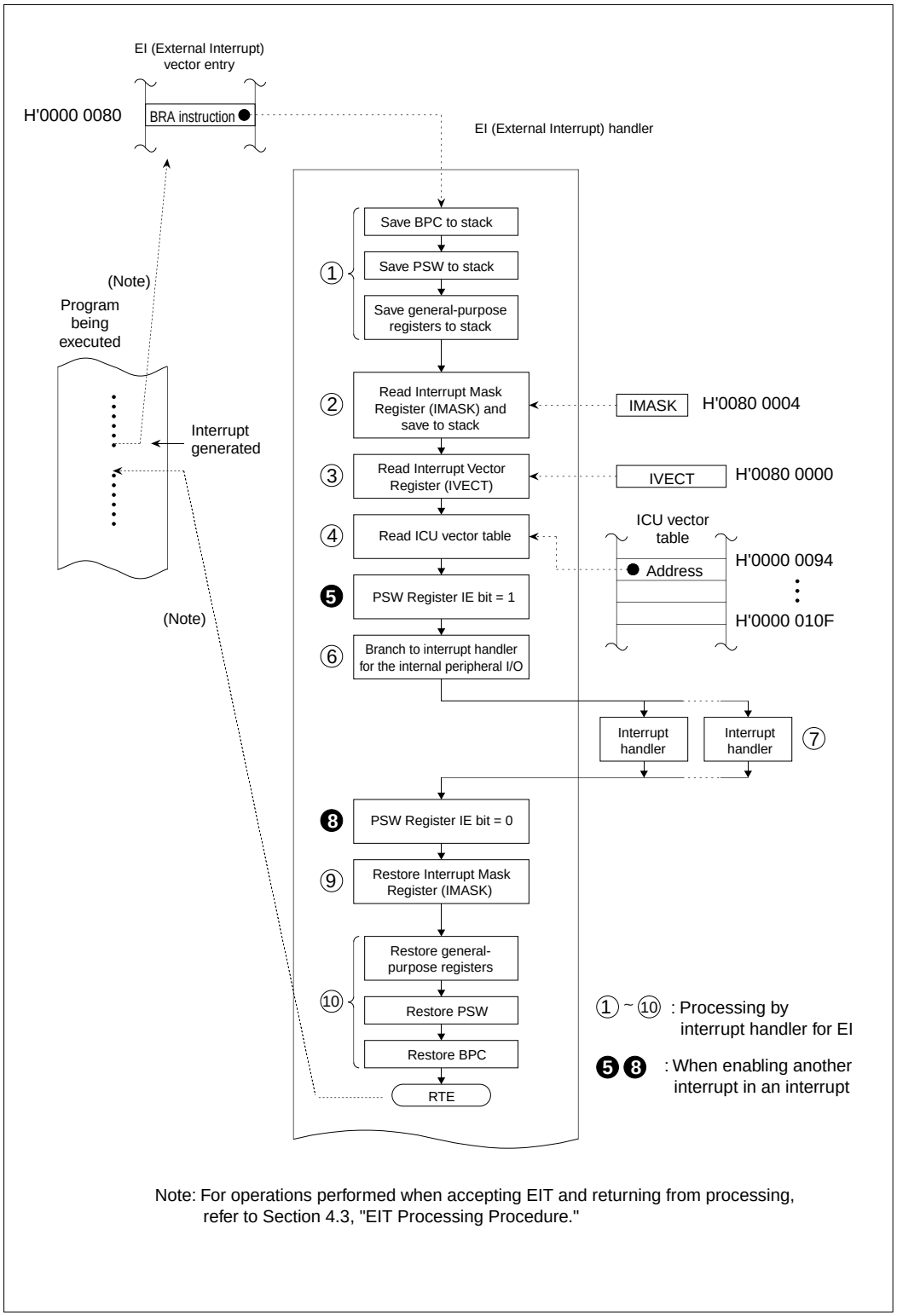
When returning from the interrupt handler, clear the PSW Register IE bit to 0 to disable interrupts before restoring the IMASK value.

(3) Identifying the interrupt source

If any internal peripheral I/O has multiple interrupt sources, inspect the Interrupt Status Register for that internal peripheral I/O to identify the interrupt source.

(4) Enabling multiple interrupts

To enable another interrupt within an interrupt handler, set the PSW Register IE (Interrupt Enable) bit to 1 in that interrupt handler to enable the interrupt to be accepted. However, be sure to save each register (BPC, PSW, general-purpose registers, and IMASK) to the stack before setting the IE bit.



Note: For operations performed when accepting EIT and returning from processing, refer to Section 4.3, "EIT Processing Procedure."

Figure 5.5.2 Example of Operations Performed for Interrupts from Internal Peripheral I/O

5.6 Description of System Break Interrupt (SBI) Operation

5.6.1 Accepting SBI Interrupt

The System Break Interrupt (SBI) is an emergency interrupt issued when power outage is detected or a fault condition is notified from an external watchdog timer. The SBI interrupt is always accepted by a falling edge of the $\overline{SBI}$ signal regardless of how the PSW Register IE bit is set, and cannot be masked.

5.6.2 SBI Processing by Handler

After processing for the SBI is finished, always be sure to terminate or reset the system without returning to the program the CPU was executing when the interrupt occurred.

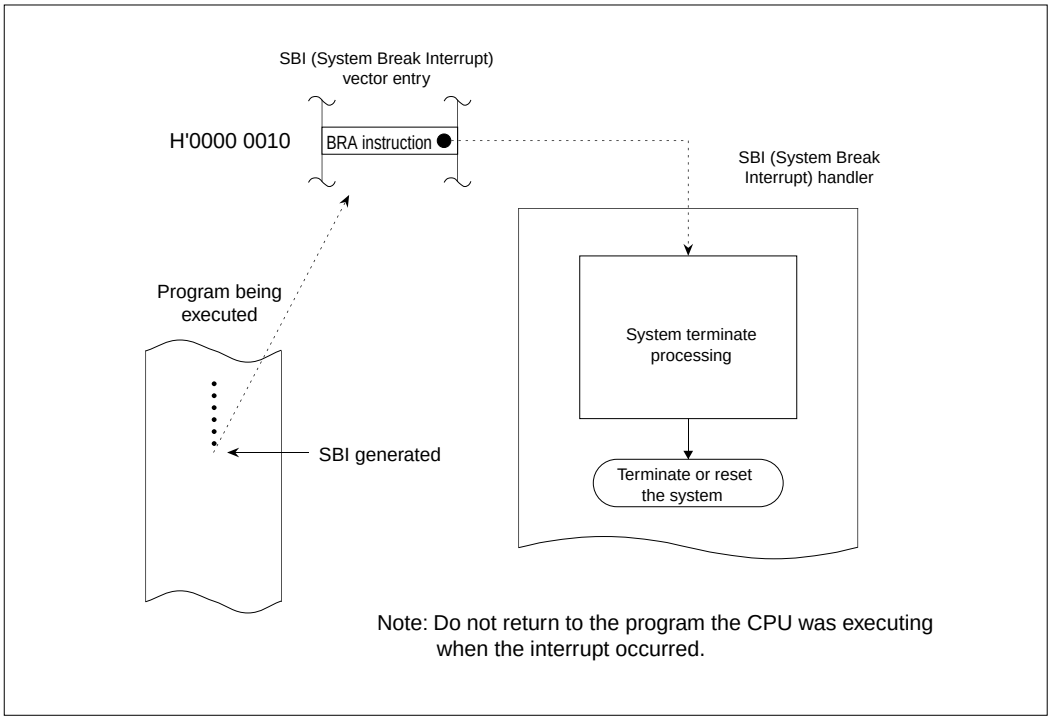


Figure 5.6.1 Example of Operation for SBI Interrupt

CHAPTER 6

INTERNAL MEMORY

6.1 Outline of the Internal Memory

The M32172F2/M32173F2 contains the following types of memory:

- 16-Kbyte RAM for the M32172F2 or 32-Kbyte RAM for the M32173F2
- 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
Capacity	M32172F2: 16 Kbytes; M32173F2: 32 Kbytes
Location address	M32172F2: H'0080 4000-H'0080 7FFF M32173F2: H'0080 4000-H'0080 BFFF
Wait insertion	Operates with no wait cycles (when CPU clock = 40 MHz)
Internal bus connection	Connects to 32-bit bus
Dual port	By using the RTD (Real-time Debugger), data can be read out (monitored) or written to any area of the internal RAM via serial communication from the outside independently of the CPU. (For details, refer to Chapter 14, "Real-time Debugger.")

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
Capacity	M32172F2, M32173F2: 256 Kbytes
Location address	M32172F2, M32173F2: H'0000 0000-H'0003 FFFF
Wait insertion	Operates with no wait cycles (when CPU clock = 40 MHz)
Durability	Can be rewritten 100 times
Internal bus connection	Connects to 32-bit bus
Other	Virtual-flash emulation function is included. (For details, refer to Section 6.7, "Virtual-flash Emulation Function.")

6.4 Internal Flash Memory Related Registers

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

Address	+0 address D0 – D7	+1 address D8 – D15
H'0080 07E0	Flash Mode Register (FMOD)	Flash Status Register 1 (FSTAT1)
H'0080 07E2	Flash Control Register 1 (FCNT1)	Flash Control Register 2 (FCNT2)
H'0080 07E4	Flash Control Register 3 (FCNT3)	Flash Control Register 4 (FCNT4)
H'0080 07E6		
H'0080 07E8	Virtual-flash L Bank Register 0 (FELBANK0)	
H'0080 07EA	Virtual-flash L Bank Register 1 (FELBANK1)	
H'0080 07EC	Virtual-flash L Bank Register 2 (FELBANK2)(Note)	
H'0080 07EE		
H'0080 07F0	Virtual-flash S Bank Register 0 (FESBANK0)(Note)	
H'0080 07F2	Virtual-flash S Bank Register 1 (FESBANK1)(Note)	

Blank areas are reserved for future use.

Note: The FELBANK2, FESBANK0, and FESBANK1 Registers are nonexistent for the M32172F2.

Figure 6.4.1 Internal Flash Memory Related Register Map

6.4.1 Flash Mode Register

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

<When reset: H'0?>

D	Bit Name	Function	R	W
0-6	No functions assigned		0	–
7	FPMOD	0: FP pin = low	○	–
	(External FP pin status)	1: FP pin = high		

The Flash Mode Register (FMOD) is a read-only status register, with FPMOD showing the status of the FP (Flash Protect) pin.
The flash memory is enabled for write only when FPMOD = 1. Write to the flash memory executed when FPMOD = 0 has no effect.

6.4.2 Flash Status Registers

The 32172/32173 has two registers to show the flash memory status, one of which is Flash Status Register 1 (FSTAT1) located in the SFR area (address H'0080 07E1), and the other is Flash Status Register 2 (FSTAT2) included in the flash memory itself. Use these two status registers (FSTAT1, FSTAT2) to control operations when programming or erasing the flash memory.

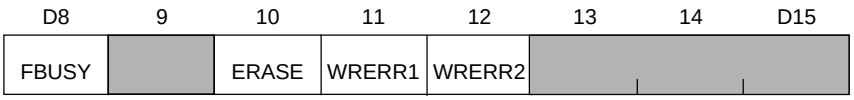
■ Flash Status Register 1 (FSTAT1) **<Address: H'0080 07E1>**

<When reset: H'01>

D	Bit Name	Function	R	W
8-14	No functions assigned		0	–
15	FSTAT	0: Busy	○	–
	(Ready/Busy status)	1: Ready		

The Flash Status Register 1 (FSTAT1) is a read-only status register used to know the status of whether the flash memory is being programmed or erased. When the FSTAT bit = 0, the flash memory is being programmed or erased, during which time the flash memory is disabled against access for programming.

■ **Flash Status Register 2 (FSTAT2)**



<When reset: H'80>

D	Bit Name	Function	R	W
8	FBUSY (Flash busy)	0: Being programmed or erased 1: Ready state	○	–
9	No functions assigned		0	–
10	ERASE (Auto erase operating status)	0: Erase normally operating or finished 1: Erase error occurred	○	–
11	WRERR1 (Program operating status 1)	0: Programming normally operating or finished 1: Programming error occurred	○	–
12	WRERR2 (Program operating status 2)	0: Programming normally operating or finished 1: Over-programming occurred	○	–
13-15	No functions assigned		0	–

The Flash Status Register 2 (FSTAT2) consists of the following four read-only status bits that indicate the operating status of the flash memory.

(1) FBUSY (flash busy) bit (D8)

The FBUSY bit is used to determine whether the operation is under way or finished when programming or erasing the flash memory.
The programming/erase operation is under way when FBUSY = 0 or finished when FBUSY = 1.

(2) ERASE (auto erase operating status) bit (D10)

The ERASE bit is used to determine whether an error occurred after the CPU has finished erasing the flash memory.
The erase operation terminated normally when ERASE = 0 or terminated in an error when ERASE = 1.

(3) WRERR1 (program operating status 1) bit (D11)

The WRERR1 bit is used to determine whether an error occurred after the CPU has finished programming the flash memory. The programming operation terminated normally when WRERR1 = 0 or terminated in an error when WRERR1 = 1.
The condition under which WRERR1 is set to 1 is when any bit other than those that must be 0 is found to be a 0 by comparison between the write data and the data in the flash memory.

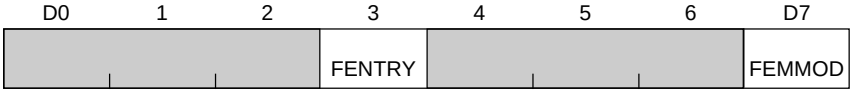
(4) WRERR2 (program operating status 2) bit (D12)

The WRERR2 bit is used to determine whether an error occurred after the CPU has finished programming the flash memory. The programming operation terminated normally when WRERR2 = 0 or terminated in an error when WRERR2 = 1.
The condition for which this bit is set to 1 is that even when the write operation was executed a specified number of times repeatedly, the flash memory could not be programmed.

Note: This status register is provided within the flash memory, so that by writing a read status command (H'7070) to any address of the internal flash memory, this register is enabled for readout. For details, refer to Section 6.5, "Programming the Internal Flash Memory."

6.4.3 Flash Control Registers

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



<When reset: H'00>

D	Bit Name	Function	R	W
0-2	No functions assigned		0	–
3	FENTRY (Flash mode entry)	0: Normal read 1: Erase/program possible	○	○
4-6	No functions assigned		0	–
7	FEMMOD (Virtual-flash emulation mode)	0: Normal mode 1: Virtual-flash emulation mode	○	○

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

(1) FENTRY (flash mode entry) bit (D3)

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

To set the FENTRY bit to 1, write 0 and then 1 to this bit successively while the FP pin = high.

The FENTRY bit is cleared under the following conditions:

- When the microcomputer is reset
- When a 0 is written to the FENTRY bit
- When the FP pin status is changed from high to low

When using a program in the flash memory while the FENTRY bit = 0, the EI vector entry is at the flash memory address H'0000 0080. To run the flash rewrite program in RAM while the FENTRY bit = 1, the EI vector entry is at the RAM address H'0080 4000, in which case flash rewrite operation can be controlled using an interrupt.

Table 6.4.1 EI Vector Entry Transition by FENTRY

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

(2) FEMMOD (virtual-flash emulation mode) bit (D7)

The FEMMOD bit controls state transition to virtual-flash emulation mode. The CPU is placed in virtual-flash emulation mode by setting the FEMMOD bit to 1 while the FENTRY bit = 0.

(For details, refer to Section 6.7, "Virtual-flash Emulation Function.")

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

<When reset: H'00>

D	Bit Name	Function	R	W
8-14	No functions assigned		0	–
15	FPROT	0: Enables protection by lock bits	○	○
	(Unlock)	1: Disables protection by lock bits		

The Flash Control Register 2 (FCNT2) controls flash memory protection by lock bits (disables the internal flash memory against erase/programming). Protection is disabled by setting the FPROT bit to 1, so that blocks of the flash memory protected by lock bits can be erased/programmed.

To set the FPROT bit to 1, write 0 and then 1 to this bit successively while the FENTRY bit = 1.

The FPROT bit is cleared to 0 immediately after chip reset or by writing 0 to this bit, pulling input on the FP pin low, or resetting the FENTRY bit to 0.

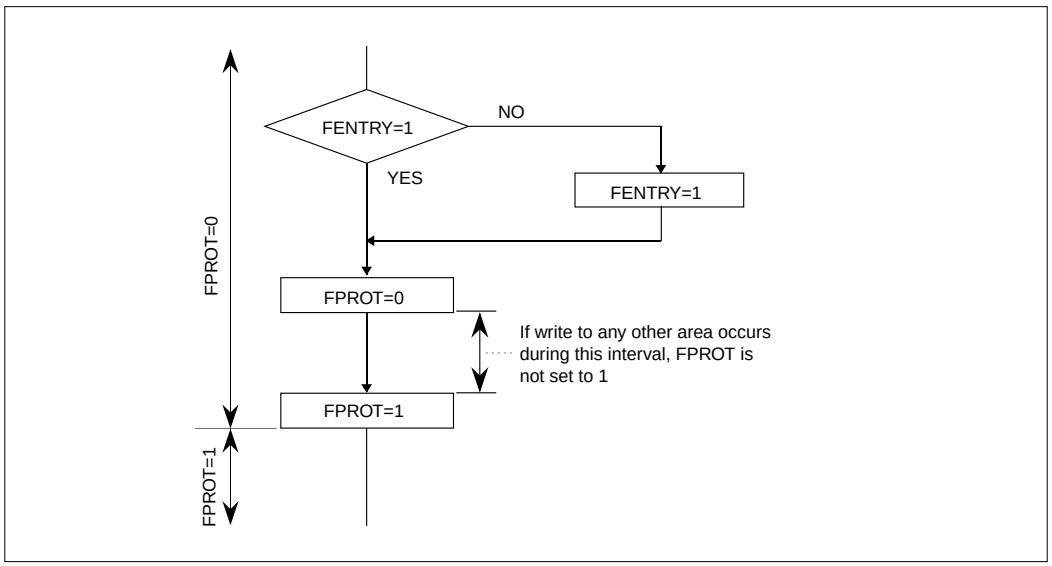
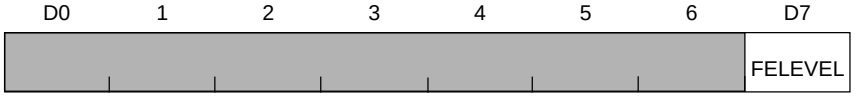


Figure 6.4.2 Protection Unlocking Flow

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



<When reset: H'00>

D	Bit Name	Function	R	W
0-6	No functions assigned		0	–
7	FELEVEL	0: Normal level	○	○
	(Erase margin up)	1: Increases erase margin		

The Flash Control Register 3 (FCNT3) controls the depth of erase levels when erasing the internal flash memory with an erase command. By setting the FELEVEL bit is set to 1, the flash memory erase level can be deepened, which will result in an increased reliability margin.

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

<When reset: H'00>

D	Bit Name	Function	R	W
8-14	No functions assigned		0	–
15	FRESET	0: No operation	○	○
	(Reset flash)	1: Resets the flash memory		

The Flash Control Register 4 (FCNT4) controls canceling the erase or program operation in the middle or initializing each status bit of Flash Status Register 2 (FSTAT2).
Setting the FRESET bit to 1 cancels the erase or program operation in the middle and initializes each status bit of the FSTAT2 Register (to H'80).

The FRESET bit is effective only when the FENTRY bit = 1. Information of the FRESET bit is ignored unless the FENTRY bit = 1.

When programming or erasing the flash memory, make sure the FRESET bit = 0.

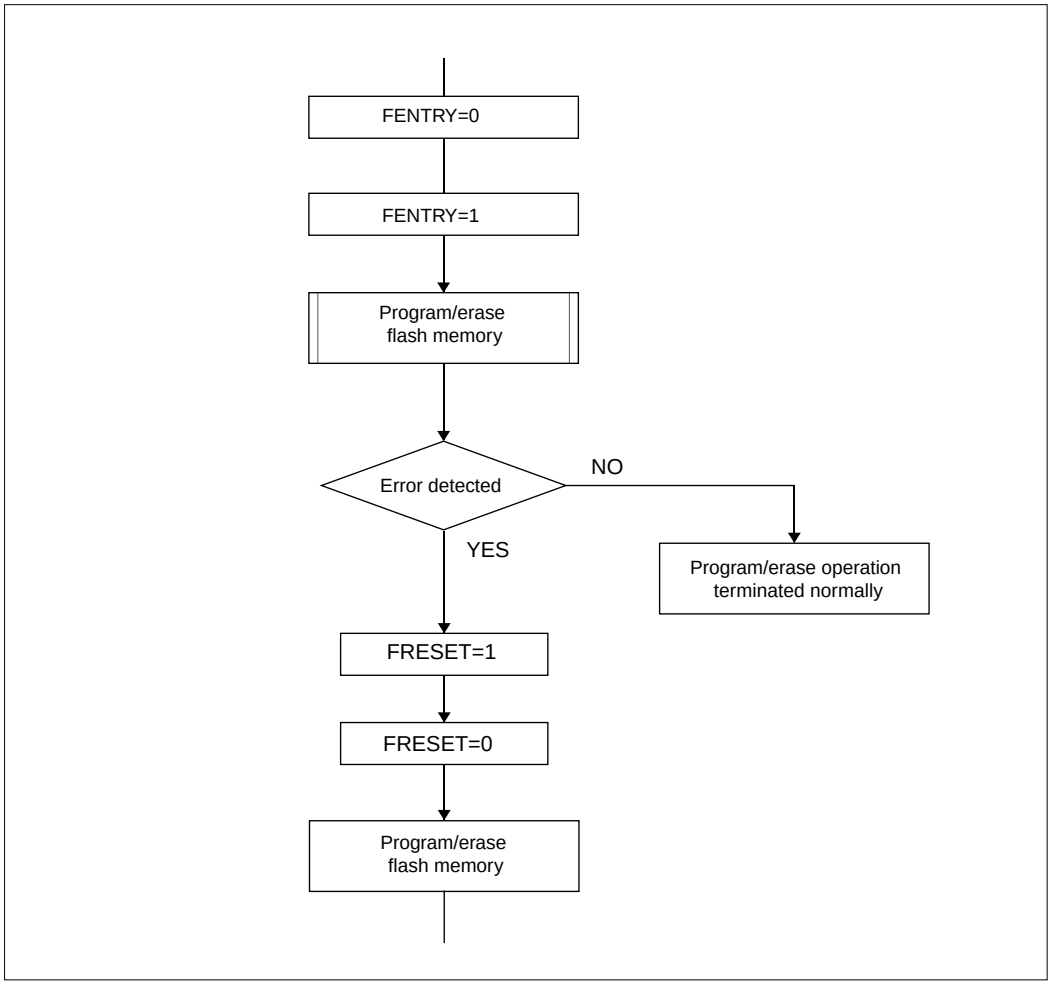


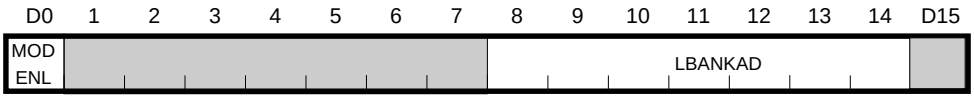
Figure 6.4.3 Example for Using the FCNT4 Register

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>**

■ **Virtual-flash L Bank Register 2 (FELBANK2)** **<Address: H'0080 07EC>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0	MODENL (Virtual-flash emulation enable)	0: Disables virtual-flash function 1: Enables virtual-flash function	○	○
1-7	No functions assigned		0	–
8-14	LBANKAD (L bank address)	Start address A12-A18 of the desired L bank	○	○
15	No functions assigned		0	–

Note: This register must always be accessed in halfwords.

(1) MODENL (virtual-flash emulation enable) bit (D0)

After entering virtual-flash emulation mode (by setting the FEMMOD bit to 1 while the FENTRY bit = 0), set the MODENL bit to 1 and the virtual-flash emulation function is enabled for the L bank area selected with LBANKAD bits.

(2) LBANKAD (L bank address) bits (D8-D14)

The LBANKAD bits select one of the L banks divided in units of 8 KB. Use these seven LBANKAD bits to set the start address A12-A18 (of the 32-bit address) of the desired L bank.

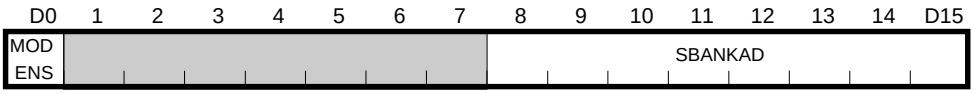
(For details, refer to Section 6.7, "Virtual-flash Emulation Function.")

Note: The M32172F2 does not have Virtual-flash L Bank Register 2 (FELBANK2).

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>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0	MODENS (Virtual-flash emulation enable)	0: Disables virtual-flash function 1: Enables virtual-flash function	○	○
1-7	No functions assigned		0	–
8-15	SBANKAD (S bank address)	Start address A12-A19 of the relevant S bank	○	○

(1) MODENS (virtual-flash emulation enable) bit (D0)

After entering virtual-flash emulation mode (by setting the FEMMOD bit to 1 while the FENTRY bit = 0), set the MODENS bit to 1 and the virtual-flash emulation function is enabled for the S bank area selected with SBANKAD bits.

(2) SBANKAD (S bank address) bits (D8-D15)

The SBANKAD bits select one of the S banks divided in units of 4 KB. Use these eight SBANKAD bits to set the start address A12-A19 (of the 32-bit address) of the desired S bank.

(For details, refer to Section 6.7, "Virtual-flash Emulation Function.")

Note: The M32172F2 does not have the Virtual-flash S Bank Registers (FESBANK0 and FESBANK1).

6.5 Programming the Internal Flash Memory

6.5.1 Outline of Flash Memory Programming

There are following two methods for writing to the internal flash memory.

(1) When no write programs exist in the internal flash memory

Set the FP pin = high, MOD0 = high, and MOD1 = low to enter boot mode. In this case, the reset vector entry is at the beginning of the boot program area (H'8000 0000). (Normally the reset vector entry is at the start address of the internal flash memory.)

Use the boot program to transfer the "flash write program" from the boot area into the internal RAM. Then jump to the RAM and by using the program in the RAM, set the Flash Control Register 1 FENTRY bit to 1 to enable the flash memory for write (flash E/W enable mode).

After that, use the "flash write program" that has been transferred into the RAM to write to the internal flash memory.

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

Set the FP pin = high, MOD0 = low, and MOD1 = low to enter single-chip mode. Transfer the "flash write program" from the internal flash memory in which it was prepared beforehand into the internal RAM. Then jump to the RAM and by using the program in the RAM, set the Flash Control Register 1 (FCNT1) FENTRY bit to 1 to enable the flash memory for write (flash E/W enable mode).

After that, use the "flash write program" that has been transferred into the RAM to write to the internal flash memory.

Or you can set the FP pin = high, MOD0 = low, and MOD1 = high to enter flash E/W enable mode in external extended mode.

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

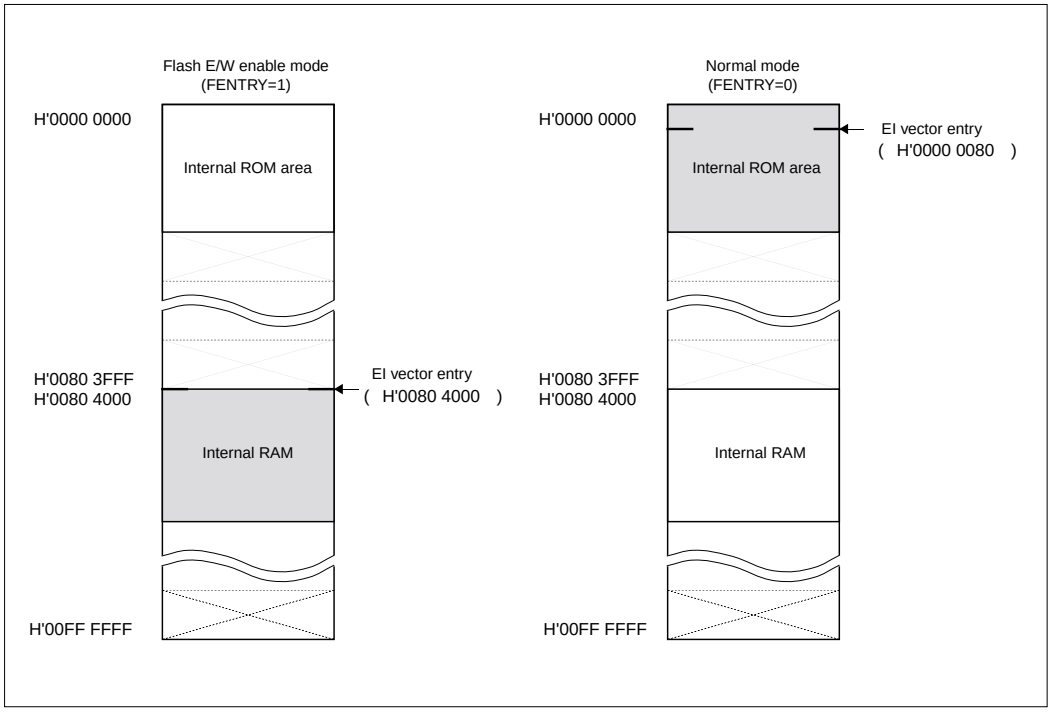


Figure 6.5.1 EI Vector Entry when in Flash E/W Enable Mode

(1) When the write programs do not exist in the internal flash memory

Use the program in the boot ROM located on the 32172/32173 memory map to write to the flash memory. To transfer the write data, use serial I/O1 in clock-synchronized serial mode.
This method is used for writing to the flash memory using a flash programmer.

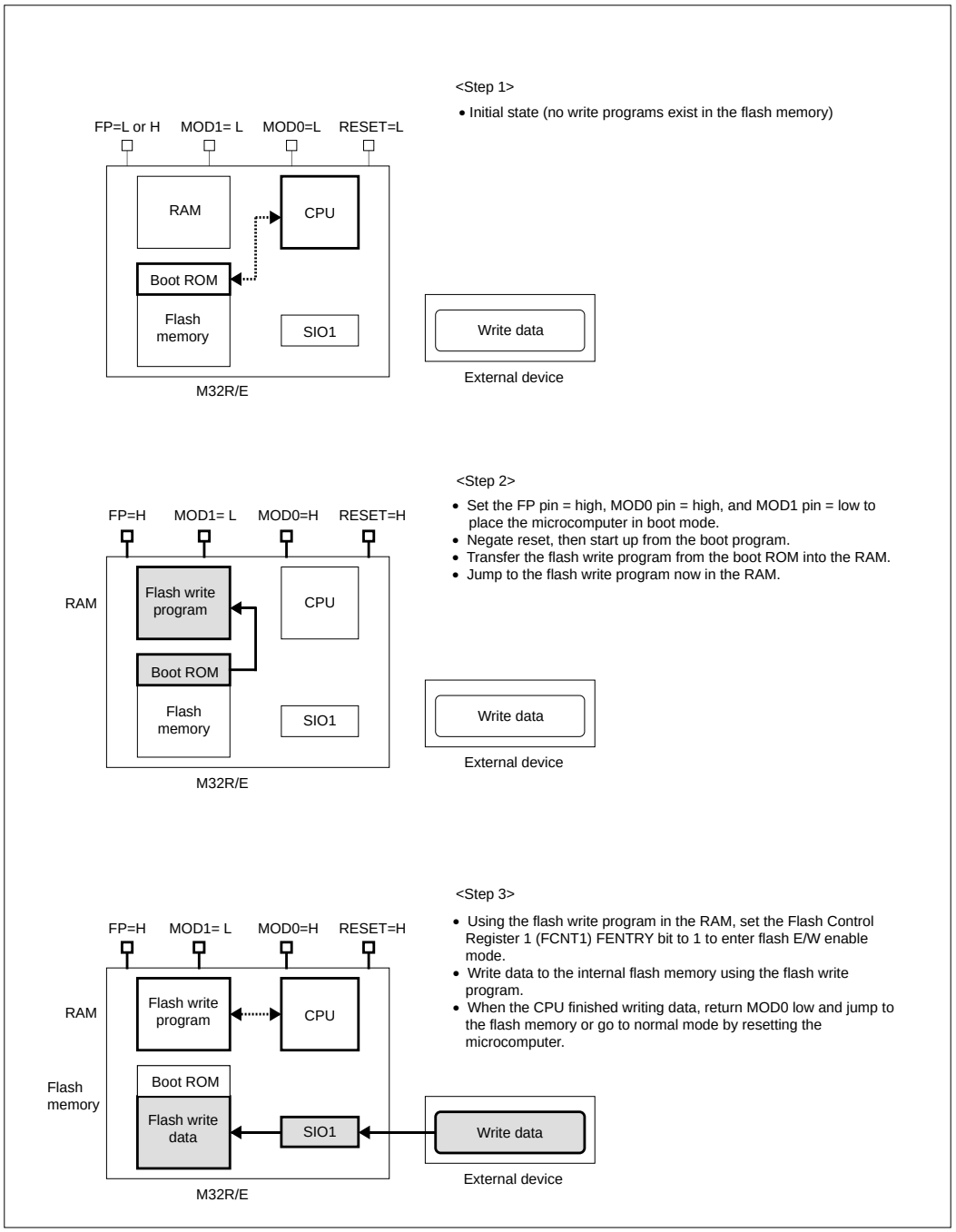


Figure 6.5.2 Procedure for Writing to the Internal Flash Memory (when no write programs exist in the flash memory)

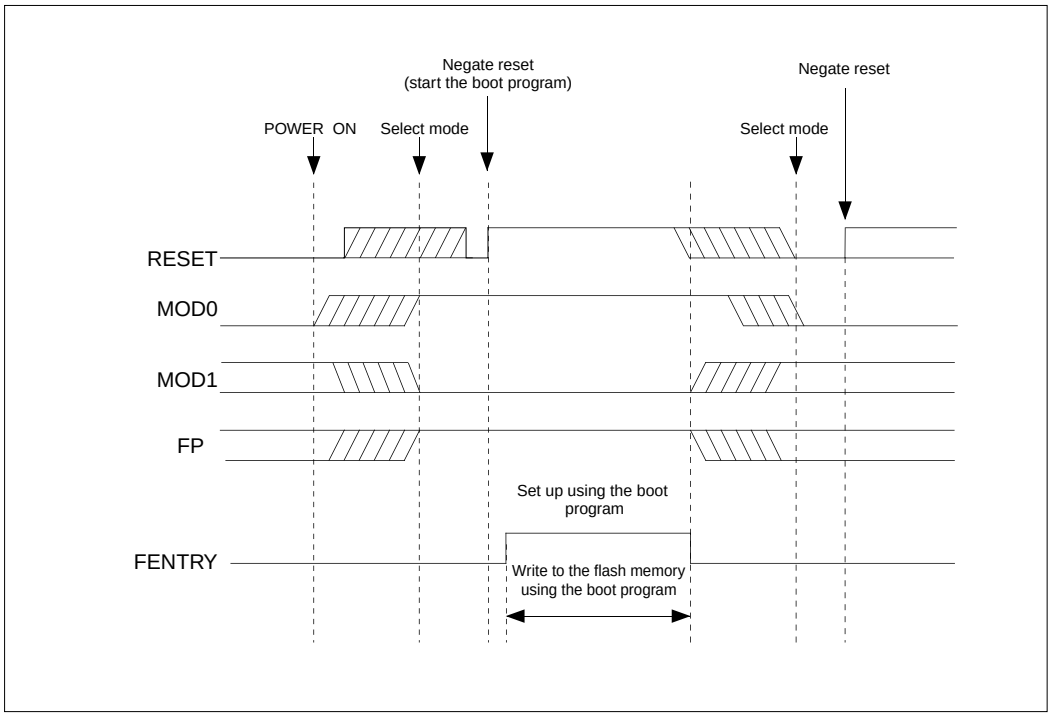


Figure 6.5.3 Timing at Which Writing to the Internal Flash Memory (when no write programs exist in the flash memory)

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

Use the write program available in the internal flash memory to write to the flash memory.
For this write operation, use the internal peripheral circuits as needed for the system to which you are writing. (The data bus, serial I/O, and ports can be used.)
The following shows an example where the flash memory is written to in single-chip mode using serial I/O0.

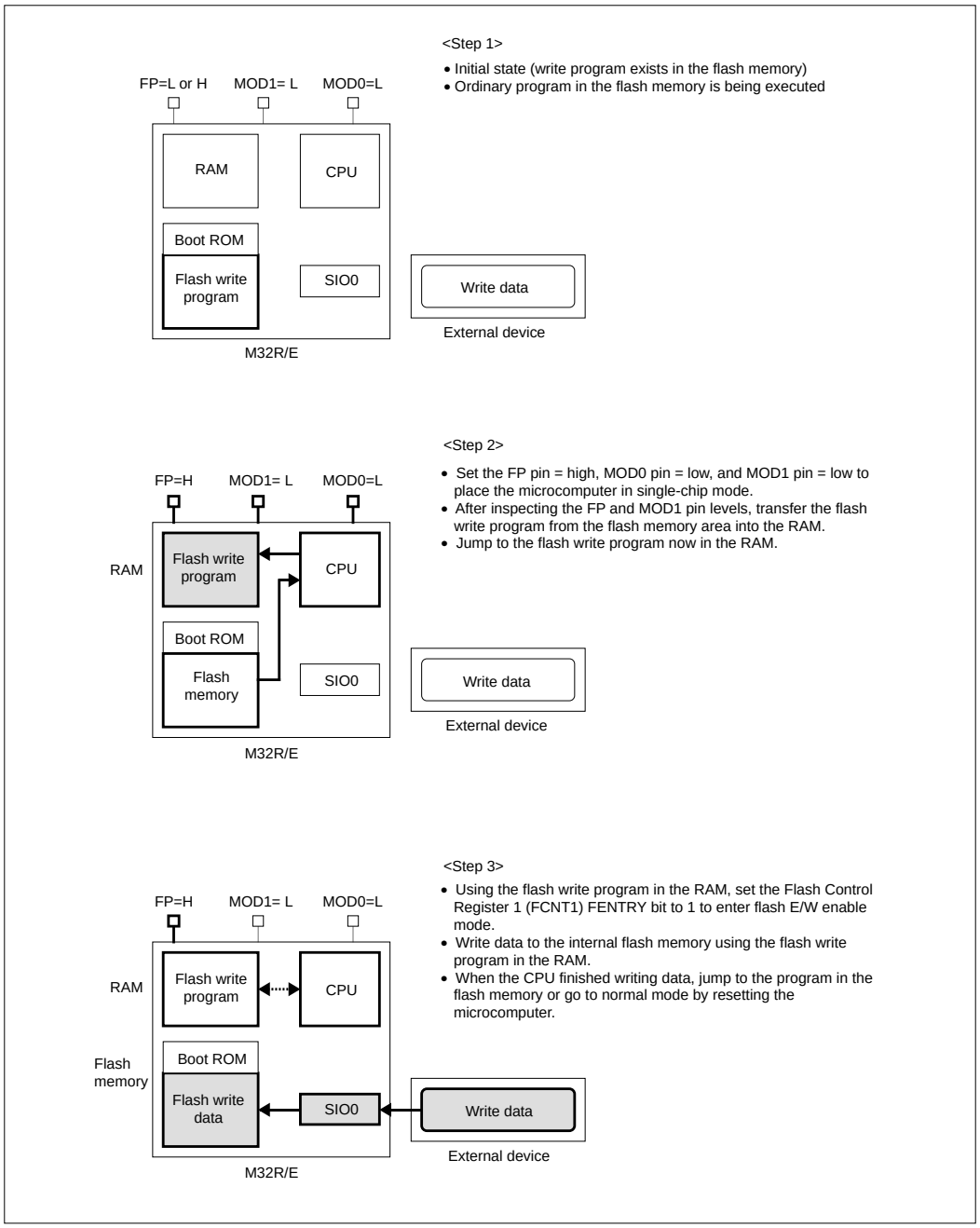


Figure 6.5.4 Procedure for Writing to the Internal Flash Memory (when write program exists in the flash memory)

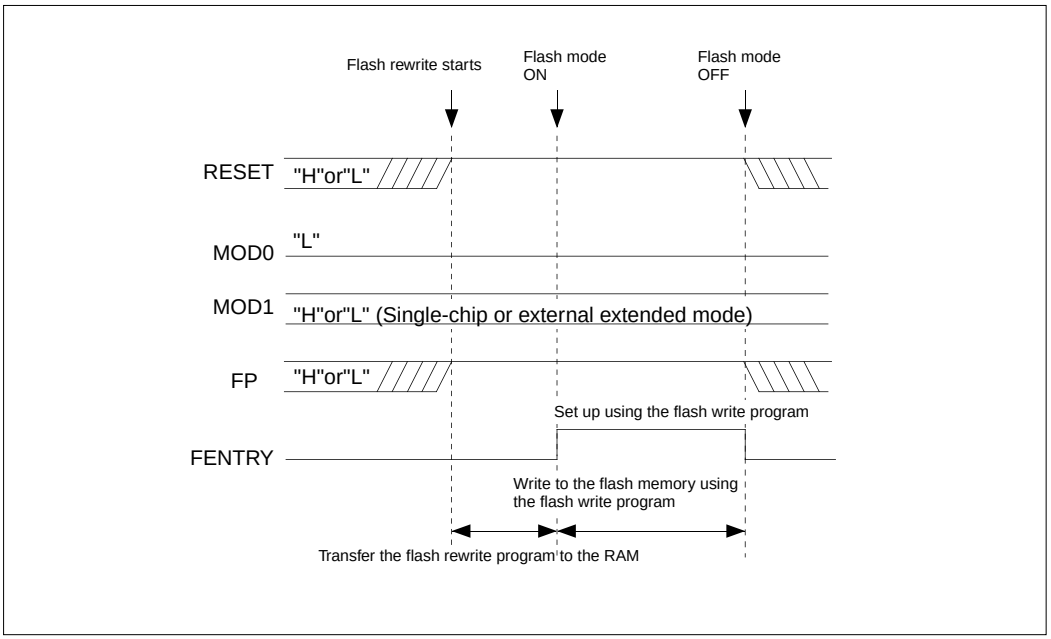


Figure 6.5.5 Timing at Which Writing to the Internal Flash Memory (when write program exists in the flash memory)

6.5.2 Controlling Operation Modes during Flash Programming

Chip operation modes are set using MOD0, MOD1, and Flash Control Register 1 (FCNT1) FENTRY bit. Operation modes during flash programming are listed below.

Table 6.5.1 Setting Operation Modes during Flash Programming

FP	MOD0	MOD1	FENTRY (Note)	Operation Mode	Reset Vector Entry	EI Vector Entry
0	0	0	–	Single-chip mode	Flash memory start address (H'0000 0000)	Flash area (H'0000 0080)
1	0	0	0			
0	1	0	–	Processor mode	External area start address (H'0000 0000)	External area (H'0000 0080)
0	0	1	–	External extended mode	Flash memory start address (H'0000 0000)	Flash area (H'0000 0080)
1	0	1	0			
1	0	0	1	Single-chip mode + flash E/W enable	Flash memory start address (H'0000 0000)	Internal RAM start address (H'0080 4000)
1	1	0	0	Boot mode	Boot program area start address (H'8000 0000)	Flash area (H'0000 0080)
1	1	0	1	Boot mode + flash E/W enable	Boot program area start address (H'8000 0000)	Internal RAM start address (H'0080 4000)
1	0	1	1	External extended mode + flash E/W enable	Flash memory start address (H'0000 0000)	Internal RAM start address (H'0080 4000)
–	1	1	–	reserved	–	–

Note: Shows the status of the Flash Control Register 1 (FCNT1) FENTRY bit. (- denotes "Don't care")

(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, you need to transfer the necessary program into the internal RAM before entering flash E/W enable mode, so the program can be run from the RAM.

(2) Entering flash E/W enable mode

The microcomputer can enter flash E/W enable mode only when it is operating in single-chip or external extended mode. The microcomputer enters flash E/W enable mode only when the FP pin is at the high level and the Flash Control Register 1 (FCNT1) FENTRY bit = 1, and cannot enter flash E/W enable mode when operating in processor mode or the FP pin is low.

(3) Detecting the MOD0 and MOD1 pin levels

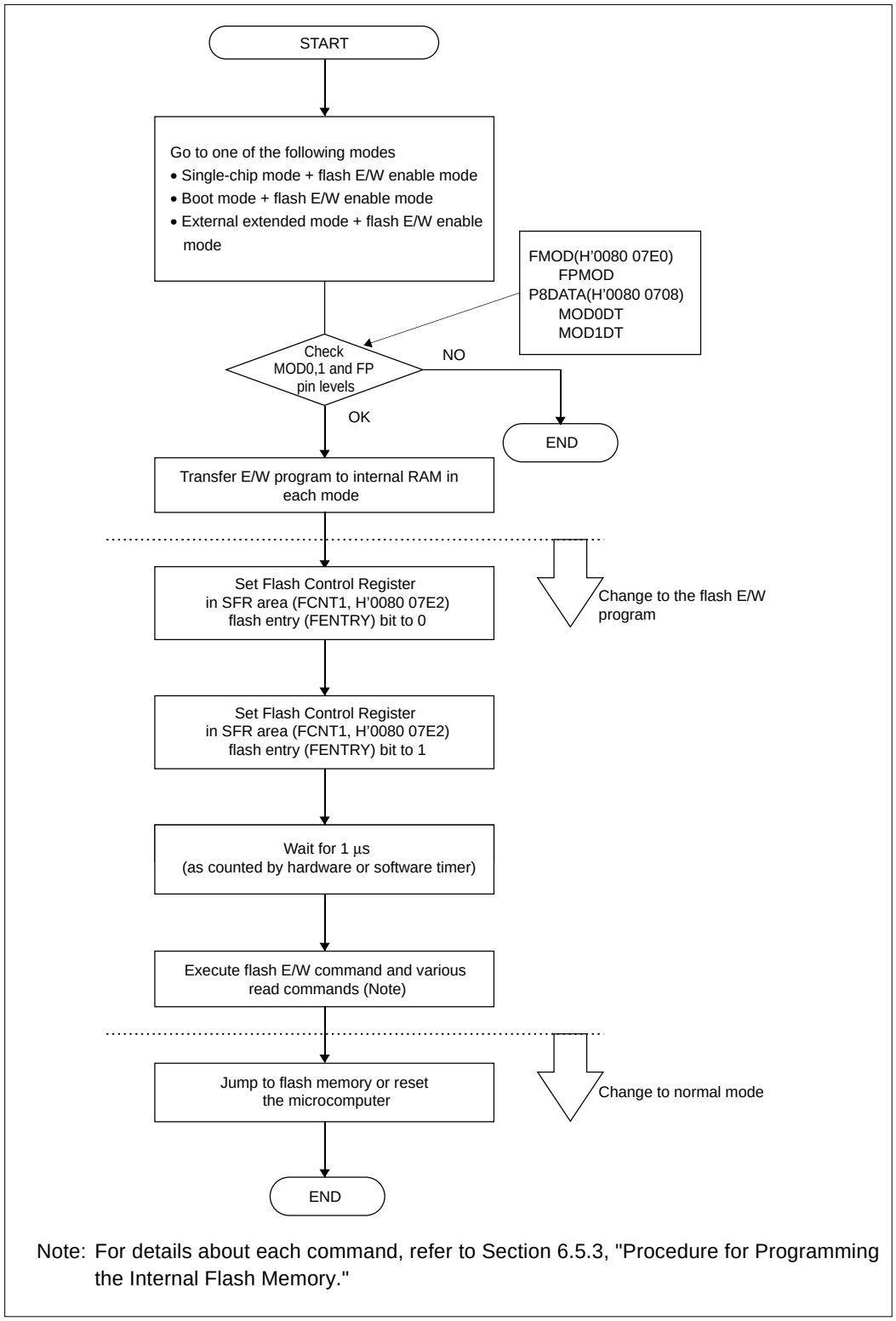
Whether the MOD0 and MOD1 pin levels (high or low) can be detected by inspecting the P8 Port Data Register (H'0080 0708) MOD0DT and MOD1DT bits.

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

D0	1	2	3	4	5	6	D7
MOD0DT	MOD1DT	P82DT	P83DT	P84DT	P85DT	P86DT	P87DT

<When reset: Indeterminate>

D	Bit Name	Function	R	W
0	MOD0DT (MOD0 data)	0: MOD0 pin = low 1: MOD0 pin = high	○	–
1	MOD1DT (MOD1 data)	0: MOD1 pin = low 1: MOD1 pin = high	○	–
2	P82DT (Port P82 data)	• When the direction bit is set to 0 (input mode) by Port Direction Register 0: Port input pin = low 1: Port input pin = high	○	○
3	P83DT (Port P83 data)		○	○
4	P84DT (Port P84 data)	• When the direction bit is set to 1 (output mode) by Port Direction Register 0: Port output latch = low 1: Port output latch = high	○	○
5	P85DT (Port P85 data)		○	○
6	P86DT (Port P86 data)		○	○
7	P87DT (Port P87 data)		○	○



Note: For details about each command, refer to Section 6.5.3, "Procedure for Programming the Internal Flash Memory."

Figure 6.5.6 Procedure for Entering Flash E/W Enable Mode

6.5.3 Procedure for Programming the Internal Flash Memory

To program the internal flash memory, use the flash write program which has been transferred from the flash memory into the internal RAM after entering flash E/W enable mode.

In flash E/W enable mode, unlike in normal mode, data cannot be read from the internal flash memory and, hence, no programs in the flash memory can be executed. Therefore, the flash write program must be prepared in the internal RAM before entering flash E/W enable mode. (Once flash E/W enable mode is entered into, accessing the flash memory is inhibited unless accessed with flash commands.)

To access the internal flash memory in flash E/W enable mode, issue a command for the internal flash memory address to be accessed. The commands that can be issued in flash E/W enable mode are listed below.

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

Table 6.5.2 Commands in Flash E/W Enable Mode

Command Name	Command Data Issued
Read Array command	H'FFFF
Page Program command	H'4141
Lock Bit Program command	H'7777
Block Erase command	H'2020
Erase All Unlock Blocks command	H'A7A7
Read Status Register command	H'7070
Clear Status Register command	H'5050
Read Lock Bit Status command	H'7171
Verify command (Note)	H'D0D0

Note: Use this command you performed Lock Bit Program, Block Erase, or Erase All Unlock Blocks operation.

(1) Read Array command

The internal flash memory is placed in read mode by writing command data H'FFFF to any flash memory address. Read the desired address, and the memory content at that address is read out.

To exit flash E/W enable mode, be sure to run the Read Array command before exiting flash E/W enable mode.

(2) Page Program command

The flash memory is programmed in units of pages, each page consisting of 256 bytes (lower address H'00-H'FF).
To write data to the flash memory (i.e., to program the flash memory), write the program command H'4141 to any address of the internal flash memory and then write the program data to the desired flash memory address.

Protected blocks of the flash memory cannot be accessed for write with the Page Program command.

Page Program is automatically performed by an internal control circuit, and the conpletion of programming can be verified by checking the Flash Status Register 1 (FSTAT1) FSTAT1 bit. (Refer to Section 6.4.2, "Flash Status Registers.") While the FSTAT1 bit = 0, the next programming can not be performed.

(3) Lock Bit Program command

The flash memory can be protected (against write and erase) in units of blocks. The Lock Bit Program command is used to protect any memory block.
Write the lock bit command data H'7777 to any address of the internal flash memory. Next, write the Verify command data H'D0D0 to the last even address of the memory block to be protected, and this memory block is protected (against write and erase). To remove protection, disable lock bit-effectuated protection using the Flash Control Register 2(FCNT2) FPROT bit (see Section 6.4.3, "Flash Control Registers") and erase the block whose protection you want to remove. (The content of this memory block is also erased.

Table 6.5.3 M32172F2/M32173F2 Target Blocks and the Addresses to Specify

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

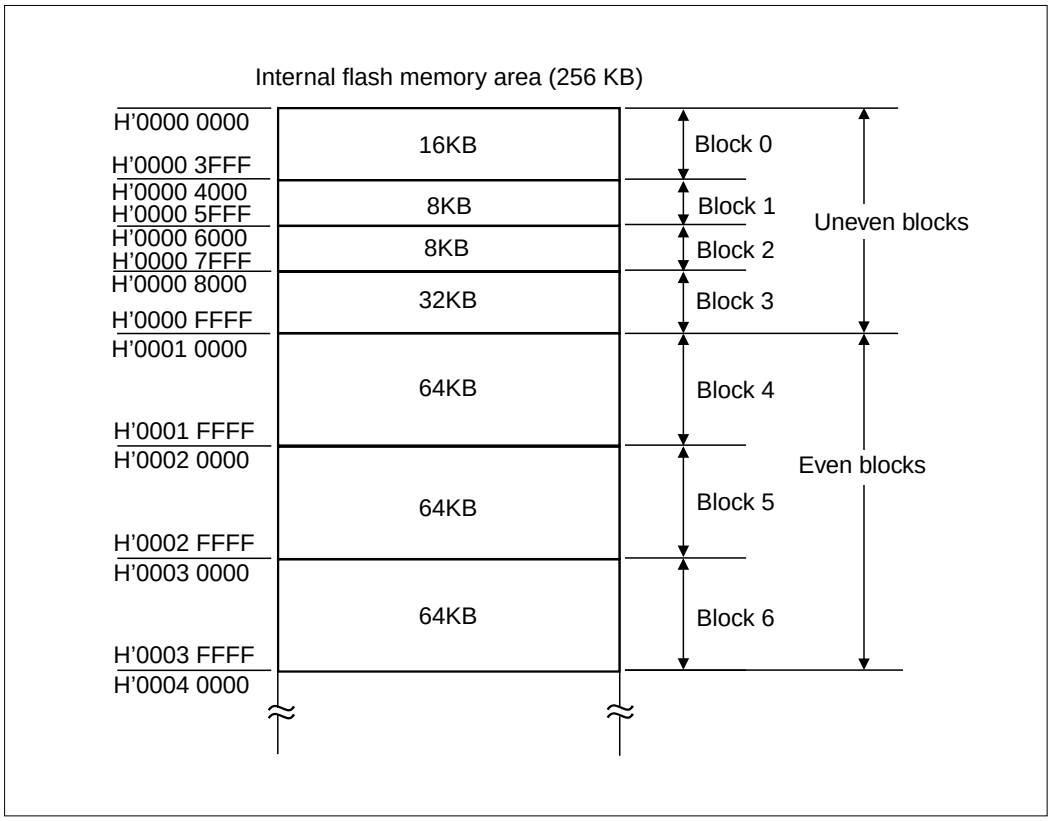


Figure 6.5.7 M32172F2/M32173F2 Flash Memory Block Configuration

(4) Block Erase command

The Block Erase command erases the contents of the internal flash memory one block at a time. For Block Erase, write the command data H'2020 to any address of the internal flash memory. Next, write the confirm command data H'D0D0 to the last even address of the memory block to be erased. (See Table 6.5.3, "Target Blocks and the Addresses to Specify.") The content of the memory block is erased.

Protected blocks of the flash memory cannot be accessed for erase with the Block Erase command.

Block Erase is automatically performed by the internal control circuit, and the completion of Block Erase can be verified by checking the Flash Status Register 1 (FSTAT1) FSTAT1 bit. (Refer to Section 6.4.2, "Flash Status Registers.") While the FSTAT1 bit = 0, you cannot erase the next block.

(5) Erase All Unlock Blocks command

The Erase All Unlock Blocks command erases all unprotected memory blocks. To erase all unprotected memory blocks, write the command data H'A7A7 to any address of the internal flash memory. Next, write the command data H'D0D0 to any address of the internal flash memory, and all of the unprotected memory blocks are erased.

(6) Read Status Register command

The Read Status Register command reads the content of Flash Status Register 2 (FSTAT2) that indicates whether programming or erase operation on flash memory has terminated normally or in an error. To read Flash Status Register 2, write the command data H'7070 to any address of the internal flash memory. Next, read any address of the internal flash memory, and Flash Status Register 2 (FSTAT2) is read out.

(7) Clear Status Register command

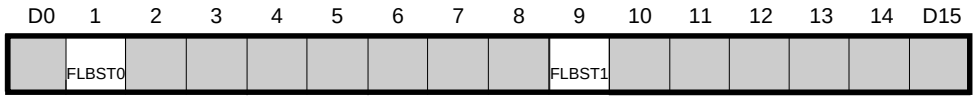
The Clear Status Register command clears the Flash Status Register 2 (FSTAT2) ERASE (auto erase operating status), WRERR1 (program operating status 1), and WRERR2 (program operating status 2) bits to 0. Write the command data H'5050 to any address of the internal flash memory, and Flash Status Register 2 is cleared to 0.

If an error occurs when programming or erasing the flash memory and the Flash Status Register 2 (FSTAT2) ERASE (auto erase operating status), WRERR1 (program operating status 1), or WRERR2 (program operating status 2) bit is set to 1, you cannot perform the next program or erase operation unless WRERR1 (Program operating condition 1) or WRERR2 (Program operating condition 2) is cleared to 0.

(8) Read Lock Bit Status command

The Read Lock Bit Status command is used to verify whether memory blocks are protected (against write/erase) or not protected. Write the command data H'7171 to any address of the internal flash memory. Next, read the last even address of the desired block (See Table 6.5.3, "Target Blocks and the Addresses to Specify."), and you know whether the block you read is protected or not.
If the FLBST0 (lock bit 0) and FLBST1 (lock bit 1) bits of the data you read are 0, it means that the memory block is protected. If the FLBST0 (lock bit 0) and FLBST1 (lock bit 1) bits of the data you read are 1, it means that the memory block is not protected.

■ **Lock Bit Status Register (FLBST)**



<When reset: Indeterminate>

D	Bit Name	Function	R	W
0	No functions assigned		?	–
1	FLBST0 (Lock bit 0)	0: Protected 1: Not protected	○	–
2-8	No functions assigned		?	–
9	FLBST1 (Lock bit 1)	0: Protected 1: Not protected (Same content as FLBST0 is output)	○	–
10-15	No functions assigned		?	–

The Lock Bit Status Register is a read-only register which contains lock bits independently for each block.

Follow the procedure below to write to the lock bits.

a) Setting the lock bits to 0 (protect a memory block)

Issue the lock bit program command (H'7777) to the memory block to be protected.

b) Setting the lock bits to 1 (unprotect a memory block)

Set the Flash Control Register 2 FPROT bit to 1 to disable protection by lock bits, then use the Block Erase command (H'2020) or Erase All Unlock Blocks command (H'A7A7) to erase the memory block you want to unprotect.
This is the only way to remove protection from a memory block. The lock bits cannot alone be set to 1.

c) Lock bit status when reset

The lock bits are nonvolatile bits, so that they are unaffected by resetting or powering off the microcomputer.

(9) Command execution flow

The diagrams below show command execution flow for each command used.

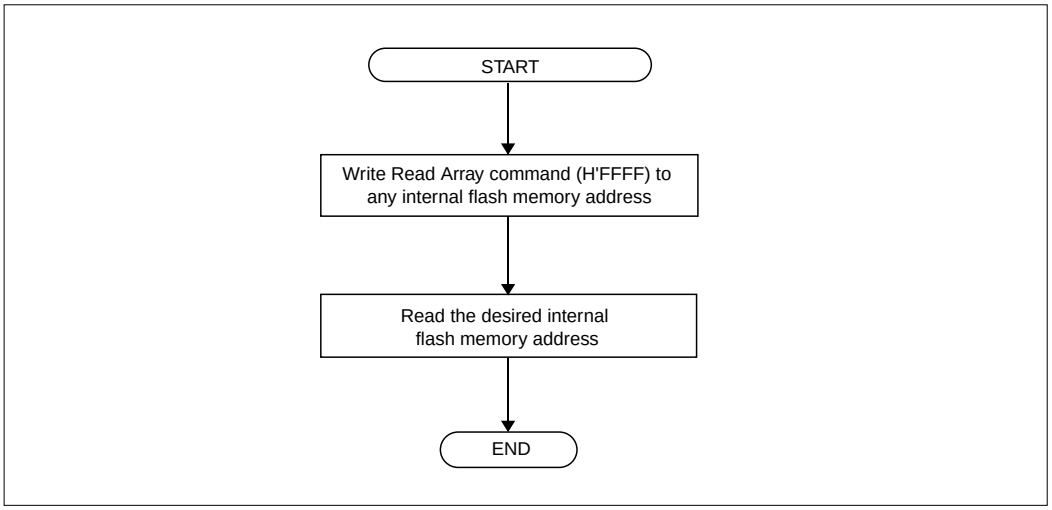
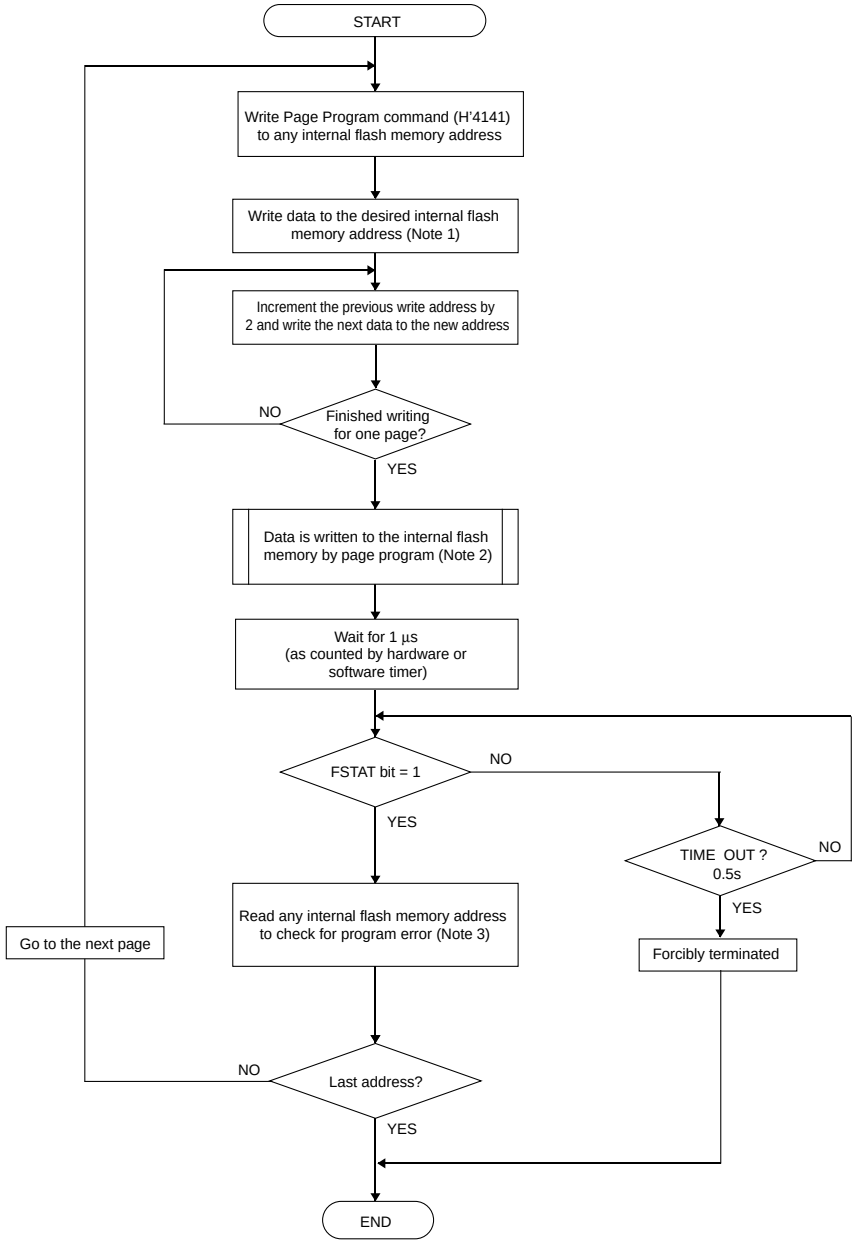


Figure 6.5.8 Read Array Command

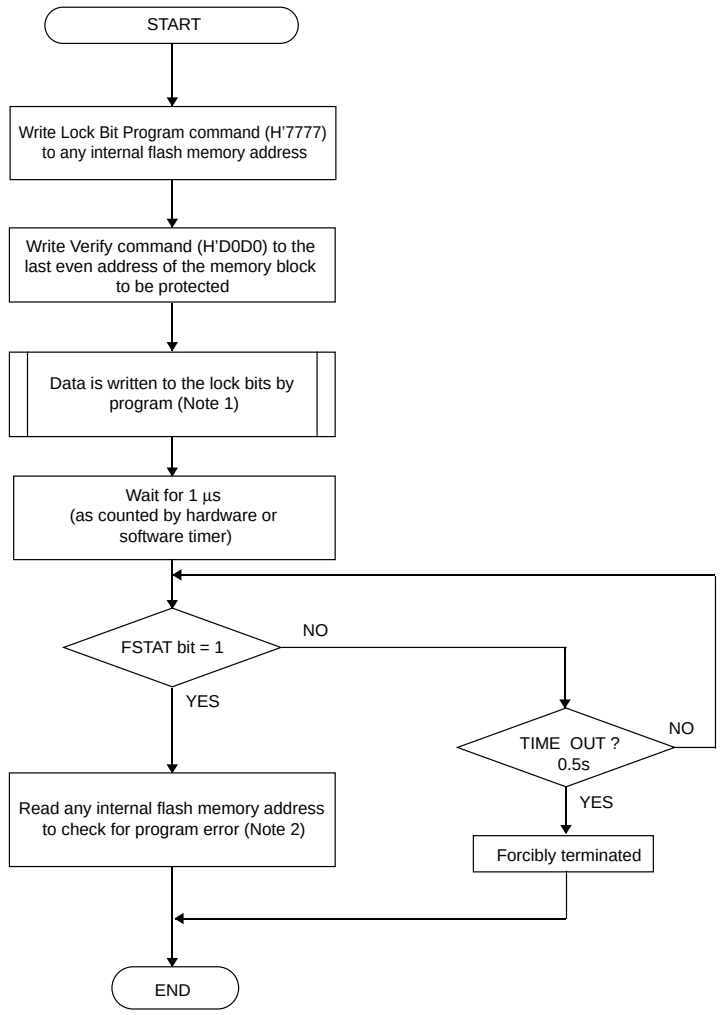


Note 1: Start writing from the beginning of the 256-byte boundary (lower address H'00).

Note 2: After programming operation starts, the flash memory is automatically in the same condition as when the Read Status Register command is entered. (There is no need to issue the Read Status Register command until entering any other command.)

Note 3: Inspect the Flash Status Register 2 (FSTAT2) ERASE (auto erase operating status), WRERR1 (program operating status 1), and WRERR2 (program operating status 2) bits to check for program errors.

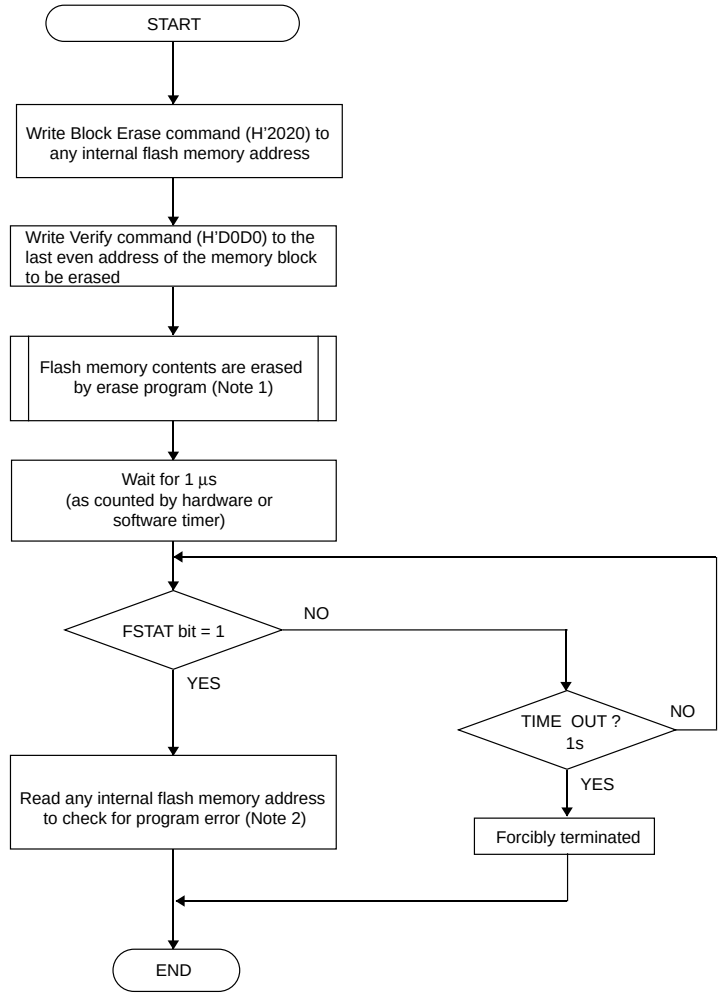
Figure 6.5.9 Page Program Command



Note 1: After programming operation starts, the flash memory is automatically in the same condition as when the Read Status Register command is entered. (There is no need to issue the Read Status Register command until entering any other command.)

Note 2: Inspect the Flash Status Register 2 (FSTAT2) ERASE (auto erase operating status), WRERR1 (program operating status 1), and WRERR2 (program operating status 2) bits to check for program errors.

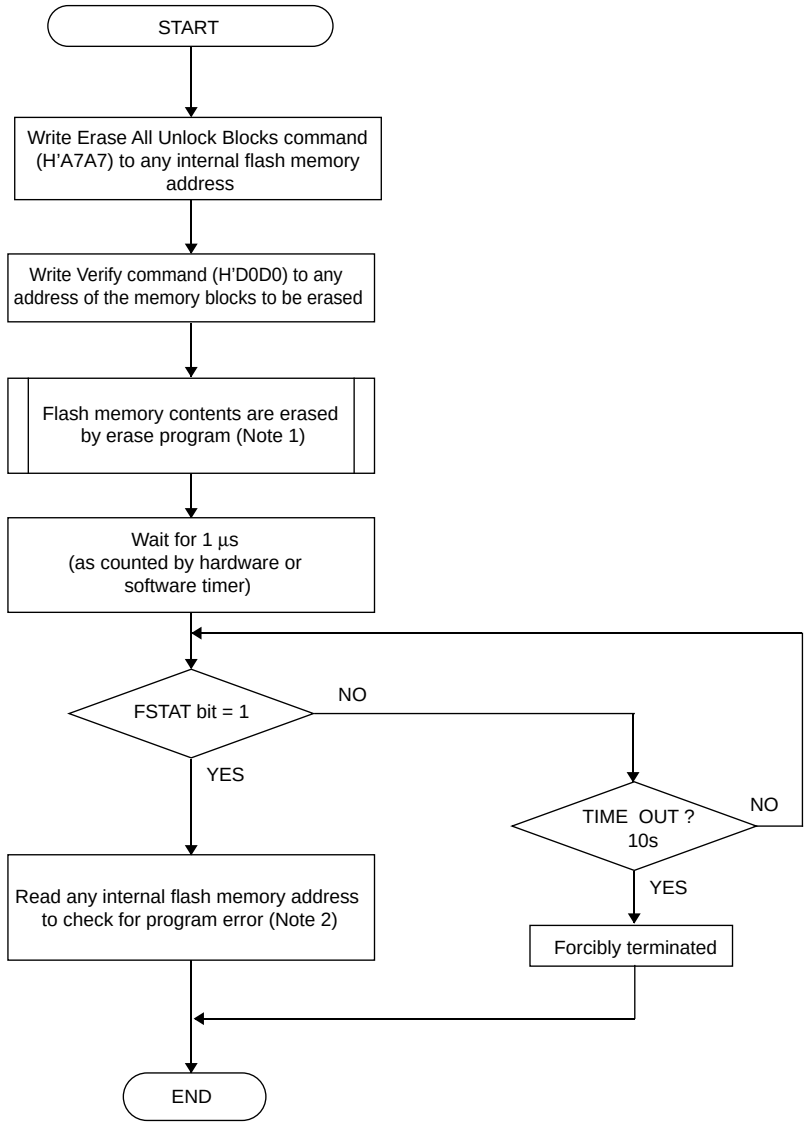
Figure 6.5.10 Lock Bit Program Command



Note 1: After erase operation starts, the flash memory is automatically in the same condition as when the Read Status Register command is entered. (There is no need to issue the Read Status Register command until entering any other command.)

Note 2: Inspect the Flash Status Register 2 (FSTAT2) ERASE (auto erase operating status), WRERR1 (program operating status 1), and WRERR2 (program operating status 2) bits to check for program errors.

Figure 6.5.11 Block Erase Command



Note 1: After erase operation starts, the flash memory is automatically in the same condition as when the Read Status Register command is entered. (There is no need to issue the Read Status Register command until entering any other command.)

Note 2: Inspect the Flash Status Register 2 (FSTAT2) ERASE (auto erase operating status), WRERR1 (program operating status 1), and WRERR2 (program operating status 2) bits to check for program errors.

Figure 6.5.12 Erase All Unlock Blocks Command

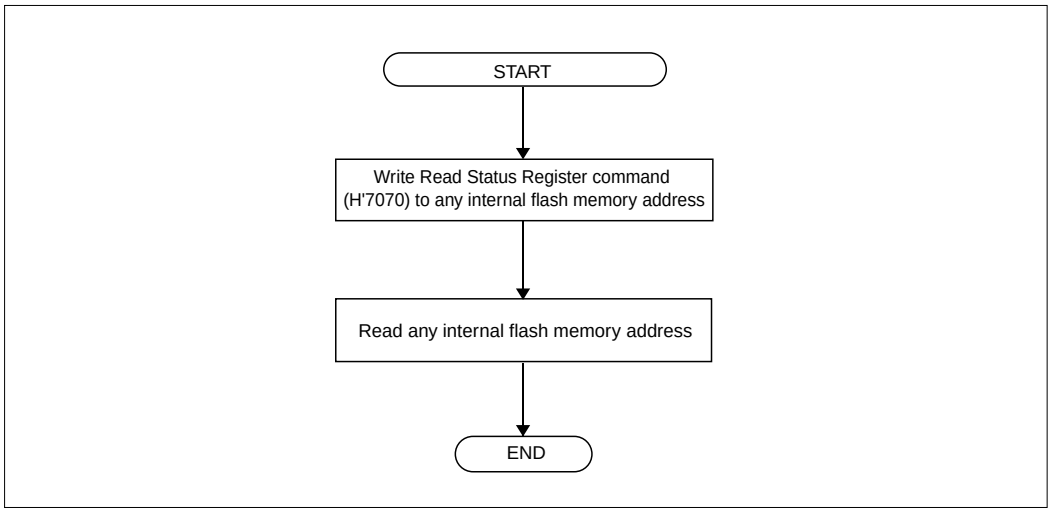


Figure 6.5.13 Read Status Register Command

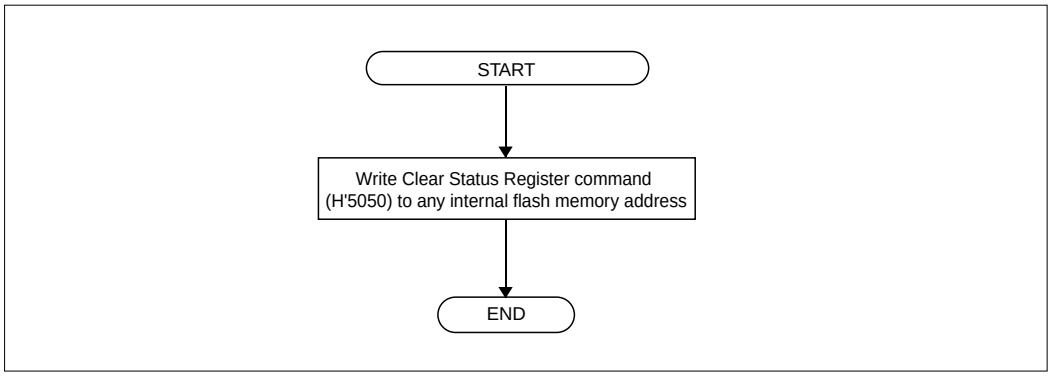


Figure 6.5.14 Clear Status Register Command

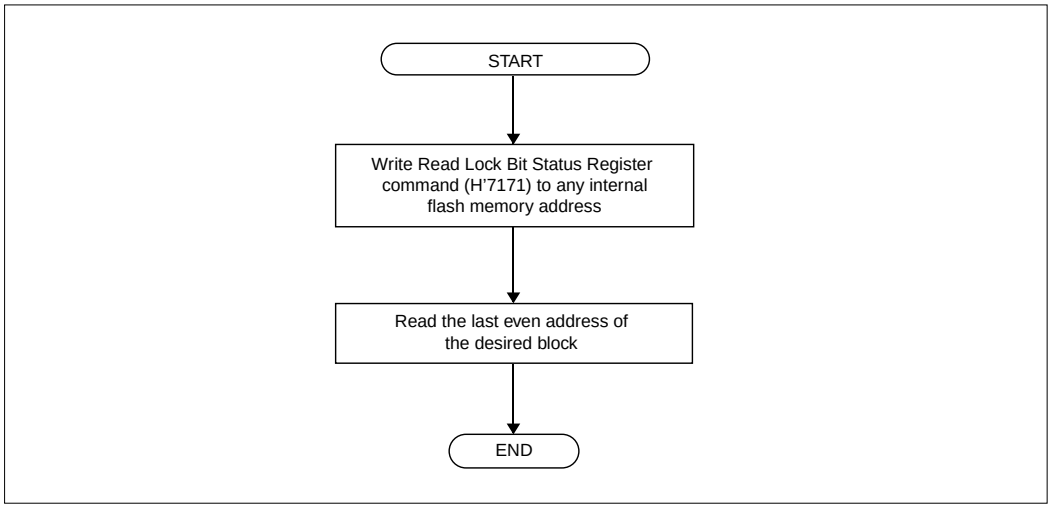


Figure 6.5.15 Read Lock Bit Status Register Command

6.5.4 Flash Programming Time (Reference Data)

The time required for programming the internal flash memory is described below for your reference.

(1)M32172F2 and M32173F2

① SIO transfer time (when transfer data size = 256 KB)

1/57600bpsx 1 (frame) x 11 (number of transfer bits) x 256 KB ≒ 50.1 [s]

② Flash programming time

256 KB/ 256-byte blocks x 8 ms ≒ 8.2 [s]

③ Erase time (entire area)

50 ms x number of blocks ≒ 350 [ms]

④ Total flash programming time (entire 256 KB area)

- When communicating in UART mode at a rate of 57,600 bps, the flash programming time is extremely short as compared with the serial communication time and can, therefore, be ignored. In this case, the flash programming time can be calculated using the equation below

  ① + ③ ≒ 50.5 [s]

- When writing data at high speed by increasing the serial communication rate or by other means, the fastest flash programming time possible is calculated using the equation below

  ② + ③ ≒ 8.6 [s]

6.6 Boot ROM

Specifications of the boot ROM are shown below.

Table 6.6.1 Specifications of the boot ROM

Item	Specification
Capacity	8 Kbytes
Location address	H'8000 0000 - H'8000 1FFF
Wait insertion	Operates with no wait cycles (when internal CPU memory clock = 40 MHz)
Internal bus connection	Connects to a 32-bit bus
Readout	Can only be read out when FP = 1, MOD0 = 1, and MOD1 = 0. If read out in any other mode, the read value is indeterminate. Cannot be accessed for write.
Other	Because the boot ROM area is a reserved area that can only be used during boot mode, the program in it cannot be modified.

6.7 Virtual-flash Emulation Function

The microcomputer can map 8-Kbyte blocks of internal RAM (up to two for the M32172F2 or up to three for the M32173F2) beginning with the start address into the internal flash memory area divided in units of 8-kbytes (L banks), and can map 4-Kbyte blocks of internal RAM (up to two for the M32173F2) beginning with address H'0080 A000 into the internal flash memory area divided in units of 4-kbyts (S banks). (The M32172F2 does not have the function for mapping into S banks of the internal flash memory.) This capability is referred to as the "virtual-flash emulation" function.
This function allows the data located in 8-Kbyte or 4-Kbyte blocks of the internal RAM to be used by switching to the L or S banks of flash memory specified by the Virtual-Flash Bank Register. Therefore, applications that require changes of data during program operation can have data dynamically changed using 8 or 4 Kbytes of RAM area. The RAM used for virtual-flash emulation can be accessed for read and write from both the internal RAM and the internal flash memory areas.
When this function is used in combination with the internal Realtime Debugger (RTD), the data tables created in the internal flash memory can be referenced or rewritten from the outside, allowing for each data table tuning from the outside.

Before writing to the internal flash memory, always be sure to terminate this virtual-flash emulation mode.

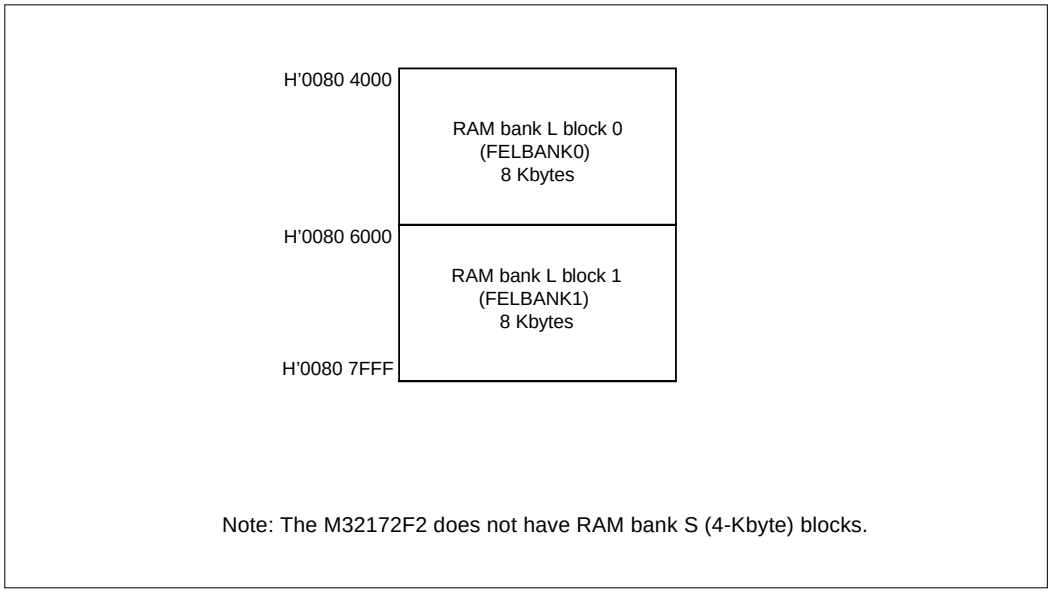


Figure 6.7.1 Configuration of the M32172F2's Internal RAM Banks

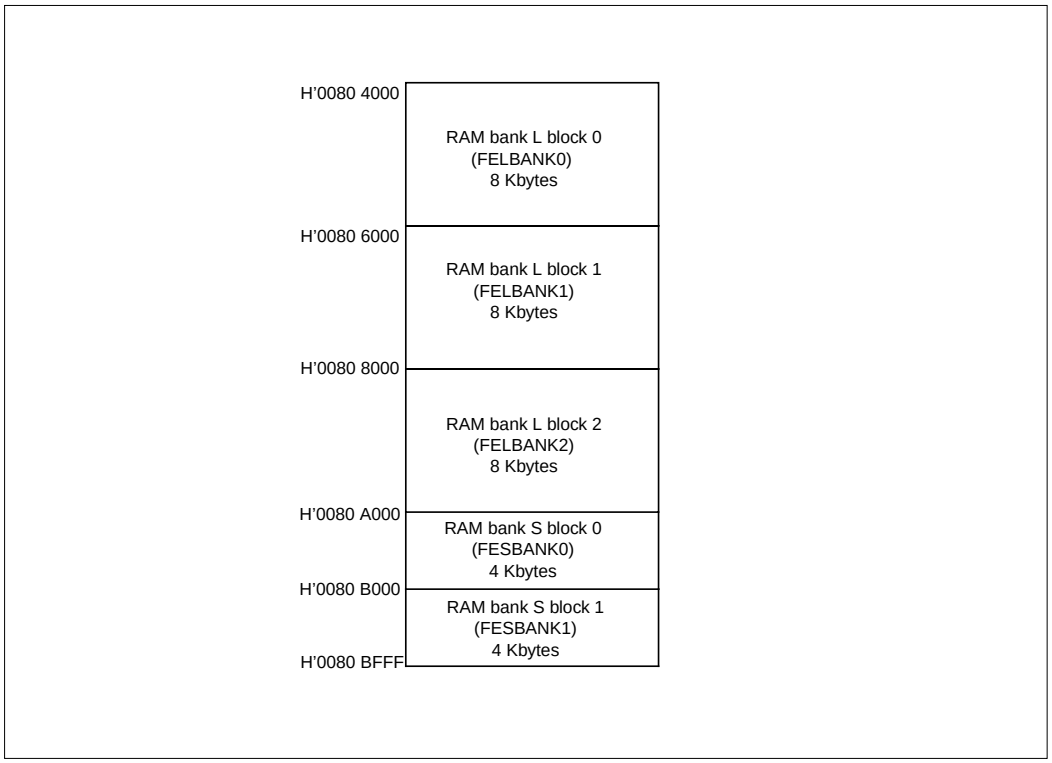


Figure 6.7.2 Configuration of the M32173F2's Internal RAM Banks

6.7.1 Virtual-Flash Emulation Areas

The following shows the areas effective for the virtual-flash emulation function.

Using the Virtual-Flash L Bank Registers, select 8-Kbyte blocks L banks of flash memory, one for each register (by setting the seven bits A12-A18 of start address of the desired L bank in the Virtual-Flash L Bank Register LBANKAD bits). Then set the Virtual-Flash L Bank Register MODENL bit to 1. The selected L bank areas can be replaced with 8-Kbyte blocks of the internal RAM beginning with its start address (up to two blocks for the M32172F2 or up to three blocks for the M32173F2). Furthermore, the M32173F2 has Virtual-Flash S Bank Registers (FESBANK0, FESBANK1). Using these registers, select 4-Kbyte blocks S banks of flash memory, one for each register (by setting the eight bits A12-A19 of start address of each desired S bank in the Virtual-Flash S Bank Register SBANKAD bits). Then set the Virtual-Flash S Bank Register MODENS0 and MODENS1 bits to 1. The selected S bank areas can be replaced with up to two 4-Kbyte blocks of the internal RAM beginning with the address H'0080 A000.
In this way, the M32172F2 can have two 8-Kbyte L banks selected. For the M32173F2, three 8-Kbyte L banks and two 4-Kbyte S banks for a total of up to five banks can be selected.

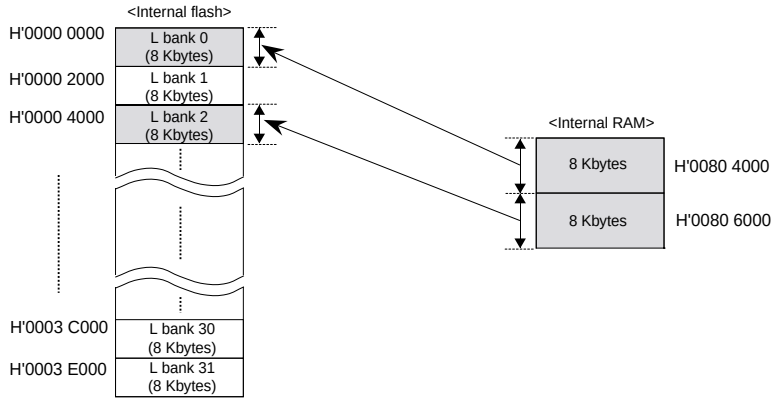
Note1: If the Virtual-Flash Emulation Enable bit is enabled while the same bank area is set in two or more Virtual-Flash Bank Registers, the internal RAM area (8 or 4 Kbyte) to be replaced with is selected by priority as follows:

- M32172F2
  FELBANK0 > FELBANK1
- M32173F2
  FELBANK0 > FELBANK1 > FELBANK2 > FESBANK0 > FESBANK1

Note 2: During virtual-flash emulation mode, the RAM can be accessed for read and write from the internal RAM area and the area that has been set as a virtual-flash area.

Note 3: When performing virtual-flash read after setting Flash Control Register 1's Virtual-Flash Emulation Mode bit to 1, be sure to wait for three CPU clock periods or more before performing virtual-flash read after setting the said bit to 1.

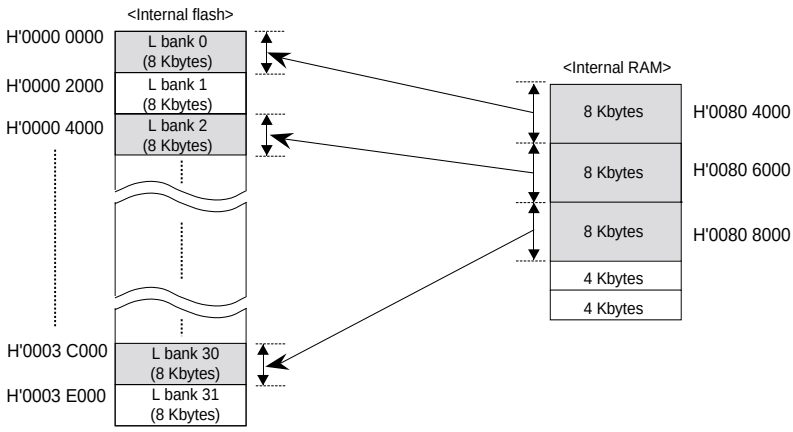
Note 4: When performing virtual-flash read after setting the Virtual-Flash Bank Register (LBank or S Bank Register)'s Virtual-Flash Emulation Enable bit and bank address bits, be sure to wait for three CPU clock periods or more before performing virtual-flash read after setting the Virtual-Flash Bank Register.



Note 1: If the Virtual-Flash Emulation Enable bit is enabled while the same bank area is set in two or more Virtual-Flash Bank Registers, the internal RAM area to be replaced with is selected by priority: FELBANK0 > FELBANK1.

Note 2: When access is made to the 8-Kbyte area (L bank) selected by Virtual-Flash L Bank Register 0 or 1, what is actually accessed is the internal RAM area. During virtual-flash emulation mode, the RAM can be read and written to from both the internal RAM and the selected virtual-flash memory areas.

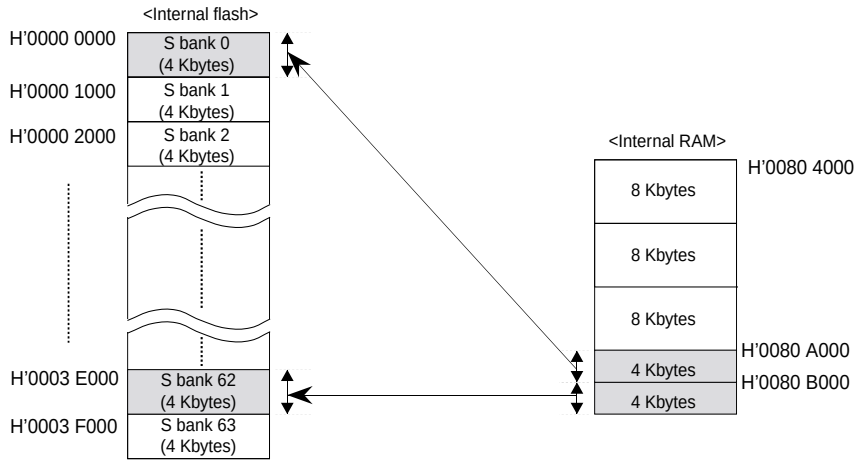
Figure 6.7.3 Virtual-Flash Emulation Areas of the M32172F2 Divided in Units of 8 Kbytes



Note 1: If the Virtual-Flash Emulation Enable bit is enabled while the same bank area is set in two or more Virtual-Flash Bank Registers, the internal RAM area to be replaced with is selected by priority: FELBANK0 > FELBANK1 > FELBANK2 > FESBANK0 > FESBANK1.

Note 2: When access is made to the 8-Kbyte area (L bank) selected by Virtual-Flash L Bank Registers 0-2, what is actually accessed is the internal RAM area. During virtual-flash emulation mode, the RAM can be read and written to from both the internal RAM and the selected virtual-flash memory areas.

Figure 6.7.4 Virtual-Flash Emulation Areas of the M32172F2 Divided in Units of 8 Kbytes



Note 1: If the Virtual-Flash Emulation Enable bit is enabled while the same bank area is set in two or more Virtual-Flash Bank Registers, the internal RAM area to be replaced with is selected by priority: FELBANK0 > FELBANK1 > FELBANK2 > FESBANK0 > FESBANK1.

Note 2: When access is made to the 4-Kbyte area (S bank) selected by Virtual-Flash S Bank Register 0 or 1, what is actually accessed is the internal RAM area. During virtual-flash emulation mode, the RAM can be read and written to from both the internal RAM and the selected virtual-flash memory areas.

Figure 6.7.5 Virtual-Flash Emulation Areas of the M32173F2 Divided in Units of 4 Kbytes

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

Note: Set seven bits A12-A18 of the start address (32-bit) of each L bank of flash memory divided every 8-kbyte in the Virtual-Flash L Bank Register L bank address (LBANKAD) bits.

Figure 6.7.6 Values Set in the M32172F2's Virtuar-Flash Bank Register when Divided in Units of 8-kbyte

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

Note: Set eight bits A12-A19 of the start address (32-bit) of each S bank of flash memory divided every 4-kbyte in the Virtual-Flash S Bank Register S bank address (SBANKAD) bits.

Figure 6.7.7 Values Set in the M32173F2's Virtual-Flash Bank Register when Divided in Units of 8-kbyte

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

Note: Set eight bits A12-A19 of the start address (32-bit) of each S bank of flash memory divided every 4-Kbyte in the Virtual-Flash S Bank Register S bank address (SBANKAD) bits.

Figure 6.7.8 Values Set in the M32173F2's Virtual-Flash Bank Register when Divided in Units of 4-Kbyte

6.7.2 Transition to Virtual-Flash Emulation Mode

To enter virtual-flash emulation mode, set the Flash Control Register 1 (FCNT1) FEMMOD bit to 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 7FFF for the M32172F2 or H'0080 4000 through H'0080 BFFF for the M32173F2) can be accessed as internal RAM.

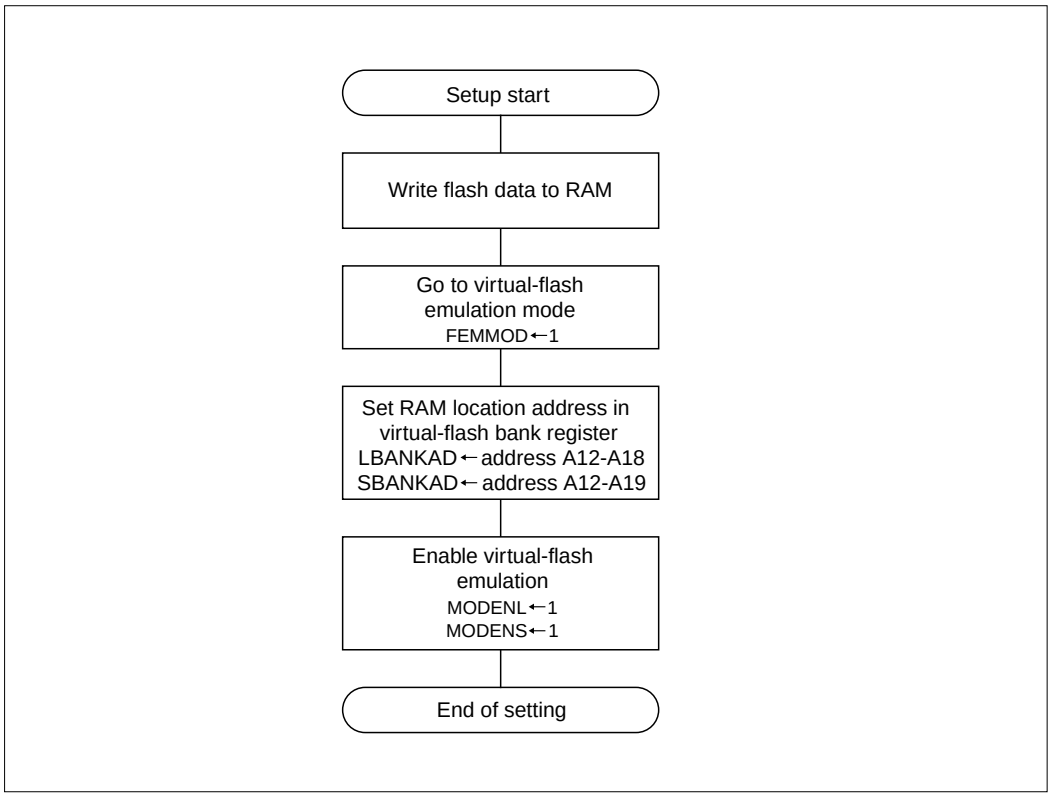


Figure 6.7.9 Setup Sequence for Virtual-Flash Emulation Mode

6.7.3 Application Example for Virtual-Flash Emulation Mode

By locating two RAM areas in the same virtual-flash area using the Virtual-Flash Emulation Function, you can rewrite data in the flash memory successively.

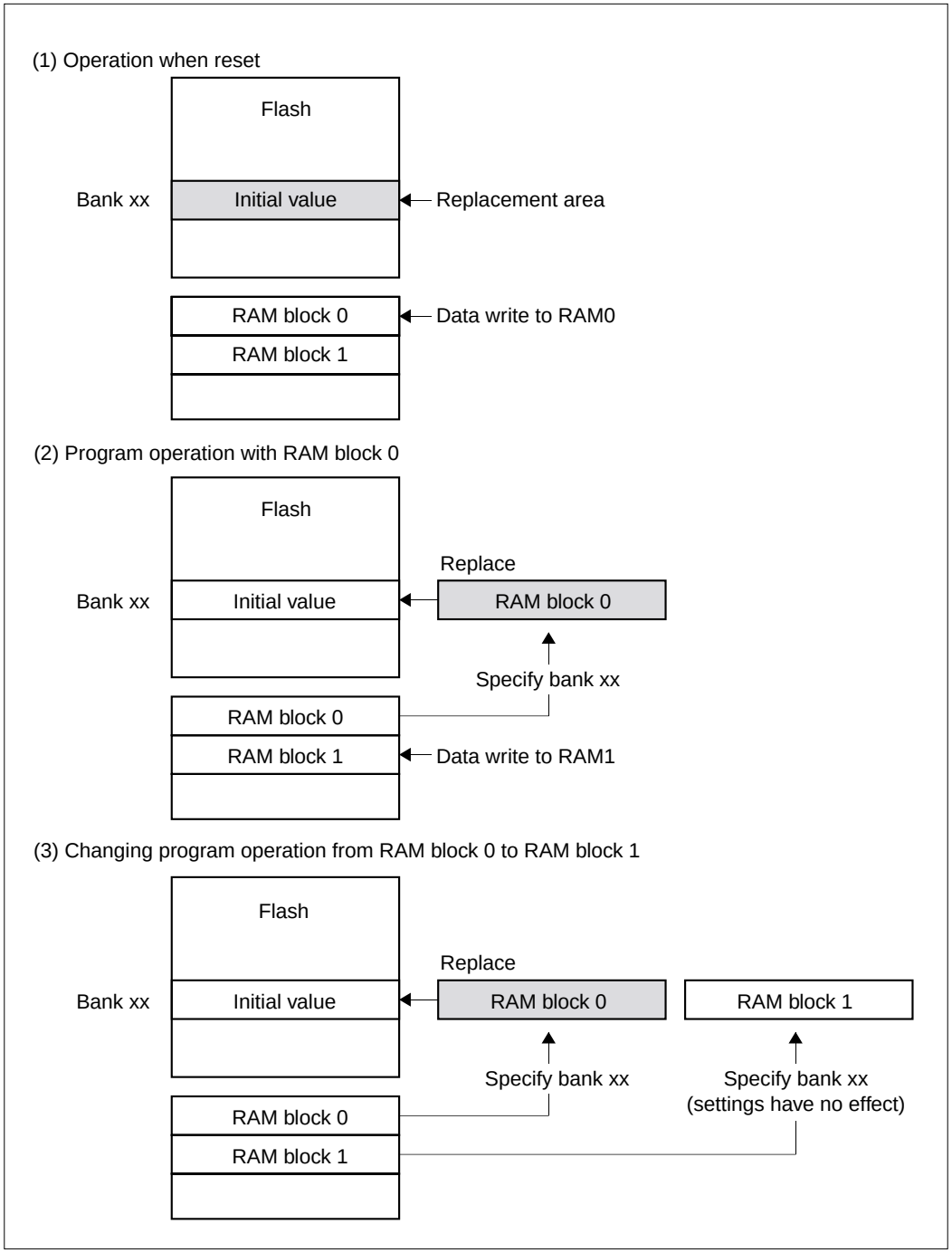


Figure 6.7.10 Usage Example for Virtual-Flash Emulation (1/2)

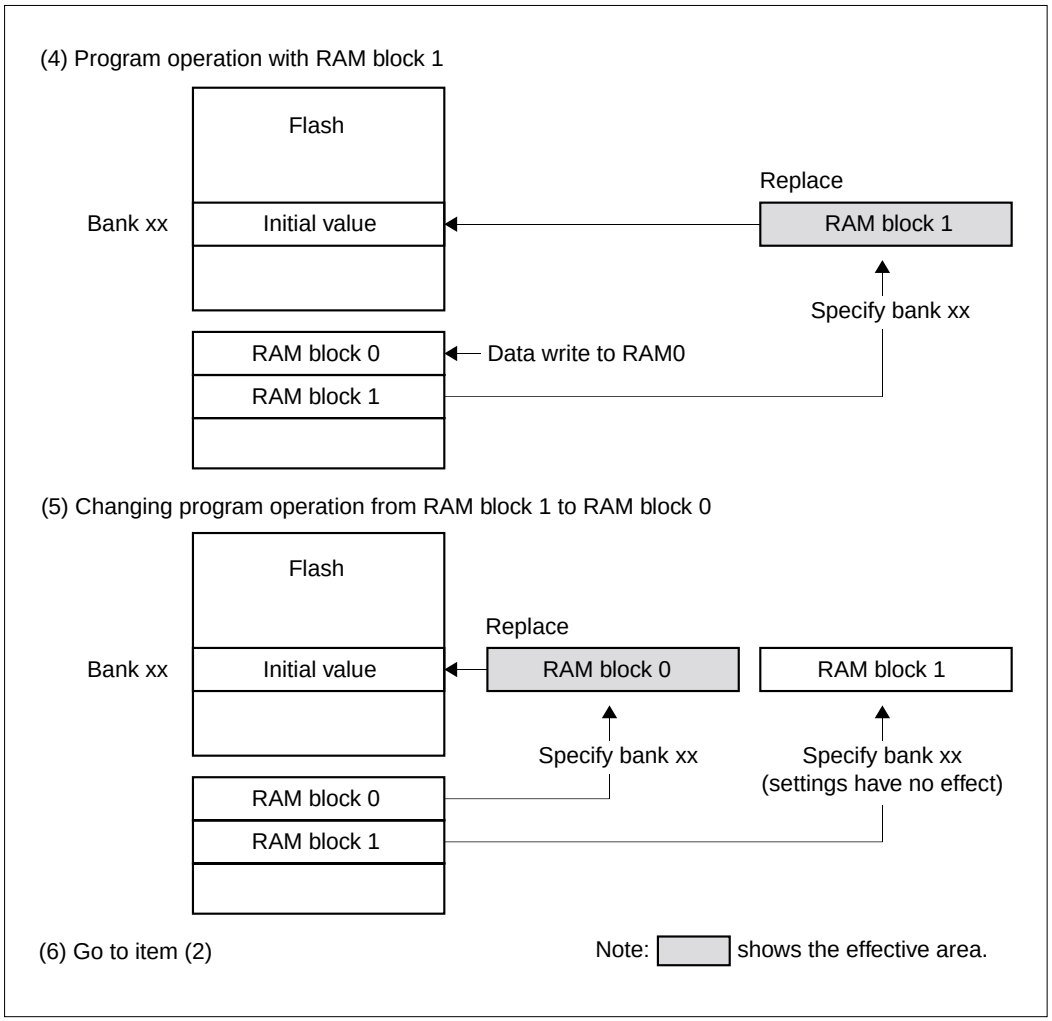


Figure 6.7.11 Usage Example for Virtual-Flash Emulation (2/2)

6.8 Connecting a Serial Programmer

To rewrite the internal flash memory using a general-purpose serial programmer in boot mode + flash E/W enable mode, pins on the microcomputer listed below need to be processed to be suitable for use with the serial programmer.

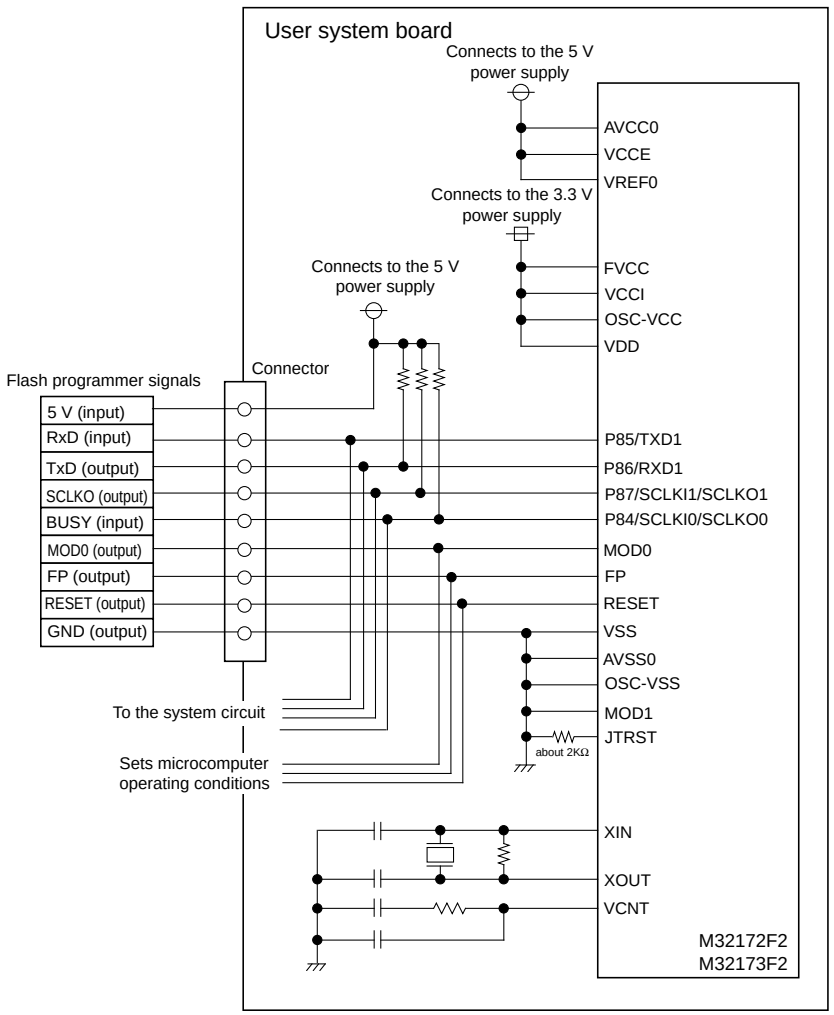
Table 6.8.1 Processing Pins when Using a Serial Programmer

Pin Name	Pin No.	Function	Remark
SCLKI1	71	Transfer clock input	Pullup required
RXD1	70	Serial data input (received data)	Pullup required
TXD1	69	Serial data output (transmit data)	
P84	68	Transmit/receive enable output	Pullup required
FP	94	Flash memory protect	
MOD0	92	Operation mode 0	
MOD1	93	Operation mode 1	Connect to ground
$\overline{\text{RESET}}$	91	Reset	
XIN	4	Clock input	
XOUT	5	Clock output	
VCNT	7	PLL circuit control input	
OSC-VCC	6	PLL circuit power supply	Connect to 3.3 V power supply
OSC-VSS	3	PLL circuit ground	Connect to ground
VREF0	42	A-D converter reference voltage input	Connect to 5 V power supply
AVCC0	43	Analog power supply	Connect to 5 V power supply
AVSS0	60	Analog ground	Connect to ground
FVCC	73	Flash memory power supply	Connect to 3.3 V power supply
VDD	108	RAM backup power supply	Connect to 3.3 V power supply
VCCE	20,65,95,132	5 V power supply	
VCCI	61,123,137	3.3 V power supply	
VSS	21,62,72,96,138	Ground	

Note: Any other pins do not need to be processed.

The diagram below shows a typical user system configuration with a serial programmer connected. After the user system is powered on, the serial programmer is used in clock-synchronized serial mode to write data to the flash memory. In this configuration, no oscillation frequency-dependent problems with communication may occur.

If the pins connecting to the serial programmer are used by the system, measures must be taken to prevent any adverse effect on operation of the serial programmer. Note that addresses H'0000 0084 through H'0000 0093 are used as a flash memory protect ID check area by the serial programmer.



Note 1: Turn on the power to the user system before writing to the flash memory.
Note 2: When using P84-P87 in the system circuit, consideration must be taken for connection to the serial programmer.
Note 3: Measures must be taken for P64/SBI not to generate interrupts by fixing it high or low.
Note 4: Select the appropriate pullup resistance values for P84, P86, and P87 according to the system design conditions.
Note 5: The typical pullup resistance values for P84, P86, and P87 are 4.7 to 10 kΩ.
Note 6: The status of the ports not shown above do not affect writing to the flash memory.

Figure 6.8.1 Pin Connection Diagram for the 32172

6.9 Precautions on Rewriting Flash Memory

The following describes precautions to be observed when rewriting the internal flash memory using a serial programmer in boot mode + flash E/W enable mode.

- When you use the pins with the system that are used by a serial programmer, take measures not to affect the system when connecting a serial programmer.
- If the flash memory needs to be protected, set any ID in the flash memory protect ID check area (H'0000 0084-H'0000 0093).
- If the flash memory does not need to be protected, fill the flash memory protect ID check area (H'0000 0084-H'0000 0093) with data H'FF.

CHAPTER 7
RESET

7.1 Outline of Reset

The microcomputer is reset by applying a low-level signal to the $\overline{\text{RESET}}$ input pin. The microcomputer is gotten out of a reset state by releasing the $\overline{\text{RESET}}$ input back high, upon which the reset vector entry address is set in the Program Counter (PC) and the program starts executing from the reset vector entry.

7.2 Reset Operation

7.2.1 Power-on Reset

When powering on the microcomputer, hold the input signal on the $\overline{\text{RESET}}$ pin low until the internal multiply-by-4 clock generator becomes oscillating stably.

7.2.2 Reset during Operation

To reset the microcomputer during operation, hold the $\overline{\text{RESET}}$ input low for more than four clock periods of XIN signal.

7.2.3 Reset Vector Movement during Flash Rewrite

When placed in boot mode, the microcomputer's reset vector entry address changes to the start address of the boot program space (address H'8000 0000). For details, refer to Section 6.5, "Programming the Internal Flash Memory."

7.3 Internal State Immediately after Reset

The table below shows the internal state of the microcomputer immediately after reset. For details about the initial state of internal peripheral I/O registers, refer to the relevant section of this manual where each internal peripheral I/O in interest is described.

Table 7.3.1 Internal State Immediately after Reset

Register		State after reset
PSW	(CR0)	B'0000 0000 0000 0000 ??00 000? 0000 0000 (BSM, BIE, BC bits = indeterminate)
CBR	(CR1)	H'0000 0000 (C bit = 0)
SPI	(CR2)	Indeterminate
SPU	(CR3)	Indeterminate
BPC	(CR6)	Indeterminate
PC		H'0000 0000 (Executed beginning with address H'0000 0000) (Note)
ACC (Accumulator)		Indeterminate

Note: During boot mode, this is located at the start address of the boot program space (address H'8000 0000).

Table 7.3.2 Pin Status When Reset

Pin name		Single chip mode	External extended mode	Microprocessor mode	Boot
RESET, MOD0, MOD1, and FP		Input	Input	Input	Input
Port	P0, P1	Input	Input	Input	Input
	P2, P3	Input	Input	Hi-Z	Input
	P41 P42 P43 P45 P46	Input	Input	Hi-Z	Input
	P47	Input	Input	Hi-Z	Input
	P6, P7	Input	Input	Input	Input
	P8, P12 P9, P13 P10, P15 P11, P17 P220, P221	Input	Input	Input	Input
	P225	Input	Input	Hi-Z	Input
	DA0, DA1	Input	Input	Input	Input
	AD0IN0-7 AD1IN0-3	Input	Input	Input	Input
JTAG	JTMS	Input	Input	Input	Input
	JTCK	Input	Input	Input	Input
	JTRST	Input	Input	Input	Input
	JTDO	Indeterminate	Indeterminate	Indeterminate	Indeterminate
	JTDI	Input	Input	Input	Input

Note: The state of JTAG pins cannot be initialized by a reset.When the low-level signal is applied to JTRST pin, the state of JTAG pins is initialized.

7.4 Precautions to Be Taken Immediately after Reset

- **Input/output ports**

  After reset release, its input/output ports are disabled against input in order to prevent electric current from flowing through the pins. To use any ports in input mode, enable them for input using the Port Input Function Enable Register (PIEN)'s PIEN0 bit. For details, refer to Section 8.3, "Input/ Output Port Related Registers."

*** This is a blank page ***

CHAPTER 8

INPUT/OUTPUT PORTS AND PIN FUNCTIONS

8.1 Outline of Input/Output Ports

The 32172/32173 has a total of 99 input/output ports connecting to external pins, comprised of P0-P13, P15, P17, and P22 (with P5 reserved for future use). These input/output ports can be used as input ports or output ports by setting up the direction registers.

The input/output ports each are shared with other internal peripheral I/O or external extended bus signal lines, thus comprising dual-function or triple-function pins. The pin functions are selected depending on selected chip operation mode or by using the input/output port operation mode registers. (If any internal peripheral I/O has still another function, the register for that internal peripheral I/O needs to be set to select the desired pin function.)
Each input port contains a port input function enable bit that may be used to prevent electric current from flowing into the input port. This helps to simplify hardware and software processing to be performed immediately after reset or when rewriting the flash memory.
For any port to be used in input mode, a port input function enable bit for that port needs to be set.

The input/output ports are outlined in the next page.

Table 8.1.1 Outline of the Input/Output Ports

Item	Specification
Number of ports (Note 1)	Total 99 bits P0 : P00-P07 (8 bits) P1 : P10-P17 (8 bits) P2 : P20-P27 (8 bits) P3 : P30-P37 (8 bits) P4 : P41-P47 (7 bits) P6 : P61-P64 (4 bits) P7 : P70-P77 (8 bits) P8 : P82-P87 (6 bits) P9 : P93-P97 (5 bits) P10 : P100-P107 (8 bits) P11 : P110-P117 (8 bits) P12 : P124-P127 (4 bits) P13 : P130-P137 (8 bits) P15 : P150, P153 (2 bits) P17 : P172-P175 (4 bits) P22 : P220, P221, P225, (3 bits)
Port function	Each port can be set for input or output mode by using the input/output port direction control register. (However, P64 is a $\overline{SBI}$ input-only port; P97 is a CAN1 input-only port; P93, P124-P127, P130-P137, P172, and P173 are input-only ports; P221 is a CAN0 input-only port.)
Pin function	Dual functions with peripheral I/O or external extended signals (or multiple functions with two or more peripheral I/O functions)
Pin function selection	P0-P4(Note2), P225 : Depends on CPU operation mode (determined by setting MOD0 and MOD1 pins) P6(Note3)-P22 : Selected by setting input/output port operation mode registers. (However, peripheral I/O pin functions are selected using peripheral I/O registers.)

Note 1: P14, P16, and P18-P21 are nonexistent.

Note 2: Only when the CPU is operating in external extended mode, P0-P4 (except P46) have their pin functions switched by setting operation mode registers. (When the CPU is operating in single-chip or processor mode, the pin functions of these ports are switched depending on CPU operation mode.)

Note 3: P16-P63 are always input/output ports (single function pins).

8.2 Selecting Pin Functions

Each input/output port serves dual functions sharing the pin with other internal peripheral I/O or extended external bus signal lines (or triple functions sharing the pin with two or more functions of peripheral I/O). Pin functions are selected depending on the operation modes set or by using the input/output port operation mode registers.

When the CPU is set to opreate in external extended or processor mode, P0-P4 and P225 all are switched to the signal pins needed for external access. The CPU operation mode is determined by setting the MOD0 and MOD1 pins. (See the table below.)

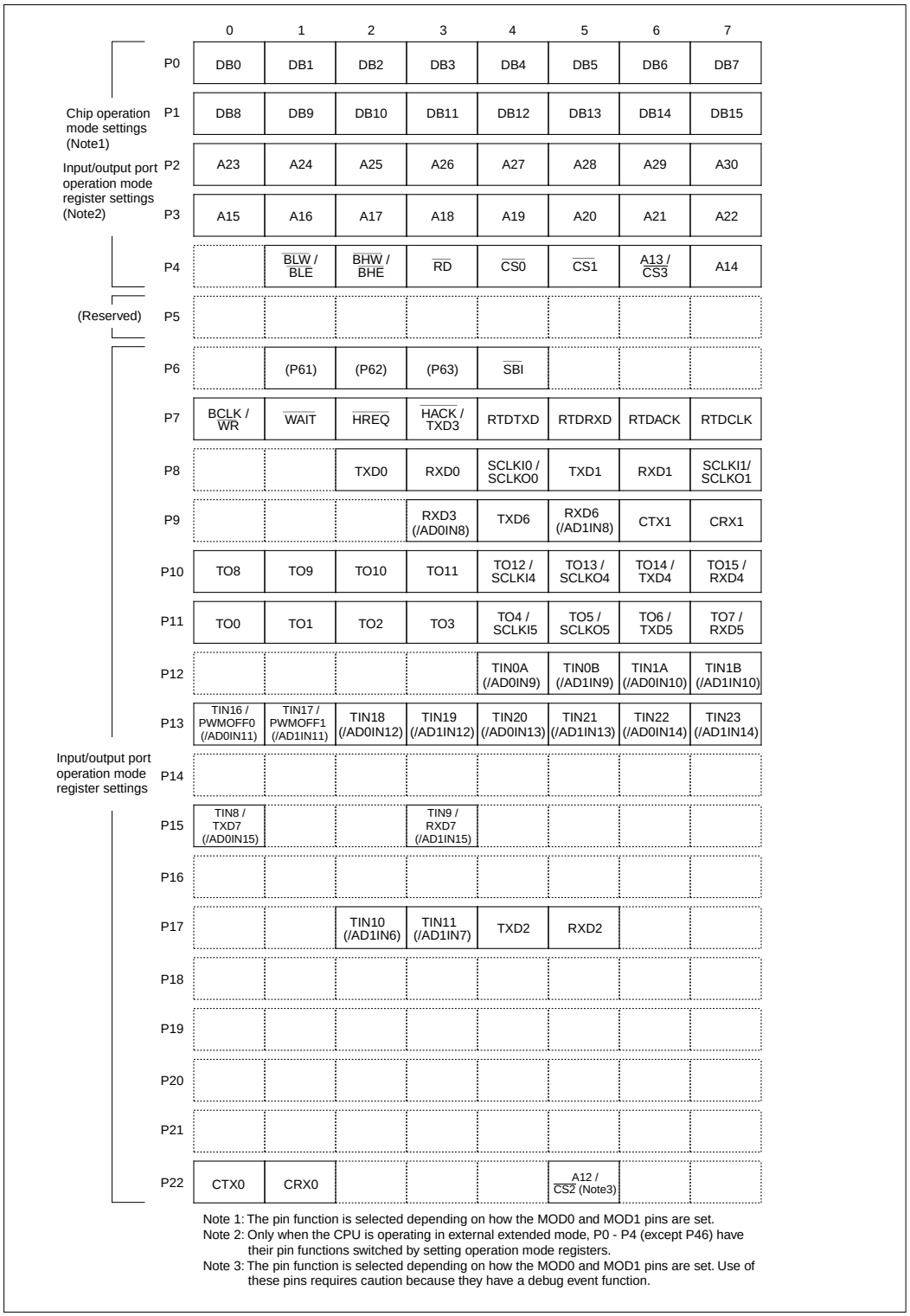
Table 8.2.1 CPU Operation Modes and P0-P4 and P225 Pin Functions

MOD0	MOD1	Operation Mode	P0-P4 Pin Function (except P46)	P46 and P225 Pin Function
VSS(Note1)	VSS	Single-chip mode	Input/output port pin	Input/output port pin
VSS	VCC	External extended mode	External extended signal pin or input/output port pin (Note 2)	External extended signal pin
VCC	VSS	Processor mode	External extended signal pin	External extended signal pin
VCC	VCC	Reserved (use inhibited)	—	—

Note 1: VCC connects to +5 V, and VSS connects to GND.
Note 2: Only when the CPU is operating in external extended mode, P0-P4 (except P46) have their pin functions switched by setting operation mode registers.

Only when the PCU is operating in external extended mode, P0-P4 (except P46) have their pin functions switched between input/output port pins and internal peripheral I/O pins by setting up the input/output port operation mode registers. Ports P6-P13, P15, P17, and P22 (except P6-P64, P97, P221, P225) have their pin functions switched between input/output port and internal peripheral I/O pins by setting up the input/output port operation mode registers. If any internal peripheral I/O has multiple functions, select the desired pin function using the relevant internal peripheral I/O register.

Operation on FP and MOD1 pins during write to the internal flash memory does not affect the pin functions.



	0	1	2	3	4	5	6	7
P0	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7
P1	DB8	DB9	DB10	DB11	DB12	DB13	DB14	DB15
P2	A23	A24	A25	A26	A27	A28	A29	A30
P3	A15	A16	A17	A18	A19	A20	A21	A22
P4		$\overline{BLW}$ / BLE	$\overline{BHW}$ / BHE	$\overline{RD}$	$\overline{CS0}$	$\overline{CS1}$	A13 / $\overline{CS3}$	A14
P5								
P6		(P61)	(P62)	(P63)	$\overline{SBI}$			
P7	BCLK / $\overline{WR}$	$\overline{WAIT}$	$\overline{HREQ}$	$\overline{HACK}$ / TXD3	RTDTXD	RTDRXD	RTDACK	RTDCLK
P8			TXD0	RXD0	SCLKI0 / SCLKO0	TXD1	RXD1	SCLKI1/ SCLKO1
P9				RXD3 (/AD0IN8)	TXD6	RXD6 (/AD1IN8)	CTX1	CRX1
P10	TO8	TO9	TO10	TO11	TO12 / SCLKI4	TO13 / SCLKO4	TO14 / TXD4	TO15 / RXD4
P11	TO0	TO1	TO2	TO3	TO4 / SCLKI5	TO5 / SCLKO5	TO6 / TXD5	TO7 / RXD5
P12					TIN0A (/AD0IN9)	TIN0B (/AD1IN9)	TIN1A (/AD0IN10)	TIN1B (/AD1IN10)
P13	TIN16 / PWMOFF0 (/AD0IN11)	TIN17 / PWMOFF1 (/AD1IN11)	TIN18 (/AD0IN12)	TIN19 (/AD1IN12)	TIN20 (/AD0IN13)	TIN21 (/AD1IN13)	TIN22 (/AD0IN14)	TIN23 (/AD1IN14)
P14								
P15	TIN8 / TXD7 (/AD0IN15)			TIN9 / RXD7 (/AD1IN15)				
P16								
P17			TIN10 (/AD1IN6)	TIN11 (/AD1IN7)	TXD2	RXD2		
P18								
P19								
P20								
P21								
P22	CTX0	CRX0				A12 / $\overline{CS2}$ (Note3)		

Note 1: The pin function is selected depending on how the MOD0 and MOD1 pins are set.
Note 2: Only when the CPU is operating in external extended mode, P0 - P4 (except P46) have their pin functions switched by setting operation mode registers.
Note 3: The pin function is selected depending on how the MOD0 and MOD1 pins are set. Use of these pins requires caution because they have a debug event function.

Figure 8.2.1 Input/Output Ports and Pin Function Assignments

8.3 Input/Output Port Related Registers

The input/output port related registers consist of the Port Data Register, Port Direction Register, and Port Operation Mode Register. Ports P0-P4 and P225 have their pin functions determined depending on CPU operation mode (selected with the FP, MOD0, and MOD1 pins).
Port P5 is reserved for future use. An input/output port related register map is shown below.

Note: When the CPU is operating in single-chip or processor mode, the pin functions of these ports are switched depending on CPU operation mode.

Address	+0 Address D0 – D7	+1 Address D8 – D15
H'0080 0700	P0 Data Register (P0DATA)	P1 Data Register (P1DATA)
H'0080 0702	P2 Data Register (P2DATA)	P3 Data Register (P3DATA)
H'0080 0704	P4 Data Register (P4DATA)	
H'0080 0706	P6 Data Register (P6DATA)	P7 Data Register (P7DATA)
H'0080 0708	P8 Data Register (P8DATA)	P9 Data Register (P9DATA)
H'0080 070A	P10 Data Register (P10DATA)	P11 Data Register (P11DATA)
H'0080 070C	P12 Data Register (P12DATA)	P13 Data Register (P13DATA)
H'0080 070E		P15 Data Register (P15DATA)
H'0080 0710		P17 Data Register (P17DATA)
H'0080 0712		
H'0080 0714		
H'0080 0716	P22 Data Register (P22DATA)	
H'0080 0720	P0 Direction Register (P0DIR)	P1 Direction Register (P1DIR)
H'0080 0722	P2 Direction Register (P2DIR)	P3 Direction Register (P3DIR)
H'0080 0724	P4 Direction Register (P4DIR)	
H'0080 0726	P6 Direction Register (P6DIR)	P7 Direction Register (P7DIR)
H'0080 0728	P8 Direction Register (P8DIR)	P9 Direction Register (P9DIR)
H'0080 072A	P10 Direction Register (P10DIR)	P11 Direction Register (P11DIR)
H'0080 072C		
H'0080 072E		P15 Direction Register (P15DIR)
H'0080 0730		P17 Direction Register (P17DIR)
H'0080 0732		
H'0080 0734		
H'0080 0736	P22 Direction Register (P22DIR)	

Blank areas are reserved for future use.

Figure 8.3.1 Input/Output Port Related Register Map (1/2)

Address	D0 +0 Address D7	D8 +1 Address D15
H'0080 0740	P0 Operation Mode Register (P0MOD)	P1 Operation Mode Register (P1MOD)
H'0080 0742	P2 Operation Mode Register (P2MOD)	P3 Operation Mode Register (P3MOD)
H'0080 0744	P4 Operation Mode Register (P4MOD)	Port Input Function Enable Register (PIEN)
H'0080 0746		P7 Operation Mode Register (P7MOD)
H'0080 0748	P8 Operation Mode Register (P8MOD)	P9 Operation Mode Register (P9MOD)
H'0080 074A	P10 Operation Mode Register (P10MOD)	P11 Operation Mode Register (P11MOD)
H'0080 074C	P12 Operation Mode Register (P12MOD)	P13 Operation Mode Register (P13MOD)
H'0080 074E		P15 Operation Mode Register (P15MOD)
H'0080 0750		P17 Operation Mode Register (P17MOD)
H'0080 0752		
H'0080 0754		
H'0080 0756	P22 Operation Mode Register (P22MOD)	
	≈	≈
H'0080 0764	P4 Peripheral Output Select Register (P4SMOD)	
H'0080 0766		P7 Peripheral Output Select Register (P7SMOD)
H'0080 0768		
H'0080 076A	P10-P11 Peripheral Output Select Register (P1011SMOD)	
H'0080 076C		
H'0080 076E		P15 Peripheral Output Select Register (P15SMOD)
H'0080 0770		
H'0080 0772		
H'0080 0774		
H'0080 0776	P22 Peripheral Output Select Register (P22SMOD)	
H'0080 0778		
	≈	≈

Blank areas are reserved for future use.
Note: The registers enclosed in the thick frames must always be accessed in halfwords.

Figure 8.3.2 Input/Output Port Related Register Map (2/2)

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>**

D0	1	2	3	4	5	6	D7
(D8	9	10	11	12	13	14	D15)
Pn0DT	Pn1DT	Pn2DT	Pn3DT	Pn4DT	Pn5DT	Pn6DT	Pn7DT

Note: n = 0-4, 6-13, 15, 17, and 22

<When reset: indeterminate>

D	Bit Name	Function	R	W
0	Pn0DT (Port Pn0 data)	• When the direction bit is set to 0 (input	○	○
1	Pn1DT (Port Pn1 data)	mode) by Port Direction Register	○	○
2	Pn2DT (Port Pn2 data)	0: Port input pin = low	○	○
3	Pn3DT (Port Pn3 data)	1: Port input pin = high	○	○
4	Pn4DT (Port Pn4 data)	• When the direction bit is set to 1 (output	○	○
5	Pn5DT (Port Pn5 data)	mode) by Port Direction Register	○	○
6	Pn6DT (Port Pn6 data)	0: Port output latch = low	○	○
7	Pn7DT (Port Pn7 data)	1: Port output latch = high	○	○

Note 1: The bits listed below have no functions assigned. (They show a 0 when read; writing to these bits has no effect.)
P40, P60, P65-P67, P90-P92, P120-P123, P151, P152, P154-P157, P170, P171, P176, P177, P222-P224, P226, P227

Note 2: Port P64 is available for only input mode. Writing to the P64DT bit has no effect.

Note 3: Ports P80 and P81 are available for only input mode. Writing to the P80DT and P81DT bits has no effect. When read, P80 and P81 show the MOD0 and MOD1 pin levels, respectively.

Note 4: Ports P93, P97, P124-P127, P130-P137, P172, P173 and P221 are available for only input mode. Writing to the P93DT, P97DT, P124-127DT, P130-P137DT, P172DT, P173DT and 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>**
■ **P15 Direction Register (P15DIR)** **<Address : H'0080 072F>**
■ **P17 Direction Register (P17DIR)** **<Address : H'0080 0731>**
■ **P22 Direction Register (P22DIR)** **<Address : H'0080 0736>**

D0	1	2	3	4	5	6	D7
(D8	9	10	11	12	13	14	D15)
Pn0DIR	Pn1DIR	Pn2DIR	Pn3DIR	Pn4DIR	Pn5DIR	Pn6DIR	Pn7DIR

Note: n = 0-4, 6-11, 15, 17, and 22

<When reset: indeterminate>

D	Bit Name	Function	R	W
0	Pn0DIR (Port Pn0 direction bit)	0: Input mode (when reset)	○	○
1	Pn1DIR (Port Pn1 direction bit)	1: Output mode	○	○
2	Pn2DIR (Port Pn2 direction bit)		○	○
3	Pn3DIR (Port Pn3 direction bit)		○	○
4	Pn4DIR (Port Pn4 direction bit)		○	○
5	Pn5DIR (Port Pn5 direction bit)		○	○
6	Pn6DIR (Port Pn6 direction bit)		○	○
7	Pn7DIR (Port Pn7 direction bit)		○	○

Note 1: The bits listed below have no functions assigned. (They show a 0 when read; writing to these bits has no effect.)
P40, P60, P64-P67, P90-P92, P97, P120 -P123, P151, P152, P154-P157, P170, P171, P176, P177, P221-P224, P226, P227

Note 2: When reset, all ports are set for input mode.

Note 3: Port P64 is available for only input mode. The P64DIR is nonexistent.

Note 4: Ports P93, P97, P124-127, P130-P137, P172, P173 and P221 are available for only input mode. The P93DIR, P97DIR, P124-P127DIR, P130-P137DIR, P172DIR, P173DIR and P221DIR bits are nonexistent.

Note 5: Ports P80 and P81 are available for only input mode. The P80DIR and P81DIR bits are nonexistent.

8.3.3 Port Operation Mode Registers

■ P0 Operation Mode Register (P0MOD) **<Address: H'0080 0740>**

D0	1	2	3	4	5	6	D7
P00MOD	P01MOD	P02MOD	P03MOD	P04MOD	P05MOD	P06MOD	P07MOD

<When reset: H'00>

D	Bit Name	Function	R	W
0	P00MOD	0: DB0	○	○
	(Port P00 operation mode)	1: P00		
1	P01MOD	0: DB1	○	○
	(Port P01 operation mode)	1: P01		
2	P02MOD	0: DB2	○	○
	(Port P02 operation mode)	1: P02		
3	P03MOD	0: DB3	○	○
	(Port P03 operation mode)	1: P03		
4	P04MOD	0: DB4	○	○
	(Port P04 operation mode)	1: P04		
5	P05MOD	0: DB5	○	○
	(Port P05 operation mode)	1: P05		
6	P06MOD	0: DB6	○	○
	(Port P06 operation mode)	1: P06		
7	P07MOD	0: DB7	○	○
	(Port P07 operation mode)	1: P07		

Note: The value set in the P0 Operation Mode Register takes effect only when the CPU operates in external extended mode

■ **P1 Operation Mode Register (P1MOD)** **<Address: H'0080 0741>**

D8	9	10	11	12	13	14	D15
P10MOD	P11MOD	P12MOD	P13MOD	P14MOD	P15MOD	P16MOD	P17MOD

<When reset: H'00>

D	Bit Name	Function	R	W
8	P10MOD (Port P10 operation mode)	0: DB8 1: P10	○	○
9	P11MOD (Port P11 operation mode)	0: DB9 1: P11	○	○
10	P12MOD (Port P12 operation mode)	0: DB10 1: P12	○	○
11	P13MOD (Port P13 operation mode)	0: DB11 1: P13	○	○
12	P14MOD (Port P14 operation mode)	0: DB12 1: P14	○	○
13	P15MOD (Port P15 operation mode)	0: DB13 1: P15	○	○
14	P16MOD (Port P16 operation mode)	0: DB14 1: P16	○	○
15	P17MOD (Port P17 operation mode)	0: DB15 1: P17	○	○

Note: The value set in the P1 Operation Mode Register takes effect only when the CPU operates in external extended mode.

■ **P2 Operation Mode Register (P2MOD)** **<Address: H'0080 0742>**

D0	1	2	3	4	5	6	D7
P20MOD	P21MOD	P22MOD	P23MOD	P24MOD	P25MOD	P26MOD	P27MOD

<When reset: H'00>

D	Bit Name	Function	R	W
0	P20MOD (Port P20 operation mode)	0: A23 1: P20	○	○
1	P21MOD (Port P21 operation mode)	0: A24 1: P21	○	○
2	P22MOD (Port P22 operation mode)	0: A25 1: P22	○	○
3	P23MOD (Port P23 operation mode)	0: A26 1: P23	○	○
4	P24MOD (Port P24 operation mode)	0: A27 1: P24	○	○
5	P25MOD (Port P25 operation mode)	0: A28 1: P25	○	○
6	P26MOD (Port P26 operation mode)	0: A29 1: P26	○	○
7	P27MOD (Port P27 operation mode)	0: A30 1: P27	○	○

Note: The value set in the P2 Operation Mode Register takes effect only when the CPU operates in external extended mode.

■ **P3 Operation Mode Register (P3MOD)** **<Address: H'0080 0743>**

D8	9	10	11	12	13	14	D15
P30MOD	P31MOD	P32MOD	P33MOD	P34MOD	P35MOD	P36MOD	P37MOD

<When reset: H'00>

D	Bit Name	Function	R	W
8	P30MOD (Port P30 operation mode)	0: A15 1: P30	○	○
9	P31MOD (Port P31 operation mode)	0: A16 1: P31	○	○
10	P32MOD (Port P32 operation mode)	0: A17 1: P32	○	○
11	P33MOD (Port P33 operation mode)	0: A18 1: P33	○	○
12	P34MOD (Port P34 operation mode)	0: A19 1: P34	○	○
13	P35MOD (Port P35 operation mode)	0: A20 1: P35	○	○
14	P36MOD (Port P36 operation mode)	0: A21 1: P36	○	○
15	P37MOD (Port P37 operation mode)	0: A22 1: P37	○	○

Note: The value set in the P3 Operation Mode Register takes effect only when the CPU operates in external extended mode

■ **P4 Operation Mode Register (P4MOD)** **<Address: H'0080 0744>**

D0	1	2	3	4	5	6	D7
	P41MOD	P42MOD	P43MOD	P44MOD	P45MOD		P47MOD

<When reset: H'00>

D	Bit Name	Function	R	W
0	No functions assigned		0	–
1	P41MOD (Port P41 operation mode)	0: $\overline{BLW}/\overline{BLE}$ 1: P41	○	○
2	P42MOD (Port P42 operation mode)	0: $\overline{BHW}/\overline{BHE}$ 1: P42	○	○
3	P43MOD (Port P43 operation mode)	0: $\overline{RD}$ 1: P43	○	○
4	P44MOD (Port P44 operation mode)	0: $\overline{CS0}$ 1: P44	○	○
5	P45MOD (Port P45 operation mode)	0: $\overline{CS1}$ 1: P45	○	○
6	No functions assigned		0	–
7	P47MOD (Port P47 operation mode)	0: A14 1: P47	○	○

Note: The value set in the P4 Operation Mode Register takes effect only when the CPU operates in external extended mode.

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

D8	9	10	11	12	13	14	D15
P70MOD	P71MOD	P72MOD	P73MOD	P74MOD	P75MOD	P76MOD	P77MOD

<When reset: H'00>

D	Bit Name	Function	R	W
8	P70MOD (Port P70 operation mode)	0: P70 1: BCLK/$\overline{WR}$	○	○
9	P71MOD (Port P71 operation mode)	0: P71 1: $\overline{WAIT}$	○	○
10	P72MOD (Port P72 operation mode)	0: P72 1: $\overline{HREQ}$	○	○
11	P73MOD (Port P73 operation mode)	0: P73 1: $\overline{HACK}$/TXD3	○	○
12	P74MOD (Port P74 operation mode)	0: P74 1: RTDTXD	○	○
13	P75MOD (Port P75 operation mode)	0: P75 1: RTDRXD	○	○
14	P76MOD (Port P76 operation mode)	0: P76 1: RTDACK	○	○
15	P77MOD (Port P77 operation mode)	0: P77 1: RTDCLK	○	○

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

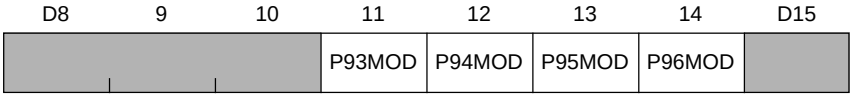
D0	1	2	3	4	5	6	D7
		P82MOD	P83MOD	P84MOD	P85MOD	P86MOD	P87MOD

<When reset: H'00>

D	Bit Name	Function	R	W
0,1	No functions assigned		0	–
2	P82MOD (Port P82 operation mode)	0: P82 1: TXD0	○	○
3	P83MOD (Port P83 operation mode)	0: P83 1: RXD0	○	○
4	P84MOD (Port P84 operation mode)	0: P84 1: SCLKI0/SCLKO0	○	○
5	P85MOD (Port P85 operation mode)	0: P85 1: TXD1	○	○
6	P86MOD (Port P86 operation mode)	0: P86 1: RXD1	○	○
7	P87MOD (Port P87 operation mode)	0: P87 1: SCLKI1/SCLKO1	○	○

Note: Ports P80 and P81 are nonexistent.

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



<When reset: H'00>

D	Bit Name	Function	R	W
8-10	No functions assigned		0	–
11	P93MOD (Port P93 operation mode)	0: P93 1: RXD3	○	○
12	P94MOD (Port P94 operation mode)	0: P94 1: TDX6	○	○
13	P95MOD (Port P95 operation mode)	0: P95 1: RXD6	○	○
14	P96MOD (Port P96 operation mode)	0: P96 1: CTX1	○	○
15	No functions assigned		0	–

Note 1: Ports P90-P92 are nonexistent.
Note 2: P97 is a CAN1 input-only pin.

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

D0	1	2	3	4	5	6	D7
P100MOD	P101MOD	P102MOD	P103MOD	P104MOD	P105MOD	P106MOD	P107MOD

<When reset: H'00>

D	Bit Name	Function	R	W
0	P100MOD (Port P100 operation mode)	0: P100 1: TO8	○	○
1	P101MOD (Port P101 operation mode)	0: P101 1: TO9	○	○
2	P102MOD (Port P102 operation mode)	0: P102 1: TO10	○	○
3	P103MOD (Port P103 operation mode)	0: P103 1: TO11	○	○
4	P104MOD (Port P104 operation mode)	0: P104 1: TO12/SCLKI4	○	○
5	P105MOD (Port P105 operation mode)	0: P105 1: TO13/SCLKO4	○	○
6	P106MOD (Port P106 operation mode)	0: P106 1: TO14/TXD4	○	○
7	P107MOD (Port P107 operation mode)	0: P107 1: TO15/RXD4	○	○

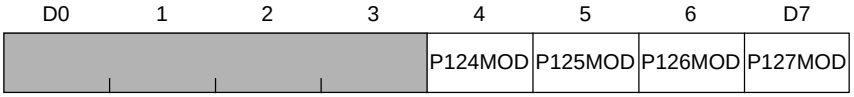
■ **P11 Operation Mode Register (P11MOD)** **<Address: H'0080 074B>**

D8	9	10	11	12	13	14	D15
P110MOD	P111MOD	P112MOD	P113MOD	P114MOD	P115MOD	P116MOD	P117MOD

<When reset: H'00>

D	Bit Name	Function	R	W
8	P110MOD (Port P110 operation mode)	0: P110 1: TO0	○	○
9	P111MOD (Port P111 operation mode)	0: P111 1: TO1	○	○
10	P112MOD (Port P112 operation mode)	0: P112 1: TO2	○	○
11	P113MOD (Port P113 operation mode)	0: P113 1: TO3	○	○
12	P114MOD (Port P114 operation mode)	0: P114 1: TO4/SCLKI5	○	○
13	P115MOD (Port P115 operation mode)	0: P115 1: TO5/SCLKO5	○	○
14	P116MOD (Port P116 operation mode)	0: P116 1: TO6/TXD5	○	○
15	P117MOD (Port P117 operation mode)	0: P117 1: TO7/RXD5	○	○

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



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4	P124MOD (Port P124 operation mode)	0: P124 1: TIN0A	○	○
5	P125MOD (Port P125 operation mode)	0: P125 1: TIN0B	○	○
6	P126MOD (Port P126 operation mode)	0: P126 1: TIN1A	○	○
7	P127MOD (Port P127 operation mode)	0: P127 1: TIN1B	○	○

Note : Ports P120-P123 are nonexistent.

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

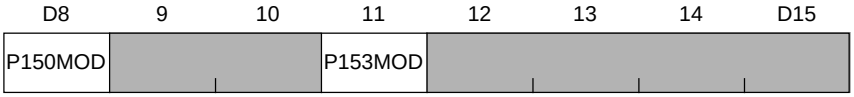
D8	9	10	11	12	13	14	D15
P130MOD	P131MOD	P132MOD	P133MOD	P134MOD	P135MOD	P136MOD	P137MOD

<When reset: H'00>

D	Bit Name	Function	R	W
8	P130MOD (Port P130 operation mode)	0: P130 1: TIN16/PWMOFF0	○	○
9	P131MOD (Port P131 operation mode)	0: P131 1: TIN17/PWMOFF1	○	○
10	P132MOD (Port P132 operation mode)	0: P132 1: TIN18	○	○
11	P133MOD (Port P133 operation mode)	0: P133 1: TIN19	○	○
12	P134MOD (Port P134 operation mode)	0: P134 1: TIN20	○	○
13	P135MOD (Port P135 operation mode)	0: P135 1: TIN21	○	○
14	P136MOD (Port P136 operation mode)	0: P136 1: TIN22	○	○
15	P137MOD (Port P137 operation mode)	0: P137 1: TIN23	○	○

Note: Ports P130-P137 are input-only pins.

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

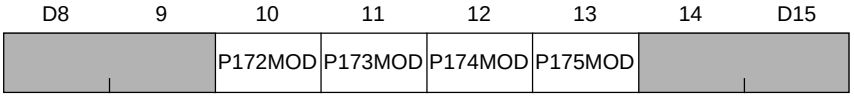


<When reset: H'00>

D	Bit Name	Function	R	W
8	P150MOD (Port P150 operation mode)	0: P150 1: TIN8/TXD7	○	○
9,10	No functions assigned		0	–
11	P153MOD (Port P153 operation mode)	0: P153 1: TIN9/TXD7	○	○
12-15	No functions assigned		0	–

Note: Ports P151, P152, P154-P157 are nonexistent.

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

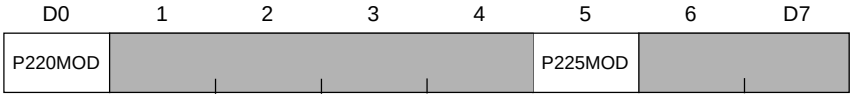


<When reset: H'00>

D	Bit Name	Function	R	W
8,9	No functions assigned		0	–
10	P172MOD (Port P172 operation mode)	0: P172 1: TIN10	○	○
11	P173MOD (Port P173 operation mode)	0: P173 1: TIN11	○	○
12	P174MOD (Port P174 operation mode)	0: P174 1: TXD2	○	○
13	P175MOD (Port P175 operation mode)	0: P175 1: RXD2	○	○
14,15	No functions assigned		0	–

Note: Ports P170-P171, P176, P177 are nonexistent.

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



<When reset: H'00>

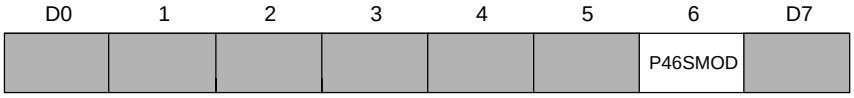
D	Bit Name	Function	R	W
0	P220MOD (Port P220 operation mode)	0: P220 1: CTX0	○	○
1-4	No functions assigned		0	–
5	P225MOD (Port P225 operation mode)	0: P225 1: use inhibited	○	○
6-7	No functions assigned		0	–

Note 1: P221 is a CAN0 input-only pin.

Note 2: P225 has its pin function altered depending on how the MOD0 and MOD1 pins are set. Use of this port requires caution because it has a debug event function.

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

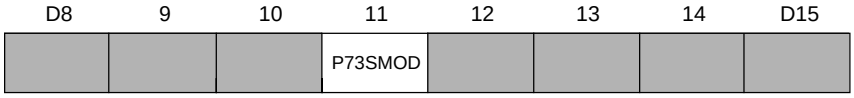
■ P4 Peripheral Output Select Register (P4SMOD) <Address: H'0080 0764>



<When reset: H'00>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6	P46SMOD (Selects port P46 peripheral output)	0:A13 1:$\overline{CS3}$	○	○
7	No functions assigned		0	–

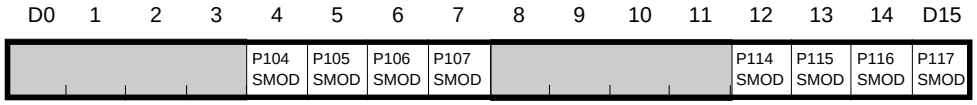
■ **P7 Peripheral Output Select Register (P7SMOD)** **<Address: H'0080 0767>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-10	No functions assigned		0	–
11	P73SMOD (Selects port P73 peripheral output)	0: $\overline{\text{HACK}}$ 1: TXD3	○	○
12-15	No functions assigned		0	–

■ **P10-P11 Peripheral Output Select Register (P1011SMOD)** **<Address: H'0080 076A>**

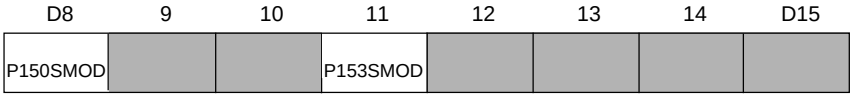


<When reset: H'0000>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4	P104SMOD (Selects port P104 peripheral output)	0: TO12 1: SCLKI4	○	○
5	P105SMOD (Selects port P105 peripheral output)	0: TO13 1: SCLKO4	○	○
6	P106SMOD (Selects port P106 peripheral output)	0: TO14 1: TXD4	○	○
7	P107SMOD (Selects port P107 peripheral output)	0: TO15 1: RXD4	○	○
8-11	No functions assigned		0	–
12	P114SMOD (Selects port P114 peripheral output)	0: TO4 1: SCLKI5	○	○
13	P115SMOD (Selects port P115 peripheral output)	0: TO5 1: SCLKO5	○	○
14	P116SMOD (Selects port P116 peripheral output)	0: TO6 1: TXD5	○	○
15	P117SMOD (Selects port P117 peripheral output)	0: TO7 1: RXD5	○	○

Note: This register must always be accessed in halfwords.

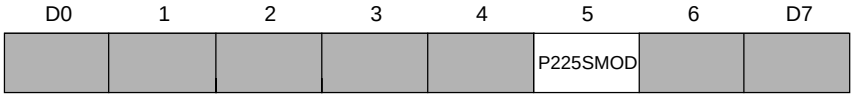
■ **P15 Peripheral Output Select Register (P15SMOD)** **<Address: H'0080 076F>**



<When reset: H'00>

D	Bit Name	Function	R	W
8	P150SMOD (Port P150 operation mode)	0: TIN8 1: TXD7	○	○
9,10	No functions assigned			
11	P153SMOD (Port P153 operation mode)	0: TIN9 1: RXD7	○	○
12-15	No functions assigned		0	–

■ **P22 Peripheral Output Select Register (P22SMOD)** **<Address: H'0080 0776>**

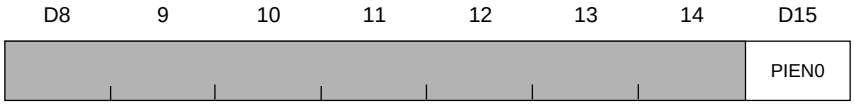


<When reset: H'00>

D	Bit Name	Function	R	W
0-4	No functions assigned		0	–
5	P225SMOD (Selects port P225 peripheral output)	0: A12 1: $\overline{CS2}$	○	○
6-7	No functions assigned		0	–

Note: The value set in the P22 Peripheral Output Select Register takes effect only when the CPU operates in external extended mode.

■ **Port Input Function Enable Register (PIEN)** **<Address: H'0080 0745>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-14	No functions assigned		0	–
15	PIEN0 (Port input function enable bit)	0: Disables input (to prevent current from flowing in) 1: Enables input	○	○

This register is used to prevent electric current from flowing into the port input pin. Because the ports are disabled against input after reset, they need to be enabled for input by setting this register bit to 1.

During boot mode, the pins shared with serial I/O function are enabled for input. Therefore, when rewriting the flash memory via serial communication, they can be protected against the current flowing in from pins other than the serial I/O function by setting this register bit to 0.

The pins that can be controlled by the port input function enable bit in each mode are listed below.

Table 8.3.1 Pins Controllable by Port Input Function Enable Bit

Mode Name	Controllable Pin	Noncontrollable Pin
Single chip	P00-P07, P10-P17, P20-P27 P30-P37, P41-P47, P61-P63 P70-P77, P82-P87, P94-P96 P100-P107, P110-P117, P150, P153, P174, P175 P220, P225	P64, P93, P97 P124-P127 P130-P137, P221 FP, MOD0, MOD1, RESET
External extended Microprocessor	P61-P63, P70-P77, P82-P87 P94-P96, P100-P107, P110-P117 P150, P153, P174, P175, P220	P00-P07, P10-P17 P20-P27, P30-P37 P41-P47, P64, P93,P97 P124-P127, P130-P137 P221, P225, FP,MOD0,MOD1,RESET
Boot (Single chip)	P00-P07, P10-P17, P20-P27 P30-P37, P41-P47, P61-P63 P67, P70-P77,P94-P96 P100-P107, P110-P117, P140-P147, P150-P157 P160-P167, P172-P173, P180-P187 P190-P197, P210-P217, P220 P222-P225	P64, P82-P87, P93,P97, P124-P127, P130-P137 P174, P175, P221, FP MOD0, MOD1, RESET

8.4 Port Peripheral Circuits

Figures 8.4.1 through 8.4.4 show the peripheral circuit diagrams of the ports on the microcomputer.

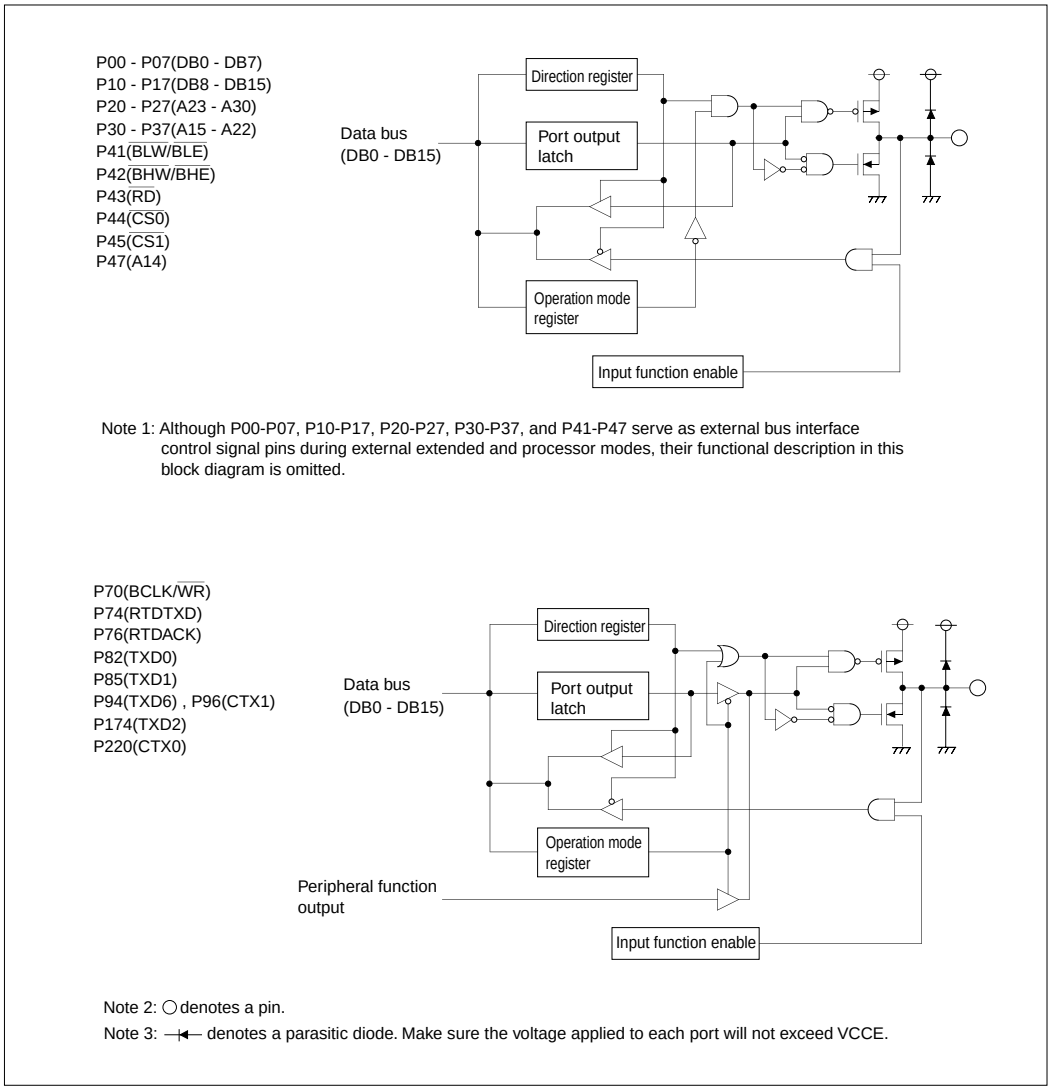


Figure 8.4.1 Port Peripheral Circuit Diagram (1)

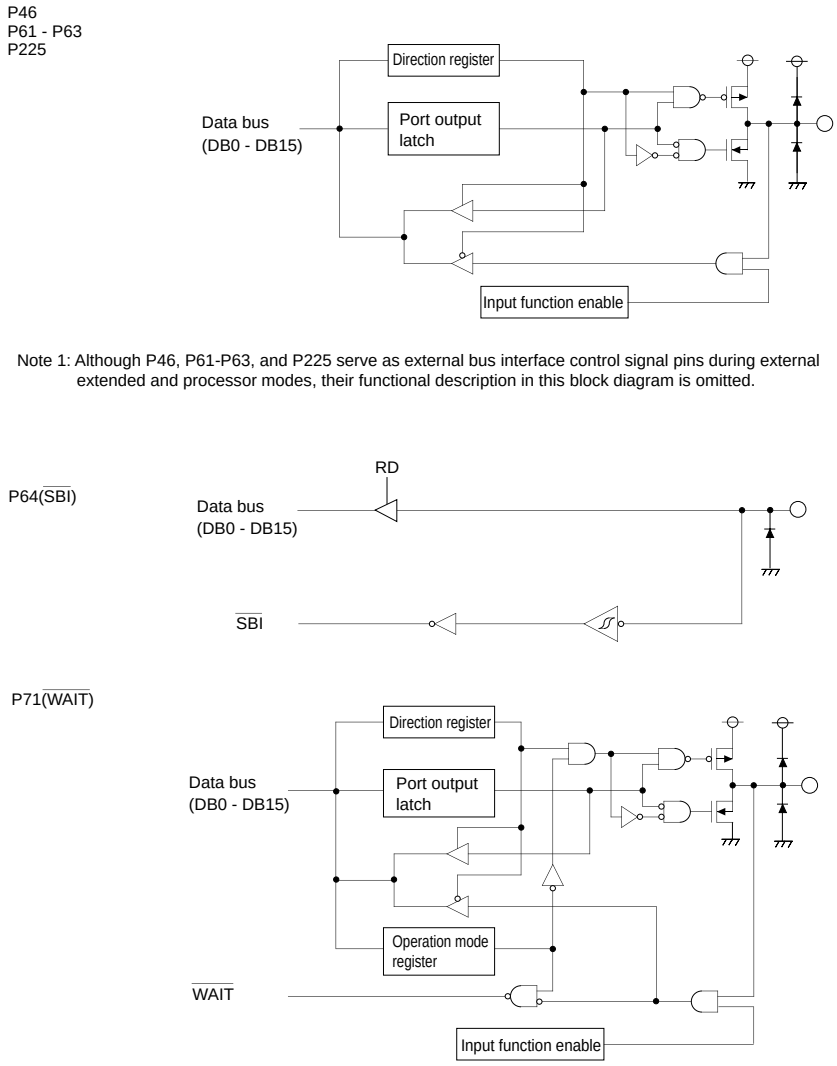
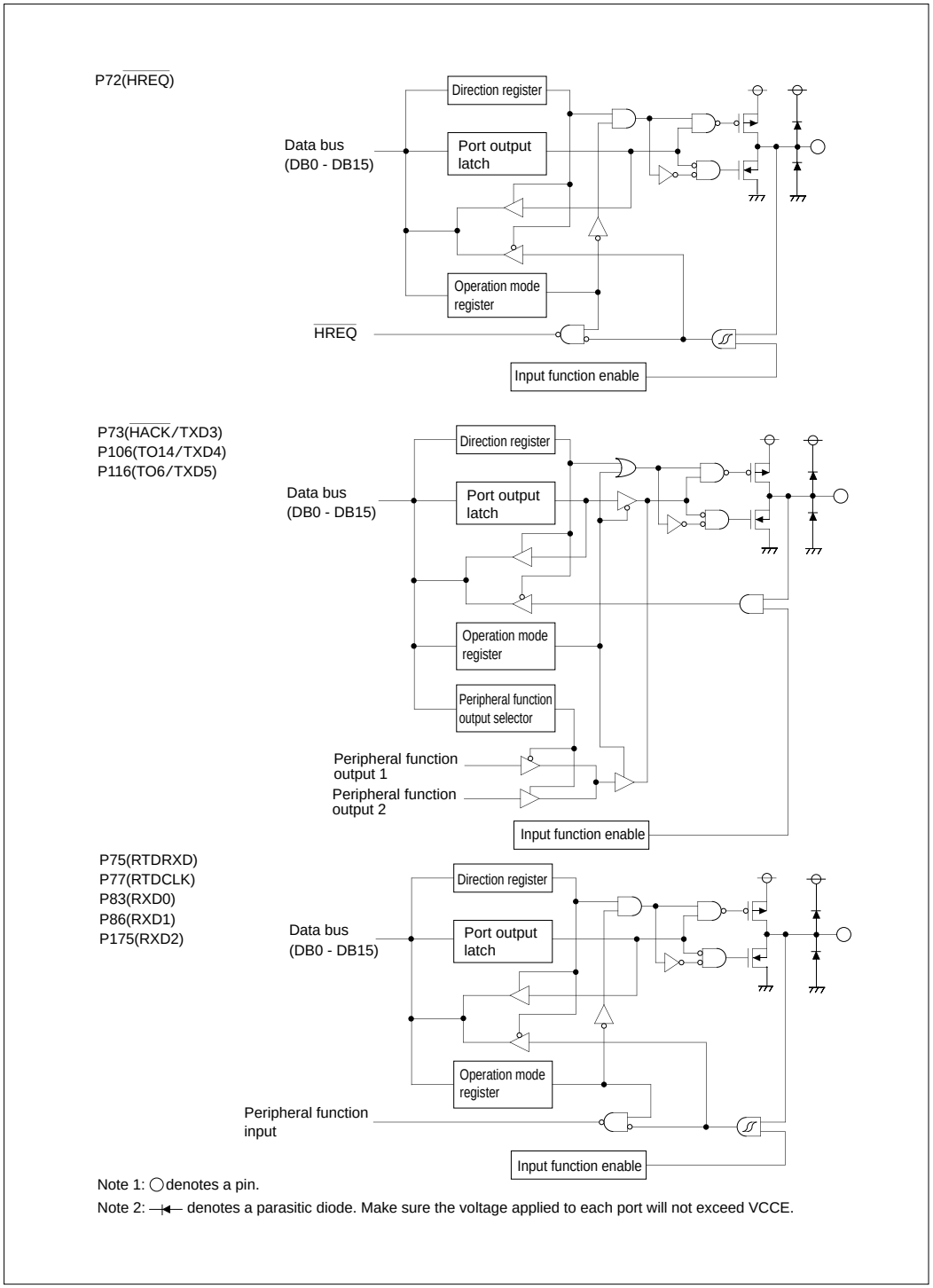


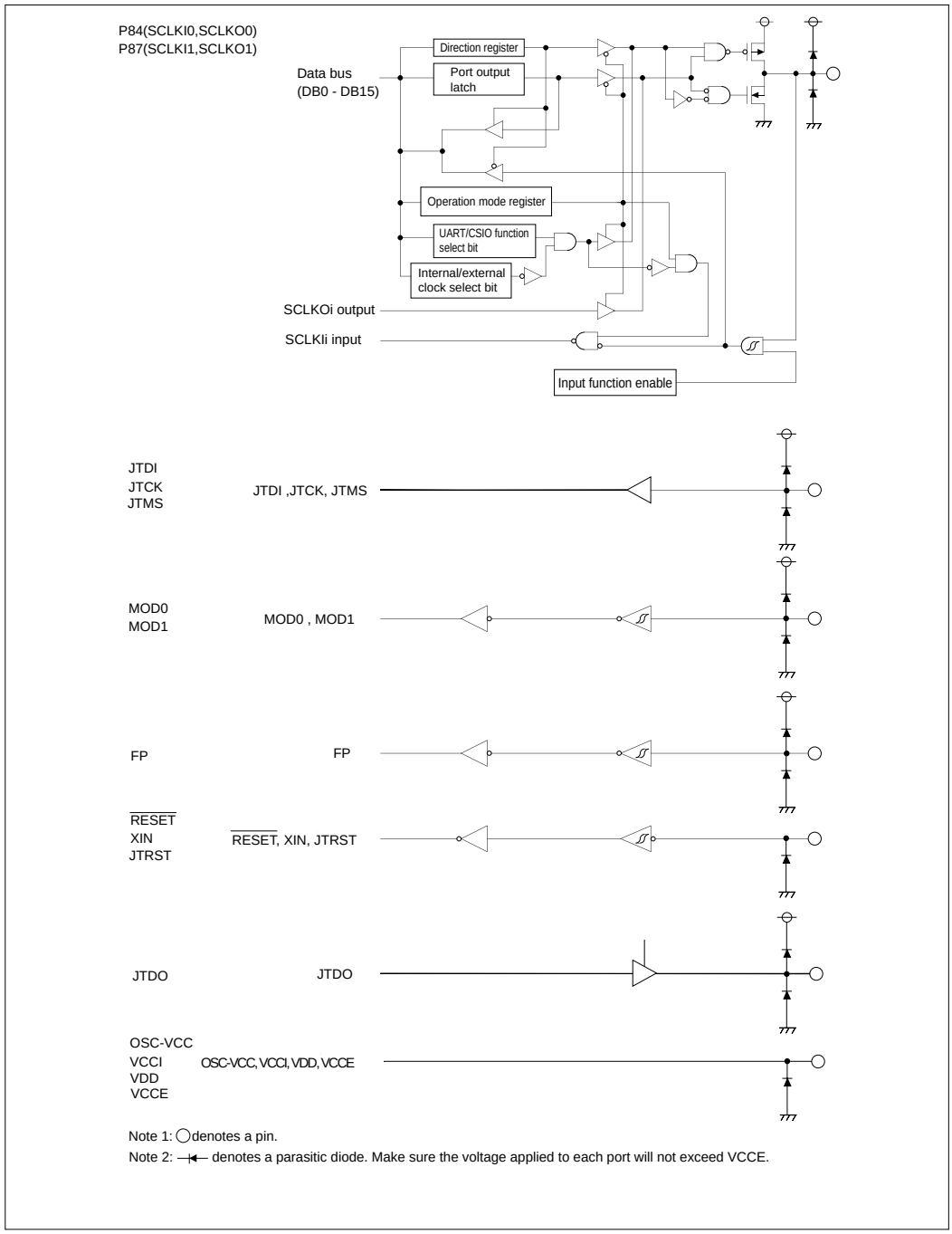
Figure 8.4.2 Port Peripheral Circuit Diagram (2)



Note 1: ○ denotes a pin.

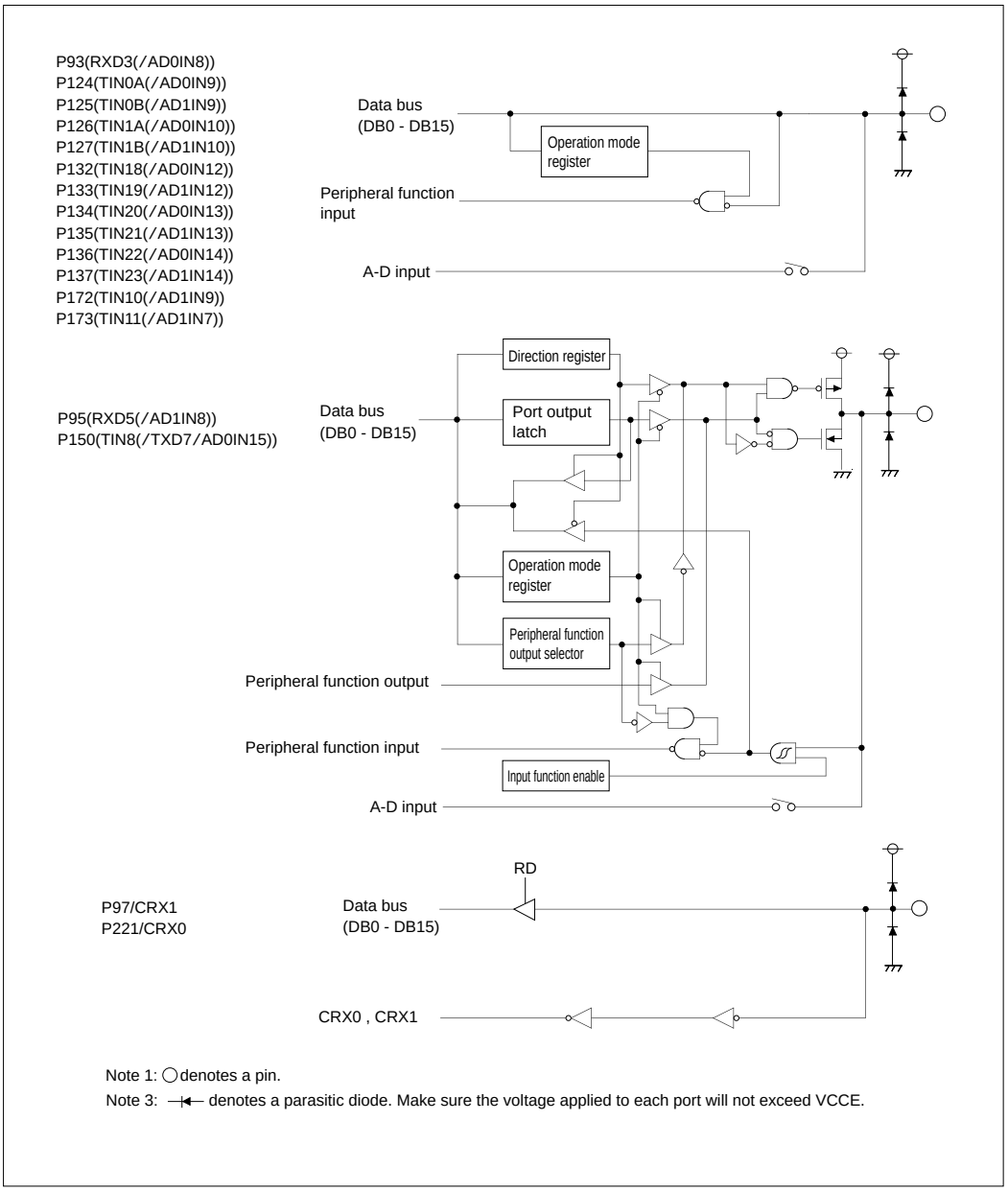
Note 2: —|◄— denotes a parasitic diode. Make sure the voltage applied to each port will not exceed VCCE.

Figure 8.4.3 Port Peripheral Circuit Diagram (3)



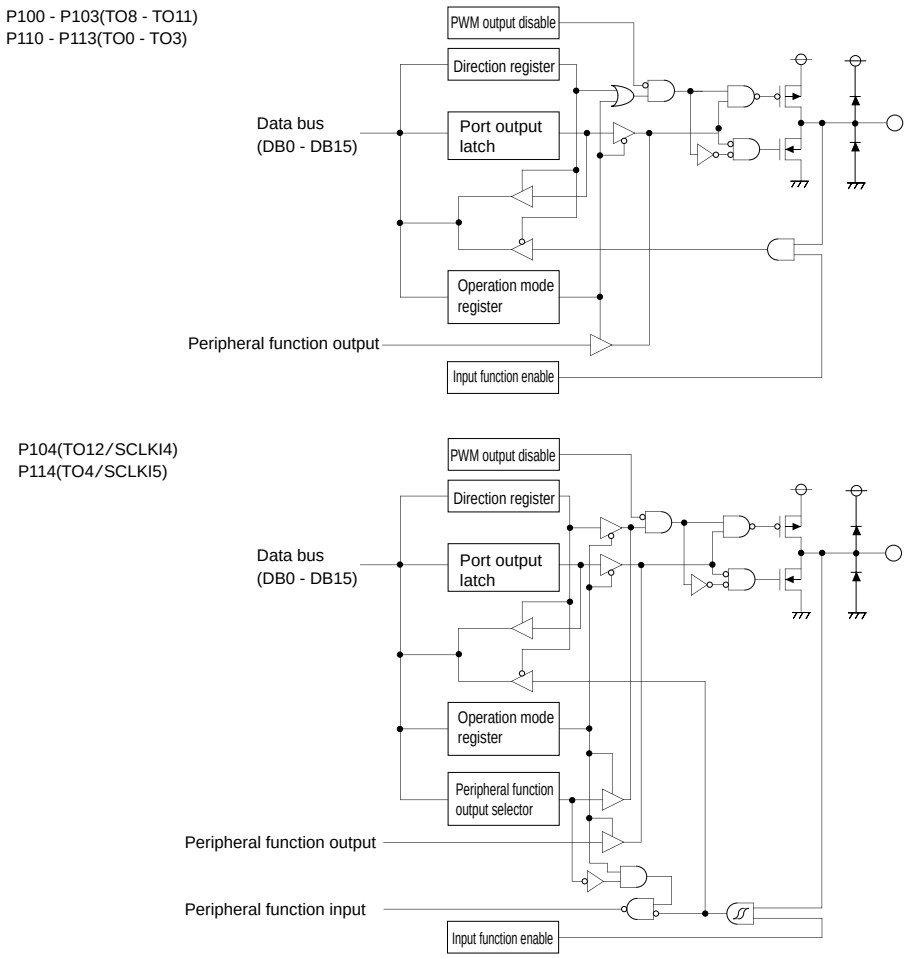
Note 1: ○denotes a pin.
Note 2: —|◄— denotes a parasitic diode. Make sure the voltage applied to each port will not exceed VCCE.

Figure 8.4.4 Port Peripheral Circuit Diagram (4)



Note 1: ○ denotes a pin.
Note 3: ⊣◂ denotes a parasitic diode. Make sure the voltage applied to each port will not exceed VCCE.

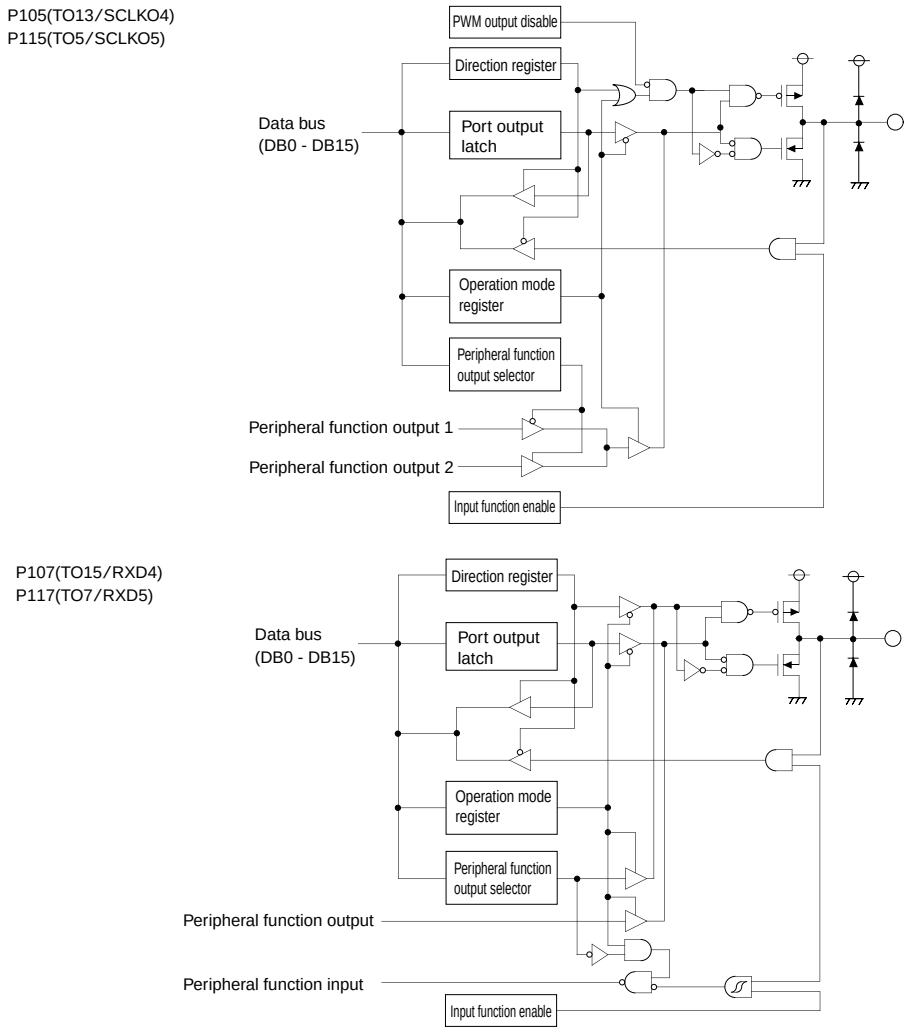
Figure 8.4.5 Port Peripheral Circuit Diagram (5)



Note 1: ◯ denotes a pin.

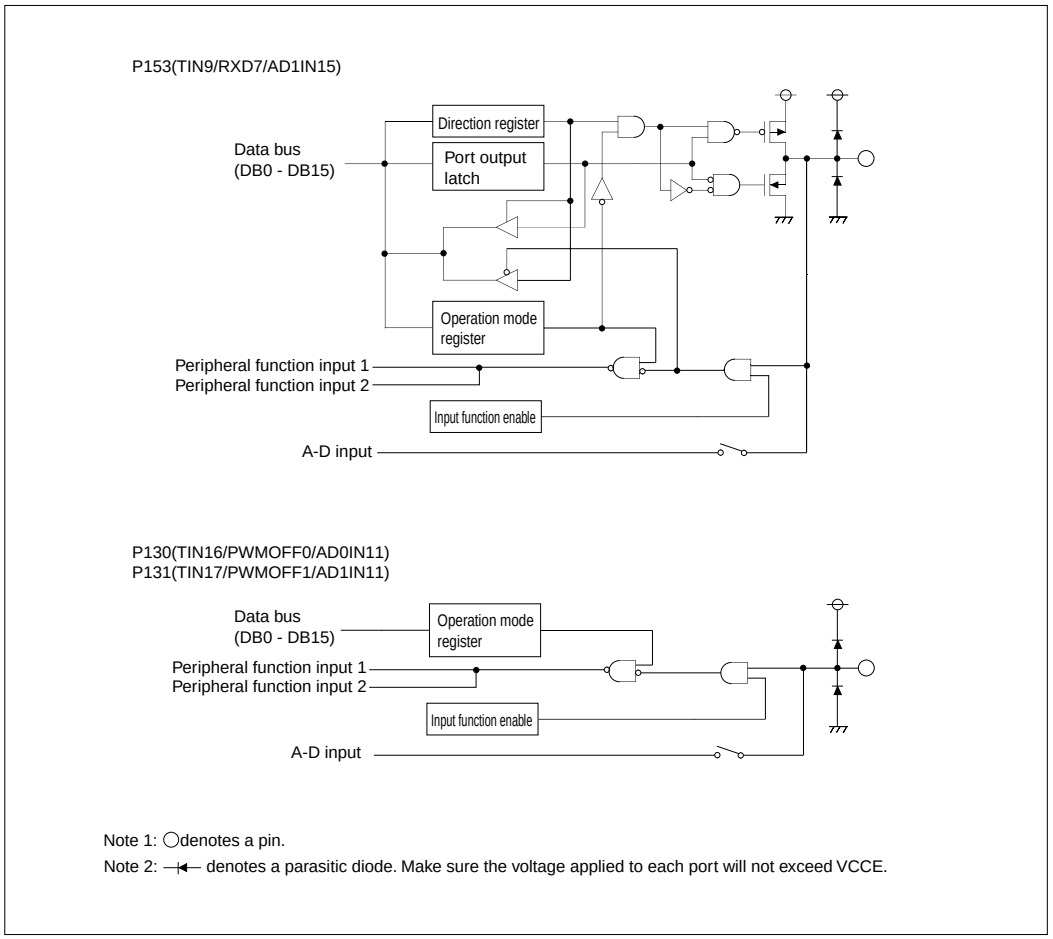
Note 2: —|◄— denotes a parasitic diode. Make sure the voltage applied to each port will not exceed VCCE.

Figure 8.4.6 Port Peripheral Circuit Diagram (6)



Note 1: ○denotes a pin.
Note 2: —|◄— denotes a parasitic diode. Make sure the voltage applied to each port will not exceed VCCE.

Figure 8.4.7 Port Peripheral Circuit Diagram (7)



Note 1: ○denotes a pin.

Note 2: ⊣◂ denotes a parasitic diode. Make sure the voltage applied to each port will not exceed VCCE.

Figure 8.4.8 Port Peripheral Circuit Diagram (8)

8.5 Precautions on Input/Output Ports

- **When using ports in output mode**

Immediately after reset, the Port Data Register values are indeterminate.Therefore, write the initial output value to the Port Data Register before setting the Port Direction Register for output. Note that if the Port Direction Register is set for output before writing to the Port Data Register, an indeterminate value may be output for a while until the write data is set in the Port Data Register.

*** This is a blank page ***

CHAPTER 9

DMAC

9.1 Outline of DMAC

The microcomputer has 10-channel DMA (Direct Memory Access) Controller allowing data to be transferred at high speed between internal peripheral I/Os, between internal RAM and internal peripheral I/O, and between internal RAMs when triggered in software or by request from internal peripheral I/O.

Table 9.1.1 Outline of DMAC

Item	Content
Number of channels	10 channels
Transfer request	• Software trigger • Request from internal peripheral I/O: A-D converter, input/output timer, serial I/O (reception complete, transmit buffer empty), or PD controller • Cascaded operation between DMA channels (Note)
Maximum transfer count	256 times
Transferable address space	• 64 Kbytes (address space in H'0080 0000 through H'0080 FFFF) • Supports transfers between internal peripheral I/Os, between internal RAM and internal peripheral I/O, and between internal RAMs
Transfer data size	16 or 8 bits
Transfer method	Single-transfer method DMA (control of internal bus released for each transfer performed), dual-address transfer
Transfer mode	Single transfer mode
Direction of transfer	Selectable among three modes for the source and destination • Address fixed • Address increment • Ring buffer
Channel priority	Channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7 > channel 8 > channel 9 > (fixed priority)
Maximum transfer rate	13.3 Mbytes per second (when internal peripheral clock = 20 MHz)
Interrupt request	Group interrupt request can be generated when any transfer count register underflows
Transfer area	64 Kbytes in H'0080 0000 through H'0080 FFFF (transferable in the entire internal RAM and SFR area)

Note: The DMA channels can be cascaded in the manner shown below.
Completion of one DMA transfer on channel 0 starts a DMA transfer on channel 1.
Completion of one DMA transfer on channel 1 starts a DMA transfer on channel 2.
Completion of one DMA transfer on channel 2 starts a DMA transfer on channel 0.
Completion of one DMA transfer on channel 3 starts a DMA transfer on channel 4.
Completion of one DMA transfer on channel 5 starts a DMA transfer on channel 6.
Completion of one DMA transfer on channel 6 starts a DMA transfer on channel 7.
Completion of one DMA transfer on channel 7 starts a DMA transfer on channel 5.
Completion of one DMA transfer on channel 8 starts a DMA transfer on channel 9.
Completion of one DMA transfer on channel 9 starts a DMA transfer on channel 1-9.
Completion of all DMA transfers on channel 0 (i.e., the transfer count register underflows) starts a DMA transfer on channel 5.
Completion of all DMA transfers on channel 1 (i.e., the transfer count register underflows) starts DMA transfers on channels 0-9.
Completion of all DMA transfers on channel 3 (i.e., the transfer count register underflows) starts DMA transfers on channel 8.

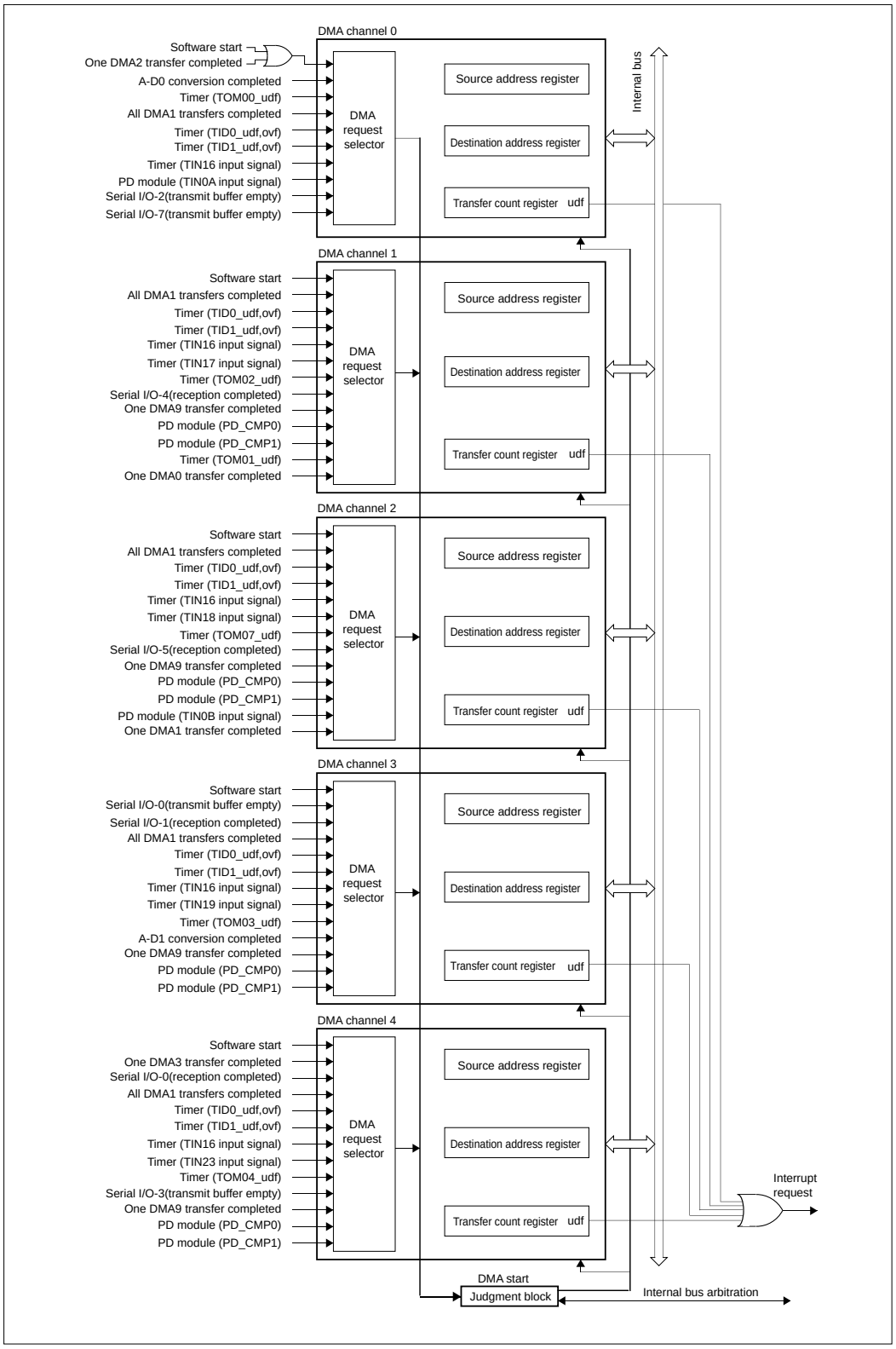


Figure 9.1.1 DMA Block Diagram (1/2)

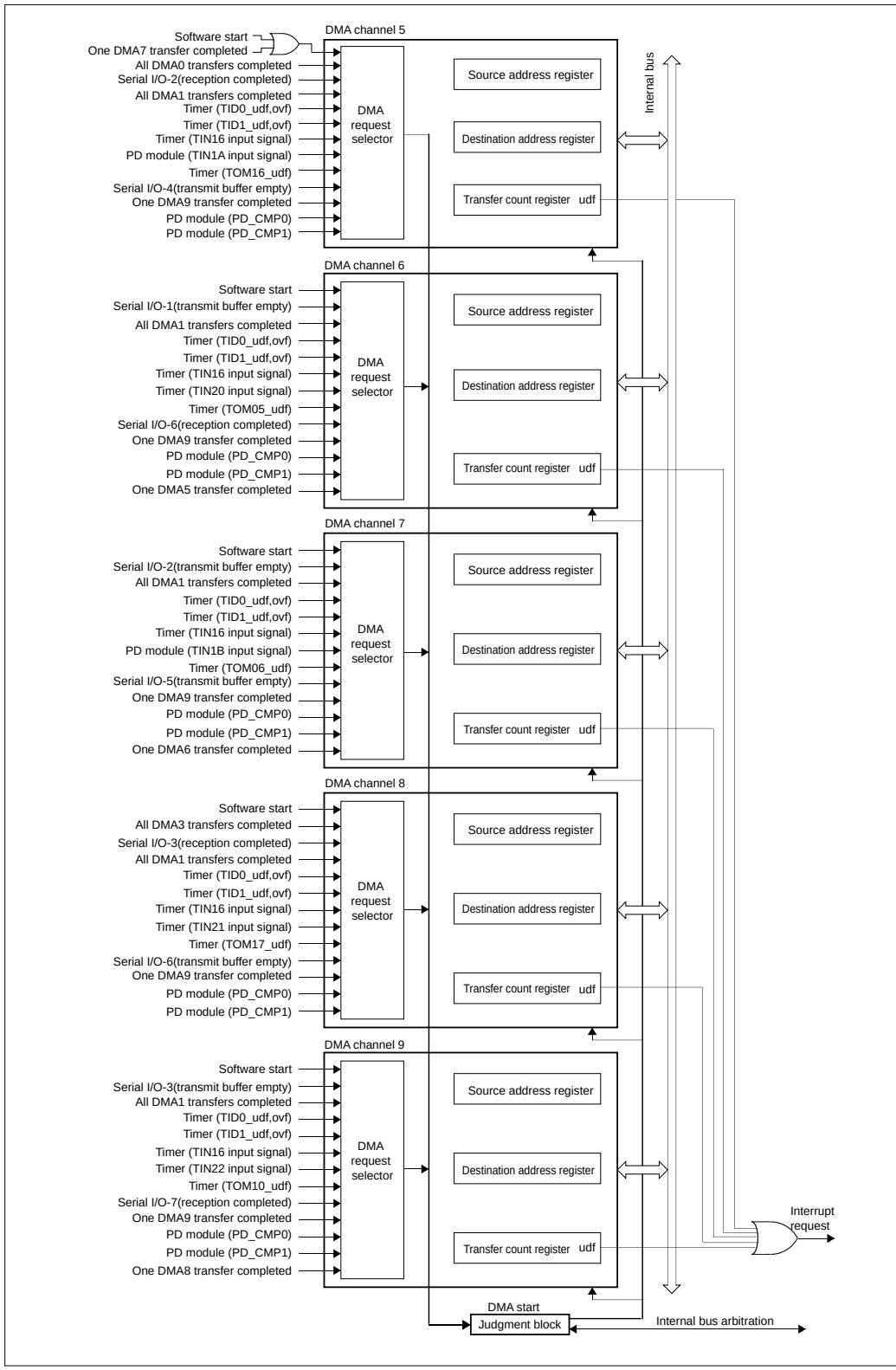


Figure 9.1.2 DMA Block Diagram (2/2)

9.2 DMAC Related Registers

A memory map of DMA related registers is shown below.

Address	+0 address D0 – D7	+1 address D8 – D15
H'0080 0400	DMA0-4 Interrupt Request Status Register (DM04ITST)	DMA0-4 Interrupt Mask Register (DM04ITMK)
H'0080 0408	DMA5-9 Interrupt Request Status Register (DM59ITST)	DMA5-9 Interrupt Mask Register (DM59ITMK)
H'0080 0410	DMA0 Channel Control Register (DM0CNT)	DMA0 Transfer Count Register (DM0TCT)
H'0080 0412	DMA0 Source Address Register (DM0SA)	
H'0080 0414	DMA0 Destination Address Register (DM0DA)	
H'0080 0416	DMA0 Request Extended Cause Register (DM0REQ)	
H'0080 0418	DMA5 Channel Control Register (DM5CNT)	DMA5 Transfer Count Register (DM5TCT)
H'0080 041A	DMA5 Source Address Register (DM5SA)	
H'0080 041C	DMA5 Destination Address Register (DM5DA)	
H'0080 041E	DMA5 Request Extended Cause Register (DM5REQ)	
H'0080 0420	DMA1 Channel Control Register (DM1CNT)	DMA1 Transfer Count Register (DM1TCT)
H'0080 0422	DMA1 Source Address Register (DM1SA)	
H'0080 0424	DMA1 Destination Address Register (DM1DA)	
H'0080 0426	DMA1 Request Extended Cause Register (DM1REQ)	
H'0080 0428	DMA6 Channel Control Register (DM6CNT)	DMA6 Transfer Count Register (DM6TCT)
H'0080 042A	DMA6 Source Address Register (DM6SA)	
H'0080 042C	DMA6 Destination Address Register (DM6DA)	
H'0080 042E	DMA6 Request Extended Cause Register (DM6REQ)	
H'0080 0430	DMA2 Channel Control Register (DM2CNT)	DMA2 Transfer Count Register (DM2TCT)
H'0080 0432	DMA2 Source Address Register (DM2SA)	
H'0080 0434	DMA2 Destination Address Register (DM2DA)	
H'0080 0436	DMA2 Request Extended Cause Register (DM2REQ)	
H'0080 0438	DMA7 Channel Control Register (DM7CNT)	DMA7 Transfer Count Register (DM7TCT)
H'0080 043A	DMA7 Source Address Register (DM7SA)	
H'0080 043C	DMA7 Destination Address Register (DM7DA)	
H'0080 043E	DMA7 Request Extended Cause Register (DM7REQ)	

Blank areas are reserved for future use.

Note: The registers enclosed in the thick frames can only be accessed in halfwords.

Figure 9.2.1 DMA Related Register Map (1/2)

Address	+0 address D0 – D7	+1 address D8 – D15
H'0080 0440	DMA3 Channel Control Register (DM3CNT)	DMA3 Transfer Count Register (DM3TCT)
H'0080 0442	DMA3 Source Address Register (DM3SA)	
H'0080 0444	DMA3 Destination Address Register (DM3DA)	
H'0080 0446	DMA3 Request Extended Cause Register (DM3REQ)	
H'0080 0448	DMA8 Channel Control Register (DM8CNT)	DMA8 Transfer Count Register (DM8TCT)
H'0080 044A	DMA8 Source Address Register (DM8SA)	
H'0080 044C	DMA8 Destination Address Register (DM8DA)	
H'0080 044E	DMA8 Request Extended Cause Register (DM8REQ)	
H'0080 0450	DMA4 Channel Control Register (DM4CNT)	DMA4 Transfer Count Register (DM4TCT)
H'0080 0452	DMA4 Source Address Register (DM4SA)	
H'0080 0454	DMA4 Destination Address Register (DM4DA)	
H'0080 0456	DMA4 Request Extended Cause Register (DM4REQ)	
H'0080 0458	DMA9 Channel Control Register (DM9CNT)	DMA9 Transfer Count Register (DM9TCT)
H'0080 045A	DMA9 Source Address Register (DM9SA)	
H'0080 045C	DMA9 Destination Address Register (DM9DA)	
H'0080 045E	DMA9 Request Extended Cause Register (DM9REQ)	
H'0080 0460	DMA0 Software Request Generation Register (DM0SRI)	
H'0080 0462	DMA1 Software Request Generation Register (DM1SRI)	
H'0080 0464	DMA2 Software Request Generation Register (DM2SRI)	
H'0080 0466	DMA3 Software Request Generation Register (DM3SRI)	
H'0080 0468	DMA4 Software Request Generation Register (DM4SRI)	
H'0080 0470	DMA5 Software Request Generation Register (DM5SRI)	
H'0080 0472	DMA6 Software Request Generation Register (DM6SRI)	
H'0080 0474	DMA7 Software Request Generation Register (DM7SRI)	
H'0080 0476	DMA8 Software Request Generation Register (DM8SRI)	
H'0080 0478	DMA9 Software Request Generation Register (DM9SRI)	

Blank areas are reserved for future use.

Note: The registers enclosed in the thick frames can only be accessed in halfwords.

Figure 9.2.2 DMA Related Register Map (2/2)

9.2.1 DMA Channel Control Registers

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

D0	1	2	3	4	5	6	D7
MDSEL0	TREQF0	REQSL0		TENL0	TSZSL0	SADSL0	DADSL0

<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL0 (Selects DMA0 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF0 (DMA0 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL0 (Selects cause of DMA0 request)	00: Software start or one DMA2 transfer completed 01: A-D0 conversion completed 10: Timer (TOM00_udf) 11: Extended request cause (Note)	○	○
4	TENL0 (DMA0 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL0 (Selects DMA0 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL0 (Selects DMA0 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL0 (Selects DMA0 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA0 Request Extended Cause Register (DM0REQ).

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

D0	1	2	3	4	5	6	D7
MDSEL1	TREQF1	REQSL1		TENL1	TSZSL1	SADSL1	DADSL1

<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL1 (Selects DMA1 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF1 (DMA1 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL1 (Selects cause of DMA1 request)	00: Software start 01: Extended request cause (Note) 10: Timer (TOM01_udf) 11: One DMA0 transfer completed	○	○
4	TENL1 (DMA1 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL1 (Selects DMA1 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL1 (Selects DMA1 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL1 (Selects DMA1 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA1 Request Extended Cause Register (DM1REQ).

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

D0	1	2	3	4	5	6	D7
MDSEL2	TREQF2	REQSL2		TENL2	TSZSL2	SADSL2	DADSL2

<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL2 (Selects DMA2 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF2 (DMA2 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL2 (Selects cause of DMA2 request)	00: Software start 01: Extended request cause (Note) 10: PD module (TIN0B input signal) 11: One DMA1 transfer completed	○	○
4	TENL2 (DMA2 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL2 (Selects DMA2 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL2 (Selects DMA2 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL2 (Selects DMA2 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA2 Request Extended Cause Register (DM2REQ).

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

D0	1	2	3	4	5	6	D7
MDSEL3	TREQF3	REQSL3		TENL3	TSZSL3	SADSL3	DADSL3

<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL3 (Selects DMA3 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF3 (DMA3 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL3 (Selects cause of DMA3 request)	00: Software start 01: Serial I/O-0 (transmit buffer empty) 10: Serial I/O-1 (reception completed) 11: Extended request cause (Note)	○	○
4	TENL3 (DMA3 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL3 (Selects DMA3 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL3 (Selects DMA3 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL3 (Selects DMA3 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA3 Request Extended Cause Register (DM3REQ).

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

D0	1	2	3	4	5	6	D7
MDSEL4	TREQF4	REQSL4		TENL4	TSZSL4	SADSL4	DADSL4

<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL4 (Selects DMA4 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF4 (DMA4 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL4 (Selects cause of DMA4 request)	00: Software start 01: One DMA3 transfer completed 10: Serial I/O-0 (reception completed) 11: Extended request cause (Note)	○	○
4	TENL4 (DMA4 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL4 (Selects DMA4 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL4 (Selects DMA4 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL4 (Selects DMA4 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA4 Request Extended Cause Register (DM4REQ).

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

D0	1	2	3	4	5	6	D7
MDSEL5	TREQF5	REQSL5		TENL5	TSZSL5	SADSL5	DADSL5

<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL5 (Selects DMA5 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF5 (DMA5 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL5 (Selects cause of DMA5 request)	00: Software start or one DMA7 transfer completed 01: All DMA0 transfers completed 10: Serial I/O-2 (reception completed) 11: Extended request cause (Note)	○	○
4	TENL5 (DMA5 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL5 (Selects DMA5 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL5 (Selects DMA5 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL5 (Selects DMA5 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA5 Request Extended Cause Register (DM5REQ).

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

D0	1	2	3	4	5	6	D7
MDSEL6	TREQF6	REQSL6		TENL6	TSZSL6	SADSL6	DADSL6

<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL6 (Selects DMA6 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF6 (DMA6 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL6 (Selects cause of DMA6 request)	00: Software start 01: Serial I/O-1 (transmit buffer empty) 10: Extended request cause (Note) 11: One DMA5 transfer completed	○	○
4	TENL6 (DMA6 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL6 (Selects DMA6 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL6 (Selects DMA6 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL6 (Selects DMA6 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA6 Request Extended Cause Register (DM6REQ).

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

D0	1	2	3	4	5	6	D7
MDSEL7	TREQF7	REQSL7		TENL7	TSZSL7	SADSL7	DADSL7

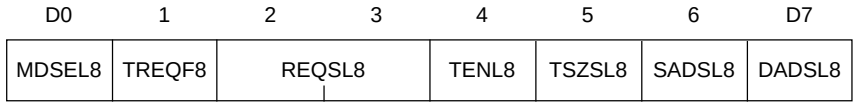
<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL7 (Selects DMA7 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF7 (DMA7 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL7 (Selects cause of DMA7 request)	00: Software start 01: Serial I/O-2 (transmit buffer empty) 10: Extended request cause (Note) 11: One DMA6 transfer completed	○	○
4	TENL7 (DMA7 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL7 (Selects DMA7 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL7 (Selects DMA7 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL7 (Selects DMA7 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA7 Request Extended Cause Register (DM7REQ).

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



<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL8 (Selects DMA8 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF8 (DMA8 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL8 (Selects cause of DMA8 request)	00: Software start 01: All DMA3 transfers completed 10: Serial I/O-3 (reception completed) 11: Extended request cause (Note)	○	○
4	TENL8 (DMA8 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL8 (Selects DMA8 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL8 (Selects DMA8 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL8 (Selects DMA8 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA8 Request Extended Cause Register (DM8REQ).

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

D0	1	2	3	4	5	6	D7
MDSEL9	TREQF9	REQSL9		TENL9	TSZSL9	SADSL9	DADSL9

<When reset: H'00>

D	Bit Name	Function	R	W
0	MDSEL9 (Selects DMA9 transfer mode)	0: Normal mode 1: Ring buffer mode	○	○
1	TREQF9 (DMA9 transfer request flag)	0: Not requested 1: Requested	○	△
2, 3	REQSL9 (Selects cause of DMA9 request)	00: Software start 01: Serial I/O-3 (transmit buffer empty) 10: Extended request cause (Note) 11: One DMA8 transfer completed	○	○
4	TENL9 (DMA9 transfer enable)	0: Disables transfer 1: Enables transfer	○	○
5	TSZSL9 (Selects DMA9 transfer size)	0: 16 bits 1: 8 bits	○	○
6	SADSL9 (Selects DMA9 source address direction)	0: Fixed 1: Increment	○	○
7	DADSL9 (Selects DMA9 destination address direction)	0: Fixed 1: Increment	○	○

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note: When "Extended request cause" is selected, be sure to set the DMA9 Request Extended Cause Register (DM9REQ).

The DMA Channel Control Register consists of a bit to select DMA transfer mode for each channel, set DMA transfer request flag, and the bits to select the cause of DMA request, enable a DMA transfer, set the transfer size, and source/destination address directions.

(1) MDSELn (DMAn transfer mode select) bit (D0)

This bit when in single transfer mode selects normal or ring buffer mode. Setting this bit to 0 selects normal mode, and setting this bit to 1 selects ring buffer mode.
In ring buffer mode, operation starts from the transfer start address and when transferred 32 times, returns to the transfer start address again, from which transfer operation restarts. In this case, the transfer count register operates in free-running mode, so that transfer operation is continued until the transfer enable bit is set to 0 (to disable transfer). No DMA transfer-finished interrupts are generated.

(2) TREQFn (DMAn transfer request flag) bit (D1)

This flag is set to 1 when a DAM transfer request occurs. Reading this flag helps to know whether there is a DMA transfer request on any channel.
The DMA transfer request is cleared by writing 0 to this bit. Writing 1 has no effect, the bit retains the value it had before writing.
Even when a new DMA transfer request occurs for a channel whose DMA transfer request flag is already set to 1, the next DMA transfer request is not accepted until after a transfer on the channel is completed.

(3) REQSLn (cause of DMAn request select) bits (D2, D3)

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

Note: If "Extended request cause" is selected for the cause of DMA request, always be sure to set the DMA Request Extended Cause Register to select a DMA request extended cause.

(4) TENLn (DMAn transfer enable) bit (D4)

Setting this bit to 1 enables transfer, making a DMA transfer ready to run. Setting this bit to 0 disables transfer. However, if a transfer request has already been accepted, transfer is not disabled until after the requested transfer is completed.

(5) TSZSLn (DMAn transfer size select) bit (D5)

This bit selects the number of data bits to be transferred in one DMA transfer operation (unit of one transfer). The unit of one transfer is 16 bits when this bit = 0, or 8 bits when this bit = 1.

(6) SADSLn (DMAn source address direction select) bit (D6)

This bit selects the direction in which the source address changes from two modes available: address fixed or address increment.

(7) DADSLn (DMAn destination address direction select) bit (D7)

This bit selects the direction in which the destination address changes from two modes available: address fixed or address increment.

9.2.2 DMA Request Extended Cause Register

■ **DMA0 Request Extended Cause Register (DM0REQ)** **<Address: H'0080 0416>**

<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL0 (Selects DMA0 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN0A input signal 0101: Serial I/O-2 (transmit buffer empty) 0110: Serial I/O-7 (transmit buffer empty) 0111: No selection 1xxx: No selection	○	○

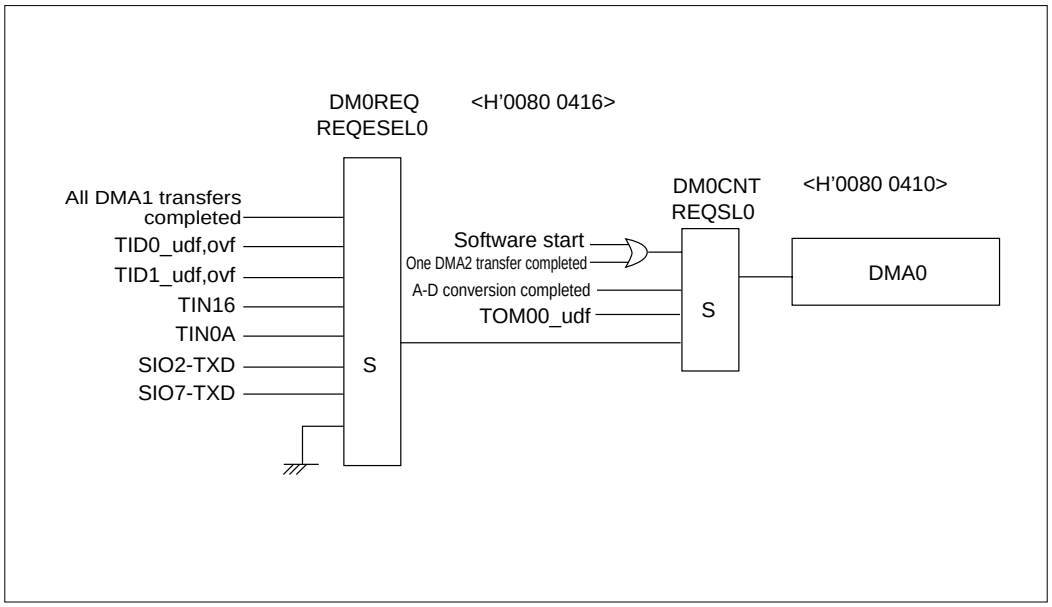
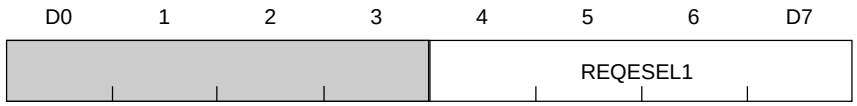


Figure 9.2.3 Block Diagram of the DMA0 Request Extended Cause

■ **DMA1 Request Extended Cause Register (DM1REQ)** **<Address: H'0080 0426>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL1 (Selects DMA1 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN17 input signal 0101: TOM02_udf 0110: Serial I/O-4 (reception completed) 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP1 101x: No selection 11xx: No selection	○	○

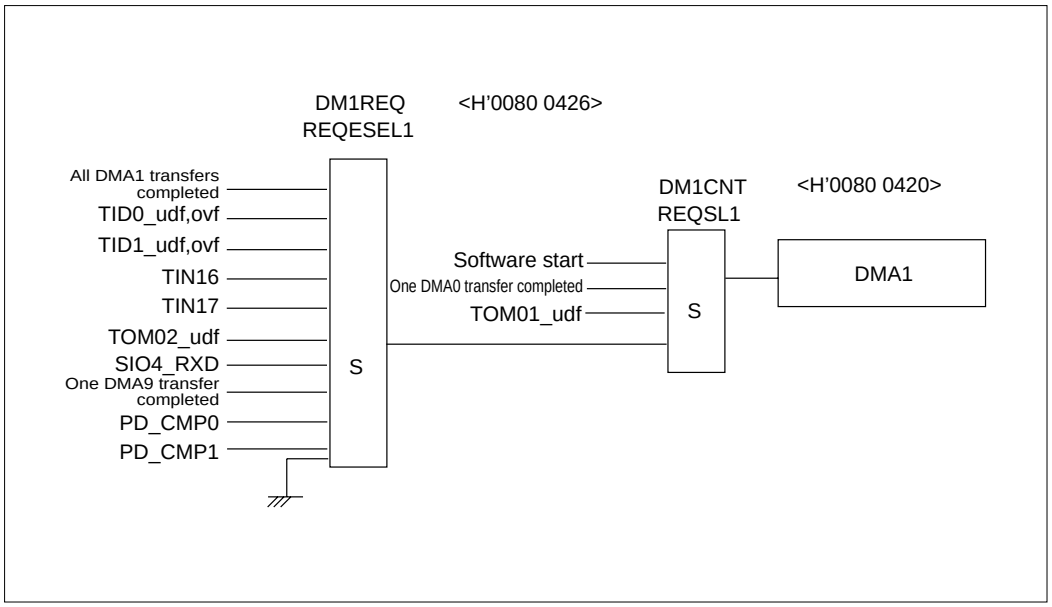
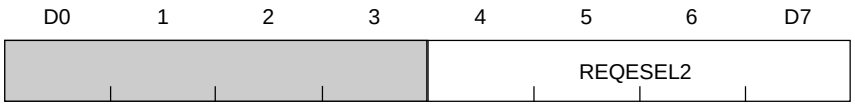


Figure 9.2.4 Block Diagram of the DMA1 Request Extended Cause

■ **DMA2 Request Extended Cause Register (DM2REQ)** **<Address: H'0080 0436>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL2 (Selects DMA2 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN18 input signal 0101: TOM07_udf 0110: Serial I/O-5 (reception completed) 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP1 101x: No selection 11xx: No selection	○	○

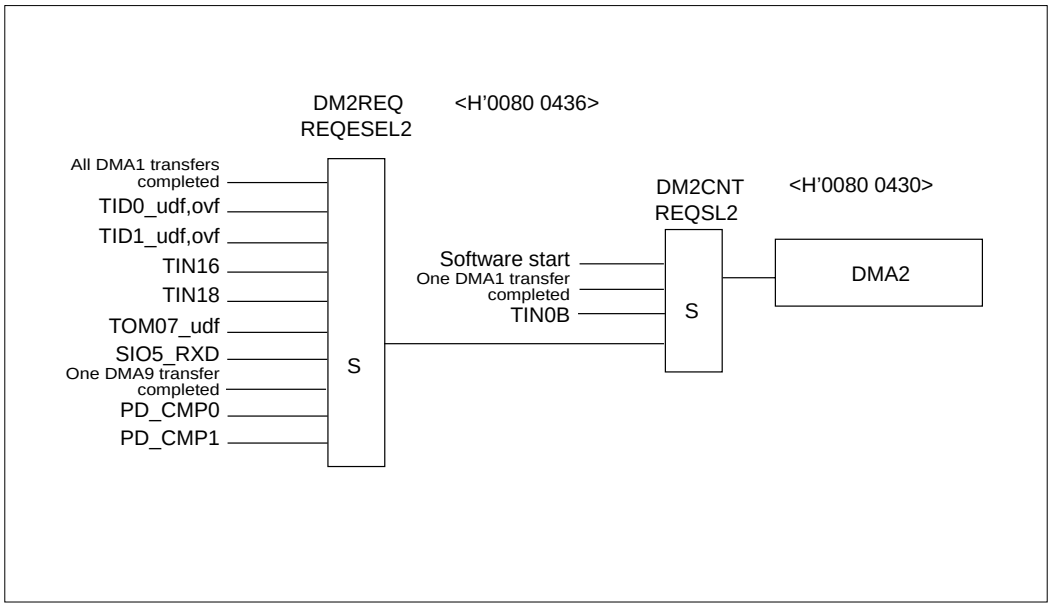
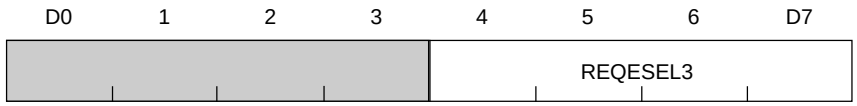


Figure 9.2.5 Block Diagram of the DMA2 Request Extended Cause

■ **DMA3 Request Extended Cause Register (DM3REQ)** **<Address: H'0080 0446>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL3 (Selects DMA3 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN19 input signal 0101: TOM03_udf 0110: A-D1 conversion completed 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP1 101x: No selection 11xx: No selection	○	○

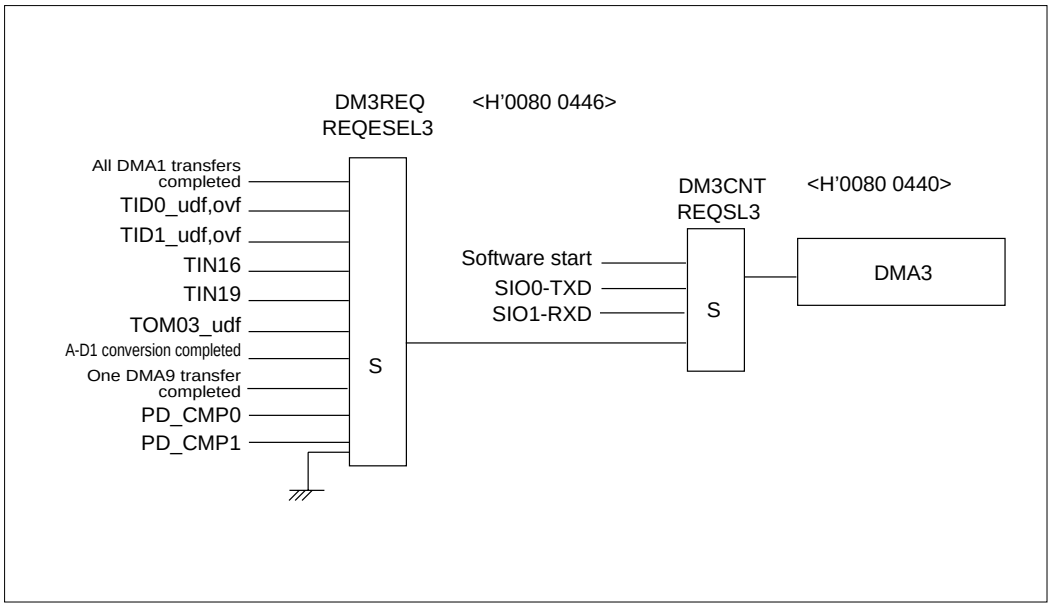
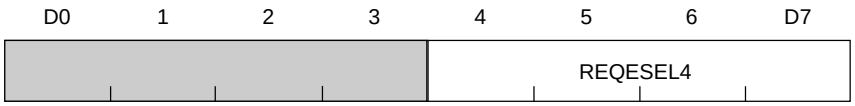


Figure 9.2.6 Block Diagram of the DMA3 Request Extended Cause

■ **DMA4 Request Extended Cause Register (DM4REQ)** **<Address: H'0080 0456>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL4 (Selects DMA4 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN23 input signal 0101: TOM04_udf 0110: Serial I/O-3 (transmit buffer empty) 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP 101x: No selection 11xx: No selection	○	○

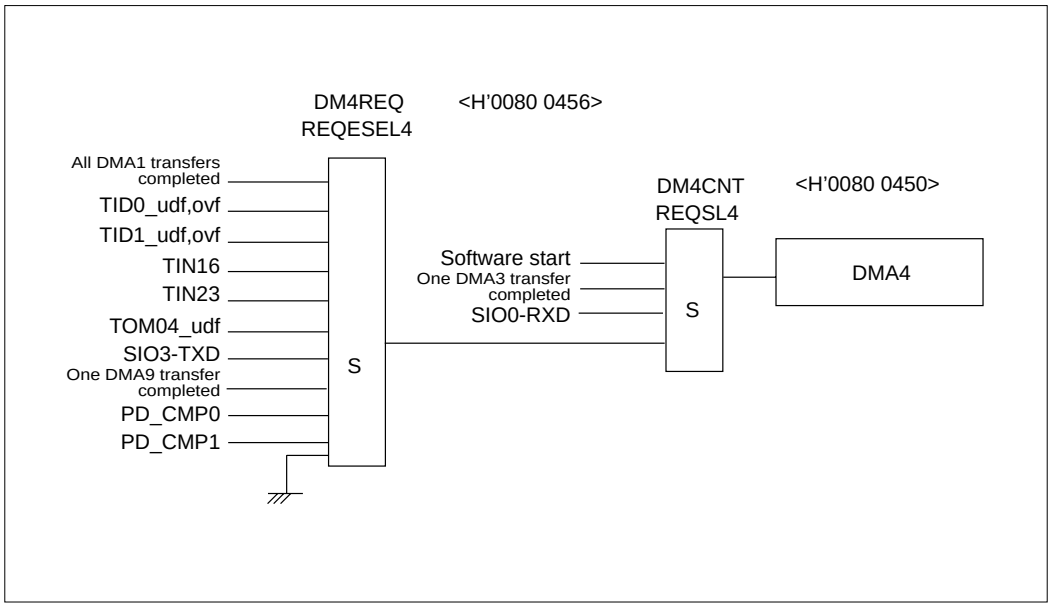
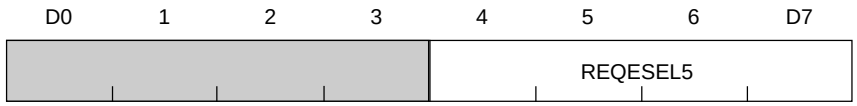


Figure 9.2.7 Block Diagram of the DMA4 Request Extended Cause

■ **DMA5 Request Extended Cause Register (DM5REQ)** **<Address: H'0080 041E>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL5 (Selects DMA5 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN1A input signal 0101: TOM16_udf 0110: Serial I/O-4 (transmit buffer empty) 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP1 101x: No selection 11xx: No selection	○	○

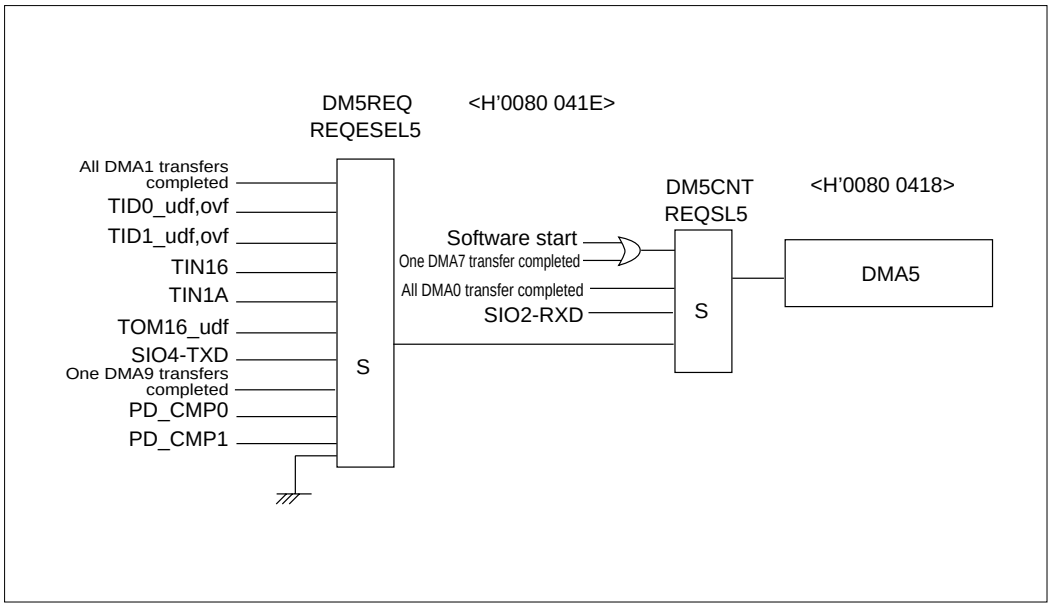
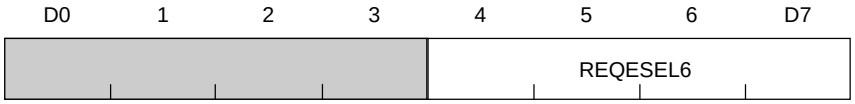


Figure 9.2.8 Block Diagram of the DMA5 Request Extended Cause

■ **DMA6 Request Extended Cause Register (DM6REQ)** **<Address: H'0080 042E>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL6 (Selects DMA6 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN20 input signal 0101: TOM05_udf 0110: Serial I/O-6 (reception completed) 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP1 101x: No selection 11xx: No selection	○	○

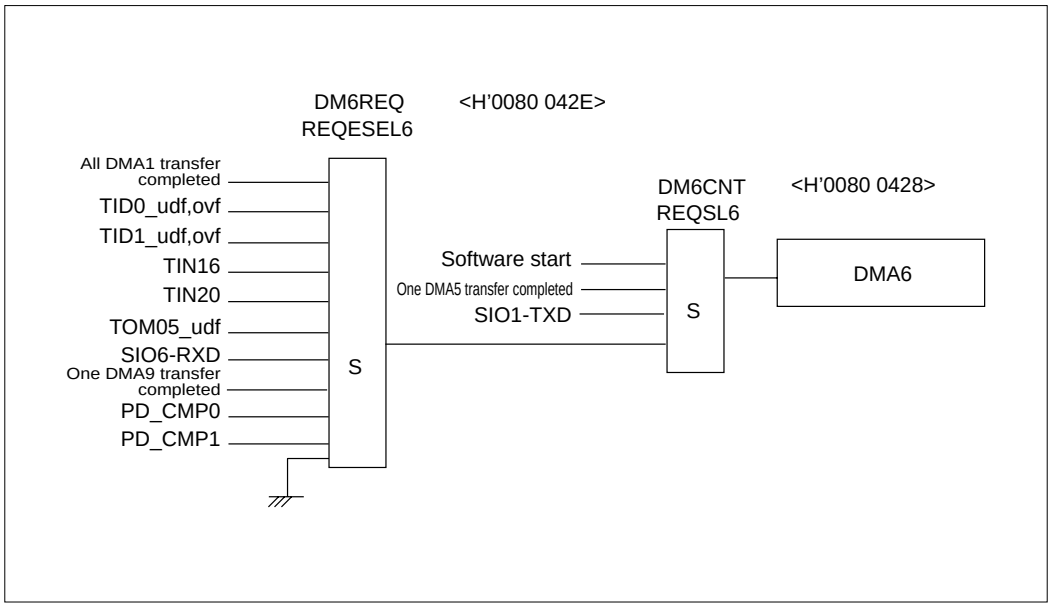
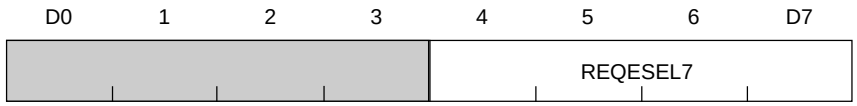


Figure 9.2.9 Block Diagram of the DMA6 Request Extended Cause

■ **DMA7 Request Extended Cause Register (DM7REQ)** **<Address: H'0080 043E>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL7 (Selects DMA7 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN1B input signal 0101: TOM06_udf 0110: Serial I/O-5 (transmit buffer empty) 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP1 101x: No selection 11xx: No selection	○	○

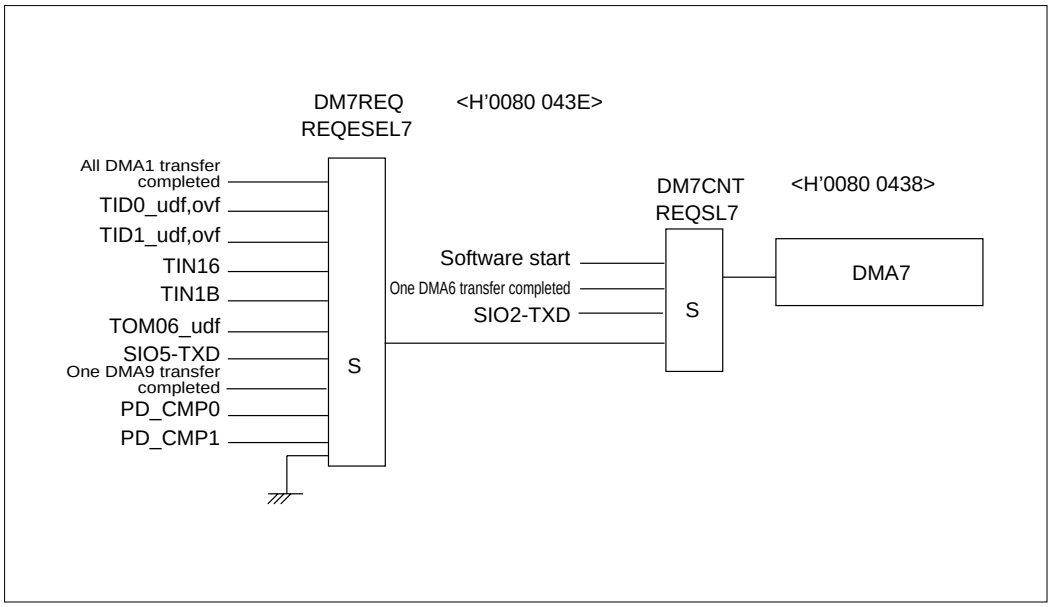
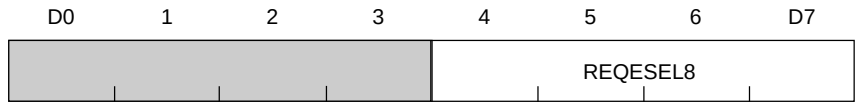


Figure 9.2.10 Block Diagram of the DMA7 Request Extended Cause

■ **DMA8 Request Extended Cause Register (DM8REQ)** **<Address: H'0080 044E>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL8 (Selects DMA8 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN21 input signal 0101: TOM17_udf 0110: Serial I/O-6 (transmit buffer empty) 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP1 101x: No selection 11xx: No selection	○	○

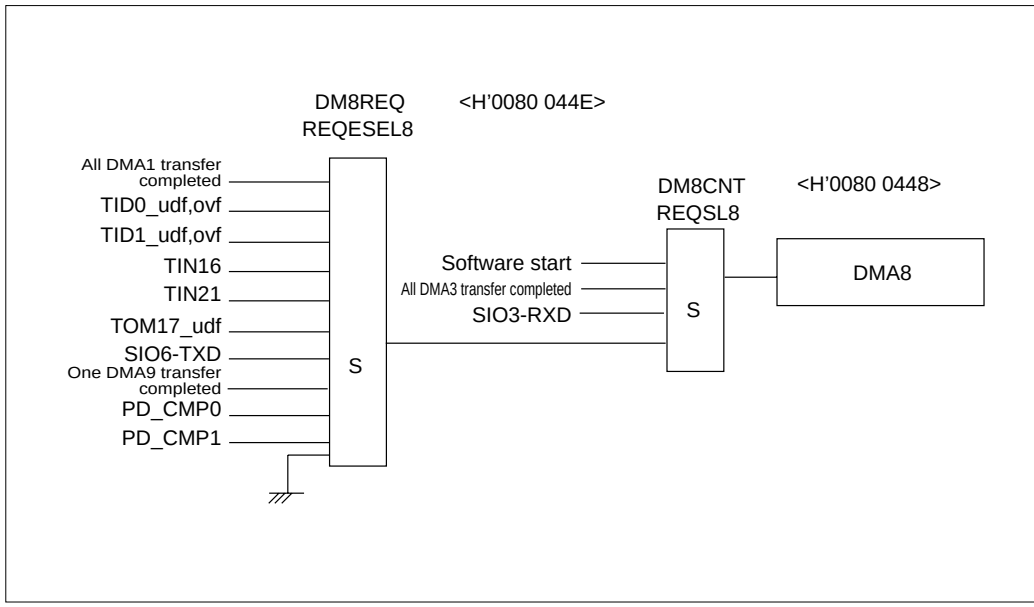
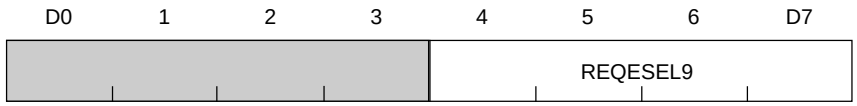


Figure 9.2.11 Block Diagram of the DMA8 Request Extended Cause

■ **DMA9 Request Extended Cause Register (DM9REQ)** **<Address: H'0080 045E>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	REQESEL9 (Selects DMA9 request extended cause)	0000: All DMA1 transfers completed 0001: TID0_udf,ovf 0010: TID1_udf,ovf 0011: TIN16 input signal 0100: TIN22 input signal 0101: TOM10_udf 0110: Serial I/O-7 (reception completed) 0111: One DMA9 transfer completed 1000: PD_CMP0 1001: PD_CMP1 101x: No selection 11xx: No selection	○	○

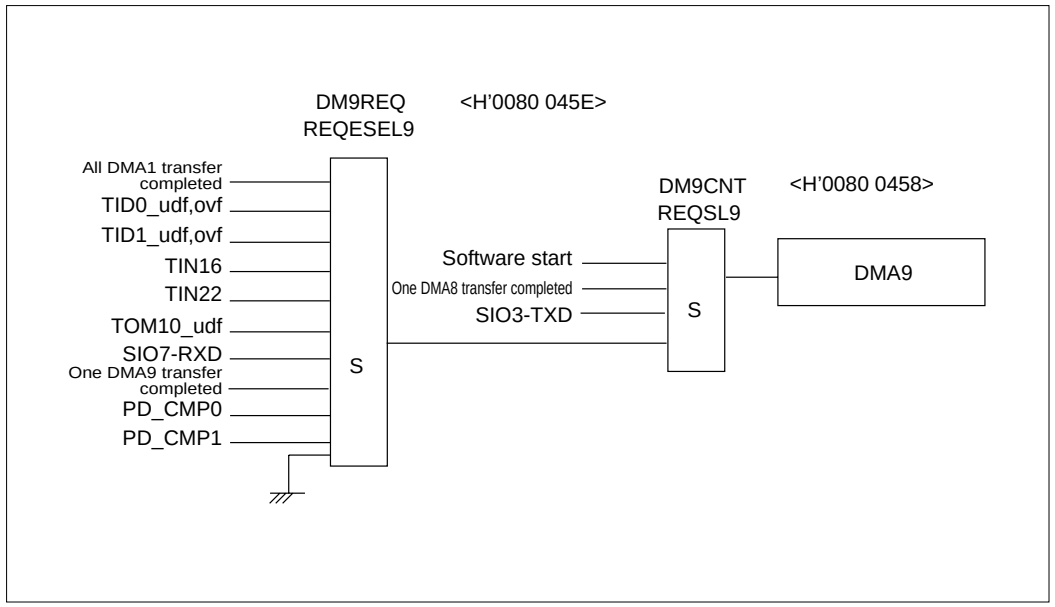


Figure 9.2.12 Block Diagram of the DMA9 Request Extended Cause

The DMA Request Extended Source Register is used to select a DMA transfer request extended source when "Extended request source" has been selected with the DMA Channel Control Register's cause of DMA request select (REQSLn) bits.

(1) REQESELn (DMAn request extended source select) bits (D4-D7)

These bits select a DMA transfer request extended source for each DMA channel.

Note: The DMA transfer request extended source selected with the REQESELn (DMAn request extended source select) bits is effective only when "Extended request source" has been selected with the DMA Channel Control Register's cause of DMA request select (REQSLn) bits.

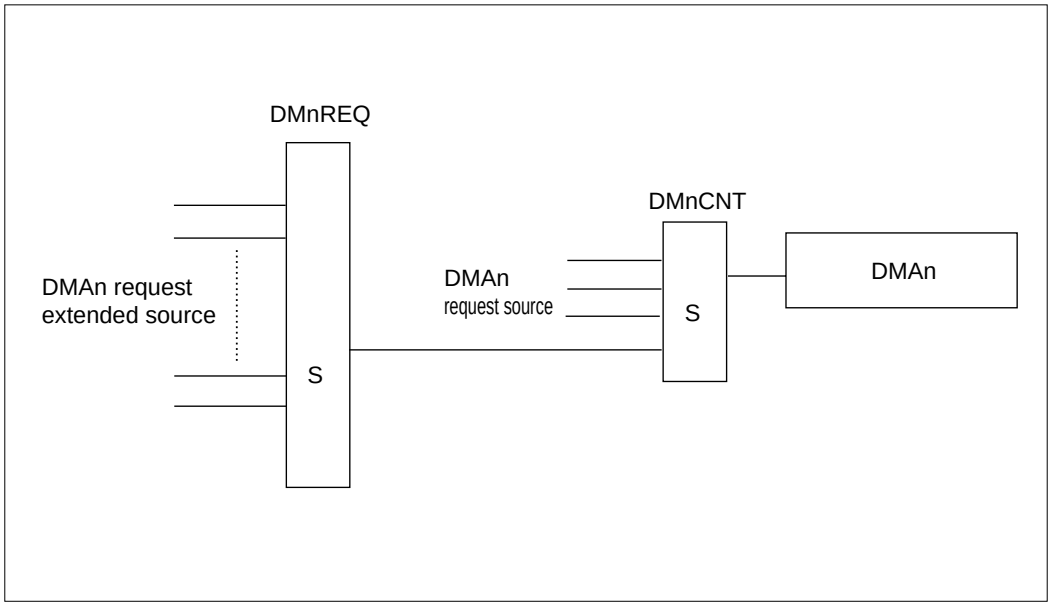
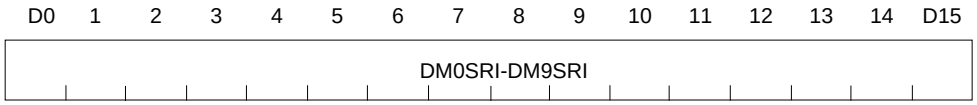


Figure 9.2.13 Block Diagram of the DMAn Request Extended Source

9.2.3 DMA Software Request Generation Registers

■ **DMA0 Software Request Generation Registers (DM0SRI)** **<Address: H'0080 0460>**
■ **DMA1 Software Request Generation Registers (DM1SRI)** **<Address: H'0080 0462>**
■ **DMA2 Software Request Generation Registers (DM2SRI)** **<Address: H'0080 0464>**
■ **DMA3 Software Request Generation Registers (DM3SRI)** **<Address: H'0080 0466>**
■ **DMA4 Software Request Generation Registers (DM4SRI)** **<Address: H'0080 0468>**
■ **DMA5 Software Request Generation Registers (DM5SRI)** **<Address: H'0080 0470>**
■ **DMA6 Software Request Generation Registers (DM6SRI)** **<Address: H'0080 0472>**
■ **DMA7 Software Request Generation Registers (DM7SRI)** **<Address: H'0080 0474>**
■ **DMA8 Software Request Generation Registers (DM8SRI)** **<Address: H'0080 0476>**
■ **DMA9 Software Request Generation Registers (DM9SRI)** **<Address: H'0080 0478>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	DM0SRI-DM9SRI	A DMA transfer request is generated	?	○
	(Generates DMA software request)	by writing any data to this register.		

Note: This register can be accessed in either bytes or halfwords.

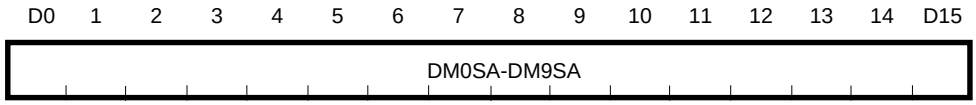
The DMA Software Request Generation Register is used to generate a DMA transfer request in software. A DMA transfer request is generated by writing any data to this register while "Software start" has been selected for the DMA request source.

DM0SRI-DM9SRI (DMA software request generation) bits

A software DMA transfer request is generated by writing any data-halfword (16 bits) or a byte (8 bits) starting from an even or odd address-to this register while "Software start" has been selected for the cause of DMA source (by setting the DMA Channel Control Register D2, D3 bits to '00').

9.2.4 DMA Source Address Registers

■ **DMA0 Source Address Registers (DM0SA)** **<Address: H'0080 0412>**
■ **DMA1 Source Address Registers (DM1SA)** **<Address: H'0080 0422>**
■ **DMA2 Source Address Registers (DM2SA)** **<Address: H'0080 0432>**
■ **DMA3 Source Address Registers (DM3SA)** **<Address: H'0080 0442>**
■ **DMA4 Source Address Registers (DM4SA)** **<Address: H'0080 0452>**
■ **DMA5 Source Address Registers (DM5SA)** **<Address: H'0080 041A>**
■ **DMA6 Source Address Registers (DM6SA)** **<Address: H'0080 042A>**
■ **DMA7 Source Address Registers (DM7SA)** **<Address: H'0080 043A>**
■ **DMA8 Source Address Registers (DM8SA)** **<Address: H'0080 044A>**
■ **DMA9 Source Address Registers (DM9SA)** **<Address: H'0080 045A>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	DM0SA-DM9SA	Source address A16-A31	○	○
	(DMA source address)	(A0-A15 are fixed to H'0080)		

Note: This register must always be accessed in halfwords.

In the DMA Source Address Register, set the source address of DMA transfer in such a way that D0 is for A16, and D15 is for A31. Because this register is comprised of a current register, its read value is always the current value.
When a DMA transfer is finished (at which the transfer count register underflows), the source address remains unchanged or changes depending on the selected transfer address direction. If "Address fixed" is selected, the source address remains the same as the set value before the DMA transfer started; if "Address increment" is selected, the source address is the last transfer address + 1 (for 8-bit transfer) or the last transfer address + 2 (for 16-bit transfer).

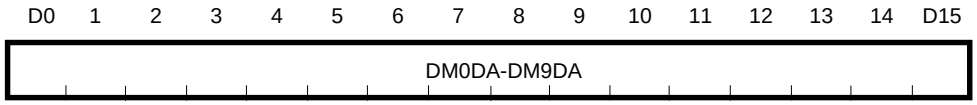
The DMA Source Address Register must always be accessed in halfwords (16 bits) beginning with an even address. If this register is accessed in bytes, its value becomes indeterminate.

DM0SA-DM9SA (source address A16-A31)

Set this register to specify the source address of DMA transfer in the internal I/O or RAM space from address H'0080 0000 to H'0080 FFFF.
The 16 high-order bits of the source address (A0 - A15) are always fixed to H'0080. Use this register to set the 16 low-order bits of the source address. (D0 corresponds to the source address A16, and D15 corresponds to the source address A31.)

9.2.5 DMA Destination Address Registers

■ **DMA0 Destination Address Registers (DM0DA)** **<Address: H'0080 0414>**
■ **DMA1 Destination Address Registers (DM1DA)** **<Address: H'0080 0424>**
■ **DMA2 Destination Address Registers (DM2DA)** **<Address: H'0080 0434>**
■ **DMA3 Destination Address Registers (DM3DA)** **<Address: H'0080 0444>**
■ **DMA4 Destination Address Registers (DM4DA)** **<Address: H'0080 0454>**
■ **DMA5 Destination Address Registers (DM5DA)** **<Address: H'0080 041C>**
■ **DMA6 Destination Address Registers (DM6DA)** **<Address: H'0080 042C>**
■ **DMA7 Destination Address Registers (DM7DA)** **<Address: H'0080 043C>**
■ **DMA8 Destination Address Registers (DM8DA)** **<Address: H'0080 044C>**
■ **DMA9 Destination Address Registers (DM9DA)** **<Address: H'0080 045C>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	DM0DA-DM9DA	Destination address A16-A31	○	○
	(DMA destination address)	(A0-A15 are fixed to H'0080)		

Note: This register must always be accessed in halfwords.

In the DMA Destination Address Register, set the destination address of DMA transfer in such a way that D0 is for A16, and D15 is for A31. Because access to this register is comprised of a current register, its read value is always the current value.
When a DMA transfer is finished (at which the transfer count register underflows), the destination address remains unchanged or changes depending on the selected transfer address direction. If "Address fixed" is selected, the destination address remains the same as the set value before the DMA transfer started; if "Address increment" is selected, the destination address is the last transfer address + 1 (for 8-bit transfer) or the last transfer address + 2 (for 16-bit transfer).

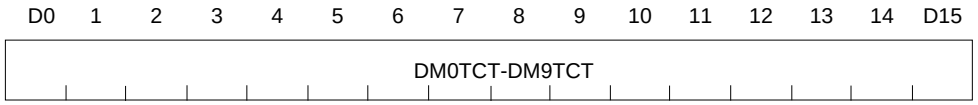
The DMA Destination Address Register must always be accessed in halfwords (16 bits) beginning with an even address. If this register is accessed in bytes, its value becomes indeterminate.

DM0DA-DM9DA (destination address A16-A31)

Set this register to specify the destination address of DMA transfer in the internal I/O or RAM space from address H'0080 0000 to H'0080 FFFF.
The 16 high-order bits of the destination address (A0 - A15) are always fixed to H'0080. Use this register to set the 16 low-order bits of the destination address. (D0 corresponds to the destination address A16, and D15 corresponds to the destination address A31.)

9.2.6 DMA Transfer Count Registers

■ **DMA0 Transfer Count Registers (DM0TCT)** **<Address: H'0080 0411>**
■ **DMA1 Transfer Count Registers (DM1TCT)** **<Address: H'0080 0421>**
■ **DMA2 Transfer Count Registers (DM2TCT)** **<Address: H'0080 0431>**
■ **DMA3 Transfer Count Registers (DM3TCT)** **<Address: H'0080 0441>**
■ **DMA4 Transfer Count Registers (DM4TCT)** **<Address: H'0080 0451>**
■ **DMA5 Transfer Count Registers (DM5TCT)** **<Address: H'0080 0419>**
■ **DMA6 Transfer Count Registers (DM6TCT)** **<Address: H'0080 0429>**
■ **DMA7 Transfer Count Registers (DM7TCT)** **<Address: H'0080 0439>**
■ **DMA8 Transfer Count Registers (DM8TCT)** **<Address: H'0080 0449>**
■ **DMA9 Transfer Count Registers (DM9TCT)** **<Address: H'0080 0459>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
8-15	DM0TCT-DM9TCT	Number of times a DMA transfer is performed	○	○
	(DMA transfer count)	(Ignored during 32-channel ring buffer mode)		

The DMA Transfer Count Register is used to set the number of times data is transferred in each channel. However, the value in this register is ignored 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 value you get by reading this register is always the current value. (However, if you read this register in a cycle immediately after a transfer, the value you get is the value that was in the count register before the transfer began.) When transfer finishes, this count register underflows, so that the read value you get is H'FF.

If any cascaded channel exists, each time one DMA transfer (byte or halfword) is completed, or when all transfers are completed (at which the transfer count register underflows), transfer in the cascaded channel starts.

9.2.7 DMA Interrupt Request Status Registers

■ DMA0-4 Interrupt Request Status Register (DM04ITST) **<Address: H'0080 0400>**

D0	1	2	3	4	5	6	D7
			DMITST4	DMITST3	DMITST2	DMITST1	DMITST0

<When reset: H'00>

D	Bit Name	Function	R	W
0-2	No functions assigned		0	–
3	DMITST4 (DMA4 interrupt request status)	0: No interrupt requested 1: Interrupt requested	○	△
4	DMITST3 (DMA3 interrupt request status)			
5	DMITST2 (DMA2 interrupt request status)			
6	DMITST1 (DMA1 interrupt request status)			
7	DMITST0 (DMA0 interrupt request status)			

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

The DMA0-4 Interrupt Request Status Register helps to know the status of interrupt requests on DMA channels 0-4. If the DMAn interrupt request status bit (n = 0-4) is set to 1, it means that the channel has a DMA interrupt request generated.

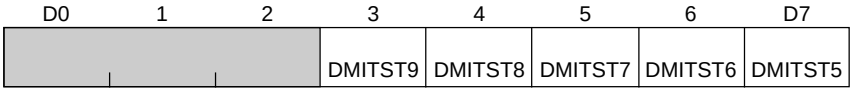
DMITSTn (DMAn interrupt request status) bit (n = 0-4)

The DMAn interrupt request status bit is set in hardware, and cannot be set in software.
The DMAn interrupt request status bit is cleared by writing 0 in software.

Note: The DMAn interrupt request status bit is not cleared by writing 0 to the DMA Interrupt Control Register "interrupt request bit" of the Interrupt Controller.

When writing to the DMA0-4 Interrupt Request Status Register, be sure to write 0 for the bits to be cleared and 1 for all other bits. Writing 1 in software does not affect any bit of this register, the bit retains the value it had before writing.

■ **DMA5-9 Interrupt Request Status Register (DM59ITST)** **<Address: H'0080 0408>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-2	No functions assigned		0	–
3	DMITST9 (DMA9 interrupt request status)	0: No interrupt requested 1: Interrupt requested	○	△
4	DMITST8 (DMA8 interrupt request status)			
5	DMITST7 (DMA7 interrupt request status)			
6	DMITST6 (DMA6 interrupt request status)			
7	DMITST5 (DMA5 interrupt request status)			

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

The DMA5-9 Interrupt Request Status Register helps to know the status of interrupt requests on DMA channels 5-9. If the DMAn interrupt request status bit (n = 5-9) is set to 1, it means that the channel has a DMA interrupt request generated.

DMITSTn (DMAn interrupt request status) bit (n = 5-9)

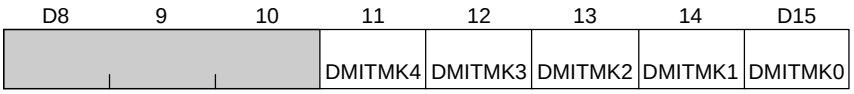
The DMAn interrupt request status bit is set in hardware, and cannot be set in software.
The DMAn interrupt request status bit is cleared by writing 0 in software.

Note: The DMAn interrupt request status bit is not cleared by writing 0 to the DMA Interrupt Control Register "interrupt request bit" of the Interrupt Controller.

When writing to the DMA5-9 Interrupt Request Status Register, be sure to write 0 for the bits to be cleared and 1 for all other bits. Writing 1 in software does not affect any bit of this register, the bit retains the value it had before writing.

9.2.8 DMA Interrupt Mask Registers

■ **DMA0-4 Interrupt Mask Register (DM04ITMK)** **<Address: H'0080 0401>**



<When reset: H'00>

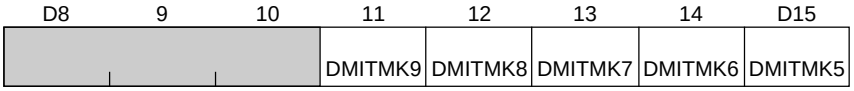
D	Bit Name	Function	R	W
8-10	No functions assigned		0	–
11	DMITMK4 (DMA4 interrupt request mask)	0: Enables interrupt request 1: Masks (disables) interrupt request	○	○
12	DMITMK3 (DMA3 interrupt request mask)			
13	DMITMK2 (DMA2 interrupt request mask)			
14	DMITMK1 (DMA1 interrupt request mask)			
15	DMITMK0 (DMA0 interrupt request mask)			

The DMA0-4 Interrupt Mask Register masks interrupt requests on DMA channels 0-4.

DMITMKn (DMAn interrupt request mask) bits (n = 0-4)

Setting the DMAn interrupt request mask bit to 1 masks a DMAn interrupt request. However, whenever an interrupt request occurs, the DMAn interrupt request status bit is set to 1 no matter how this register is set.

■ **DMA5-9 Interrupt Mask Register (DM59ITMK)** **<Address: H'0080 0409>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-10	No functions assigned		0	–
11	DMITMK9 (DMA9 interrupt request mask)	0: Enables interrupt request 1: Masks (disables) interrupt request	○	○
12	DMITMK8 (DMA8 interrupt request mask)			
13	DMITMK7 (DMA7 interrupt request mask)			
14	DMITMK6 (DMA6 interrupt request mask)			
15	DMITMK5 (DMA5 interrupt request mask)			

The DMA5-9 Interrupt Mask Register masks interrupt requests on DMA channels 5-9.

DMITMKn (DMAn interrupt request mask) bits (n = 5-9)

Setting the DMAn interrupt request mask bit to 1 masks a DMAn interrupt request. However, whenever an interrupt request occurs, the DMAn interrupt request status bit is set to 1 no matter how this register is set.

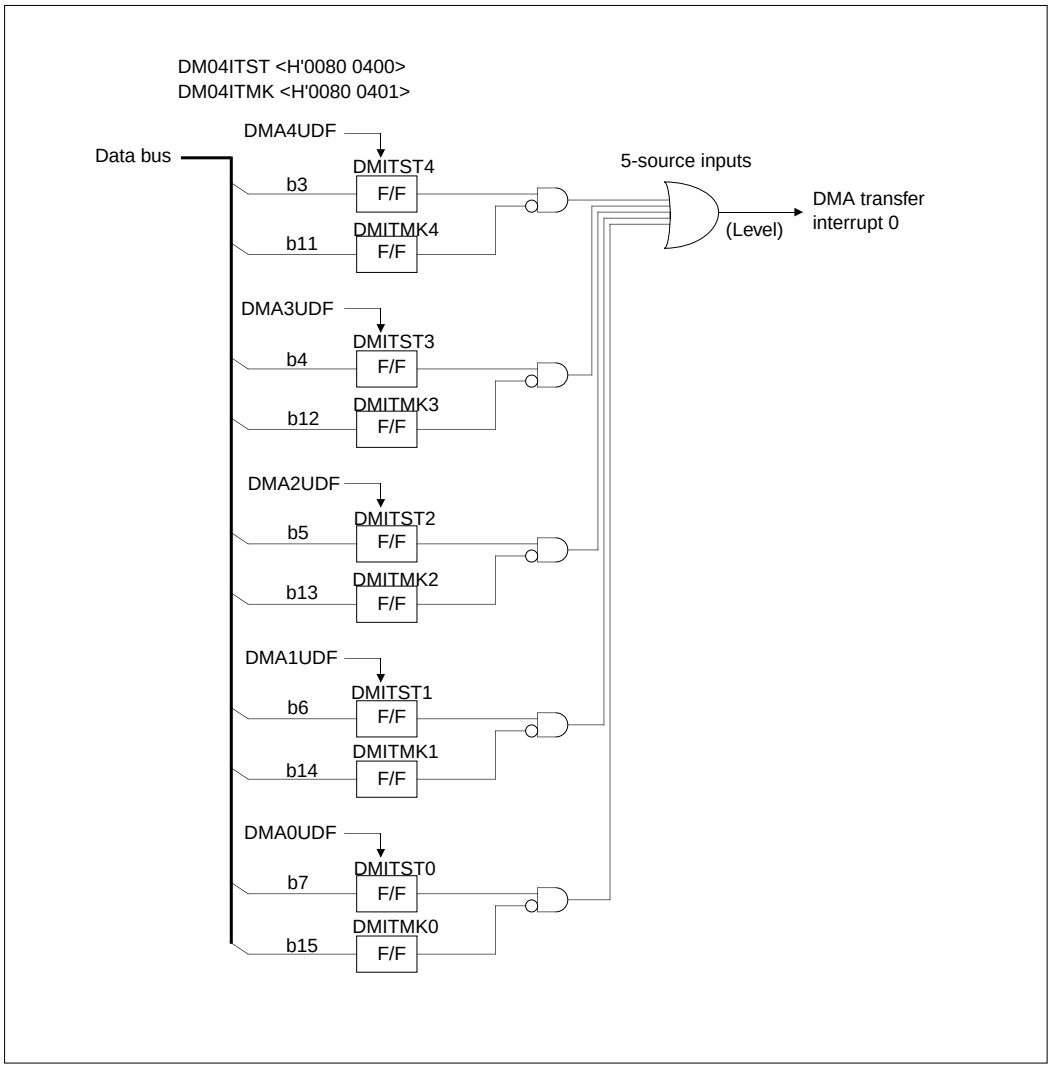


Figure 9.2.14 Block Diagram of DMA Transfer Interrupt 0

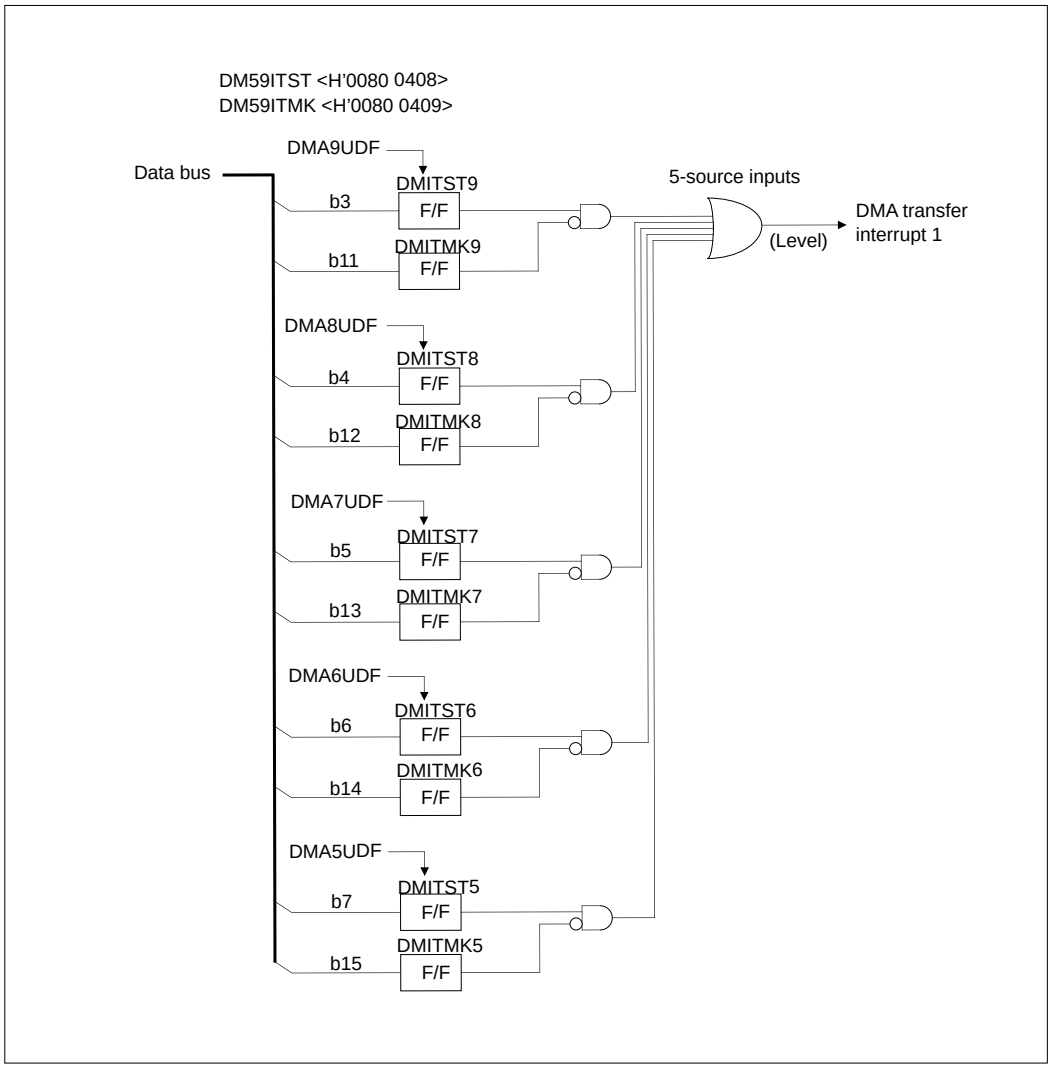


Figure 9.2.15 Block Diagram of DMA Transfer Interrupt 1

9.3 Functional Description of DMAC

9.3.1 Cause of DMA Request

A DMA transfer can be requested individually for each DMA channel (0-9 channels) from multiple sources. There are various causes of DMA transfer request, so that a DMA transfer may be started by an internal peripheral I/O, started in software by a program, or started by completion of one or all DMA transfers on another channel (cascade mode).

To select a cause of DMA request, use the cause of request select bits for the desired channel REQSLn (DMAn Channel Control Register D2, D3 bits) or the extended request cause select bits REQESELn (DMAn Request Extended Cause Register D4-D7 bits). The causes of DMA request on each channel are listed below.

Table 9.3.1 Causes of DMA Request on DMA0 and the Timing at Which Requests are Generated

REQSL0	Causes of DMA Request	DMA 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 one transfer on DMA2 is completed (cascade mode)
0 1	A-D0 conversion completed	When A-D0 conversion is completed
1 0	Timer (TOM00_udf)	When TOM00 timer underflows
1 1	Extended request cause	–

Table 9.3.2 DMA Request Extended Causes on DMA0 and the Timing at Which Requests are Generated

REQESEL0	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN0A input signal	When PD module's TIN0A input signal is generated
0101	Serial I/O-2 (transmit buffer empty)	When serial I/O-2 transmit buffer is empty
0110	Serial I/O-7 (transmit buffer empty)	When serial I/O-7 transmit buffer is empty

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA0 Channel Control Register REQSL0 bits.

Table 9.3.3 Causes of DMA Request on DMA1 and the Timing at Which Requests are Generated

REQSL1	Causes of DMA Request	DMA Request Generation Timing
0 0	Software start	When any data is written to the DMA1 Software Request Generation Register
0 1	Extended request cause	–
1 0	Timer (TOM01_udf)	When TOM01 timer underflows
1 1	One DMA0 transfer completed	When one transfer on DMA0 is completed (cascade mode)

Table 9.3.4 DMA Request Extended Causes on DMA1 and the Timing at Which Requests are Generated

REQESEL1	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN17 input signal	When the timer's TIN17 input signal is generated
0101	TOM02_udf	When TOM02 timer underflows
0110	Serial I/O-4 (reception completed)	When serial I/O-4 reception is completed
0111	One DMA9 transfer completed	When one transfer on DMA9 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA1 Channel Control Register REQSL1 bits.

Table 9.3.5 Causes of DMA Request on DMA2 and the Timing at Which Requests are Generated

REQSL2	Causes of DMA Request	DMA Request Generation Timing
0 0	Software start	When any data is written to the DMA2 Software Request Generation Register
0 1	Extended request cause	–
1 0	PD module (TIN0B input signal)	When PD module's TIN0B input signal is generated
1 1	One DMA1 transfer completed	When one transfer on DMA1 is completed (cascade mode)

Table 9.3.6 DMA Request Extended Causes on DMA2 and the Timing at Which Requests are Generated

REQESEL2	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN18 input signal	When the timer's TIN18 input signal is generated
0101	TOM07_udf	When TOM07 timer underflows
0110	Serial I/O-5 (reception completed)	When serial I/O-5 reception is completed
0111	One DMA9 transfer completed	When one transfer on DMA9 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA2 Channel Control Register REQSL2 bits.

Table 9.3.7 Causes of DMA Request on DMA3 and the Timing at Which Requests are Generated

REQSL3	Causes of DMA Request	DMA Request Generation Timing
0 0	Software start	When any data is written to the DMA3 Software Request Generation Register
0 1	Serial I/O-0 (transmit buffer empty)	When serial I/O-0 transmit buffer is empty
1 0	Serial I/O-1 (reception completed)	When serial I/O-1 reception is completed
1 1	Extended request cause	–

Table 9.3.8 DMA Request Extended Causes on DMA3 and the Timing at Which Requests are Generated

REQESEL3	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN19 input signal	When the timer's TIN19 input signal is generated
0101	TOM03_udf	When TOM03 timer underflows
0110	A-D1 conversion completed	When A-D1 conversion is completed
0111	One DMA9 transfer completed	When one transfer on DMA9 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA3 Channel Control Register REQSL3 bits.

Table 9.3.9 Causes of DMA Request on DMA4 and the Timing at Which Requests are Generated

REQSL4	Causes of DMA Request	DMA 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 transfer on DMA3 is completed (cascade mode)
1 0	Serial I/O-0 (reception completed)	When serial I/O-0 reception is completed
1 1	Extended request cause	–

Table 9.3.10 DMA Request Extended Causes on DMA4 and the Timing at Which Requests are Generated

REQESEL4	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN23 input signal	When the timer's TIN23 input signal is generated
0101	TOM04_udf	When TOM04 timer underflows
0110	Serial I/O-3 (transmit buffer empty)	When serial I/O-3 transmit buffer is empty
0111	One DMA9 transfer completed	When one transfer on DMA0 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA4 Channel Control Register REQSL4 bits.

Table 9.3.11 Causes of DMA Request on DMA5 and the Timing at Which Requests are Generated

REQSL5	Causes of DMA Request	DMA 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 one transfer on DMA7 is completed (cascade mode)
0 1	All DMA0 transfers completed	When all transfers on DMA0 are completed (cascade mode)
1 0	Serial I/O-2 (reception completed)	When serial I/O-2 reception is completed
1 1	Extended request cause	–

Table 9.3.12 DMA Request Extended Causes on DMA0 and the Timing at Which Requests are Generated

REQESEL5	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN1A input signal	When PD module's TIN1A input signal is generated
0101	TOM16_udf	When TOM16 timer underflows
0110	Serial I/O-4 (transmit buffer empty)	When serial I/O-4 transmit buffer is empty
0111	One DMA9 transfer completed	When one transfer on DMA9 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA5 Channel Control Register REQSL5 bits.

Table 9.3.13 Causes of DMA Request on DMA6 and the Timing at Which Requests are Generated

REQSL6	Causes of DMA Request	DMA Request Generation Timing
0 0	Software start	When any data is written to the DMA6 Software Request Generation Register
0 1	Serial I/O-1 (transmit buffer empty)	When serial I/O-1 transmit buffer is empty
1 0	Extended request cause	–
1 1	One DMA5 transfer completed	When one transfer on DMA5 is completed (cascade mode)

Table 9.3.14 DMA Request Extended Causes on DMA6 and the Timing at Which Requests are Generated

REQESEL6	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN20 input signal	When the timer's TIN20 input signal is generated
0101	TOM05_udf	When TOM05 timer underflows
0110	Serial I/O-6 (reception completed)	When serial I/O-6 reception is completed
0111	One DMA9 transfer completed	When one transfer on DMA9 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA6 Channel Control Register REQSL6 bits.

Table 9.3.15 Causes of DMA Request on DMA7 and the Timing at Which Requests are Generated

REQSL7	Causes of DMA Request	DMA Request Generation Timing
0 0	Software start	When any data is written to the DMA7 Software Request Generation Register
0 1	Serial I/O-2 (transmit buffer empty)	When serial I/O-2 transmit buffer is empty
1 0	Extended request cause	–
1 1	One DMA6 transfer completed	When one transfer on DMA6 is completed (cascade mode)

Table 9.3.16 DMA Request Extended Causes on DMA7 and the Timing at Which Requests are Generated

REQESEL7	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN1B input signal	When PD module's TIN1B input signal is generated
0101	TOM06_udf	When TOM06 timer underflows
0110	Serial I/O-5 (transmit buffer empty)	When serial I/O-5 transmit buffer is empty
0111	One DMA9 transfer completed	When one transfer on DMA9 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA7 Channel Control Register REQSL7 bits.

Table 9.3.17 Causes of DMA Request on DMA8 and the Timing at Which Requests are Generated

REQSL8	Causes of DMA Request	DMA Request Generation Timing
0 0	Software start	When any data is written to the DMA8 Software Request Generation Register
0 1	All DMA3 transfers completed	When all transfers on DMA3 are completed (cascade mode)
1 0	Serial I/O-3 (reception completed)	When serial I/O-3 reception is completed
1 1	Extended request cause	–

Table 9.3.18 DMA Request Extended Causes on DMA8 and the Timing at Which Requests are Generated

REQESEL8	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN21 input signal	When the timer's TIN21 input signal is generated
0101	TOM17_udf	When TOM17 timer underflows
0110	Serial I/O-6 (transmit buffer empty)	When serial I/O-6 transmit buffer is empty
0111	One DMA9 transfer completed	When one transfer on DMA9 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA8 Channel Control Register REQSL8 bits.

Table 9.3.19 Causes of DMA Request on DMA9 and the Timing at Which Requests are Generated

REQSL9	Causes of DMA Request	DMA Request Generation Timing
0 0	Software start	When any data is written to the DMA9 Software Request Generation Register
0 1	Serial I/O-3 (transmit buffer empty)	When serial I/O-3 transmit buffer is empty
1 0	Extended request cause	–
1 1	One DMA8 transfer completed	When one transfer on DMA8 is completed (cascade mode)

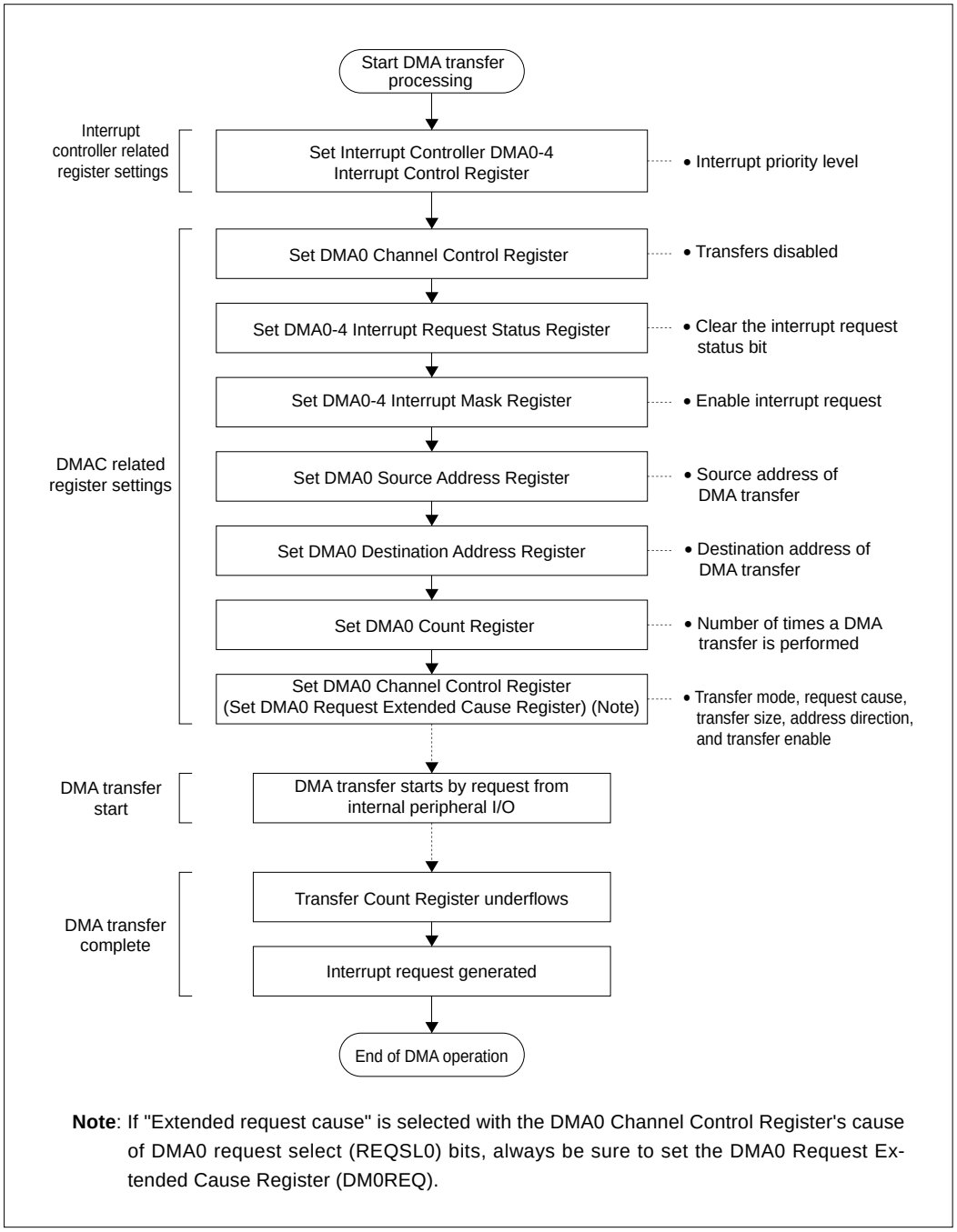
Table 9.3.20 DMA Request Extended Causes on DMA9 and the Timing at Which Requests are Generated

REQESEL9	DMA Request Extended Cause	DMA Request Generation Timing
0000	All DMA1 transfers completed	When all transfers on DMA1 are completed (cascade mode)
0001	TID0_udf,ovf	When TID0 timer underflows or overflows
0010	TID1_udf,ovf	When TID1 timer underflows or overflows
0011	TIN16 input signal	When the timer's TIN16 input signal is generated
0100	TIN22 input signal	When the timer's TIN22 input signal is generated
0101	TOM10_udf	When TOM10 timer underflows
0110	Serial I/O-7 (reception completed)	When serial I/O-7 reception is completed
0111	One DMA9 transfer completed	When one transfer on DMA9 is completed (cascade mode)
1000	PD_CMP0	When absolute data comparison on PD module channel 0 matched
1001	PD_CMP1	When absolute data comparison on PD module channel 1 matched

Note: The DMA request extended causes are effective only when "Extended request cause" is selected with the DMA9 Channel Control Register REQSL9 bits.

9.3.2 DMA Transfer Processing Procedure

The following explains how DMA transfers are performed by using DAM channel 0 as an example.



Note: If "Extended request cause" is selected with the DMA0 Channel Control Register's cause of DMA0 request select (REQSL0) bits, always be sure to set the DMA0 Request Extended Cause Register (DM0REQ).

Figure 9.3.1 DMA Transfer Processing Procedure (Example)

9.3.3 Starting DMA

Use the REQSL (cause of DMA request select) bits to select the cause of DMA request. To enable DMA, set the TENL (DMA transfer enable) bit to 1. A DMA transfer starts when the selected cause of DMA request becomes effective after setting the TENL (DMA transfer enable) bit to 1.

Note: If the request source specified with the REQSL (DMA request source select) bit is an input/output timer (TIN input signal), the time required for DMA transfer to start after detecting the rising or falling edge or both edges of the TIN input signal is three cycles at the least (150 ns when operating with the internal peripheral clock = 20 MHz). Or, depending on the bus usage condition before or after the DMA transfer, up to six cycles may be required (300 ns when operating with the internal peripheral clock = 20 MHz). (The above required time for DMA transfer to start after detecting the TIN input signal is calculated assuming that the external bus is unused, and that HOLD and the LOCK instruction are not used.)
To ensure that changes of state of the TIN input signal will be detected correctly, make sure the TIN input signal has a pulse width of at least 7tc(BCLK)/2. (For details, see Section 23.6, "AC Characteristics.")

9.3.4 Priority of DMA Channels

DMA channel 0 has the highest priority, which is followed by other channels as shown below. The channel priority is fixed.

Channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 >
channel 6 > channel 7 > channel 8 > channel 9 >

Transfer requests on channels are sampled every transfer cycle (= three DMA bus cycles), and the channel that has the highest priority among those which have transfer requests generated is selected.

9.3.5 Gaining and Releasing Control of the Internal Bus

Gaining and releasing control of the internal bus is arbitrated by a single-transfer method DMA on all channels. With the single-transfer method DMA, the DMA controller gains control of the internal bus when a DMA transfer request is accepted and returns bus control to the CPU after executing one session of DMA transfer (one read cycle + one write cycle of the internal peripheral clock). The diagram below shows the behavior of the single-transfer method DMA.

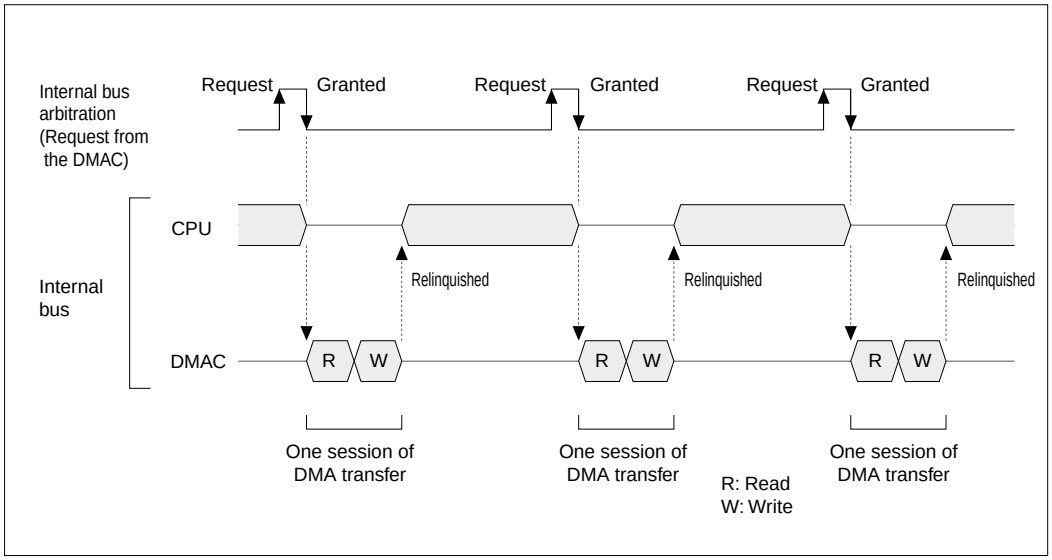


Figure 9.3.2 Gaining and Releasing Control of the Internal Bus

9.3.6 Transfer Unit

The number of data bits transferred in one DMA transfer (8 or 16 bits), referred to as the transfer unit, is selected for each channel with the TSZSL (DMA transfer size select) bit.

9.3.7 Transfer Count

The transfer count, or the number of times a DMA transfer is performed, is set for each channel by using the DMA Transfer Count Register.
DMA transfer can be performed up to 256 times. The value of the transfer count register is decremented by one for each transfer unit transferred.

In ring buffer mode, the DMA Transfer Count Register operates in free-running mode, with its set value ignored.

9.3.8 Address Space

The address space in which DMA transfers can be performed is 64 Kbytes (H'0080 0000 to H'0080 FFFF) of internal peripheral I/O or RAM for both source and destination. The source and destination addresses on each channel are set using the DMA Source Address and DMA Destination Address Registers.

9.3.9 Transfer Operation

(1) Dual-address transfer

Regardless of the transfer unit, a DMA transfer is performed with two bus cycles consisting of a source read access and a destination write access. (The transfer data is temporarily stored in the DMAC's internal temporary register before being transferred.)

(2) Bus protocol and bus timing

The bus interface is shared with the CPU. Therefore, the bus protocol and bus timing both are the same as in peripheral module access from the CPU.

(3) Transfer rate

The maximum transfer rate is calculated using the equation below.

$$\text{Maximum transfer rate [bytes/second]} = 2 \text{ bytes} \times \frac{1}{1 / f(\text{BCLK}) \times 3 \text{ cycles}}$$

(4) Address count direction and address change

The directions in which the source and destination addresses are counted (address fixed or address increment) are set for each channel by using the SADSL (source address direction select) bit and DADSL (designation address direction select) bit.

When "Address increment" is selected, the address is incremented by 2 when the transfer unit = 16 bits or by 1 when the transfer unit = 8 bits each time one DMA transfer is performed.

Table 9.3.11 Address Count Direction and Address Change

Address count direction	Transfer unit	Address change for one DMA transfer performed
Address fixed	8 bits	0
	16 bits	0
Address increment	8 bits	+1
	16 bits	+2

(5) Transfer count value

The transfer count value is decremented by one for each DMA transfer performed, regardless of whether the transfer unit is 8 or 16 bits.

(6) Transfer byte position

When the transfer unit = 8 bits, the LSB of the address register is effective for both the source and destination. (Therefore, even to odd address or odd to even address transfers, as well as even to even address and odd to odd address transfers, are possible.)

When the transfer unit = 16 bits, the LSB of the address register (address register D15 bit) is ignored, and the two bytes of data transferred are always aligned with the 16-bit bus.

The diagram below shows the valid transfer byte positions.

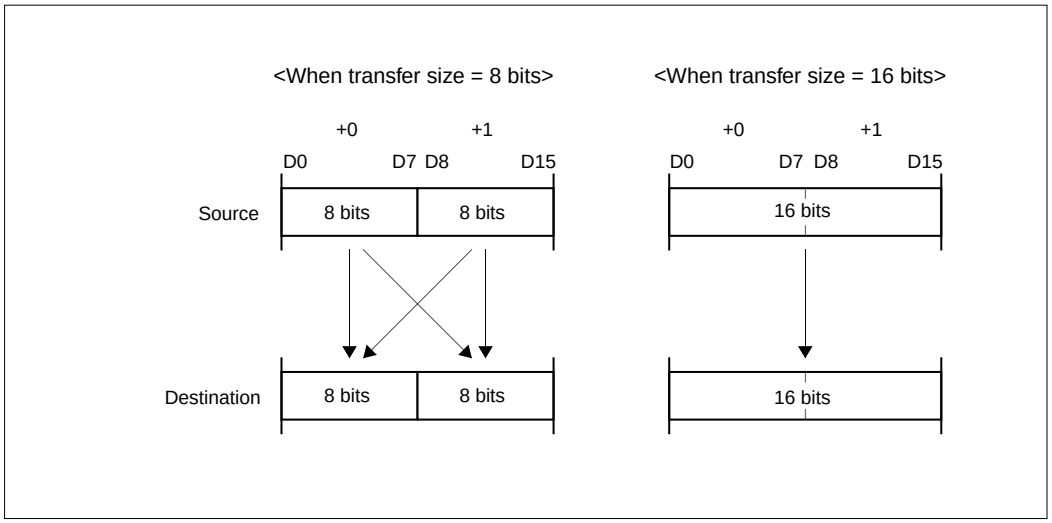


Figure 9.3.3 Transfer Byte Positions

(7) Ring buffer mode

When ring buffer mode is selected, operation starts from the transfer start address and when transferred 32 times, returns to the transfer start address again, from which transfer operation restarts. However, the 5 low-order bits of the ring buffer start address must always be B'00000. The following describes how the 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, with the 5 low-order bits incremented by one for each transfer performed. When the 5 low-order bits reach B'11111 after performing transfers beginning with the start address, the bits are reset to B'00000 by the next increment action, thus returning to the start address.

② **When the transfer size is 16 bits**

The 26 high-order bits of the transfer start address are fixed, with the 6 low-order bits incremented by two for each transfer performed. When the 6 low-order bits reach B'111110 after performing transfers beginning with the start address, the bits are reset to B'000000 by the next increment action, thus returning to the start address.

It is the source address if the source has been set to increment or the destination address if the destination has been set to increment that returns to the start address in this way.
If the source and destination addresses both have been set to increment, both addresses return to the start address. However, the start addresses on either side must always have the 5 low-order bits initially set to B'00000.

During ring buffer mode, the transfer counter register is ignored. After DMA operation starts, the counter operates in free-running mode, with transfers continued until the transfer enable bit is cleared to 0 (to disable transfers).

<When the transfer size = 8 bits>		<When the transfer size = 16 bits>	
Transfer count	Transfer address	Transfer count	Transfer address
1	H'0080 1000	1	H'0080 1000
2	H'0080 1001	2	H'0080 1002
3	H'0080 1002	3	H'0080 1004
|	|	|	|
31	H'0080 101E	31	H'0080 103C
32	H'0080 101F	32	H'0080 103E
↓	↓	↓	↓
1	H'0080 1000	1	H'0080 1000
2	H'0080 1001	2	H'0080 1002
|	|	|	|

Figure 9.3.4 Typical Address Increment Action in 32-channel Ring Buffer Mode

9.3.10 End of DMA and Interrupt

In normal mode, a DMA transfer ends when the transfer count register underflows after reaching the terminal count. When a transfer ends, the transfer enable bit is cleared to 0 to disable transfers. Also an interrupt request is generated at completion of a transfer. For channels where interrupt requests are masked with the DMA Interrupt Mask Register, no interrupts are generated, however.

During ring buffer mode, the transfer counter operates in free-running mode, so that a transfer continues until the transfer enable bit is cleared to 0 (to disable transfers). Therefore, no DMA transfer-finished interrupt requests are generated. Nor are DMA transfer-finished interrupt requests are generated when ring buffer mode transfer is terminated by clearing the transfer enable bit.

9.3.11 Register Status after End of DMA Transfer

When a DMA transfer ends, the Source Address and Destination Address Registers are in the following state.

(1) When address fixed

- The address remains fixed, i.e., the same as the set value before the DMA transfer started

(2) When address incremented

- For 8-bit transfers, the last transfer address + 1
- For 16-bit transfers, the last transfer address + 2

When a DMA transfer has finished, the transfer count register is in an underflow condition (H'FF). Therefore, to reexecute a DMA transfer, set the transfer count register back again, unless you are performing transfers 256 times (H'FF).

9.4 Precautions on Using DMAC

- **About writing to the DMA related registers**

In DMA transfers, data are exchanged via the internal bus. Therefore, make sure the DMA related registers basically are written to immediately after reset or when transfers are disabled (transfer enable bit = 0). When transfers are enabled, do not write to the DMA related registers, except for the DMA transfer enable bit, transfer request flag, and the DMA transfer count register that is protected against write in hardware. This is necessary to ensure stable operation of DMA transfers.

The table below shows whether DMA related registers can be accessed for write.

Table 9.4.1 DMA Related Registers That Can Be and Cannot Be Accessed for Write

State	Transfer enable bit	Transfer request flag	Other DMA related registers
Transfers enabled	○	○	×
Transfers disabled	○	○	○

○: Can be accessed; 5: Cannot be accessed

Even for a few exceptional registers that can be accessed for write while transfers are enabled, make sure the following conditions are met.

① **DMA Channel Control Register's transfer enable bit and transfer request flag**

For all other bits of the channel control register, write the same data as they had before writing. Note that for the transfer request flag, only writing 0 is effective.

② **DMA Transfer Count Register**

When transfers are enabled, this register is protected against write in hardware. Therefore, the data written to this register is ignored.

③ **Rewriting the source and destination addresses of DMA on other channels by a DMA transfer**

Although this means accessing the DMA related registers while transfers are enabled, it may not cause any problem. However, a DMA transfer to any DMA related registers on the current channel cannot be performed.

- **Operating on DMA related registers by a DMA transfer**

When operating on DMA related registers using a DMA transfer (e.g., reloading the DMA related registers with their initial values by a DMA transfer), do not write to any DMA related registers on the local channel, i.e., the channel on which the DMA transfer is being performed. If this precaution is neglected, device operation cannot be guaranteed.

Rewriting the DMA related registers by a DMA transfer is possible on only other channels. For example, the DMAn Source Address and DMAn Destination Address Registers on channel 1 can be rewritten by a DMA transfer through channel 0.

- **About the DMA Interrupt Request Status Register**

When writing to the DMA Interrupt Request Status Register to clear some bits, be sure to write 1 to all bits other than those to be cleared. Any bit of this register is unaffected by writing 1, the bit retains the value it had before writing.

- **About the stable operation of DMA transfers**

To ensure stable operation of DMA transfers, do not rewrite the DMA related registers unless transfers are disabled, except for the DMA Channel Control Register's transfer enable bit. It is only the DMA Source Address and Destination Address Registers that can be rewritten when transfers are enabled, provided, however, that the rewriting is performed by a DMA transfer through another channel.

*** This is a blank page ***

CHAPTER 10

INPUT/OUTPUT TIMERS

10.1 Outline of the Input/Output Timers

The 32172/32173 has four types of input/output timers, providing a total of 26 channels of timers.

Table 10.1.1 Outline of Timers

Name	Type	No. of Channels	Content
TMS (Timer Measure Small)	Input related 16-bit timers (up-counters)	4	16-bit input measurement timer. With new/old captured data hold function
TML (Timer Measure Large)	Input related 32-bit timers (up-counters)	4	32-bit input measurement timer. With new/old captured data hold function
TID (Timer Input Derivation)	Input related 16-bit timers (up/down-counters)	2	Four modes are available that can be selected in software. • Fixed period mode • Event count mode • Multiply-by-4 event count mode • Up/down event count mode
TOM (Timer Output Modification)	Output related 16-bit timers (down-counters)	16	Four output modes are available that can be selected in software. <Without correction function> • PWM output mode • Single-shot PWM output mode • Single-shot output mode • Successive output mode

Table 10.1.2 Interrupt Generating Functions of Timers

Signal Name	Timer Interrupt Request Source	Input to Interrupt Controller (ICU)	ICU Cause Input
IRQ30	PWMOFF0, PWMOFF1 inputs	PWM off input interrupt	2
IRQ28	TIN8, TIN9 inputs	Timer input interrupt 5	2
IRQ27	TIN10, TIN11 inputs	Timer input interrupt 4	2
IRQ25	TIN20, TIN21 inputs	Timer input interrupt 3	3
IRQ24	TIN22, TIN23 inputs	Timer input interrupt 2	3
IRQ23	TIN16, TIN17 inputs	Timer input interrupt 1	3
IRQ22	TIN18, TIN19 inputs	Timer input interrupt 0	3
IRQ21	TOM0_0 - TOM0_7 outputs	TOM0 output interrupt	8
IRQ20	TOM1_0 - TOM1_7 outputs	TOM1 output interrupt	8
IRQ19	TMS0 outputs	TMS0 output interrupt	1
IRQ18	TID0 outputs	TID0 output interrupt	1
IRQ17	TID1 outputs	TID1 output interrupt	1

Table 10.1.3 DMA Transfer Request Generating Functions of Timers

Signal Name	DMA Transfer Request Source	DMA Input Channel
DRQ0	TID0 underflow/overflow	Channel 0 to channel 9
DRQ1	TID1 underflow/overflow	Channel 0 to channel 9
DRQ2	TOM0_0 underflow	Channel 0
DRQ3	TOM0_1 underflow	Channel 1
DRQ4	TOM0_2 underflow	Channel 1
DRQ5	TOM0_3 underflow	Channel 3
DRQ6	TOM0_4 underflow	Channel 4
DRQ7	TOM0_5 underflow	Channel 6
DRQ8	TOM0_6 underflow	Channel 7
DRQ9	TOM0_7 underflow	Channel 2
DRQ10	TOM1_0 underflow	Channel 9
DRQ11	TOM1_6 underflow	Channel 5
DRQ12	TOM1_7 underflow	Channel 8
DRQ13	TIN16 input	Channel 0 to channel 9
DRQ14	TIN17 input	Channel 1
DRQ15	TIN18 input	Channel 2
DRQ16	TIN19 input	Channel 3
DRQ17	TIN20 input	Channel 6
DRQ18	TIN21 input	Channel 8
DRQ19	TIN22 input	Channel 9
DRQ20	TIN23 input	Channel 4

Table 10.1.4 A-D Conversion Start Request Functions of Timers

Signal Name	A-D Conversion Start Request Source	A-D Converter
AD0TRG	TIN16 input, TOM0_6 underflow, or TOM0 enable event	Can be input for A-D0 conversion start trigger
AD1TRG	TIN16 input, TOM0_6 underflow, or TOM1_6 underflow	Can be input for A-D1 conversion start trigger

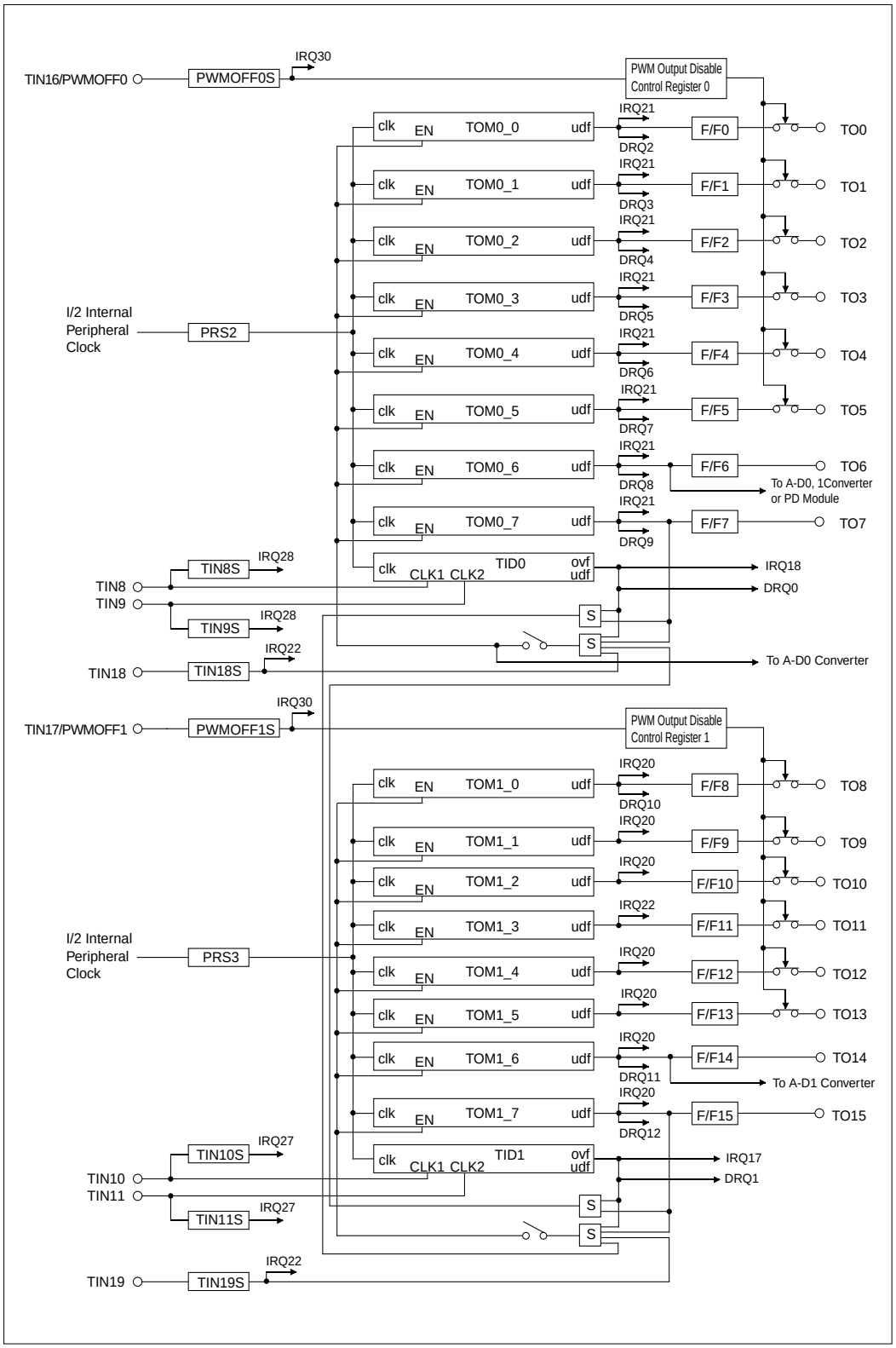


Figure 10.1.1 Block Diagram of Timers (1/3)

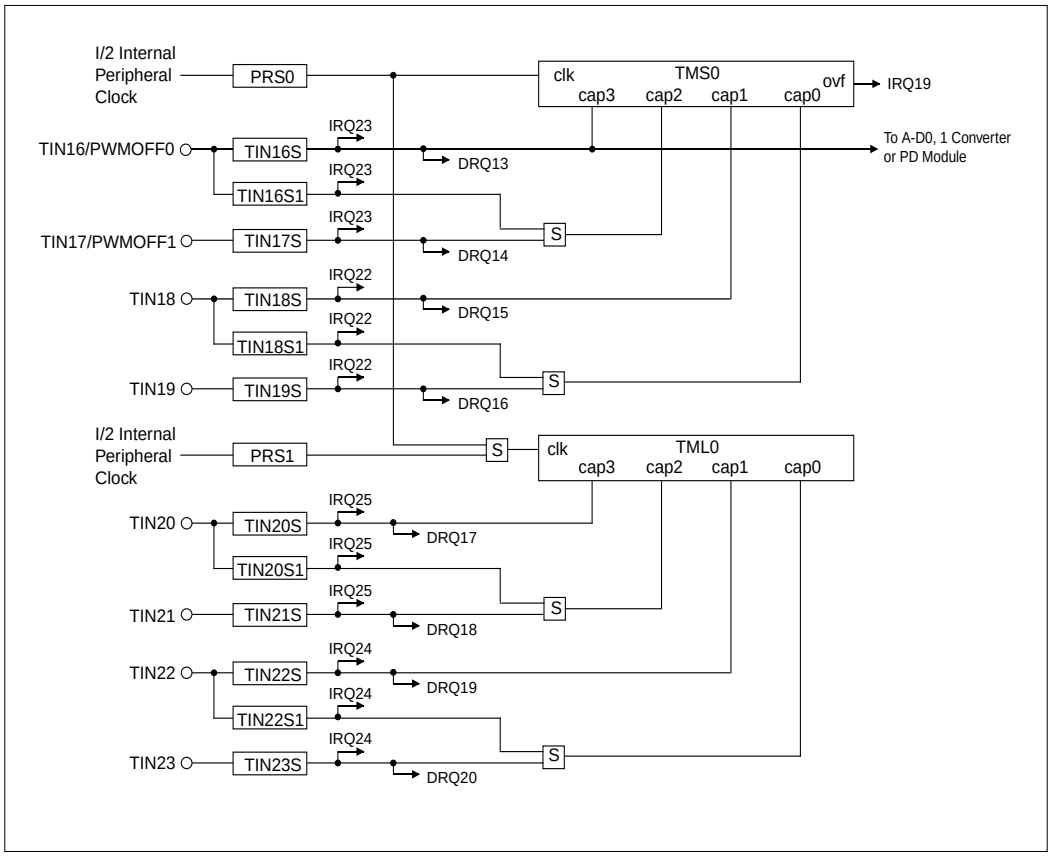


Figure 10.1.2 Block Diagram of Timers (2/3)

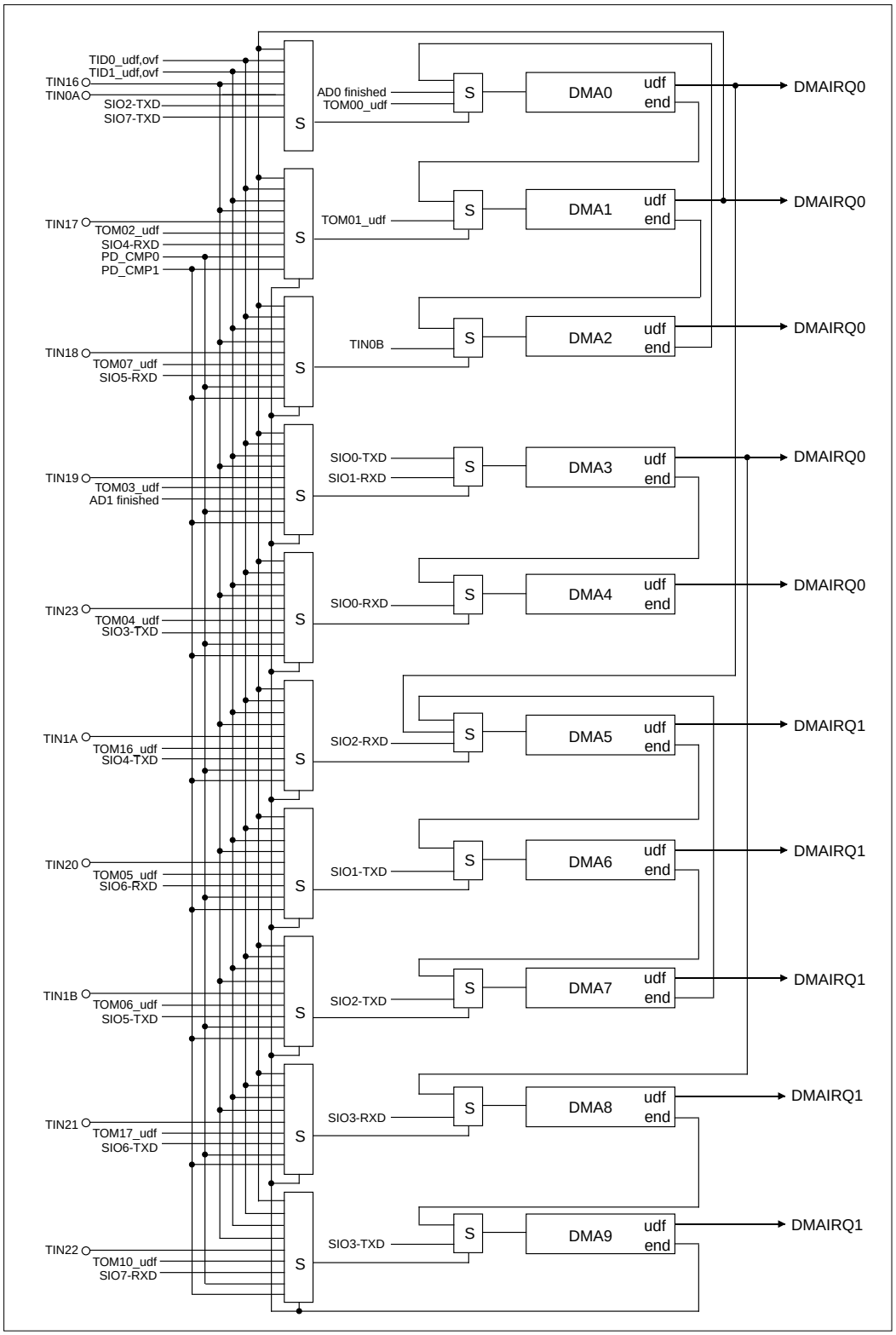


Figure 10.1.3 Block Diagram of Timers (3/3)

10.2 Common Timer Unit

The common timer unit includes the following blocks:

- Prescaler unit
- Input processing control unit
- Output flip-flop control unit
- Interrupt control unit

10.2.1 Register Map of the Common Timer Unit

The next page shows a register map of the common timer unit.

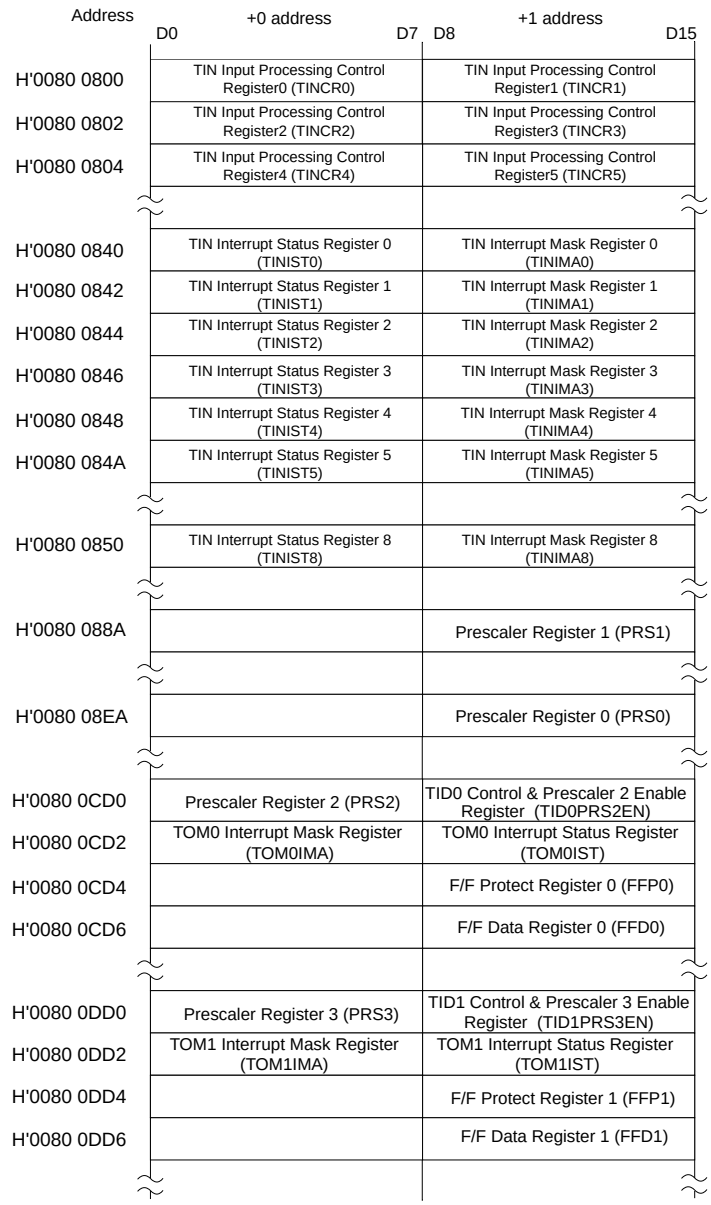


Figure 10.2.1 Register Map of the Common Timer Unit

10.2.2 Prescaler Unit

The prescalers PRS0-3 are 8-bit counter, which generates clocks supplied to each timer (TMS, TML, TID, and TOM) from the divided-by-2 frequency of the internal peripheral clock (10.0 MHz when the internal peripherals are operating at 20 MHz).

The value of each prescaler register is initialized to H'00 when reset. When the set value of the prescaler register is rewritten, the prescaler starts operating with the newly set value synchronously with the timing at which it underflows.

Values H'00 to H'FF can be set in the prescaler's counter register. The prescaler's divide ratio is given by the equation below.

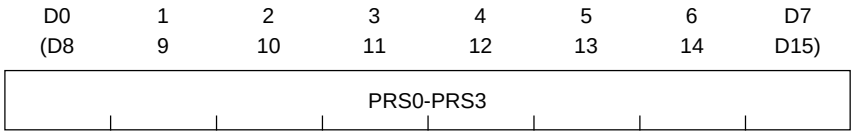
$$\text{Prescaler divide ratio} = \frac{1}{\text{Prescaler set value} + 1}$$

■ **Prescaler Register 0 (PRS0)** **<Address: H'0080 08EB>**

■ **Prescaler Register 1 (PRS1)** **<Address: H'0080 088B>**

■ **Prescaler Register 2 (PRS2)** **<Address: H'0080 0CD0>**

■ **Prescaler Register 3 (PRS3)** **<Address: H'0080 0DD0>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-7	PRS2,3	Sets the prescaler's divide-by value	○	○
8-15	PRS0,1			

Prescaler Registers 0 and 1 start counting after reset.
Prescaler Registers 2 and 3 respectively are activated by setting the TID0 Control & Prescaler 2 Enable Register and TID1 Control & Prescaler 3 Enable Register prescaler n enable (PRSnEN) bits to 1 (letting count start) and with the prescaler register value reloaded, start counting.
For details, see Section 10.5, "TID (Input Related 16-bit Timer)."

10.2.3 Input Processing Control Unit

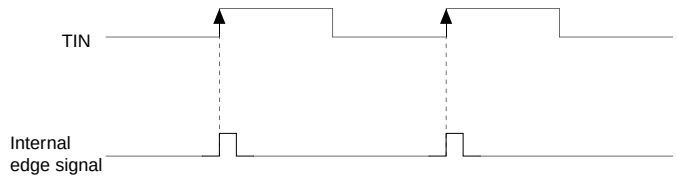
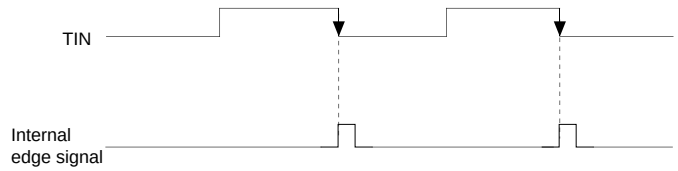
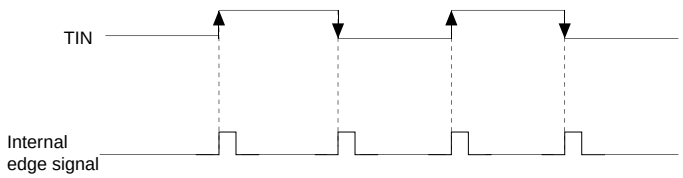
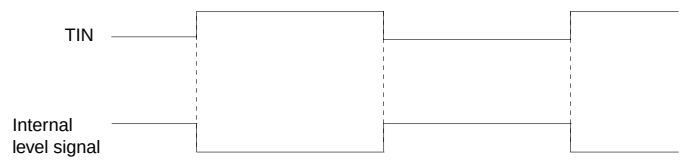
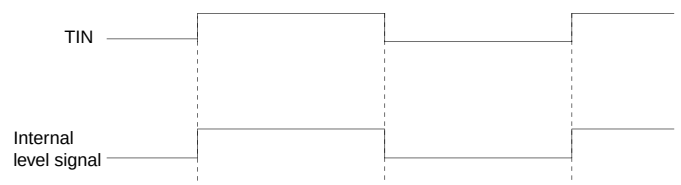
The input processing control unit performs TIN signal input processing.

In its TIN input processing unit, this unit selects the active signal edge (rising or falling edge or both edges) or the active level (high or low) at which to generate the signal to be fed to each timer for the enable, measurement, and count source signals.

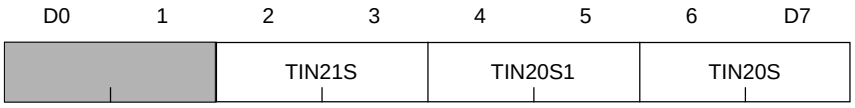
There are following input processing control registers.

- TIN Input Processing Control Register 0 (TINCR0)
- TIN Input Processing Control Register 1 (TINCR1)
- TIN Input Processing Control Register 2 (TINCR2)
- TIN Input Processing Control Register 3 (TINCR3)
- TIN Input Processing Control Register 4 (TINCR4)
- TIN Input Processing Control Register 5 (TINCR5)

Functions of TIN Input Processing Control Registers

Item	Function
Rising edge	
Falling edge	
Both edges	
Low level	
High level	

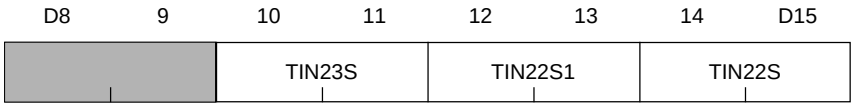
■ **TIN Input Processing Control Register 0 (TINCR0)** **<Address: H'0080 0800>**



<When reset: H'00>

D	Bit Name	Function	R	W
0,1	No functions assigned		0	–
2,3	TIN21S (Selects TIN21 input processing)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○
4,5	TIN20S1 (Selects TIN20 input processing for TML measurement 2)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○
6,7	TIN20S (Selects TIN20 input processing)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○

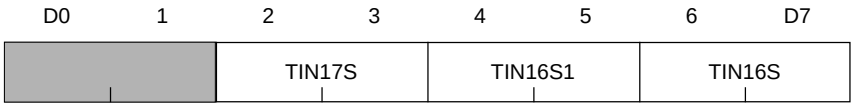
■ **TIN Input Processing Control Register 1 (TINCR1)** **<Address: H'0080 0801>**



<When reset: H'00>

D	Bit Name	Function	R	W
8,9	No functions assigned		0	–
10,11	TIN23S (Selects TIN23 input processing)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○
12,13	TIN22S1 (Selects TIN22 input processing for TML measurement 0)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○
14,15	TIN22S (Selects TIN22 input processing)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○

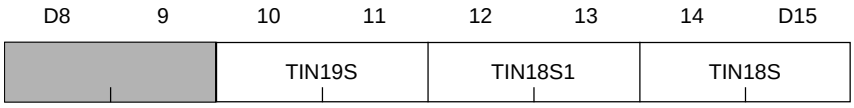
■ **TIN Input Processing Control Register 2 (TINCR2)** **<Address: H'0080 0802>**



<When reset: H'00>

D	Bit Name	Function	R	W
0,1	No functions assigned		0	–
2,3	TIN17S (Selects TIN17 input processing)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○
4,5	TIN16S1 (Selects TIN16 input processing for TML measurement 2)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○
6,7	TIN16S (Selects TIN16 input processing)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○

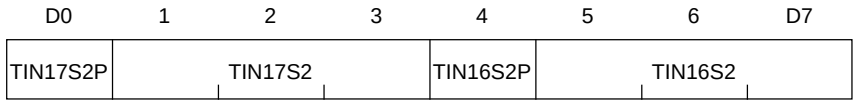
■ **TIN Input Processing Control Register 3 (TINCR3)** **<Address: H'0080 0803>**



<When reset: H'00>

D	Bit Name	Function	R	W
8,9	No functions assigned		0	–
10,11	TIN19S (Selects TIN19 input processing)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○
12,13	TIN18S1 (Selects TIN18 input processing for TML measurement 0)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○
14,15	TIN18S (Selects TIN18 input processing)	00: Ignores input 01: Rising edge 10: Falling edge 11: Both edges	○	○

■ **TIN Input Processing Control Register 4 (TINCR4)** **<Address: H'0080 0804>**



<When reset: H'00>

D	Bit Name	Function	R	W
0	TIN17S2P (Controls TIN17S2 write)	–	0	○
1-3	TIN17S2 (Selects TIN17 "PWMOFF1")	000: Ignores input 001: Rising edge 010: Falling edge 011: Both edges 10X: Low level 11X: High level	○	○
4	TIN16S2P (Controls TIN16S2 write)	–	0	○
5-7	TIN16S2 (Selects TIN16 "PWMOFF0")	000: Ignores input 001: Rising edge 010: Falling edge 011: Both edges 10X: Low level 11X: High level	○	○

For PWM output to be disabled by selecting output disable processing "PWMOFF0" by TIN16 or output disable processing "PWMOFF1" by TIN17, the related registers must be set following the procedure described below.

[Setup procedure]

- When selecting output disable processing "PWMOFF1" by TIN17
  ① Set the TIN17S2 write control (TIN17S2P) bit to 1.
  ② Subsequently after writing in (1) above, set the TIN17S2 write control (TIN17S2P) bit to 0 and the TIN17 "PWMOFF1" select (TIN17S2) bits to '000,' '001,' '010,' '011,' '10X,' or '11X.'

Note: If a write cycle for any other area occurs between (1) and (2), the values written to the TIN17 "PWMOFF1" select (TIN17S2) bits have no effect.

- When selecting output disable processing "PWMOFF0" by TIN16
  ① Set the TIN16S2 write control (TIN16S2P) bit to 1.
  ② Subsequently after writing in (1) above, set the TIN16S2 write control (TIN16S2P) bit to 0 and the TIN16 "PWMOFF0" select (TIN16S2) bits to '000,' '001,' '010,' '011,' '10X,' or '11X.'

Note: If a write cycle for any other area occurs between (1) and (2), the values written to the TIN16 "PWMOFF0" select (TIN16S2) bits have no effect.

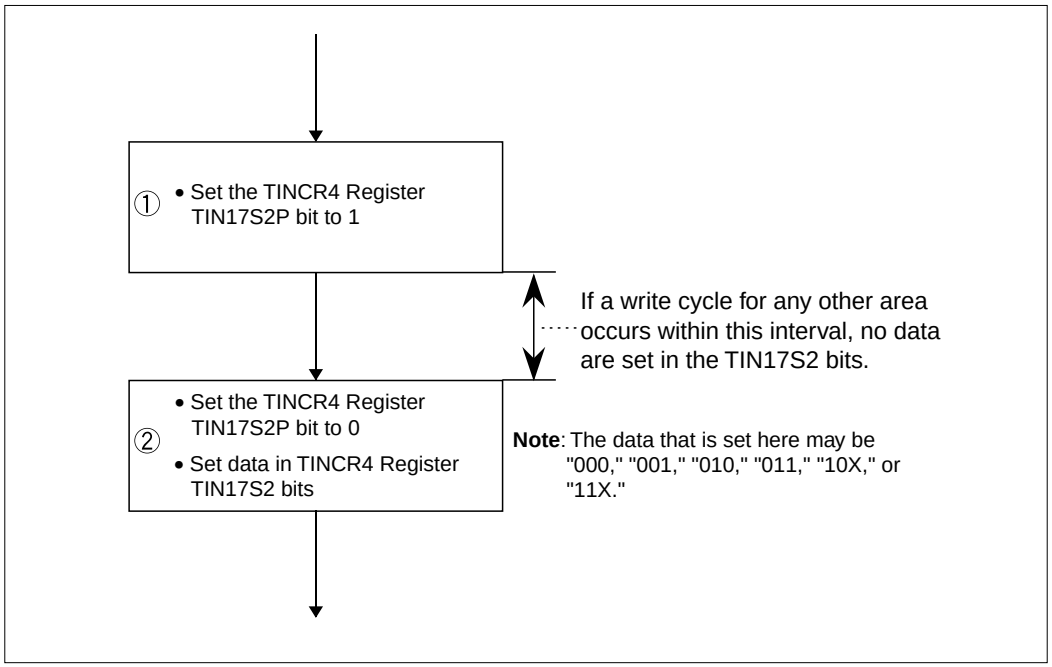


Figure 10.2.2 Procedure for Setting the TINCR4 Register when Output Disable Processing "PWMOFF1" by TIN17 is Selected

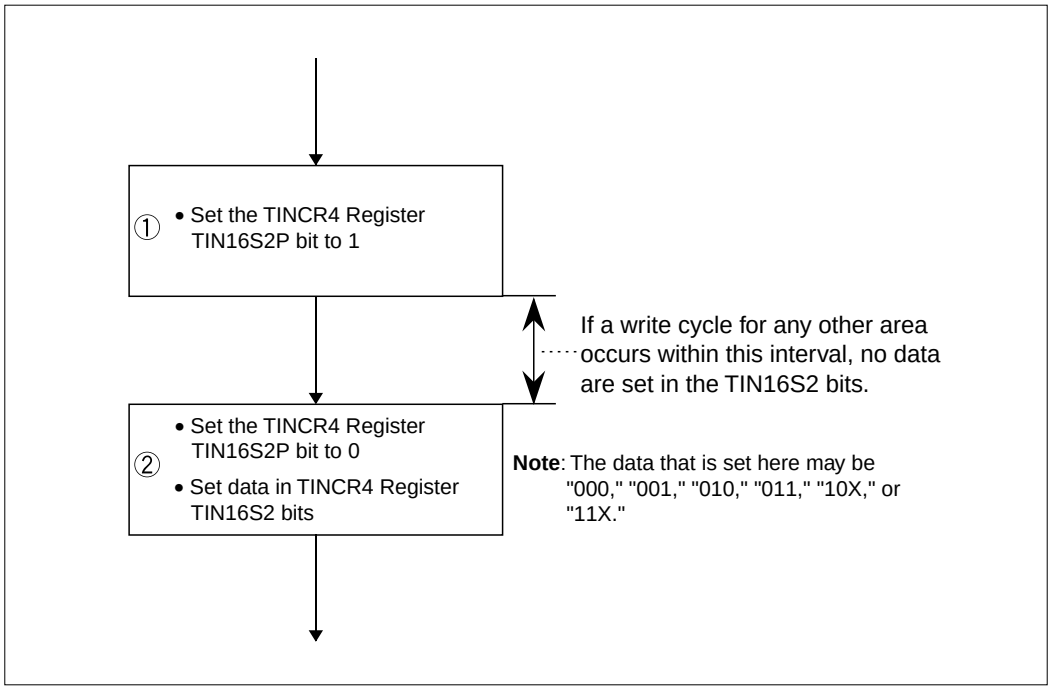
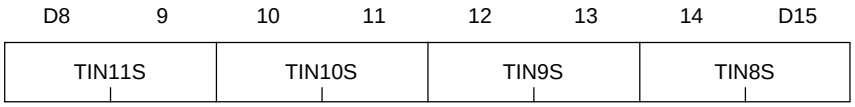


Figure 10.2.3 Procedure for Setting the TINCR4 Register when Output Disable Processing "PWMOFF0" by TIN16 is Selected

■ **TIN Input Processing Control Register 5 (TINCR5)** **<Address: H'0080 0805>**



<When reset: H'00>

D	Bit Name	Function	R	W
8,9	TIN11S	00: Ignores input	○	○
	(Selects TIN11 input processing)	01: Rising edge		
		10: Falling edge		
		11: Both edges		
10,11	TIN10S	00: Ignores input	○	○
	(Selects TIN10 input processing)	01: Rising edge		
		10: Falling edge		
		11: Both edges		
12,13	TIN9S	00: Ignores input	○	○
	(Selects TIN9 input processing)	01: Rising edge		
		10: Falling edge		
		11: Both edges		
14,15	TIN8S	00: Ignores input	○	○
	(Selects TIN8 input processing)	01: Rising edge		
		10: Falling edge		
		11: Both edges		

10.2.4 Output Flip-flop Control Unit

The output flip-flop control unit controls the flip-flop (F/F) provided for each timer output. There are following output flip-flop control registers.

- F/F Protect Register 0 (FFP0)
- F/F Protect Register 1 (FFP1)
- F/F Data Register 0 (FFD0)
- F/F Data Register 1 (FFD1)

Table 10.2.1 below lists the timing at which a signal for the output flip-flop is generated by each timer.

Table 10.2.1 Timing at Which Signal for Output Flip-flop is Generated by Each Timer

Timer	Mode	Timing at which signal for output flip-flop is generated
TMS	(16-bit measurement input)	No signal generating function
TML	(32-bit measurement input)	No signal generating function
TID	Fixed period count mode	No signal generating function
	Event count mode	No signal generating function
	Multiply-by-4 event count mode	No signal generating function
	Up/down event count mode	No signal generating function
TOM	PWM output mode	When counter is enabled or underflows
	Single-shot PWM output mode	When counter underflows
	Single-shot output mode	When counter is enabled or underflows
	Successive output mode	When counter is enabled or underflows

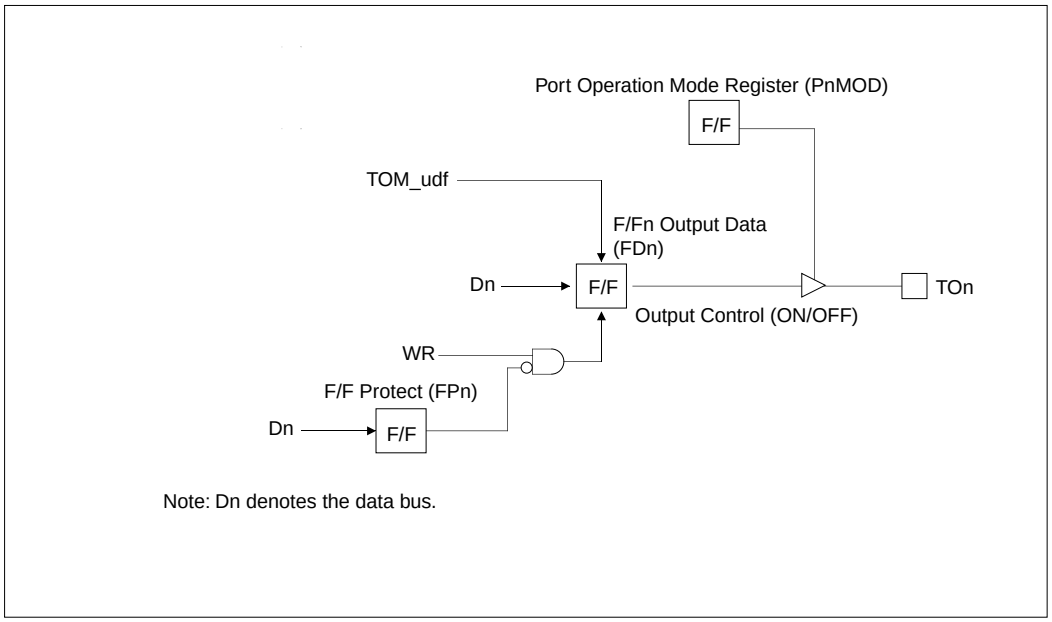


Figure 10.2.4 Configuration of the F/F Output Circuit

■ **F/F Protect Register 0 (FFP0)** <Address: H'0080 0CD5>

D8	9	10	11	12	13	14	D15
FP0	FP1	FP2	FP3	FP4	FP5	FP6	FP7

<When reset: H'00>

D	Bit Name	Function	R	W
8	FP0 (F/F0 protect)	0: Enables write to F/F output bit	○	○
9	FP1 (F/F1 protect)	1: Disables write to F/F output bit		
10	FP2 (F/F2 protect)			
11	FP3 (F/F3 protect)			
12	FP4 (F/F4 protect)			
13	FP5 (F/F5 protect)			
14	FP6 (F/F6 protect)			
15	FP7 (F/F7 protect)			

■ **F/F Protect Register 1 (FFP1)** **<Address: H'0080 0DD5>**

D8	9	10	11	12	13	14	D15
FP8	FP9	FP10	FP11	FP12	FP13	FP14	FP15

<When reset: H'00>

D	Bit Name	Function	R	W
8	FP8 (F/F8 protect)	0: Enables write to F/F output bit	○	○
9	FP9 (F/F9 protect)	1: Disables write to F/F output bit		
10	FP10 (F/F10 protect)			
11	FP11 (F/F11 protect)			
12	FP12 (F/F12 protect)			
13	FP13 (F/F13 protect)			
14	FP14 (F/F14 protect)			
15	FP15 (F/F15 protect)			

These registers enable/disable write to the respective output flip-flops (F/Fs). When disabled against write, writing to the F/F Data Register has no effect.

■ **F/F Data Register 0 (FFD0)** **<Address: H'0080 0CD7>**

D8	9	10	11	12	13	14	D15
FD0	FD1	FD2	FD3	FD4	FD5	FD6	FD7

<When reset: H'00>

D	Bit Name	Function	R	W
8	FD0 (F/F0 output data)	0: F/F output data = 0	○	○
9	FD1 (F/F1 output data)	1: F/F output data=1		
10	FD2 (F/F2 output data)			
11	FD3 (F/F3 output data)			
12	FD4 (F/F4 output data)			
13	FD5 (F/F5 output data)			
14	FD6 (F/F6 output data)			
15	FD7 (F/F7 output data)			

■ **F/F Data Register 1 (FFD1)** **<Address: H'0080 0DD7>**

D8	9	10	11	12	13	14	D15
FD8	FD9	FD10	FD11	FD12	FD13	FD14	FD15

<When reset: H'00>

D	Bit Name	Function	R	W
8	FD8 (F/F8 output data)	0: F/F output data = 0	○	○
9	FD9 (F/F9 output data)	1: F/F output data=1		
10	FD10 (F/F10 output data)			
11	FD11 (F/F11 output data)			
12	FD12 (F/F12 output data)			
13	FD13 (F/F13 output data)			
14	FD14 (F/F14 output data)			
15	FD15 (F/F15 output data)			

These registers set the output data in the respective output flip-flops (F/Fs). Although the F/F output normally varies with timer output, any F/F output (0 or 1) can be produced by setting the desired data in these registers. The F/F Data Registers can only be accessed for setting said data when the corresponding F/F Protect Registers described in the preceding page are enabled.

10.2.5 Interrupt Control Unit

The interrupt control unit controls the interrupt signals output to the Interrupt Controller by each timer. There are following 11 timer interrupt control registers for each timer.

- TIN Interrupt Control Register 0 (TINIR0)
- TIN Interrupt Control Register 1 (TINIR1)
- TIN Interrupt Control Register 2 (TINIR2)
- TIN Interrupt Control Register 3 (TINIR3)
- TIN Interrupt Control Register 4 (TINIR4)
- TIN Interrupt Control Register 5 (TINIR5)
- TIN Interrupt Control Register 8 (TINIR8)
- TOM0 Interrupt Mask Register (TOM0IMA)
- TOM0 Interrupt Status Register (TOM0IST)
- TOM1 Interrupt Mask Register (TOM1IMA)
- TOM1 Interrupt Status Register (TOM1IST)

For interrupts which have only one interrupt source for one interrupt vector table, no interrupt control registers are provided within the timer, and the interrupt status flags are automatically managed within the interrupt controller.

(For details, see Chapter 5, "Interrupt Controller.")

- TMS0 TMS0 Output Interrupt (IRQ19)
- TID0 TID0 Output Interrupt (IRQ18)
- TID1 TID1 Output Interrupt (IRQ17)

For interrupts which have two or more interrupt sources for one interrupt vector table, the interrupt control registers are used to control their interrupt requests and determine interrupt inputs. For this reason, the status flags in the Interrupt Controller only serve as the bits to determine whether interrupts are requested from the sources that have been enabled for interrupt and cannot be accessed for write.

(1) Interrupt request status bit

The status bit is used to determine whether an interrupt is requested. When an interrupt request occurs, this bit is set in hardware, and cannot be set in software. The status bit is cleared by writing 0. Writing 1 has no effect, the bit retains its status. Because the status bit is unaffected by the interrupt mask bit, it can also be used to check operation of the peripheral function. During interrupt processing, make sure that among the grouped interrupt flags, only the flag which has had its associated interrupt serviced is cleared.
If any flag for which the interrupt has not been serviced is cleared, unexecuted interrupt requests are also cleared.

(2) Interrupt mask bit

This flag is used to disable an unnecessary interrupt among the grouped interrupt requests. Setting this bit to 0 enables the interrupt; setting this bit to 1 disables the interrupt.

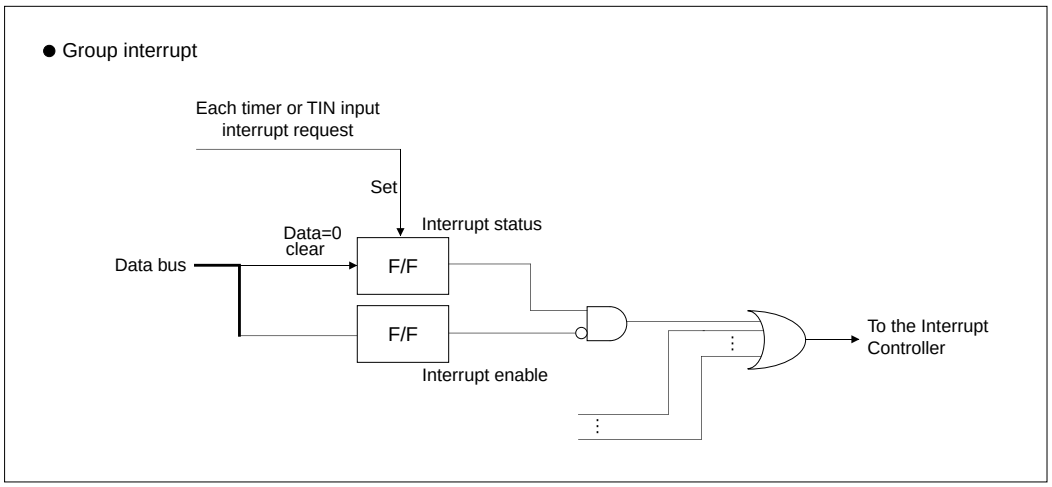


Figure 10.2.5 Interrupt Status Register and Mask Register

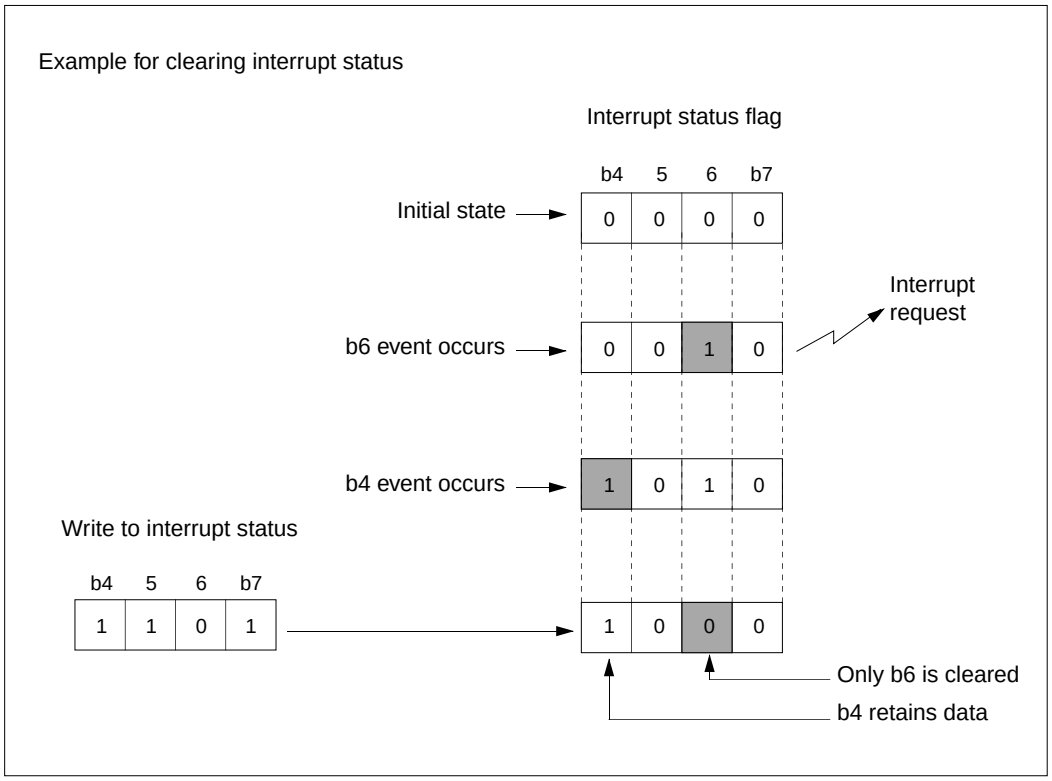


Figure 10.2.6 Example for Clearing Interrupt Status

The table below shows the relationship between the interrupt signals generated by the input/output timers and the inputs to the Interrupt Controller.

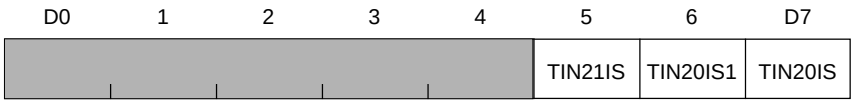
Table 10.2.2 Interrupt Signals Generated by Timers

Signal Name	Generating Source	Interrupt Input Source to ICU (Note1)	Number of Input Sources
IRQ20	TOM1_0 - TOM1_7	TOM1 output interrupt	8
IRQ21	TOM0_0 - TOM0_7	TOM0 output interrupt	8
IRQ22	TIN18, TIN19	Timer input interrupt 0	3
IRQ23	TIN16, TIN17	Timer input interrupt 1	3
IRQ24	TIN22, TIN23	Timer input interrupt 2	3
IRQ25	TIN20, TIN21	Timer input interrupt 3	3
IRQ27	TIN10, TIN11	Timer input interrupt 4	2
IRQ28	TIN8, TIN9	Timer input interrupt 5	2
IRQ30	PWMOFF0, PWMOFF1	PWM off input interrupt	2

Note 1: For details, see Chapter 5, "Interrupt Controller (ICU).")

Note 2: TMS0, TID0, and TID1 have only one interrupt source in the respective interrupt groups. Therefore, the timer interrupt control registers for these interrupts do not include the status and mask registers. (These interrupts are controlled directly by the Interrupt Controller.)

■ **TIN Interrupt Status Register 0 (TINIST0)** **<Address: H'0080 0840>**

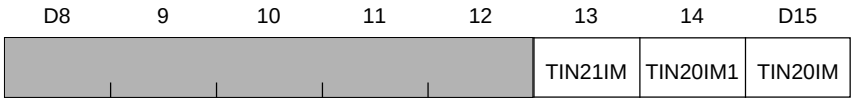


<When reset: H'00>

D	Bit Name	Function	R	W
0-4	No functions assigned		0	–
5	TIN21IS (TIN21 interrupt status)	0: Interrupt not requested	○	△
6	TIN20IS1 (TML measure 2 input detection interrupt status)	1: Interrupt requested		
7	TIN20IS (TIN20 interrupt status)			

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

■ **TIN Interrupt Mask Register 0 (TINIMA0)** **<Address: H'0080 0841>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-12	No functions assigned		0	–
13	TIN21IM (TIN21 interrupt mask)	0: Enables interrupt request	○	○
14	TIN20IM1 (TML measure 2 input detection interrupt mask)	1: Masks (disables) interrupt request		
15	TIN20IM (TIN20 interrupt mask)			

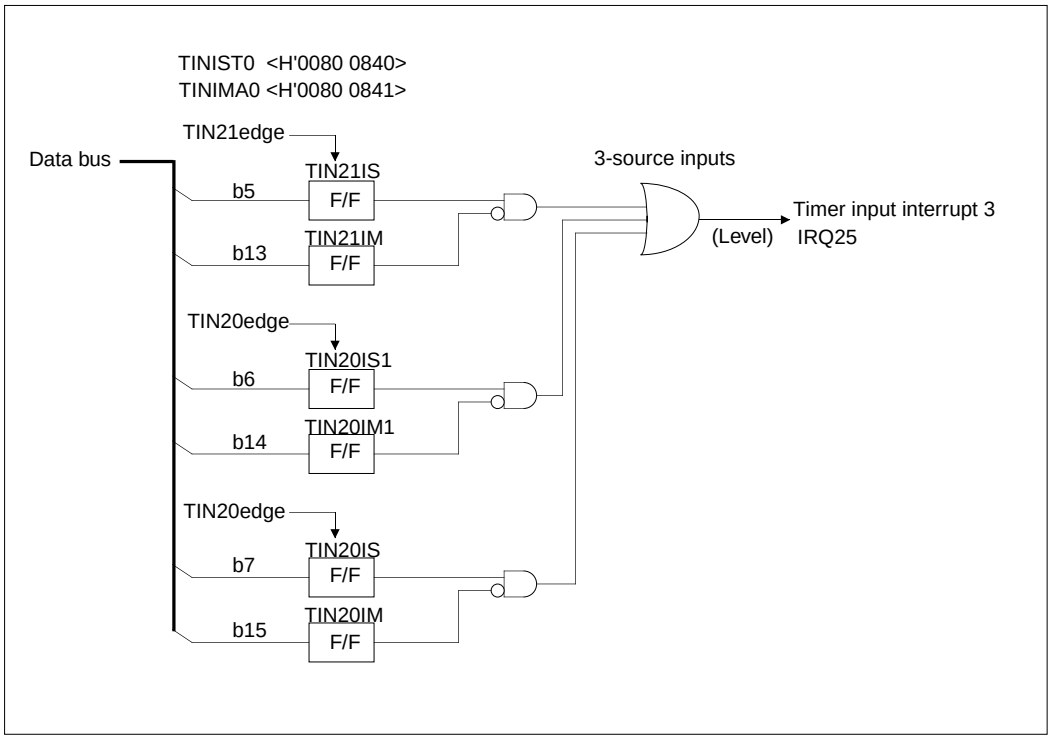
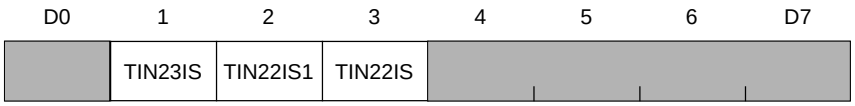


Figure 10.2.7 Block Diagram of Timer Input Interrupt 3

■ **TIN Interrupt Status Register 1 (TINIST1)** **<Address: H'0080 0842>**

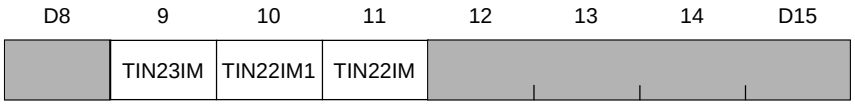


<When reset: H'00>

D	Bit Name	Function	R	W
0	No functions assigned		0	–
1	TIN23IS (TIN23 interrupt status)	0: Interrupt not requested	○	△
2	TIN22IS1 (TML measure 0 input detection interrupt status)	1: Interrupt requested		
3	TIN22IS (TIN22 interrupt status)			
4-7	No functions assigned		0	–

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

■ **TIN Interrupt Mask Register 1 (TINIMA1)** **<Address: H'0080 0843>**



<When reset: H'00>

D	Bit Name	Function	R	W
8	No functions assigned		0	–
9	TIN23IM (TIN23 interrupt mask)	0: Enables interrupt request	○	○
10	TIN22IM1 (TML measure 0 input detection interrupt mask)	1: Masks (disables) interrupt request		
11	TIN22IM (TIN22 interrupt mask)			
12-15	No functions assigned		0	–

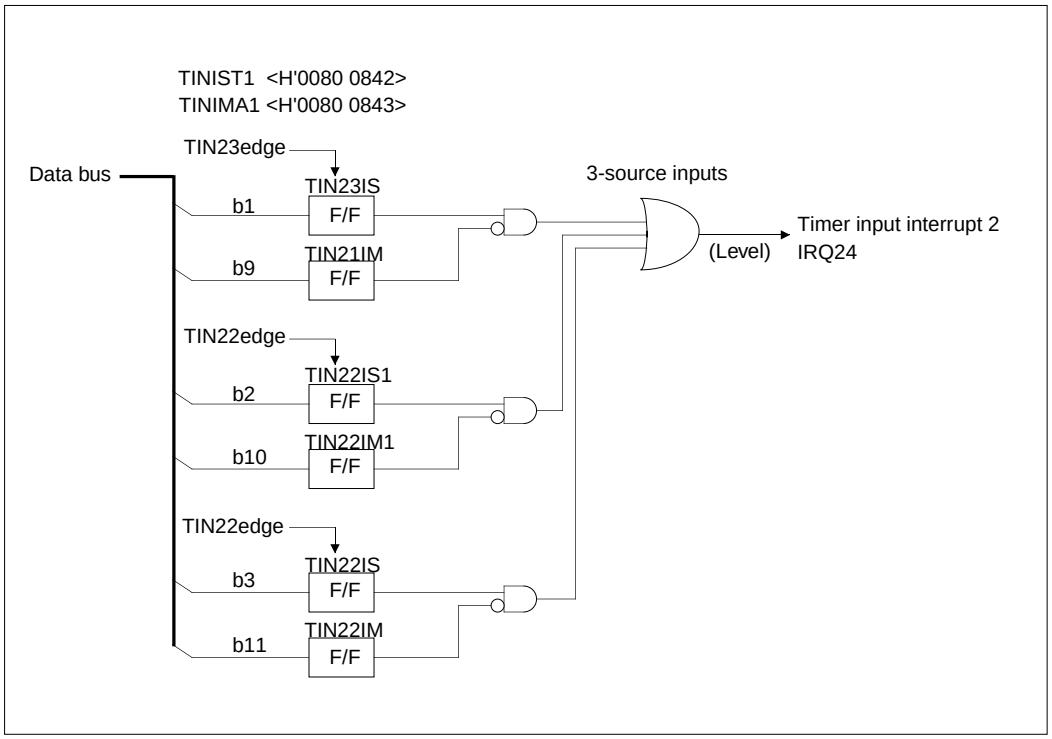
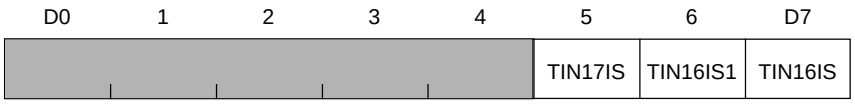


Figure 10.2.8 Block Diagram of Timer Input Interrupt 2

■ **TIN Interrupt Status Register 2 (TINIST2)** **<Address: H'0080 0844>**

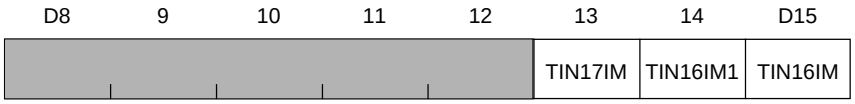


<When reset: H'00>

D	Bit Name	Function	R	W
0-4	No functions assigned		0	–
5	TIN17IS (TIN17 interrupt status)	0: Interrupt not requested	○	△
6	TIN16IS1 (TMS measure 2 input detection interrupt status)	1: Interrupt requested		
7	TIN16IS (TIN16 interrupt status)			

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

■ **TIN Interrupt Mask Register 2 (TINIMA2)** **<Address: H'0080 0845>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-12	No functions assigned		0	–
13	TIN17IM (TIN17 interrupt mask)	0: Enables interrupt request	○	○
14	TIN16IM1 (TMS measure 2 input detection interrupt mask)	1: Masks (disables) interrupt request		
15	TIN16IM (TIN16 interrupt mask)			

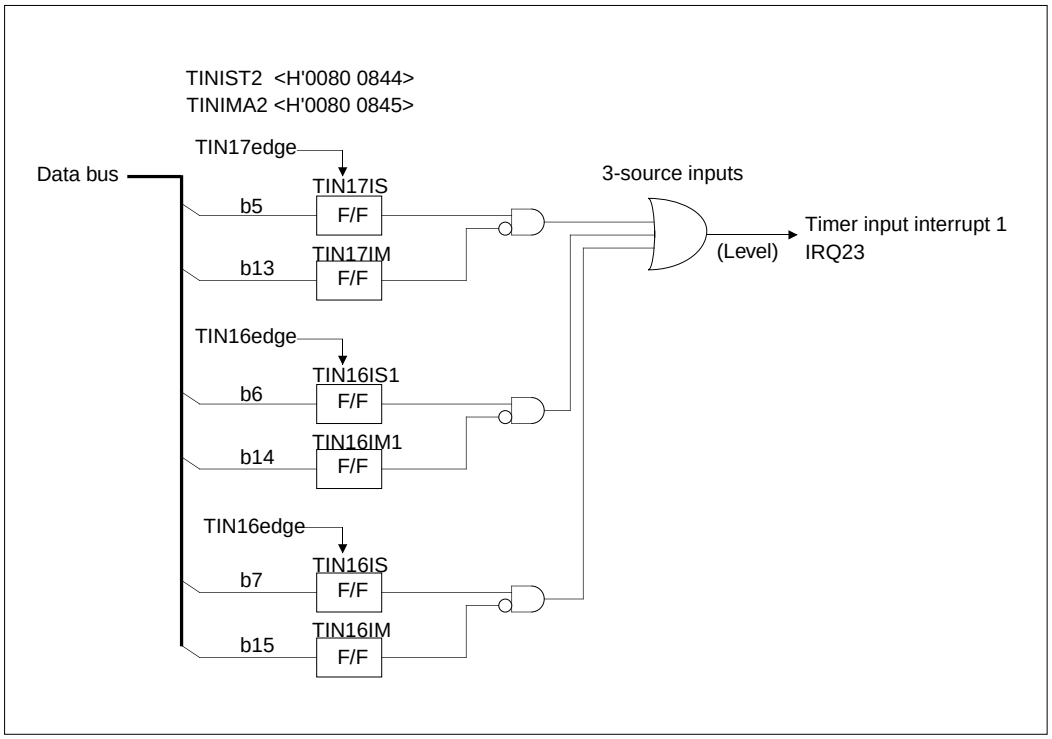
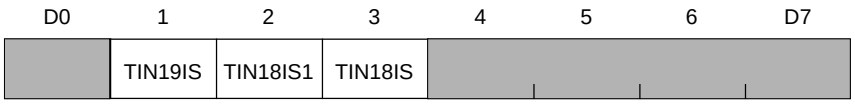


Figure 10.2.9 Block Diagram of Timer Input Interrupt 1

■ **TIN Interrupt Status Register 3 (TINIST3)** **<Address: H'0080 0846>**

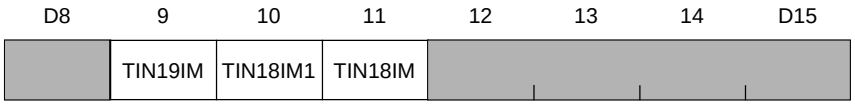


<When reset: H'00>

D	Bit Name	Function	R	W
0	No functions assigned		0	–
1	TIN19IS (TIN19 interrupt status)	0: Interrupt not requested	○	△
2	TIN18IS1 (TMS measure 0 input detection interrupt status)	1: Interrupt requested		
3	TIN18IS (TIN22 interrupt status)			
4-7	No functions assigned		0	–

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

■ **TIN Interrupt Mask Register 3 (TINIMA3)** **<Address: H'0080 0847>**



<When reset: H'00>

D	Bit Name	Function	R	W
8	No functions assigned		0	–
9	TIN19IM (TIN19 interrupt mask)	0: Enables interrupt request	○	○
10	TIN18IM1 (TMS measure 0 input detection interrupt mask)	1: Masks (disables) interrupt request		
11	TIN18IM (TIN18 interrupt mask)			
12-15	No functions assigned		0	–

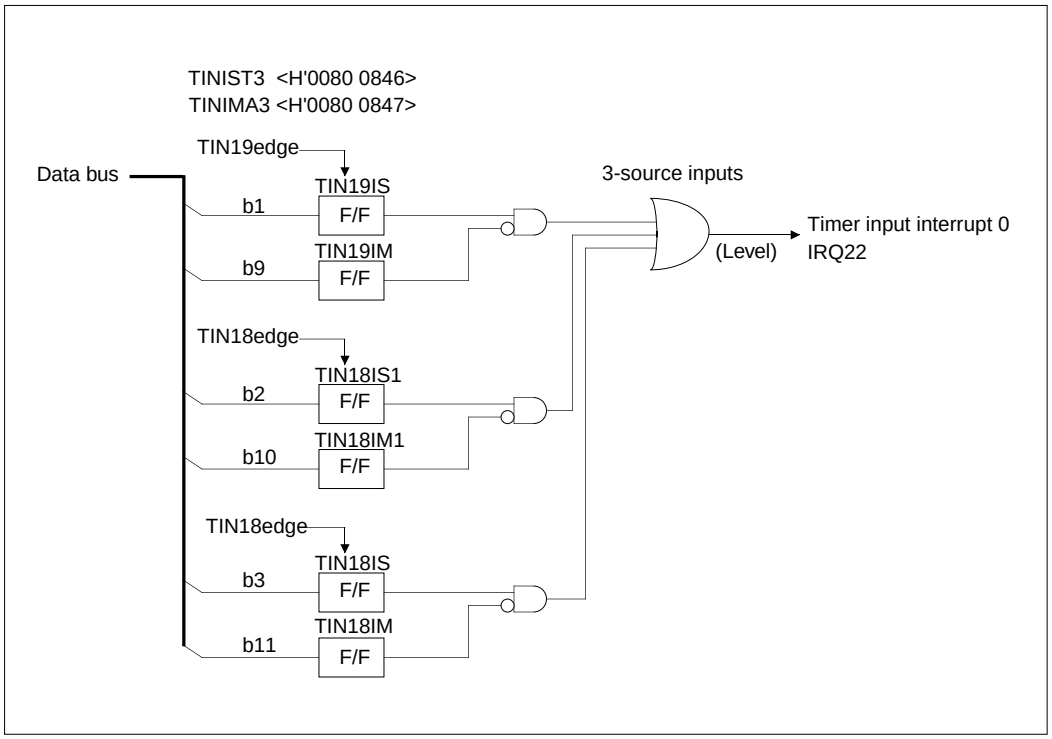
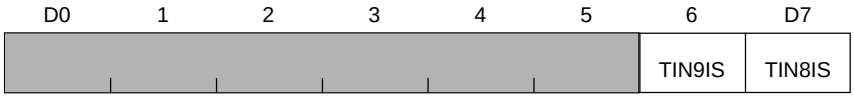


Figure 10.2.10 Block Diagram of Timer Input Interrupt 0

■ **TIN Interrupt Status Register 4 (TINIST4)** **<Address: H'0080 0848>**

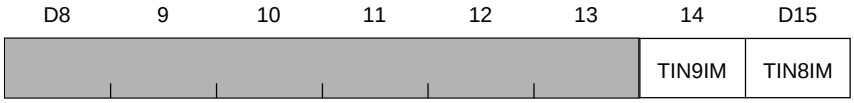


<When reset: H'00>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6	TIN9IS (TIN9 interrupt status)	0: Interrupt not requested	○	△
7	TIN8IS (TIN8 interrupt status)	1: Interrupt requested		

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

■ **TIN Interrupt Mask Register 4 (TINIMA4)** **<Address: H'0080 0849>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-13	No functions assigned		0	–
6	TIN9IM (TIN9 interrupt mask)	0: Enables interrupt request	○	○
7	TIN8IM (TIN8 interrupt mask)	1: Masks (disables) interrupt request		

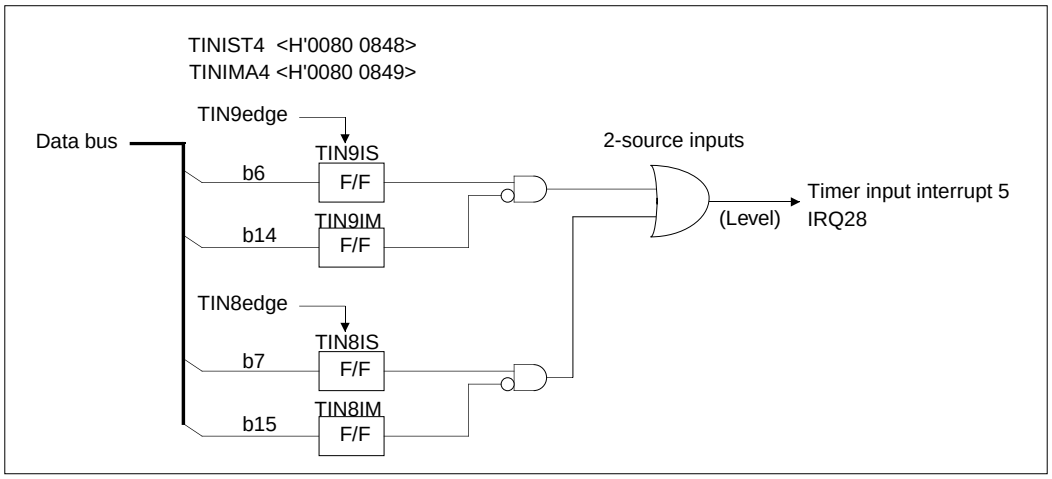


Figure 10.2.11 Block Diagram of Timer Input Interrupt 5

■ **TIN Interrupt Status Register 5 (TINIST5)** **<Address: H'0080 084A>**

D0	1	2	3	4	5	6	D7
		TIN11IS	TIN10IS				

<When reset: H'00>

D	Bit Name	Function	R	W
0, 1	No functions assigned		0	–
2	TIN11IS (TIN11 interrupt status)	0: Interrupt not requested	○	△
3	TIN10IS (TIN10 interrupt status)	1: Interrupt requested		
4-7	No functions assigned		0	–

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

■ **TIN Interrupt Mask Register 5 (TINIMA5)** **<Address: H'0080 084B>**

D8	9	10	11	12	13	14	D15
		TIN11IM	TIN10IM				

<When reset: H'00>

D	Bit Name	Function	R	W
8, 9	No functions assigned		0	–
10	TIN11IM (TIN11 interrupt mask)	0: Enables interrupt request	○	○
11	TIN10IM (TIN10 interrupt mask)	1: Masks (disables) interrupt request		
12-15	No functions assigned		0	–

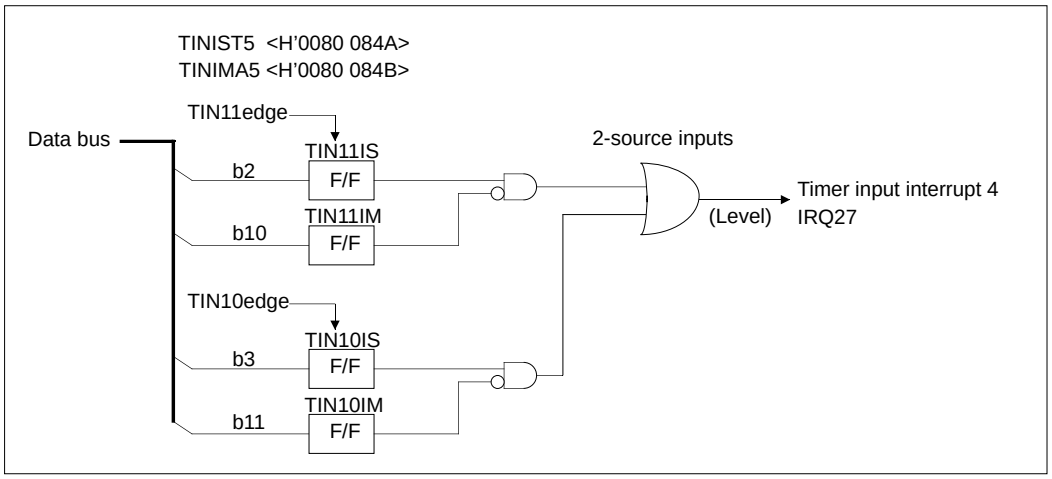
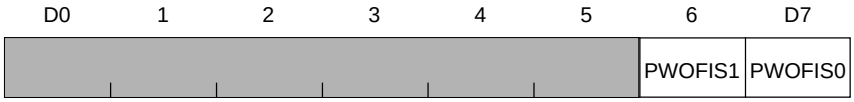


Figure 10.2.12 Block Diagram of Timer Input Interrupt 4

■ **TIN Interrupt Status Register 8 (TINIST8)** **<Address: H'0080 0850>**

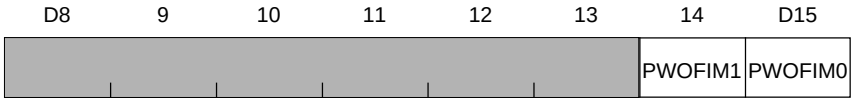


<When reset: H'00>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6	PWOFIS1 (PMW output disable interrupt status 1)	0: Interrupt not requested 1: Interrupt requested	○	△
7	PWOFIS0 (PMW output disable interrupt status 0)			

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

■ **TIN Interrupt Mask Register 8 (TINIMA8)** **<Address: H'0080 0851>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-13	No functions assigned		0	–
6	PWOFIM1 (PMW output disable interrupt mask 1)	0: Enables interrupt request 1: Masks (disables) interrupt request	○	○
7	PWOFIM0 (PMW output disable interrupt mask 0)			

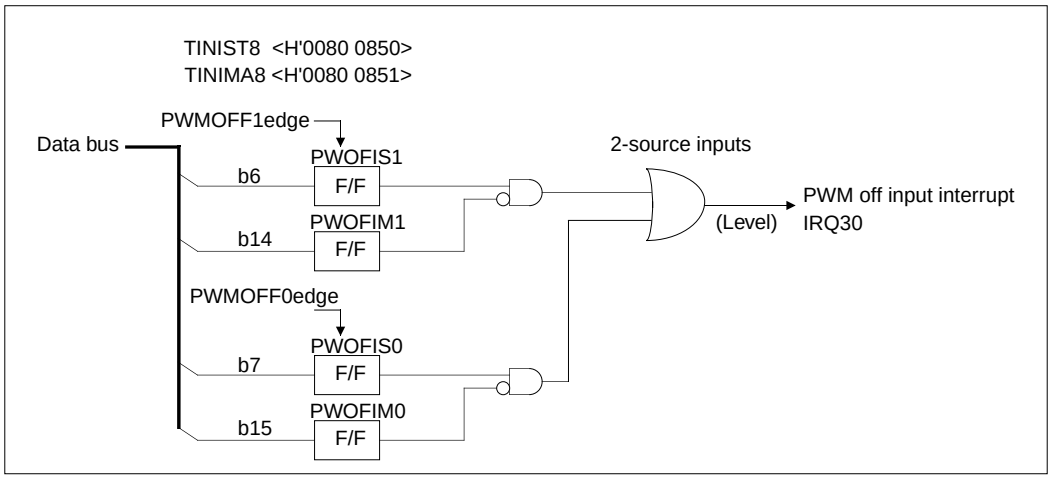


Figure 10.2.13 Block Diagram of PWM Off Input Interrupt

■ **TOM0 Interrupt Mask Register (TOM0IMA)** **<Address: H'0080 0CD2>**

D0	1	2	3	4	5	6	D7
TOM07IMA	TOM06IMA	TOM05IMA	TOM04IMA	TOM03IMA	TOM02IMA	TOM01IMA	TOM00IMA

<When reset: H'00>

D	Bit Name	Function	R	W
0	TOM07IMA (TOM0_7 interrupt mask)	0: Enables interrupt request	○	○
1	TOM06IMA (TOM0_6 interrupt mask)	1: Masks (disables) interrupt request		
2	TOM05IMA (TOM0_5 interrupt mask)			
3	TOM04IMA (TOM0_4 interrupt mask)			
4	TOM03IMA (TOM0_3 interrupt mask)			
5	TOM02IMA (TOM0_2 interrupt mask)			
6	TOM01IMA (TOM0_1 interrupt mask)			
7	TOM00IMA (TOM0_0 interrupt mask)			

■ **TOM0 Interrupt Status Register (TOM0IST)** **<Address: H'0080 0CD3>**

D8	9	10	11	12	13	14	D15
TOM07IST	TOM06IST	TOM05IST	TOM04IST	TOM03IST	TOM02IST	TOM01IST	TOM00IST

<When reset: H'00>

D	Bit Name	Function	R	W
8	TOM07IST (TOM0_7 interrupt status)	0: Interrupt not requested	○	△
9	TOM06IST (TOM0_6 interrupt status)	1: Interrupt requested		
10	TOM05IST (TOM0_5 interrupt status)			
11	TOM04IST (TOM0_4 interrupt status)			
12	TOM03IST (TOM0_3 interrupt status)			
13	TOM02IST (TOM0_2 interrupt status)			
14	TOM01IST (TOM0_1 interrupt status)			
15	TOM00IST (TOM0_0 interrupt status)			

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

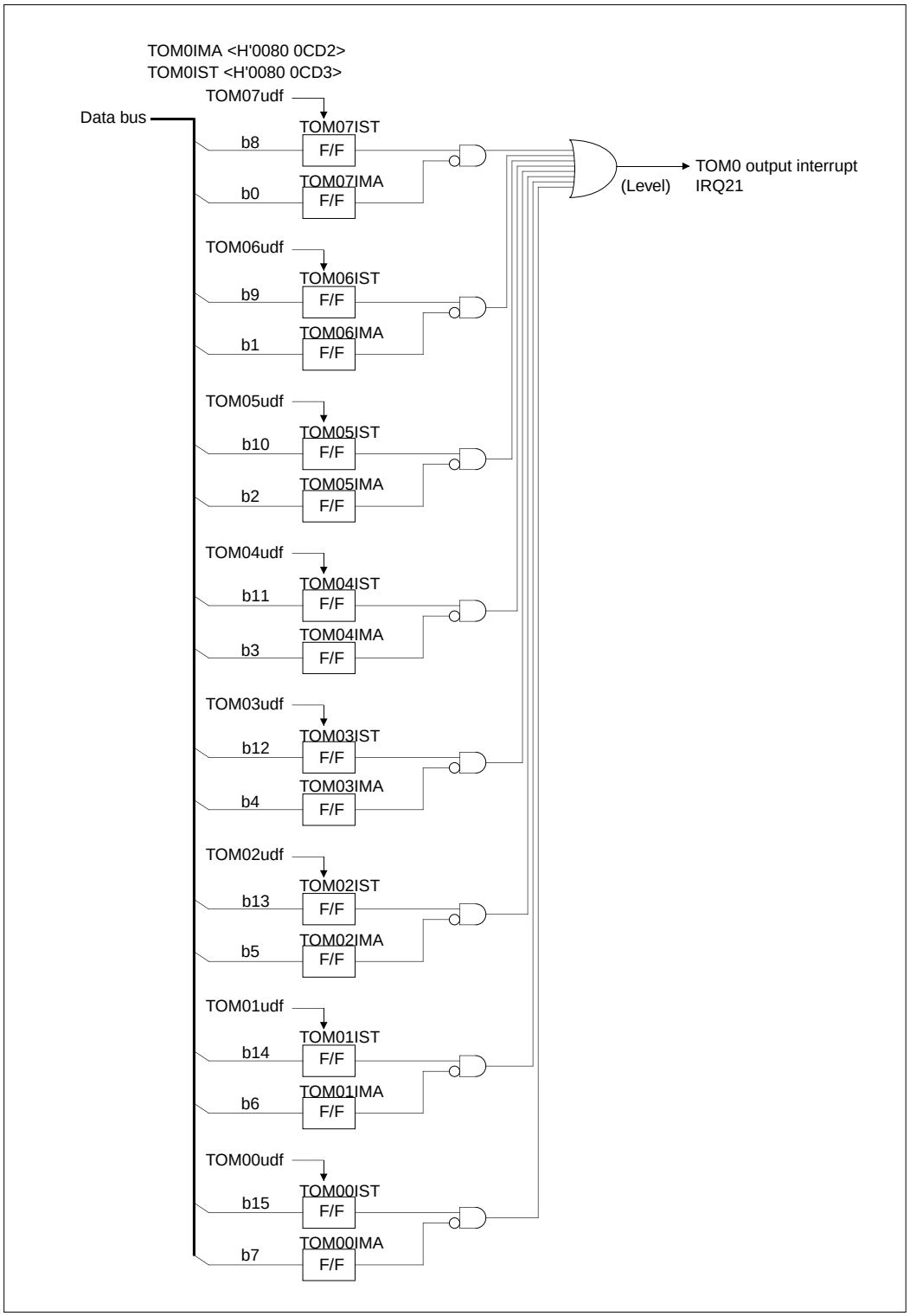


Figure 10.2.14 Block Diagram of TOM0 Output Interrupt

■ TOM1 Interrupt Mask Register (TOM1IMA) <Address: H'0080 0DD2>

D0	1	2	3	4	5	6	D7
TOM17IMA	TOM16IMA	TOM15IMA	TOM14IMA	TOM13IMA	TOM12IMA	TOM11IMA	TOM10IMA

<When reset: H'00>

D	Bit Name	Function	R	W
0	TOM17IMA (TOM1_7 interrupt mask)	0: Enables interrupt request	○	○
1	TOM16IMA (TOM1_6 interrupt mask)	1: Masks (disables) interrupt request		
2	TOM15IMA (TOM1_5 interrupt mask)			
3	TOM14IMA (TOM1_4 interrupt mask)			
4	TOM13IMA (TOM1_3 interrupt mask)			
5	TOM12IMA (TOM1_2 interrupt mask)			
6	TOM11IMA (TOM1_1 interrupt mask)			
7	TOM10IMA (TOM1_0 interrupt mask)			

■ TOM1 Interrupt Status Register (TOM1IST) <Address: H'0080 0DD3>

D8	9	10	11	12	13	14	D15
TOM17IST	TOM16IST	TOM15IST	TOM14IST	TOM13IST	TOM12IST	TOM11IST	TOM10IST

<When reset: H'00>

D	Bit Name	Function	R	W
8	TOM17IST (TOM1_7 interrupt status)	0: Interrupt not requested	○	△
9	TOM16IST (TOM1_6 interrupt status)	1: Interrupt requested		
10	TOM15IST (TOM1_5 interrupt status)			
11	TOM14IST (TOM1_4 interrupt status)			
12	TOM13IST (TOM1_3 interrupt status)			
13	TOM12IST (TOM1_2 interrupt status)			
14	TOM11IST (TOM1_1 interrupt status)			
15	TOM10IST (TOM1_0 interrupt status)			

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

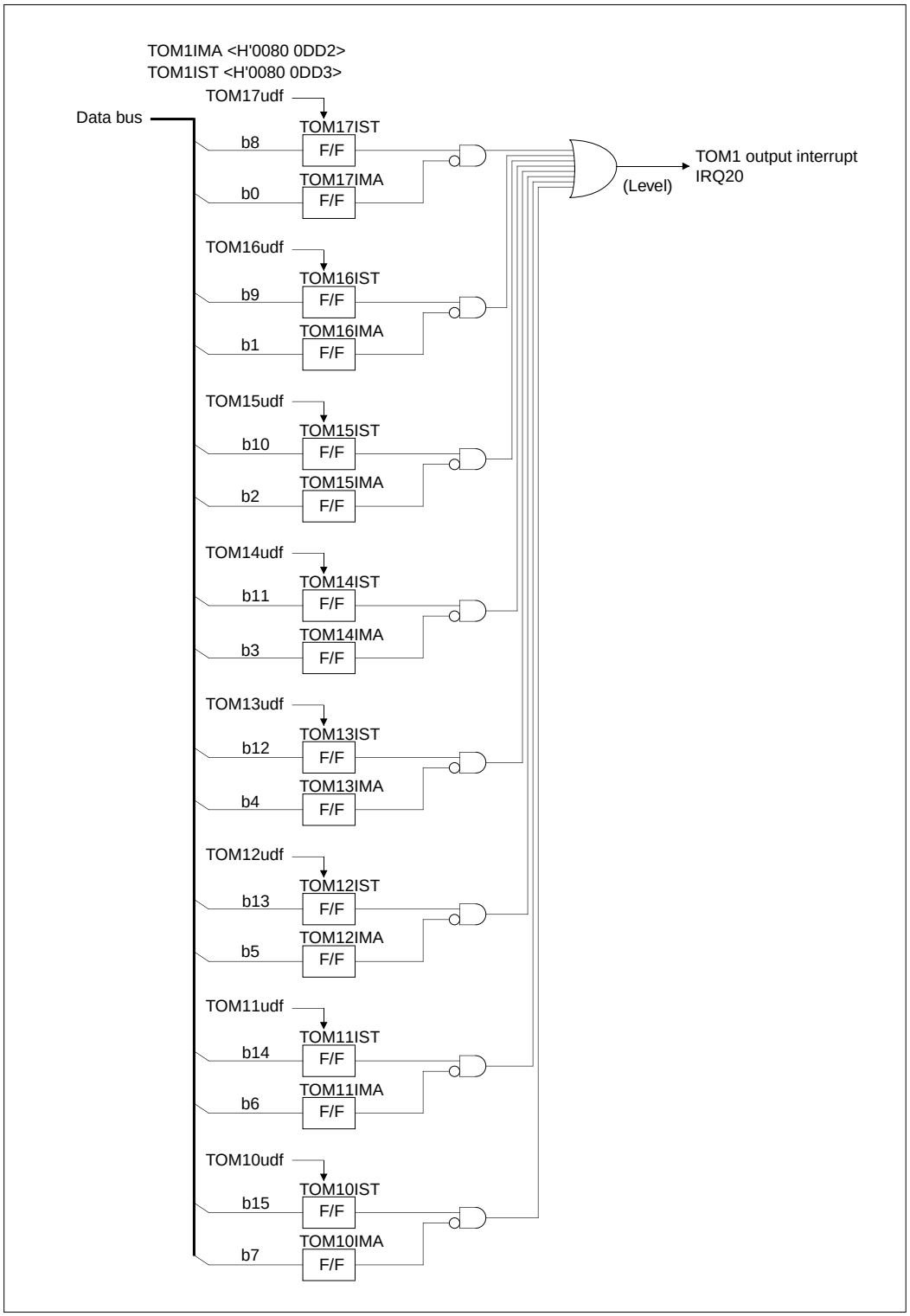


Figure 10.2.15 Block Diagram of TOM1 Output Interrupt

10.3 TMS (Input Related 16-bit Timers)

10.3.1 Outline of the TMS

TMS (Timer Measure Small) consists of input related 16-bit timers allowing input pulses on four channels to be measured. Specifications of the TMS are listed below. A block diagram of the TMS is shown in the next page.

Table 10.3.1 Specifications of TMS (Input Related 16-bit Timers)

Item	Specification
Number of channels	4 channels
Counter	16-bit up-counter (1 pc.)
Measure register	6-bit measure register (4 pcs.) and 16-bit old measure register (4 pcs.)
Timer start	Started by writing to the enable bit in software
Interrupt generation	Can be generated by counter overflow

10.3.2 Functional Outline of the TMS

In TMS, when the timer is activated (by writing to the enable bit in software), the counter starts operating. The counter is a 16-bit up-counter. When a measure signal is asserted by means of external input, the value of each measure register is latched into each old measure register, while the counter value is latched into each measure register.
Counting is stopped immediately by writing to the enable bit in software to disable counting.
A TIN and a TMS interrupt can be generated by input of an external measure signal and by occurrence a counter overflow, respectively.

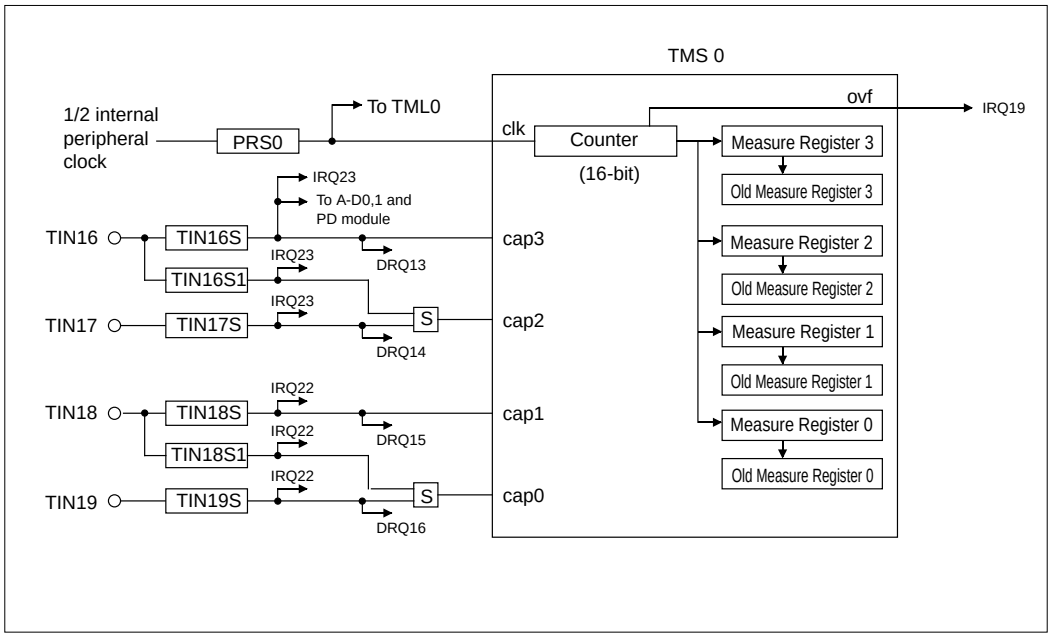


Figure 10.3.1 Block Diagram of TMS (Input Related 16-bit Timers)

10.3.3 TMS Related Register Map

A TMS related register map is shown below.

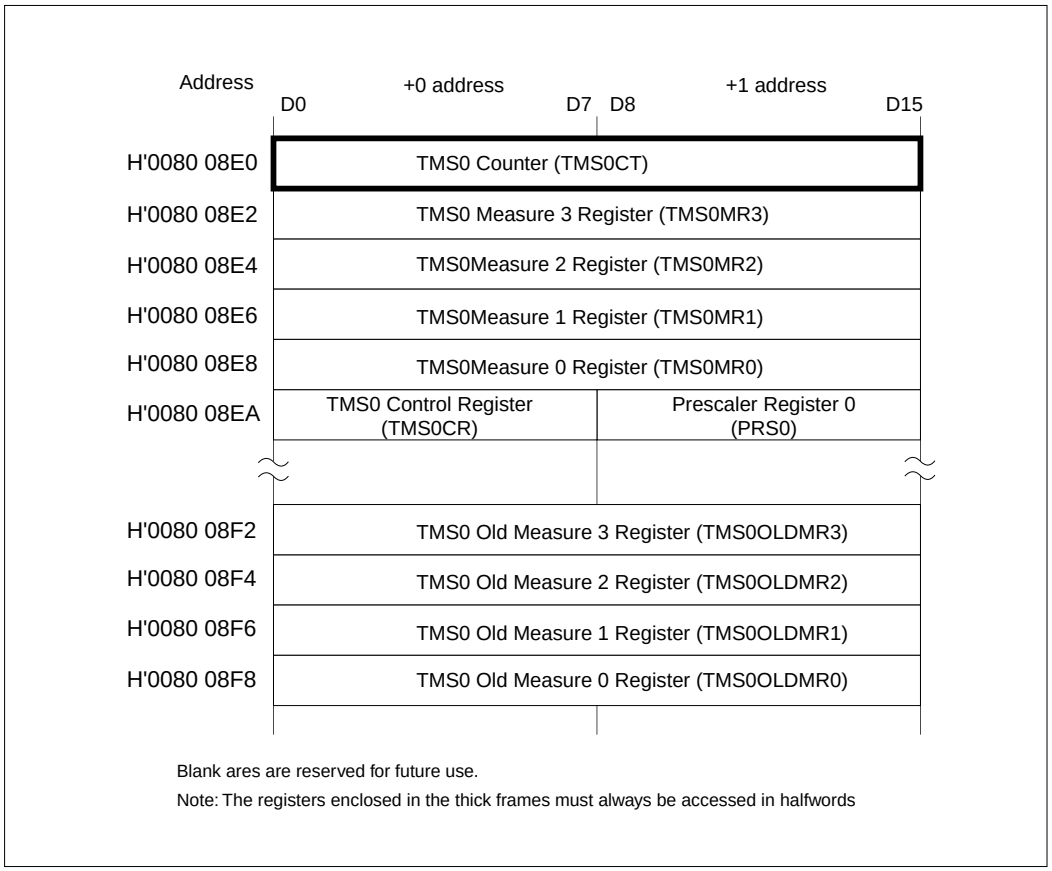


Figure 10.3.2 TMS Related Register Map

10.3.4 TMS Control Register

The TMS control register is used to select a TMS0 input event and controls TMS and TML counter startup.
Following TMS control registers are included.

- TMS0 Control Register (TMS0CR)

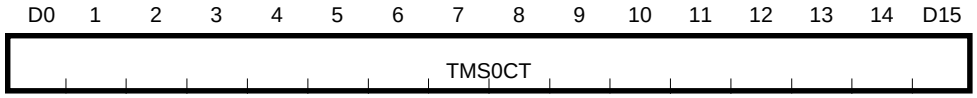
■ TMS0 Control Register (TMS0CR) **<Address: H'0080 08EA>**

<When reset: H'00>

D	Bit Name	Function	R	W
0	TMS0SS0 (Selects TMS0 measure 0 source)	0: External input TIN19 1: External input TIN18	○	○
1	No functions assigned		0	–
2	TMS0SS2 (Selects TMS0 measure 2 source)	0: External input TIN17 1: External input TIN16	○	○
3-6	No functions assigned		0	–
7	TMSLCEN (TMS/TML count enable)	0: Stops counting 1: Starts counting	○	○

10.3.5 TMS Counter (TMS0CT)

■ **TMS0 Counter (TMS0CT)** **<Address: H'0080 08E0>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TMS0CT	16-bit counter value	○	○

Note: This register must always be accessed in halfwords.

The TMS counter is a 16-bit up-counter, which starts counting when the timer is activated (by writing to the enable bit in software).
The counter can be read out on-the-fly.

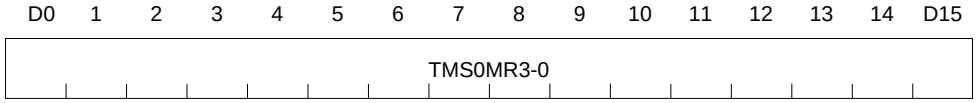
10.3.6 TMS Measure Registers (TMS0MR3-0)

■ **TMS0 Measure 3 Register (TMS0MR3)** **<Address: H'0080 08E2>**

■ **TMS0 Measure 2 Register (TMS0MR2)** **<Address: H'0080 08E4>**

■ **TMS0 Measure 1 Register (TMS0MR1)** **<Address: H'0080 08E6>**

■ **TMS0 Measure 0 Register (TMS0MR0)** **<Address: H'0080 08E8>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TMS0MR3-TMS0MR0	16-bit counter value	○	–

Note 1: This register is a read-only register.

Note 2: This register can be accessed in either bytes or halfwords.

The TMS measure registers are used to capture the content of the counter upon event input. The TMS measure registers are a read-only register.

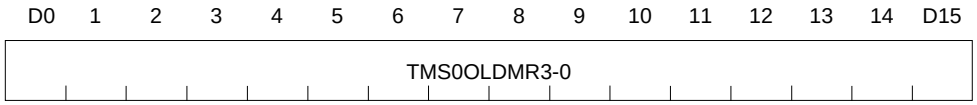
10.3.7 TMS Old Measure Registers (TMS0OLDMR3-0)

■ **TMS0 Old Measure 3 Register (TMS0OLDMR3)** **<Address: H'0080 08F2>**

■ **TMS0 Old Measure 2 Register (TMS0OLDMR2)** **<Address: H'0080 08F4>**

■ **TMS0 Old Measure 1 Register (TMS0OLDMR1)** **<Address: H'0080 08F6>**

■ **TMS0 Old Measure 0 Register (TMS0OLDMR0)** **<Address: H'0080 08F8>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TMS0OLDMR3-TMS0OLDMR0	TMS0 measure register value	○	–

Note 1: This register is a read-only register.

Note 2: This register can be accessed in either bytes or halfwords.

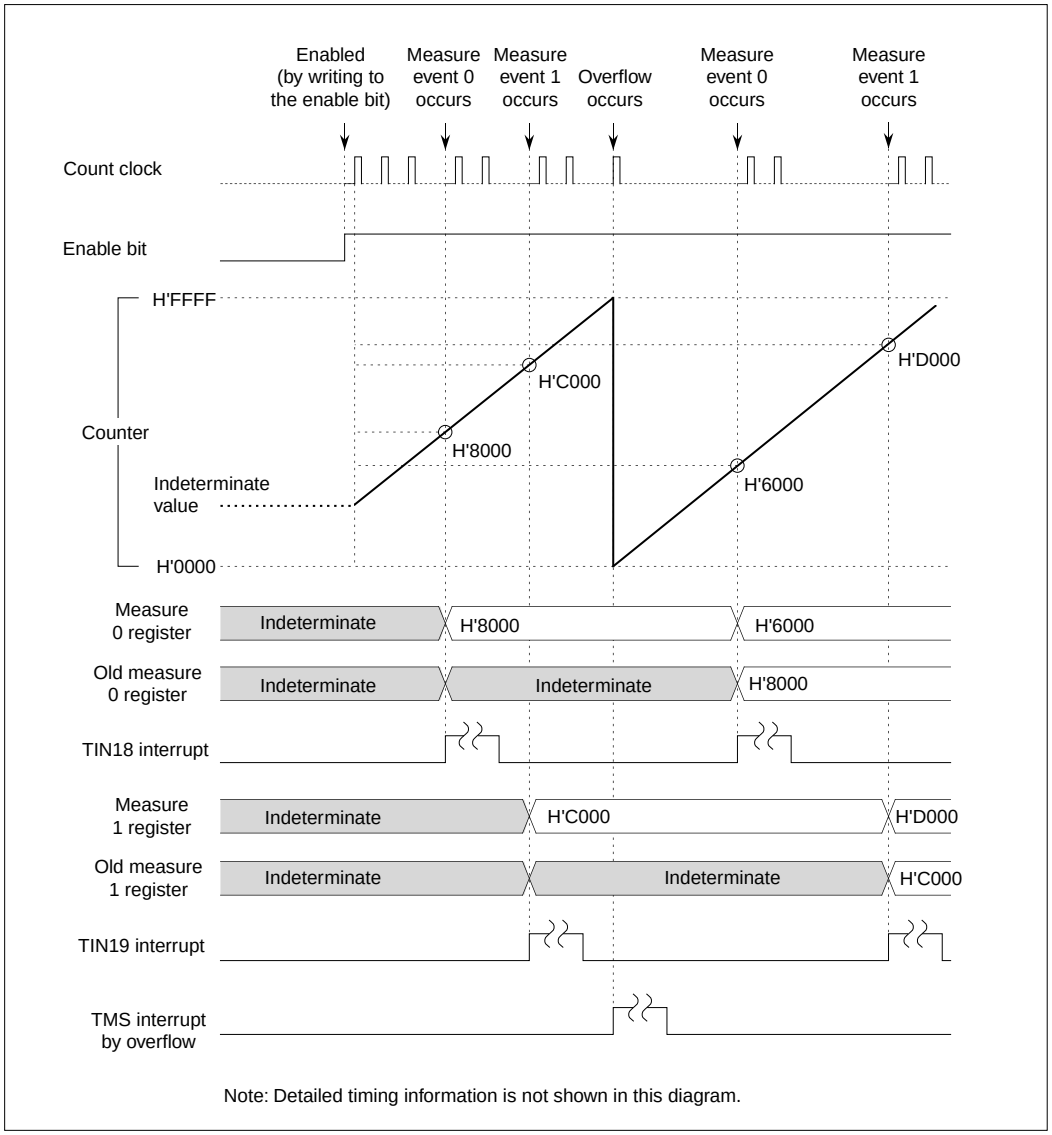
The TMS old measure registers are used to capture the contents of the respective TMS measure registers upon event input. The TMS old measure registers are a read-only register.

10.3.8 Operation of TMS Measure Input

(1) Outline of TMS measure input

For TMS measure input, the counter first starts counting up when the timer is activated (by writing to the enable bit in software). Each time there is event input to TMS while the timer is operating, the values of the measure registers 0-3 are latched into the corresponding old measure registers 0-3 and the counter value is latched into the measure registers 0-3.
The timer is stopped immediately by writing to the enable bit to disable counting.

A TIN interrupt can be generated when a measure signal is entered from an external source. A TMS interrupt can be generated when the counter overflows.



Note: Detailed timing information is not shown in this diagram.

Figure 10.3.3 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 simultaneously in the same clock cycle, the write value is set in the counter and also latched into the measure register.

10.4 TML (Input Related 32-bit Timers)

10.4.1 Outline of the TML

TML (Timer Measure Large) consists of input related 32-bit timers allowing input pulses on four channels to be measured. Specifications of the TML are listed below. A block diagram of the TML is shown in the next page.

Table 10.4.1 Specifications of TML (Input Related 32-bit Timers)

Item	Specification
Number of channels	4 channels
Counter	32-bit up-counter
Measure register	32-bit measure register (4 pcs.)
Timer start	Started by writing to the enable bit in software

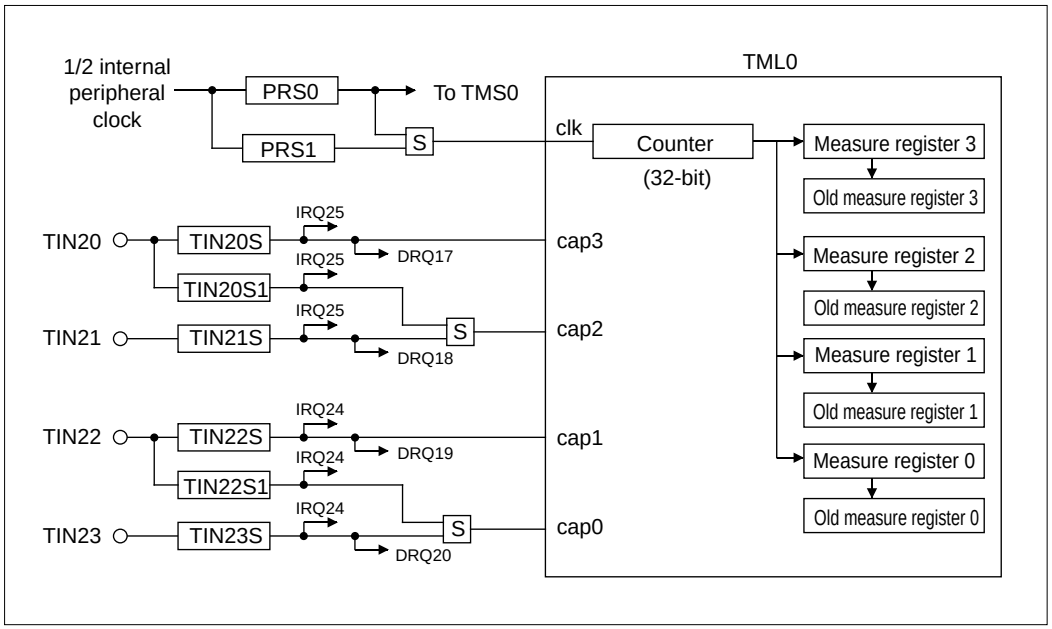


Figure 10.4.1 Block Diagram of TML (Input Related 32-bit Timers)

10.4.2 Functional Outline of the TML

The TML is designed in such a way that when the timer is activated (by writing to the enable bit in software), the counter starts operating. The counter is a 32-bit up-counter. When a measure event signal is asserted by means of external input, the content of each measure register (32-bit) is latched into each old measure register (32-bit), while the counter value is latched into each measure register (32-bit).

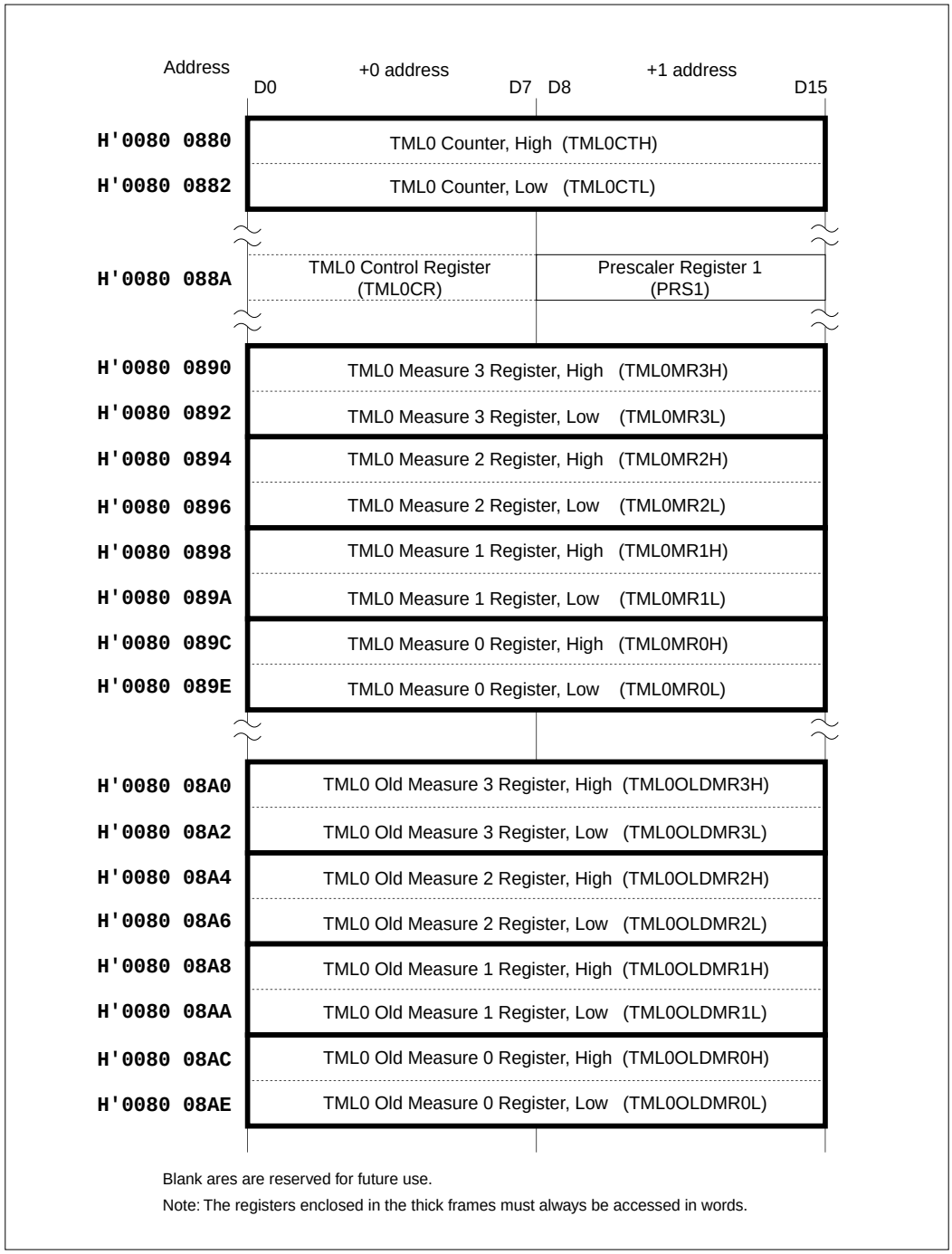
Counting is stopped immediately by writing to the enable bit in software to disable counting.

A TIN interrupt can be generated by input of an external measure signal. The TML does not have a counter overflow interrupt.

Note: The enable bit of TML timer is shared with enable bit of TMS timer.
For details, refer to section 10.3.4 "TMS Control Register".

10.4.3 TML Related Register Map

A TML related register map is shown below.



Blank ares are reserved for future use.

Note: The registers enclosed in the thick frames must always be accessed in words.

Figure 10.4.2 TML Related Register Map

10.4.4 TML Control Register

■ **TML0 Control Register (TML0CR)** **<Address: H'0080 088A>**

<When reset: H'00>

D	Bit Name	Function	R	W
8	TML0SS0	0: External input TIN23	○	○
	(Selects TML0 measure 0 source)	1: External input TIN22		
9	No functions assigned		0	–
10	TML0SS2	0: External input TIN21	○	○
	(Selects TML0 measure 2 source)	1: External input TIN20		
11-14	No functions assigned		0	–
15	TML0CKS	0: Prescaler 1 (PRS1)	○	○
	(Selects TML0 clock source)	1: Prescaler 0 (PRS0)		

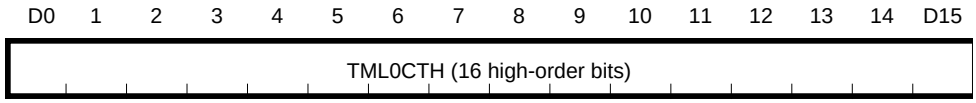
The TML0 Control Register is used to select TML0 input event and the count clock.

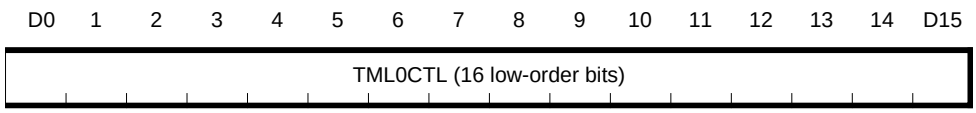
Note: The counter can be written to normally only when the selected clock source is a 1/2 internal peripheral clock. When using any other clock source, the counter cannot be written correctly. Under this condition, do not write to the counter.

10.4.5 TML Counters

■ **TML0 Counter, High (TML0CTH)** **<Address: H'0080 0880>**

■ **TML0 Counter, Low (TML0CTL)** **<Address: H'0080 0882>**





<When reset: indeterminate>

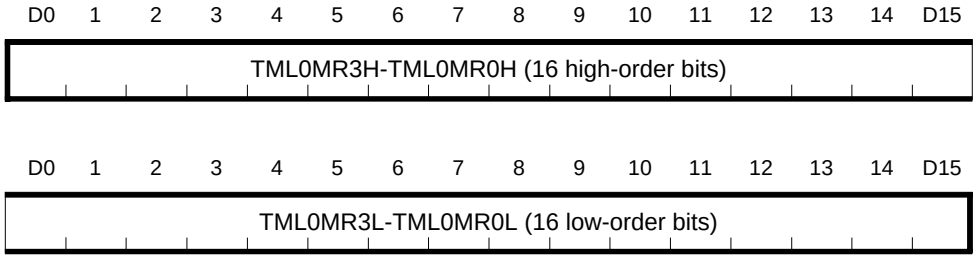
D	Bit Name	Function	R	W
0-15	TML0CTH	32-bit counter value (16 high-order bits)	○	○
	TML0CTL	32-bit counter value (16 low-order bits)		

Note: This register must always be accessed words (32 bits) beginning with the TML0CTH address.

The TML0 Counter is a 32-bit up-counter, which starts counting when the timer is activated (by writing to the enable bit in software). The TML0CTH and TML0CTL registers accommodate the 16 high-order bits and the 16 low-order bits of the 32-bit counter, respectively.
The counter can be read out on-the-fly.

10.4.6 TML Measure Registers

■ **TML0 Measure 3 Register (TML0MR3H)** **<Address: H'0080 0890>**
■ **TML0 Measure 3 Register (TML0MR3L)** **<Address: H'0080 0892>**
■ **TML0 Measure 2 Register (TML0MR2H)** **<Address: H'0080 0894>**
■ **TML0 Measure 2 Register (TML0MR2L)** **<Address: H'0080 0896>**
■ **TML0 Measure 1 Register (TML0MR1H)** **<Address: H'0080 0898>**
■ **TML0 Measure 1 Register (TML0MR1L)** **<Address: H'0080 089A>**
■ **TML0 Measure 0 Register (TML0MR0H)** **<Address: H'0080 089C>**
■ **TML0 Measure 0 Register (TML0MR0L)** **<Address: H'0080 089E>**



<When reset: indeterminate>

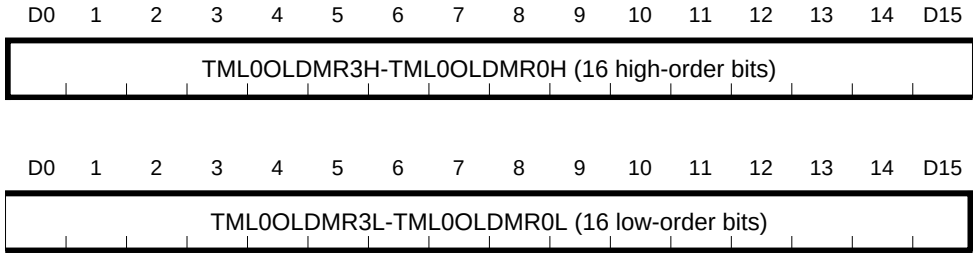
D	Bit Name	Function	R	W
0-15	TML0MR3H-0H	32-bit measure register value (16 high-order bits)	○	–
	TML0MR3L-0L	32-bit measure register value (16 low-order bits)		

Note 1: These registers are a read-only register.
Note 2: These registers must always be accessed words (32 bits) beginning with each word boundary.

The TML0 Measure Registers are used to capture the content of the counter upon event input. The TML0 Measure Registers each are configured with 32 bits, with the TML0MR3H-0H and the TML0MR3L-0L accommodating the 16 high-order bits and the 16 low-order bits, respectively. The TML0 Measure Registers are a read-only register. These registers must always be accessed in words beginning with each word boundary.

10.4.7 TML Old Measure Registers

■ **TML0 Old Measure 3 Register (TML0OLDMR3H)** **<Address: H'0080 08A0>**
■ **TML0 Old Measure 3 Register (TML0OLDMR3L)** **<Address: H'0080 08A2>**
■ **TML0 Old Measure 2 Register (TML0OLDMR2H)** **<Address: H'0080 08A4>**
■ **TML0 Old Measure 2 Register (TML0OLDMR2L)** **<Address: H'0080 08A6>**
■ **TML0 Old Measure 1 Register (TML0OLDMR1H)** **<Address: H'0080 08A8>**
■ **TML0 Old Measure 1 Register (TML0OLDMR1L)** **<Address: H'0080 08AA>**
■ **TML0 Old Measure 0 Register (TML0OLDMR0H)** **<Address: H'0080 08AC>**
■ **TML0 Old Measure 0 Register (TML0OLDMR0L)** **<Address: H'0080 08AE>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TML0OLDMR3H-0H	32-bit old measure register value (16 high-order bits)	○	–
	TML0OLDMR3L-0L	32-bit old measure register value (16 low-order bits)		

Note 1: These registers are a read-only register.
Note 2: These registers must always be accessed words (32 bits) beginning with each word boundary.

The TML0 Old Measure Registers are used to capture the contents of the corresponding TML0 Measure Registers upon event input. The TML0 Old Measure Registers each are configured with 32 bits, with the TML0OLDMR3H-0H and the TML0OLDMR3L-0L accommodating the 16 high-order bits and the 16 low-order bits, respectively. The TML0 Old Measure Registers are a read-only register. These registers must always be accessed in words beginning with each word boundary.

10.4.8 Operation of TML Measure Input

(1) Outline of TML measure input

For TML measure input, the counter starts counting up when the timer is activated (by writing to the enable bit in software). Each time there is event input to the measure registers 0-3, the counter value is latched into the measure registers.
The timer is stopped immediately by writing to the enable bit to disable counting.

A TIN interrupt can be generated by input of an external measure signal. (The TML does not have a counter overflow interrupt.)

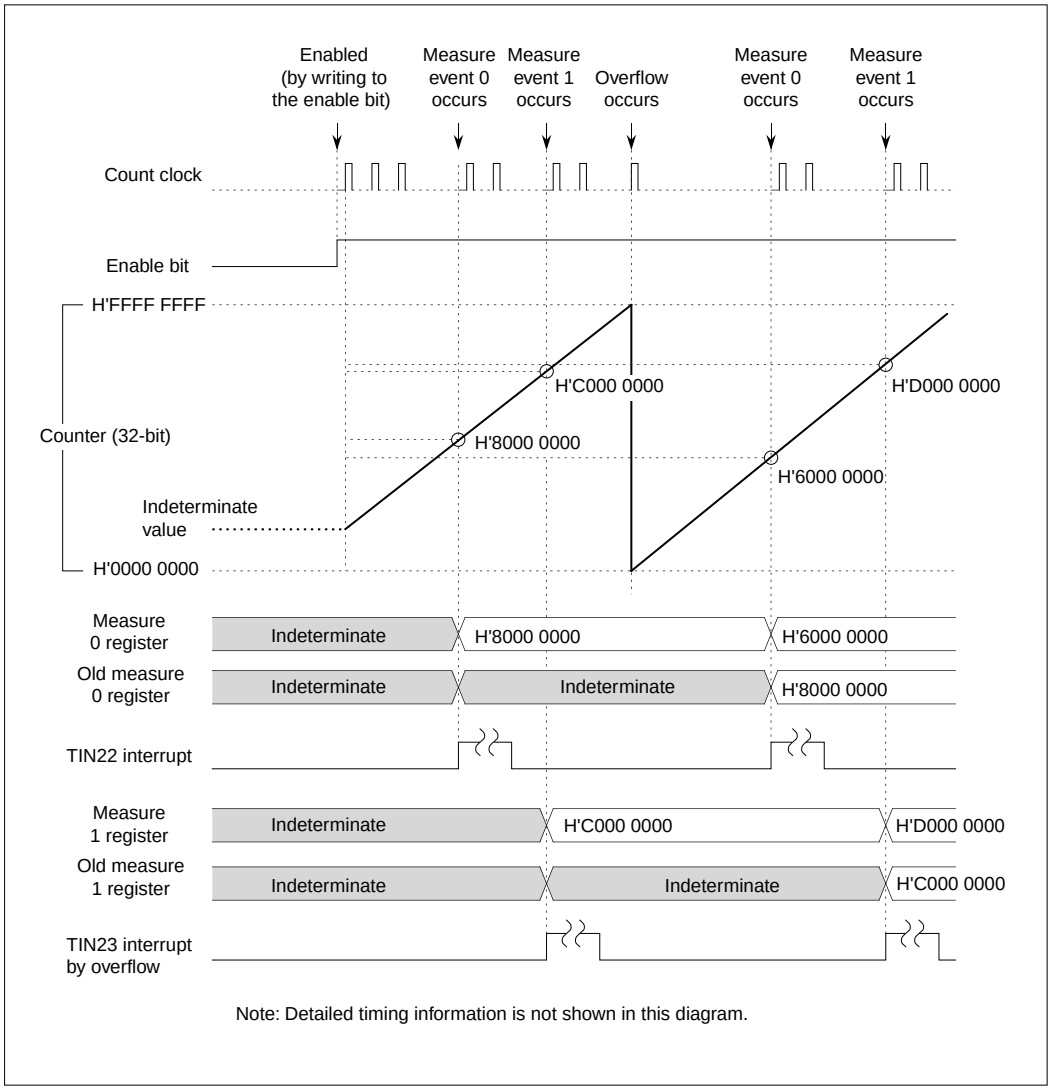


Figure 10.4.3 Typical Operation of TML Measure Input

(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 simultaneously in the same clock cycle, the write value is set in the counter, whereas the up-count value (before rewriting) is latched into the measure register.
- If the timer operates with any clock other than the 1/2 internal peripheral clock while clock bus 1 is selected for the count clock, the counter cannot be written to normally. Therefore, when operating with any clock other than the 1/2 internal peripheral clock, do not write to the counter.
- If the timer operates with any clock other than the 1/2 internal peripheral clock while clock bus 1 is selected for the count clock, the captured value is one that leads the actual counter value by one clock period. However, during the 1/2 internal peripheral clock interval from the count clock, this problem dose not occur and the counter value is captured at exact timing.

The diagram below shows the relationship between counter operation and the data that can be captured.

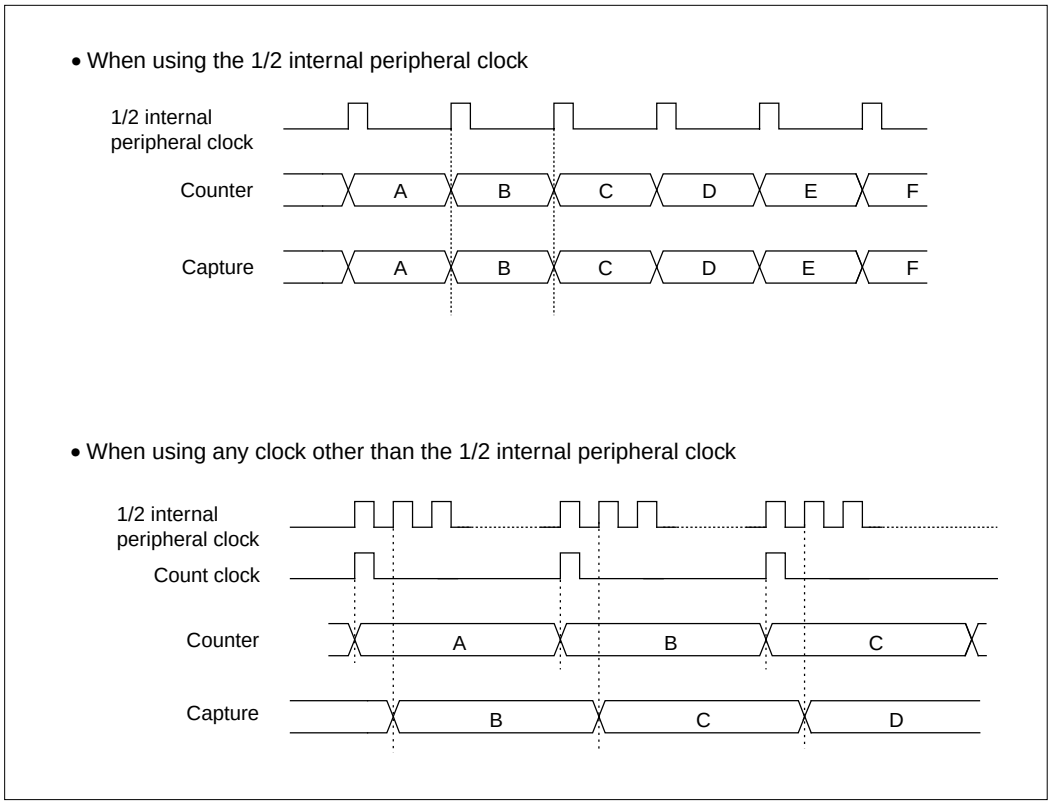


Figure 10.4.4 Mistimed Counter Value and Captured Value

10.5 TID (Input Related 16-bit Timers)

10.5.1 Outline of the TID

TID (Timer Input Derivation) consists of input related 16-bit timers which can be operated in one of the following modes as selected in software:

- Fixed period count mode
- Event count mode
- Multiply-by-4 event count mode
- Up/down event count mode

Specifications of the TID are listed below. A block diagram of the TID is shown in the next page.

Table 10.5.1 Specifications of TID (Input Related 16-bit Timers)

Item	Specification
Number of channels	2 channels
Counter	16-bit up/down-counter
Reload register	16-bit reload register
Timer start	Started by writing to the enable bit in software
Mode selection	<Input mode> • Fixed period count mode • Event count mode • Multiply-by-4 event count mode • Up/down event count mode
Interrupt generation	Can be generated by counter overflow or underflow

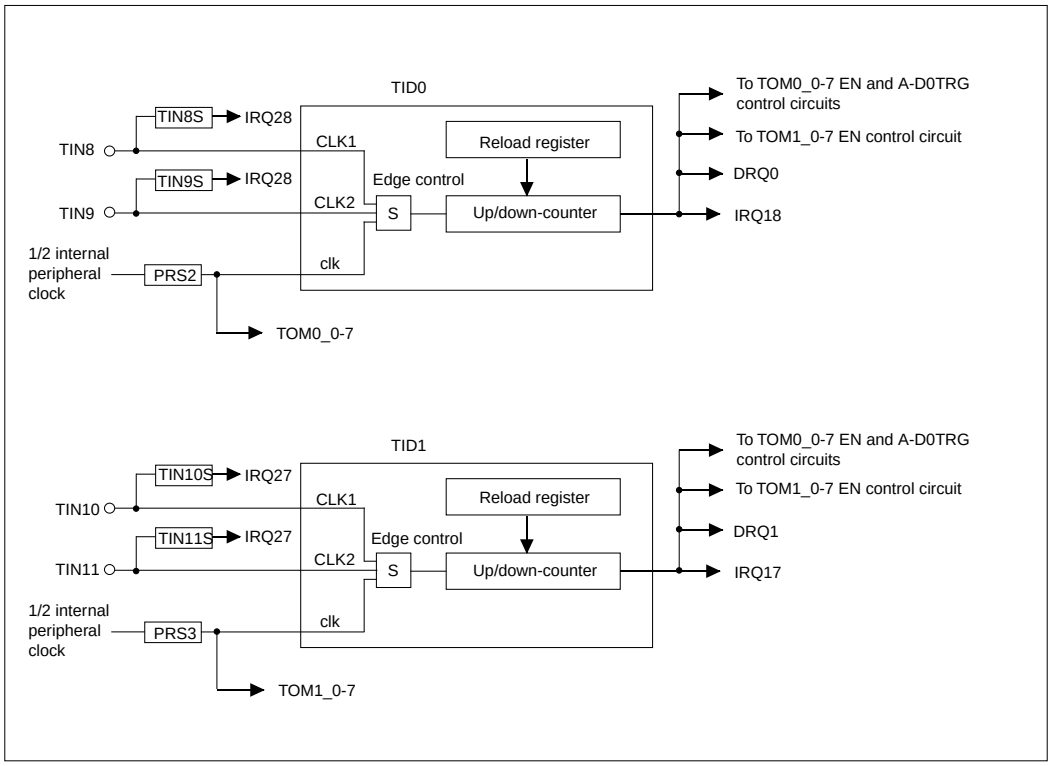


Figure 10.5.1 Block Diagram of TID (Input Related 16-bit Timers)

10.5.2 TID Related Register Map

A TID related register map is shown below.

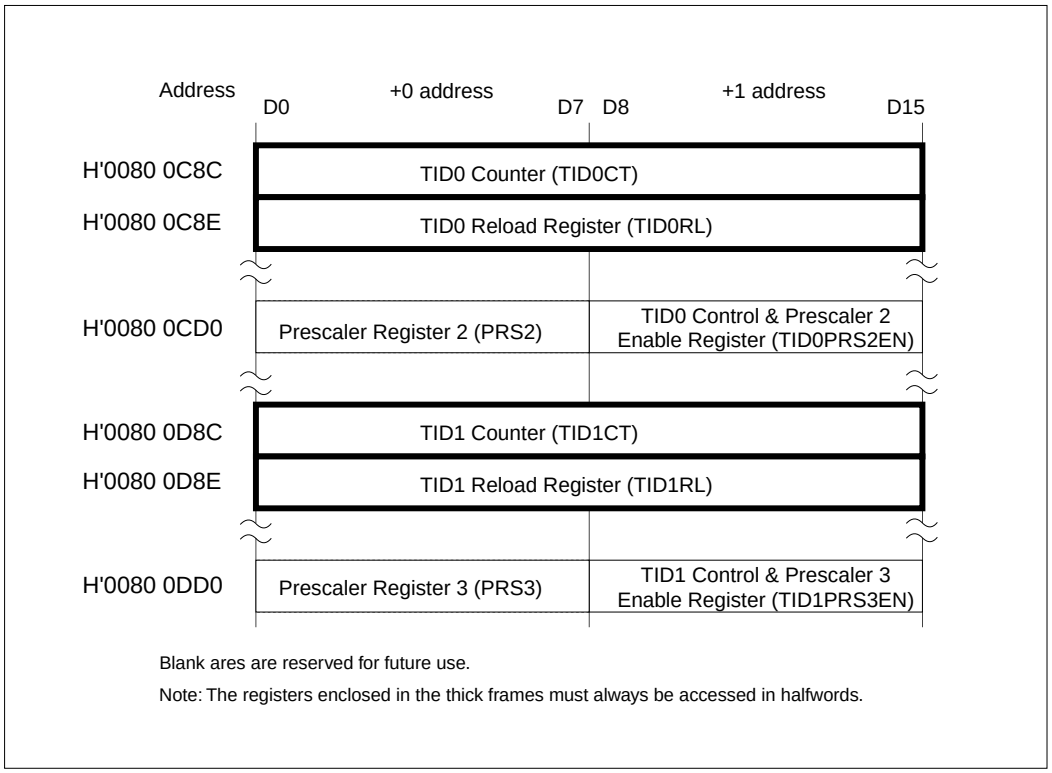
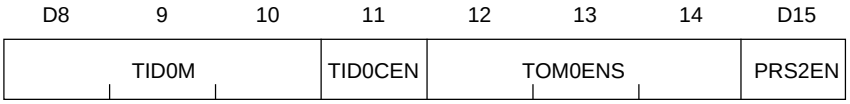


Figure 10.5.2 TID Related Register Map

10.5.3 TID Control & Prescaler Enable Registers

■ **TID0 Control & Prescaler 2 Enable Register (TID0PRS2EN)** **<Address: H'0080 0CD1>**



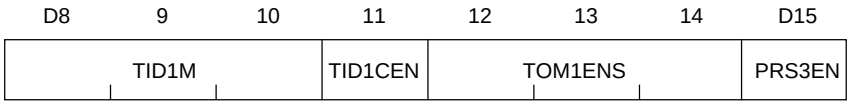
<When reset: H'00>

D	Bit Name	Function	R	W
8-10	TID0M (Selects TID0 operation mode)	X0X: Fixed period count mode X10: Multiply-by-4 event count mode 011: Event count mode 111: Up/down event count mode	○	○
11	TID0CEN (TID0 count enable)	0: Stops counting 1: Starts counting	○	○
12-14	TOM0ENS (Selects TOM0_0-7 enable source)	X0X: Disables event enable 010: TID0 output 011: TOM0_7 output 110: TID1 or TOM1_7 output 111: External input TIN18	○	○
15	PRS2EN (Prescaler 2 enable)	0: Stops counting 1: Starts counting	○	○

Note: Operation mode can only be set or changed when the counter is inactive.

The TID0 Control & Prescaler 2 Enable Register is used to select TID0 operation mode (fixed period count, event count, multiply-by-4 event count, or up/down event count mode), as well as select one of TOM0_0-7 timer enable sources and control startup of Prescaler 2.

■ **TID0 Control & Prescaler 3 Enable Register (TID0PRS3EN)** **<Address: H'0080 0DD1>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-10	TID1M (Selects TID1 operation mode)	X0X: Fixed period count mode X10: Multiply-by-4 event count mode 011: Event count mode 111: Up/down event count mode	○	○
11	TID1CEN (TID1 count enable)	0: Stops counting 1: Starts counting	○	○
12-14	TOM1ENS (Selects TOM1_0-7 enable source)	X0X: Disables event enable 010: TID1 output 011: TOM1_7 output 110: TID0 or TOM0_7 output 111: External input TIN19	○	○
15	PRS3EN (Prescaler 3 enable)	0: Stops counting 1: Starts counting	○	○

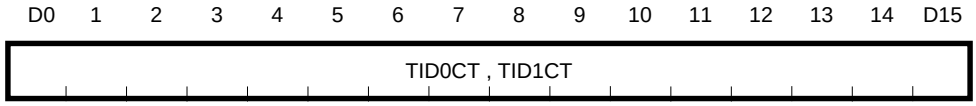
Note: Operation mode can only be set or changed when the counter is inactive.

The TID1 Control & Prescaler 3 Enable Register is used to select TID1 operation mode (fixed period count, event count, multiply-by-4 event count, or up/down event count mode), as well as select one of TOM1_0-7 timer enable sources and control startup of Prescaler 3.

10.5.4 TID Counters (TID0CT and TID1CT)

■ **TID0 Counter (TID0CT)** **<Address: H'0080 0C8C>**

■ **TID1 Counter (TID1CT)** **<Address: H'0080 0D8C>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TID0CT,TID1CT	16-bit counter value	○	○

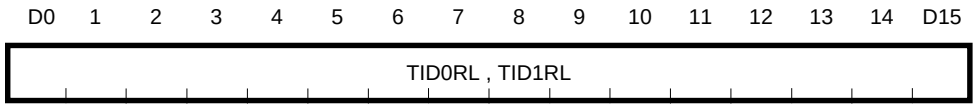
Note: This register must always be accessed in halfwords.

The TID Counters are a 16-bit up/down-counter which starts counting synchronously with the count clock after the timer is enabled (by writing to the enable bit in software).

10.5.5 TID Reload Registers (TID0RL and TID1RL)

■ **TID0 Reload Register (TID0RL)** **<Address: H'0080 0C8E>**

■ **TID1 Reload Register (TID1RL)** **<Address: H'0080 0D8E>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TID0RL,TID1RL	16-bit reload register value	○	○

Note: This register must always be accessed in halfwords.

The TID Reload Registers are used to reload the TID Counter Registers (TID0CT and TID1CT) with data.

The following shows the timing at which the content of the reload register is loaded into the corresponding counter.

- When the counter underflows in fixed count mode
- When the counter is enabled in fixed count mode

Simply because data is written to the reload register does not mean that the counter is loaded with the data.

10.5.6 Outline of Each TID Operation Mode

The following outlines each TID operation mode. When using the TID, select one of these operation modes.

(1) Fixed period count mode

In fixed period count mode, the TID uses the reload register to generate an interrupt at intervals of the reload register set value + 1.

When the timer is enabled (by writing to the enable bit in software) after setting the reload register (initially indeterminate), the counter is loaded with the content of the reload register synchronously with the count clock and thereby starts counting. The counter counts down and when it underflows after reaching the minimum count, the counter is loaded with the content of the reload register again, from which it continues counting.
To stop the counter, write to the enable bit in software to disable counting.
Also, an interrupt can be generated each time the counter underflows.

The reload register set value + 1 is the valid count value.

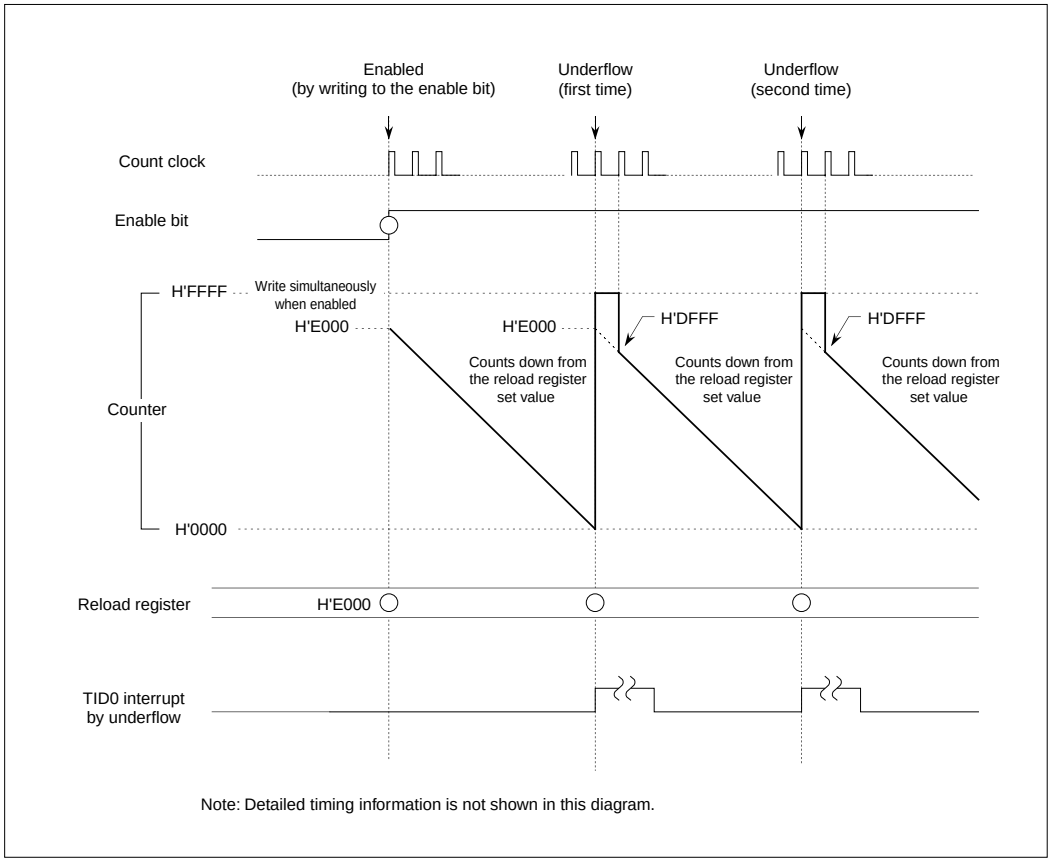


Figure 10.5.3 Typical Operation of TID in Fixed Period Count Mode

(2) Event count mode

In event count mode, the TID uses the signal (TIN8 or TIN10) entered from the outside as the clock source with which to run the counter.

Note: TIN9 and TIN11 cannot be used as the clock source for the TID.

By detecting the rising and falling edges of the externally sourced signal (TIN8 or TIN10), the TID generates a clock that is synchronized to the internal clock. When after setting the counter the timer is enabled (by writing to the enable bit in software), the counter starts counting up from its set value synchronously with the generated clock.
An interrupt can be generated when the counter overflows.

To stop the counter, write to the enable bit in software to disable counting or fix the externally sourced signal level high or low.

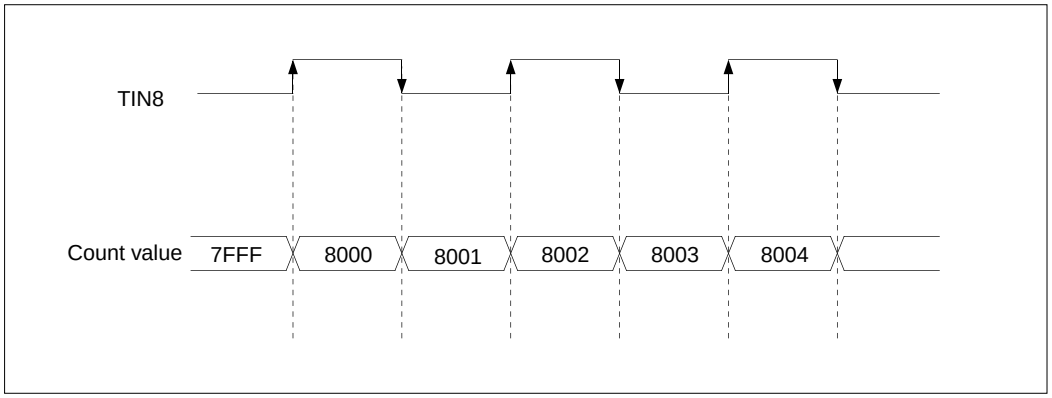


Figure 10.5.4 Typical Operation of the TID in Event Count Mode (Basic Operation)

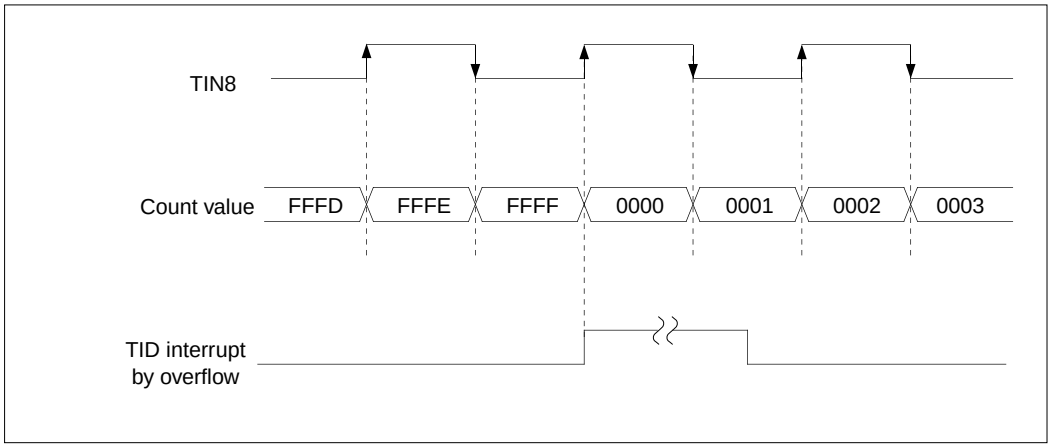


Figure 10.5.5 Typical Operation of the TID in Event Count Mode (when the Counter Overflows)

(3) Multiply-by-4 event count mode

In Multiply-by-4 event count mode, the TID uses two train of signals (TIN8 and TIN9, TIN10 and TIN11) entered from the outside as the clock sources with which to run the counter. The counter is switched between up-count and down-count by the input state of the two signals.

By detecting the rising and falling edges of both of the two externally sourced signals, the TID generates a clock that is synchronized to the internal clock. When after setting the counter the timer is enabled (by writing to the enable bit in software), the counter starts counting from its set value synchronously with the generated clock. For details on whether the counter counts up or down, see Table 10.5.2.

An interrupt can be generated when the counter overflows or underflows.

To stop the counter, write to the enable bit in software to disable counting or fix the externally sourced signal levels high or low.

Table 10.5.2 Count Direction during Multiply-by-4 Event Count Mode

Input	Count Direction							
	Up-count				Down-count			
TIN8 (TIN10)	H	↓	L	↑	H	↓	L	↑
TIN9 (TIN11)	↑	H	↓	L	↓	L	↑	H

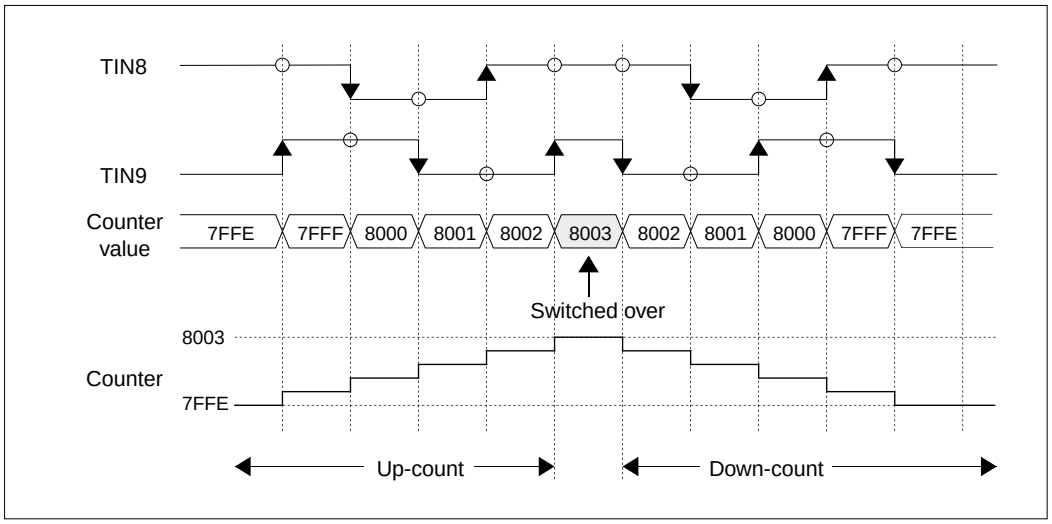


Figure 10.5.6 Up/Down Count Operation (Switchover Timing)

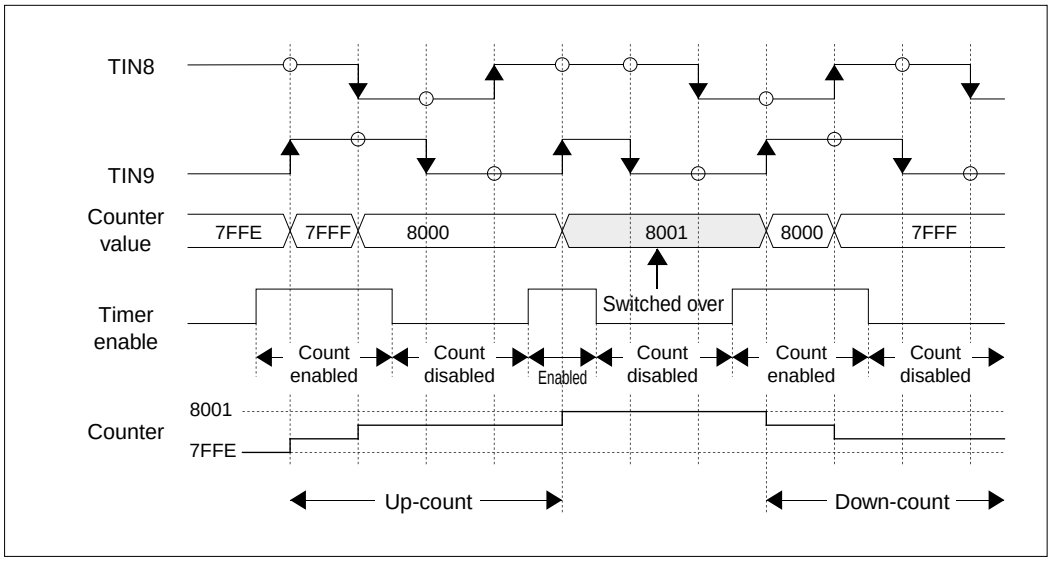


Figure 10.5.7 Up/Down Count Operation (Count Enabled/Disabled)

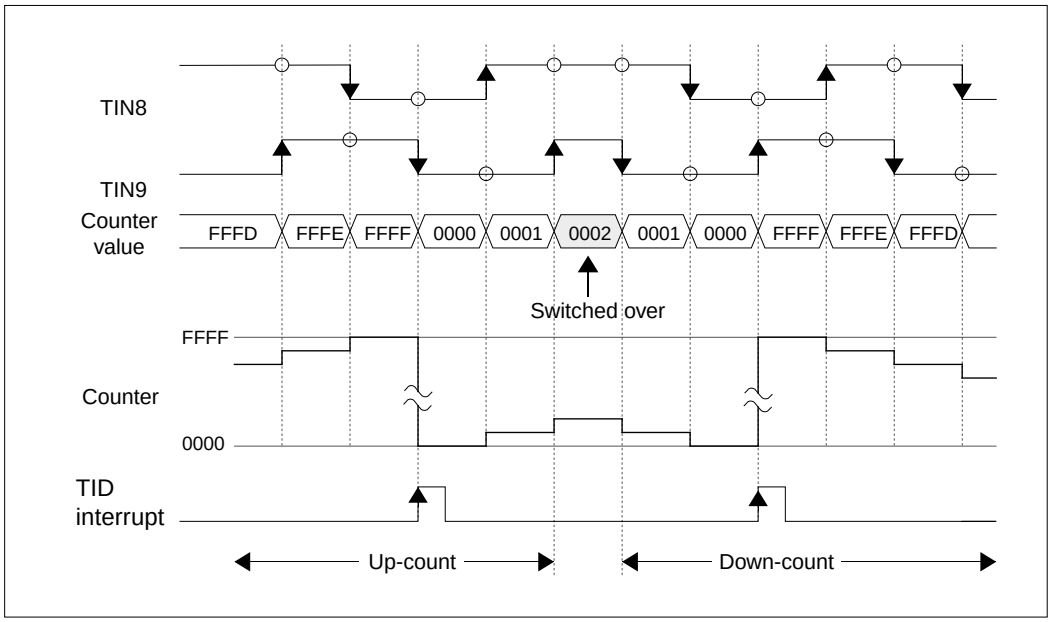


Figure 10.5.8 Up/Down Count Operation (Interrupt Timing)

(5) Up/down event count mode

In up/down event count mode, the TID uses one of two train of signals (TIN8 or TIN10) entered from the outside as the clock source and the other (TIN9 or TIN11) as the up/down signal with which to run the counter. The counter is switched between up-count and down-count by the input state of the up/down signal.

By detecting the rising and falling edges of the externally sourced clock signal, the TID generates a clock that is synchronized to the internal clock. When after setting the counter the timer is enabled (by writing to the enable bit in software), the counter starts counting up or down from its set value synchronously with the generated clock.

The direction in which the counter counts is determined by the input level of the up/down signal (see Table 10.5.3). An interrupt can be generated when the counter overflows or underflows.

To stop the counter, write to the enable bit in software to disable counting or fix the externally sourced clock signal level high or low.

Note: TIN9 and TIN11 cannot be used as the clock source.

Table 10.5.3 Count Direction during Up/Down Event Count Mode

Input	Count Direction			
	Up-count		Down-count	
TIN8 (TIN10)	↑	↓	↑	↓
TIN9 (TIN11)	Low level		High level	

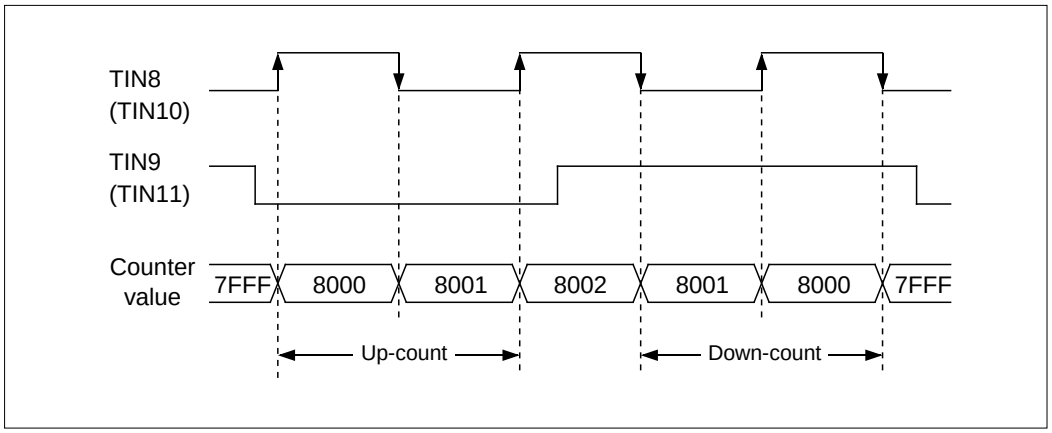


Figure 10.5.9 Up/Down Count Operation

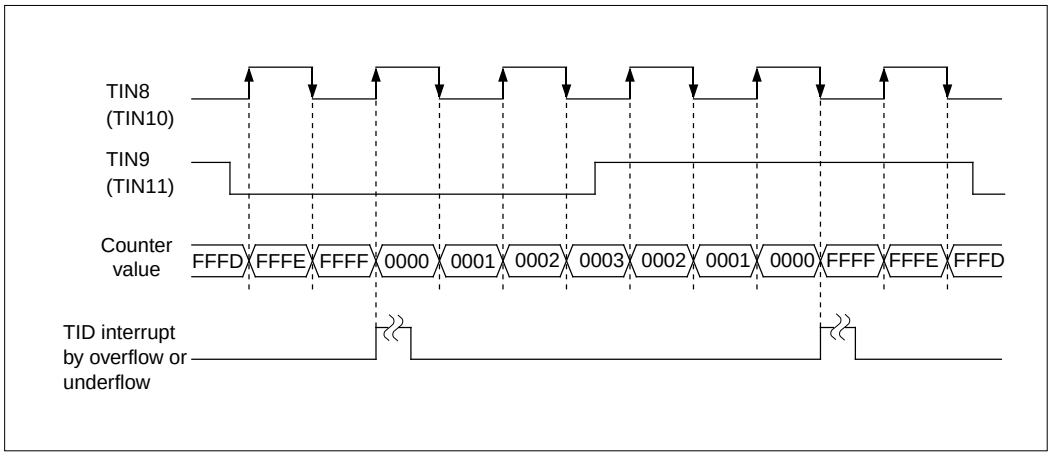


Figure 10.5.10 Up/Down Count Operation (Interrupt Timing)

10.6 TOM (Output Related 16-bit Timers)

10.6.1 Outline of the TOM

TOM (Timer Output Modification) consists of output related 16-bit timers which can be run in one of the following modes as selected in software:

<Output modes without correction function>
- PWM output mode
- Single-shot output mode
- Single-shot PWM output mode
- Successive output mode

Specifications of the TOM are listed below. A block diagram of the TOM is shown in the next page.

Table 10.6.1 Specifications of TOM (Output Related 16-bit Timers)

Item	Specification
Number of channels	16 channels (two blocks, 8 channels each, for a total of 16 channels)
Counter	16-bit down-counter (16 pcs.)
Reload register	16-bit reload register (16 pcs.)
Timer start	TOM0_0-7: Started by writing to the enable bit in software or by an underflow/overflow signal from TID0 timer, an underflow signal from TOM0_7, an underflow/overflow signal from TID1 timer, or an underflow signal from TOM1_7 TOM1_0-7: Started by writing to the enable bit in software or by an underflow/overflow signal from TID1 timer, an underflow signal from TOM1_7, an underflow/overflow signal from TID0 timer, or an underflow signal from TOM0_7
Mode selection	<Output modes without correction function> • PWM output mode • Single-shot output mode • Single-shot PWM output mode • Successive output mode
Interrupt generation	Can be generated by counter underflow

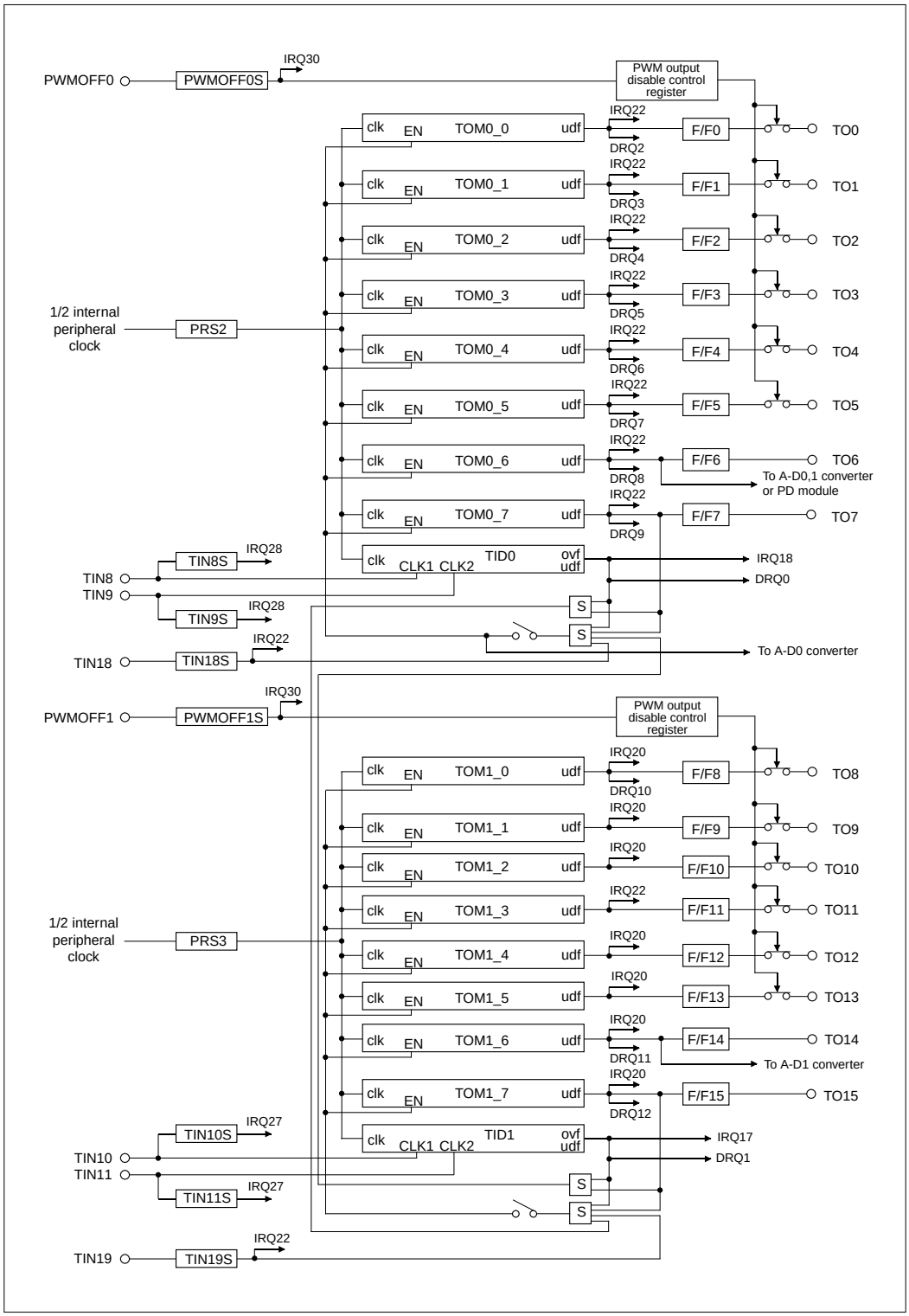


Figure 10.6.1 Block Diagram of TOM (Output Related 16-bit Timers)

10.6.2 Outline of Each TOM Operation Mode

The following outlines each TOM operation mode. When using the TOM, select one of these operation modes.

(1) PWM output mode (without correction function)

In PWM output mode, the TOM uses two reload registers to generate a waveform with any duty cycle.

When the timer is enabled after setting the initial value in the reload 0 and the reload 1 registers, the counter is loaded with the value of the reload 0 register synchronously with the count clock, from which it starts counting down. The first time the counter underflows after reaching the minimum count, it is loaded with the content of the reload 1 register. Thereafter, the counter is alternately reloaded by the reload 0 and the reload 1 registers each time it underflows.

The F/F output waveform in PWM output mode is inverted when the counter starts counting and each time it underflows. To stop the timer, write to the enable bit to disable counting, and the timer immediately stops (not synchronized to the PWM output period).
An interrupt and a DMA transfer request can be generated the second, the fourth, and other even-numbered times the counter underflows after being enabled.

(2) Single-shot output mode (without correction function)

In single-shot output mode, the TOM generates a pulse in duration of the reload 0 register set value + 1 only once and then stops.

When the timer is enabled after setting the reload 0 register, the counter is loaded with the content of the reload 0 register synchronously with the count clock, from which it starts counting. The counter counts down and when it underflows after reaching the minimum count, it stops.

The F/F output waveform in single-shot output mode is inverted when the counter starts counting and when it underflows. In this way, the TOM generates a single-shot pulse in duration of the reload 0 register set value + 1 only once.
An interrupt and a DMA transfer request can be generated when the counter underflows.

(3) Single-shot PWM output mode (without correction function)

In single-shot PWM output mode, the TOM uses two reload registers to generate a waveform with any duty cycle only once.

When the timer is enabled after setting the initial value in the reload 0 and the reload 1 registers, the counter is loaded with the value of the reload 0 register synchronously with the count clock, from which it starts counting down. The first time the counter underflows after reaching the minimum count, it is loaded with the content of the reload 1 register. When the counter underflows second time, it stops counting. The reload 0 register set value + 1 and the reload 1 register set value + 1 respectively are the valid count values.
To stop the timer in software, write to the enable bit to disable counting, and the timer immediately stops (not synchronized to the PWM output period).

The F/F output waveform in single-shot PWM output mode is inverted (F/F output changes level from low to high or vice versa) each time the counter underflows. (Unlike in PWM output mode, the F/F output waveform is not inverted when the counter is enabled.)
An interrupt and a DMA transfer request can be generated the second time the counter underflows after being enabled.

(4) Successive output mode (without correction function)

In successive output mode, the counter counts down from its set value and upon underflowing, it is loaded with the value of the reload 0 register. Therefore, this operation is repeated each time the counter underflows, thereby successively generating a pulse in duration of the reload 0 register set value + 1.

When the timer is enabled after setting the counter and the reload 0 register, the counter starts counting down from its set value synchronously with the count clock.
When the counter underflows after reaching the minimum count, it is loaded with the content of the reload 0 register and starts counting again. This is repeated each time the counter underflows. To stop the counter, write to the enable bit in software to disable counting.

The F/F output waveform in successive output mode is inverted when the counter starts counting and when it underflows, thereby successively generating a pulse until the count stops.
An interrupt and a DMA transfer request can be generated each time the counter underflows.

10.6.3 TOM Related Register Map

A TOM related register map is shown below.

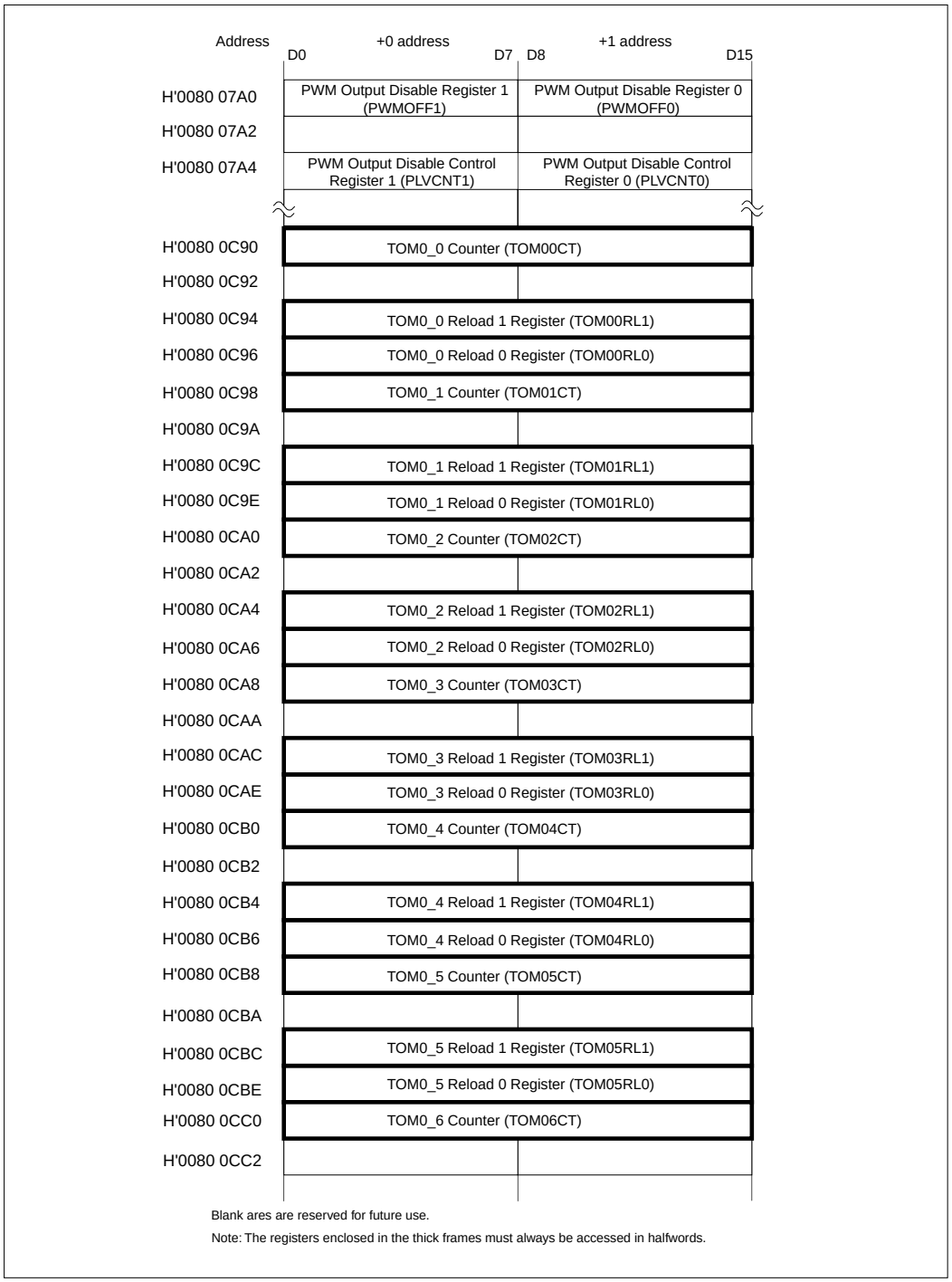
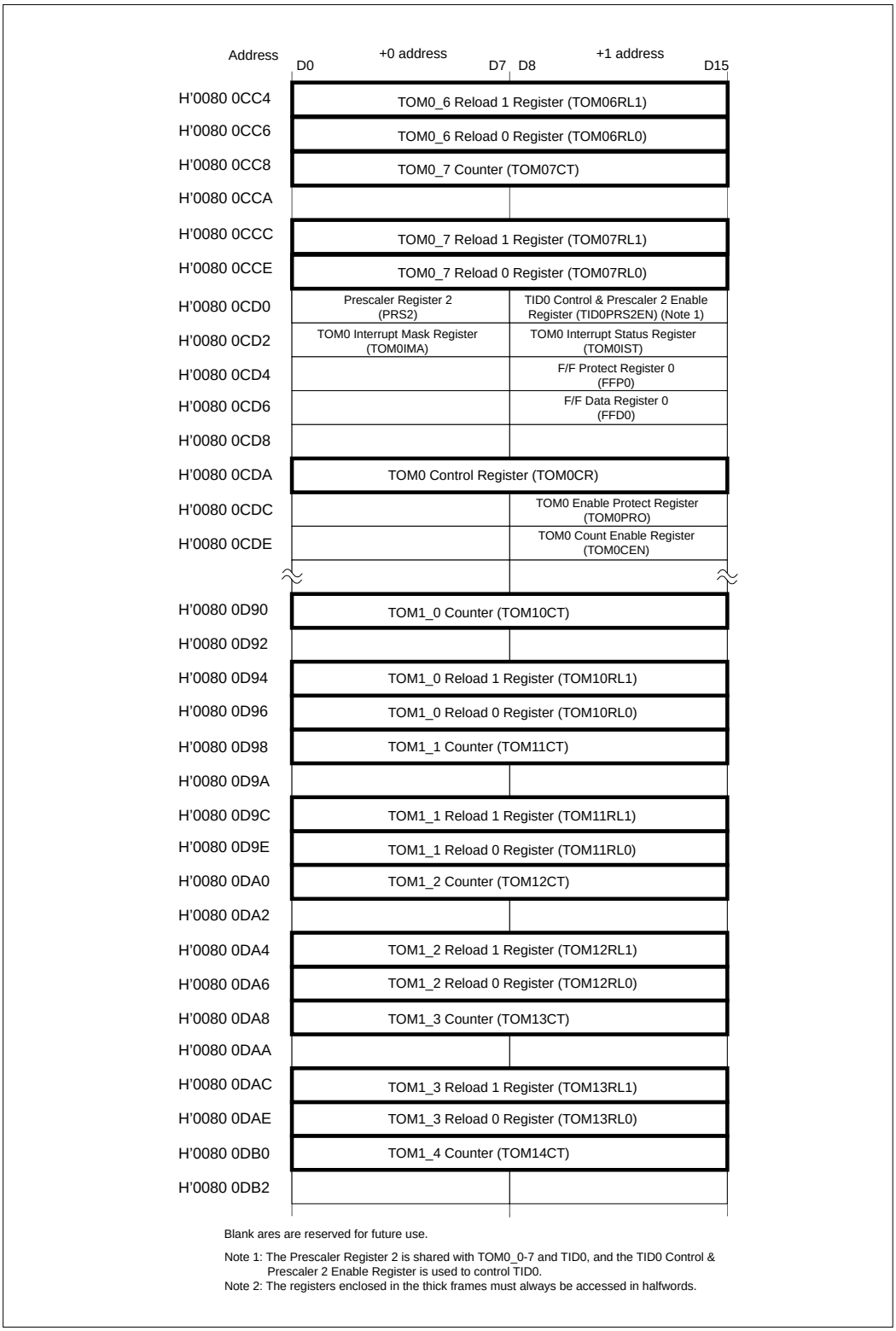


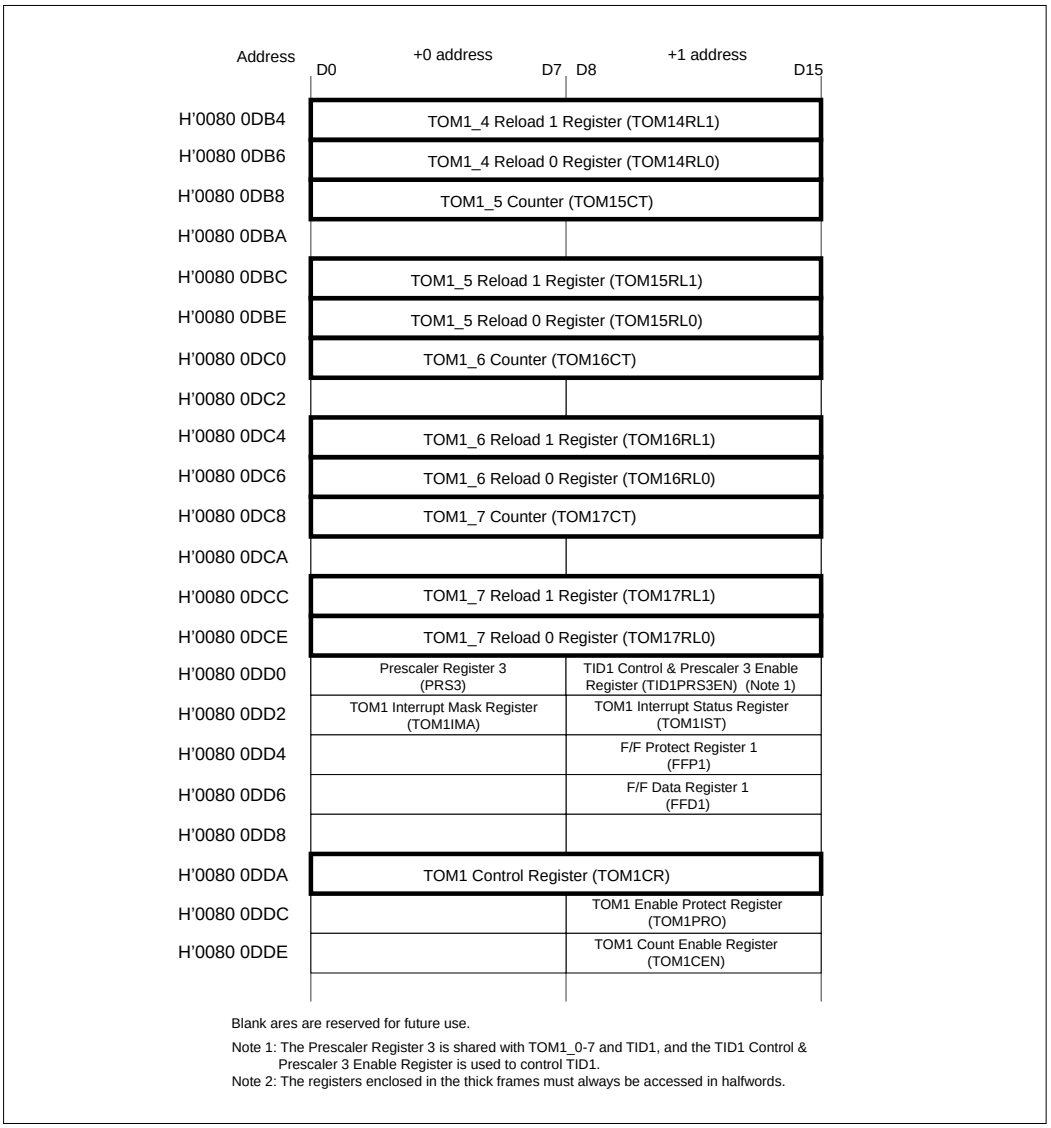
Figure 10.6.2 TOM Related Register Map (1/3)



Blank ares are reserved for future use.

Note 1: The Prescaler Register 2 is shared with TOM0_0-7 and TID0, and the TID0 Control & Prescaler 2 Enable Register is used to control TID0.
Note 2: The registers enclosed in the thick frames must always be accessed in halfwords.

Figure 10.6.3 TOM Related Register Map (2/3)



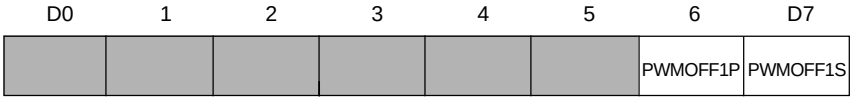
Blank ares are reserved for future use.

Note 1: The Prescaler Register 3 is shared with TOM1_0-7 and TID1, and the TID1 Control & Prescaler 3 Enable Register is used to control TID1.
Note 2: The registers enclosed in the thick frames must always be accessed in halfwords.

Figure 10.6.4 TOM Related Register Map (3/3)

10.6.4 PWM Output Disable Registers

■ **PWM Output Disable Register 1 (PWMOFF1)** **<Address: H'0080 07A0>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6	PWMOFF1P (PWMOFF1S write control)		0	○
7	PWMOFF1S (Selects port P100-P105 output disable)	0: Enables output 1: Disables output	○	○

This register controls PWM outputs from TOM1_0 through TOM1_5 timers by enabling or disabling PWM outputs on the corresponding ports P100-P105.
To ensure that the value set in the port P100-P105 output disable select (PWMOFF1S) bit is effective (0: enables output, 1: disables output), bits in this register must be written correctly by following the procedure described below.

[Write procedure]

① Set the PWMOFF1S write control (PWMOFF1P) bit to 1.
② Subsequently after writing in ① above, set the PWMOFF1S write control (PWMOFF1P) bit to 0 and the port P100-P105 output disable select (PWMOFF1S) bit to 0 or 1.

Note: If a write cycle for any other area occurs between ① and ②, the value set in the PWMOFF1S write control (PWMOFF1P) bit has no effect.

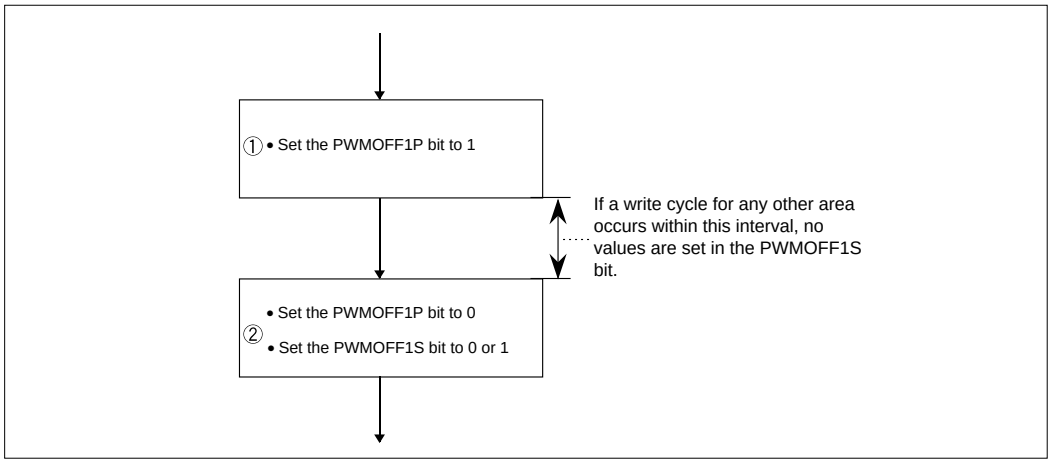


Figure 10.6.5 Procedure for Writing to the PWMOFF1 Register

■ **PWM Output Disable Register 0 (PWMOFF0)** **<Address: H'0080 07A1>**

D8	9	10	11	12	13	14	D15
						PWMOFF0P	PWMOFF0S

<When reset: H'00>

D	Bit Name	Function	R	W
8-13	No functions assigned		0	–
14	PWMOFF0P (PWMOFF0S write control)		0	○
15	PWMOFF0S (Selects port P110-P115 output disable)	0: Enables output 1: Disables output	○	○

This register controls PWM outputs from TOM0_0 through TOM0_5 timers by enabling or disabling PWM outputs on the corresponding ports P110-P115.
To ensure that the value set in the port P110-P115 output disable select (PWMOFF0S) bit is effective (0: enables output, 1: disables output), bits in this register must be written correctly by following the procedure described below.

[Write procedure]

① Set the PWMOFF0S write control (PWMOFF0P) bit to 1.
② Subsequently after writing in ① above, set the PWMOFF0S write control (PWMOFF0P) bit to 0 and the port P110-P115 output disable select (PWMOFF0S) bit to 0 or 1.

Note: If a write cycle for any other area occurs between ① and ②, the value set in the PWMOFF0S write control (PWMOFF0P) bit has no effect.

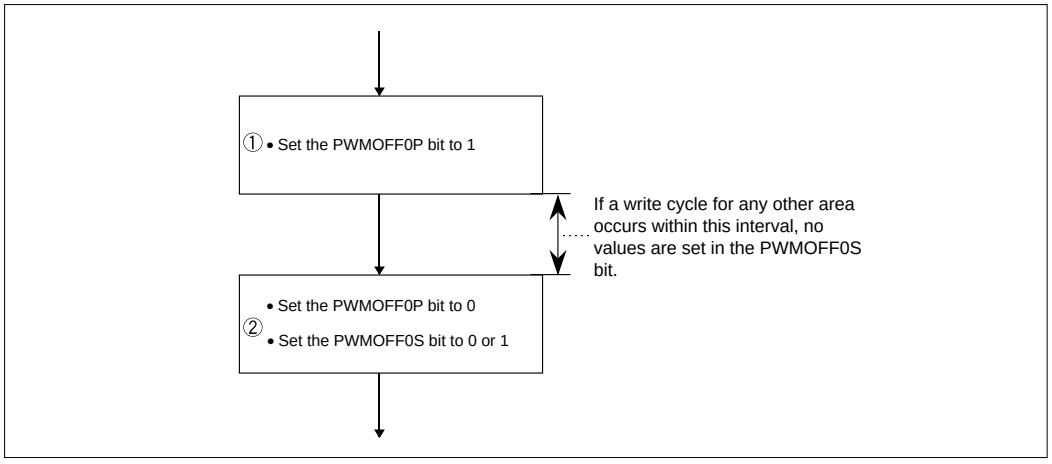
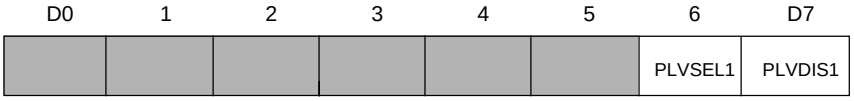


Figure 10.6.6 Procedure for Writing to the PWMOFF0 Register

10.6.5 PWM Output Disable Control Registers

■ **PWM Output Disable Control Register 1 (PLVCNT1)** **<Address: H'0080 07A4>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6	PLVSEL1 (Selects port P100-P105 output disable level)	0: Selects to disable port output when low 1: Selects to disable port output when high	○	○
7	PLVDIS1 (Invalidates/validates output disable selection)	0: Invalidates output disable selection 1: Validates output disable selection	○	○

This register controls PWM outputs from the corresponding TOM1_0 through TOM1_5 timers by enabling or disabling them depending on the port P100-P105 levels (high or low).
The port disable level select function forcibly turns port output off depending on the port output status.
This function may be used in determining whether the three-phase PWM signals turn on simultaneously. Or because its behavior depends on the port output status, it may also be used to check ports for duplicates.

(1) PLVSEL1 (port P100-P105 output disable level select) bit (D6)

This bit selects the port P100-P105 level (high or low) at which to disable PWM output.
To disable PWM output when the port level is low, set this bit to 0. To disable PWM output when the port level is high, set this bit to 1.
The following shows conditions under which port output is turned off depending on the port output status.

① **When PLVSEL1 = 0, port level = 0**

Port output is forcibly turned off when one of the following conditions holds true:

- Ports P100 and P101 both are 0
- Ports P102 and P103 both are 0
- Ports P104 and P105 both are 0

② **When PLVSEL1 = 1, port level = 1**

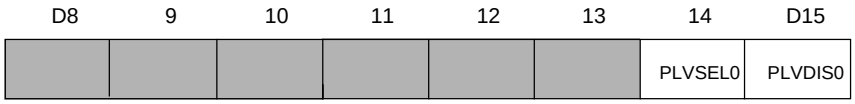
Port output is forcibly turned off when one of the following conditions holds true:

- Ports P100 and P101 both are 1
- Ports P102 and P103 both are 1
- Ports P104 and P105 both are 1

(2) PLVDIS1 (port P100-P105 output disable selection validate/invalidate) bit (D7)

This bit validates or invalidates the PWM output disable level selected with the PLVSEL1 bit. Setting this bit to 1 validates the PWM output disable level selected with the PLVSEL1 bit, so that PWM output is disabled when the port P100-P105 is at the level selected with the PLVSEL1 bit. Setting this bit to 0 invalidates the PWM output disable level selected with the PLVSEL1 bit, so that PWM output cannot be disabled according to the port P100-P105 level.

■ **PWM Output Disable Control Register 0 (PLVCNT0)** **<Address: H'0080 07A5>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-10	No functions assigned		0	–
14	PLVSEL0 (Selects port P110-P115 output disable level)	0: Selects to disable port output when low 1: Selects to disable port output when high	○	○
15	PLVDIS0 (Invalidates/validates output disable selection)	0: Invalidates output disable selection 1: Validates output disable selection	○	○

This register controls PWM outputs from the corresponding TOM0_0 through TOM0_5 timers by enabling or disabling them depending on the port P110-P115 levels (high or low).
The port disable level select function forcibly turns port output off depending on the port output status. This function may be used in determining whether the three-phase PWM signals turn on simultaneously. Or because its behavior depends on the port output status, it may also be used to check ports for duplicates.

(1) PLVSEL0 (port P110-P115 output disable level select) bit (D14)

This bit selects the port P110-P115 level (high or low) at which to disable PWM output.
To disable PWM output when the port level is low, set this bit to 0. To disable PWM output when the port level is high, set this bit to 1.
The following shows conditions under which port output is turned off depending on the port output status.

① **When PLVSEL0 = 0, port level = 0**

Port output is forcibly turned off when one of the following conditions holds true:

- Ports P110 and P111 both are 0
- Ports P112 and P113 both are 0
- Ports P114 and P115 both are 0

② **When PLVSEL0 = 1, port level = 1**

Port output is forcibly turned off when one of the following conditions holds true:

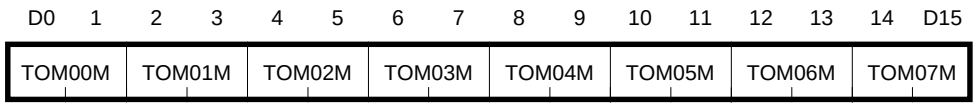
- Ports P110 and P111 both are 1
- Ports P112 and P113 both are 1
- Ports P114 and P115 both are 1

(2) PLVDIS0 (port P110-P115 output disable selection validate/invalidate) bit (D15)

This bit validates or invalidates the PWM output disable level selected with the PLVSEL1 bit. Setting this bit to 1 validates the PWM output disable level selected with the PLVSEL1 bit, so that PWM output is disabled when the port P110-P115 is at the level selected with the PLVSEL0 bit. Setting this bit to 0 invalidates the PWM output disable level selected with the PLVSEL1 bit, so that PWM output cannot be disabled according to the port P110-P115 level.

10.6.6 TOM Control Registers

■ **TOM0 Control Register (TOM0CR)** **<Address: H'0080 0CDA>**

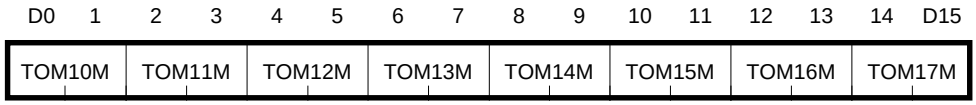


<When reset: H'0000>

D	Bit Name	Function	R	W
0,1	TOM00M (Selects TOM0_0 operation mode)	00: Single-shot output mode 01: Single-shot PWM output mode 10: Successive output mode 11: PWM output mode	○	○
2,3	TOM01M (Selects TOM0_1 operation mode)			
4,5	TOM02M (Selects TOM0_2 operation mode)			
6,7	TOM03M (Selects TOM0_3 operation mode)			
8,9	TOM04M (Selects TOM0_4 operation mode)			
10,11	TOM05M (Selects TOM0_5 operation mode)			
12,13	TOM06M (Selects TOM0_6 operation mode)			
14,15	TOM07M (Selects TOM0_7 operation mode)			

The TOM0 Control Registers are used to select operation modes of TOM0_0-7 (PWM output, single-shot output, single-shot PWM output, or successive output mode).

■ **TOM1 Control Register (TOM1CR)** **<Address: H'0080 0DDA>**



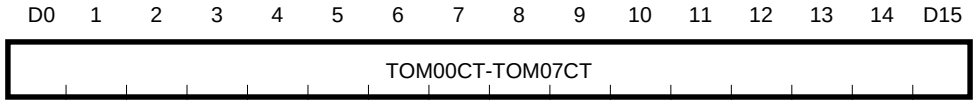
<When reset: H'0000>

D	Bit Name	Function	R	W
0,1	TOM10M (Selects TOM1_0 operation mode)	00: Single-shot output mode 01: Single-shot PWM output mode	○	○
2,3	TOM11M (Selects TOM1_1 operation mode)	10: Successive output mode 11: PWM output mode		
4,5	TOM12M (Selects TOM1_2 operation mode)			
6,7	TOM13M (Selects TOM1_3 operation mode)			
8,9	TOM14M (Selects TOM1_4 operation mode)			
10,11	TOM15M (Selects TOM1_5 operation mode)			
12,13	TOM16M (Selects TOM1_6 operation mode)			
14,15	TOM17M (Selects TOM1_7 operation mode)			

The TOM1 Control Registers are used to select operation modes of TOM1_0-7 (PWM output, single-shot output, single-shot PWM output, or successive output mode).

10.6.7 TOM Counters

■ **TOM0_0 Counter (TOM00CT)** **<Address: H'0080 0C90>**
■ **TOM0_1 Counter (TOM01CT)** **<Address: H'0080 0C98>**
■ **TOM0_2 Counter (TOM02CT)** **<Address: H'0080 0CA0>**
■ **TOM0_3 Counter (TOM03CT)** **<Address: H'0080 0CA8>**
■ **TOM0_4 Counter (TOM04CT)** **<Address: H'0080 0CB0>**
■ **TOM0_5 Counter (TOM05CT)** **<Address: H'0080 0CB8>**
■ **TOM0_6 Counter (TOM06CT)** **<Address: H'0080 0CC0>**
■ **TOM0_7 Counter (TOM07CT)** **<Address: H'0080 0CC8>**



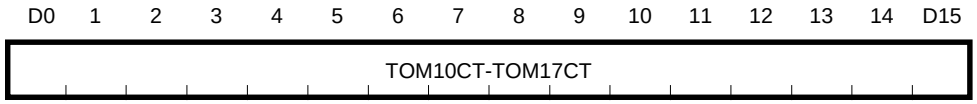
<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TOM00CT-TOM07CT	16-bit counter value	○	△

W=△: Writing to the counter in PWM output or single-shot PWM output mode has no effect.
Note: These registers must always be accessed in halfwords.

The TOM0 Counters are a 16-bit counter which after the timer is enabled, starts counting synchronously with the count clock.
Writing to these counters in PWM output or single-shot PWM output mode has no effect.

■ **TOM1_0 Counter (TOM10CT)** **<Address: H'0080 0D90>**
■ **TOM1_1 Counter (TOM11CT)** **<Address: H'0080 0D98>**
■ **TOM1_2 Counter (TOM12CT)** **<Address: H'0080 0DA0>**
■ **TOM1_3 Counter (TOM13CT)** **<Address: H'0080 0DA8>**
■ **TOM1_4 Counter (TOM14CT)** **<Address: H'0080 0DB0>**
■ **TOM1_5 Counter (TOM15CT)** **<Address: H'0080 0DB8>**
■ **TOM1_6 Counter (TOM16CT)** **<Address: H'0080 0DC0>**
■ **TOM1_7 Counter (TOM17CT)** **<Address: H'0080 0DC8>**



<When reset: indeterminate>

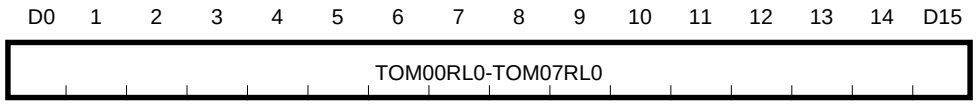
D	Bit Name	Function	R	W
0-15	TOM10CT-TOM17CT	16-bit counter value	○	△

W=△: Writing to the counter in PWM output or single-shot PWM output mode has no effect.
Note: These registers must always be accessed in halfwords.

The TOM1 Counters are a 16-bit counter which after the timer is enabled, starts counting synchronously with the count clock.
Writing to these counters in PWM output or single-shot PWM output mode has no effect.

10.6.8 TOM Reload 0 Registers

■ **TOM0_0 Reload 0 Register (TOM00RL0)** **<Address: H'0080 0C96>**
■ **TOM0_1 Reload 0 Register (TOM01RL0)** **<Address: H'0080 0C9E>**
■ **TOM0_2 Reload 0 Register (TOM02RL0)** **<Address: H'0080 0CA6>**
■ **TOM0_3 Reload 0 Register (TOM03RL0)** **<Address: H'0080 0CAE>**
■ **TOM0_4 Reload 0 Register (TOM04RL0)** **<Address: H'0080 0CB6>**
■ **TOM0_5 Reload 0 Register (TOM05RL0)** **<Address: H'0080 0CBE>**
■ **TOM0_6 Reload 0 Register (TOM06RL0)** **<Address: H'0080 0CC6>**
■ **TOM0_7 Reload 0 Register (TOM07RL0)** **<Address: H'0080 0CCE>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TOM00RL0-TOM07RL0	16-bit reload register value	○	○

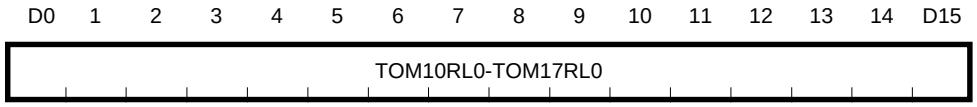
Note: These registers must always be accessed in halfwords.

The TOM0 Reload 0 Registers are used to reload the TOM0 Counter Registers (TOM00CT through TOM07CT) with data.
The following shows the timing at which the content of the reload 0 register is loaded into the corresponding counter.

- When the counter is enabled in single-shot output, PWM output, or single-shot PWM output mode
- When the counter underflows in successive output mode
- When the counter underflows after reaching the count value set by reload 1 register in PWM output mode

Simply because data is written to the reload 0 register does not mean that the counter is loaded with the data.

■ **TOM1_0 Reload 0 Register (TOM10RL0)** **<Address: H'0080 0D96>**
■ **TOM1_1 Reload 0 Register (TOM11RL0)** **<Address: H'0080 0D9E>**
■ **TOM1_2 Reload 0 Register (TOM12RL0)** **<Address: H'0080 0DA6>**
■ **TOM1_3 Reload 0 Register (TOM13RL0)** **<Address: H'0080 0DAE>**
■ **TOM1_4 Reload 0 Register (TOM14RL0)** **<Address: H'0080 0DB6>**
■ **TOM1_5 Reload 0 Register (TOM15RL0)** **<Address: H'0080 0DBE>**
■ **TOM1_6 Reload 0 Register (TOM16RL0)** **<Address: H'0080 0DC6>**
■ **TOM1_7 Reload 0 Register (TOM17RL0)** **<Address: H'0080 0DCE>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TOM10RL0 - TOM17RL0	16-bit reload register value	○	○

Note: These registers must always be accessed in halfwords.

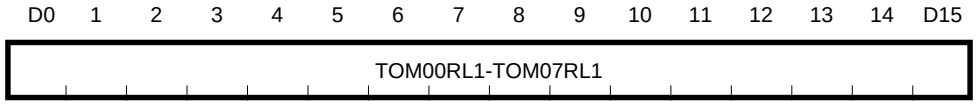
The TOM1 Reload 0 Registers are used to reload the TOM1 Counter Registers (TOM10CT through TOM17CT) with data.
The following shows the timing at which the content of the reload 0 register is loaded into the corresponding counter.

- When the counter is enabled in single-shot output, PWM output, or single-shot PWM output mode
- When the counter underflows in successive output mode
- When the counter underflows after reaching the count value set by reload 1 register in PWM output mode

Simply because data is written to the reload 0 register does not mean that the counter is loaded with the data.

10.6.9 TOM Reload 1 Registers

■ **TOM0_0 Reload 1 Register (TOM00RL1)** **<Address: H'0080 0C94>**
■ **TOM0_1 Reload 1 Register (TOM01RL1)** **<Address: H'0080 0C9C>**
■ **TOM0_2 Reload 1 Register (TOM02RL1)** **<Address: H'0080 0CA4>**
■ **TOM0_3 Reload 1 Register (TOM03RL1)** **<Address: H'0080 0CAC>**
■ **TOM0_4 Reload 1 Register (TOM04RL1)** **<Address: H'0080 0CB4>**
■ **TOM0_5 Reload 1 Register (TOM05RL1)** **<Address: H'0080 0CBC>**
■ **TOM0_6 Reload 1 Register (TOM06RL1)** **<Address: H'0080 0CC4>**
■ **TOM0_7 Reload 1 Register (TOM07RL1)** **<Address: H'0080 0CCC>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TOM00RL1-TOM07RL1	16-bit reload register value	○	○

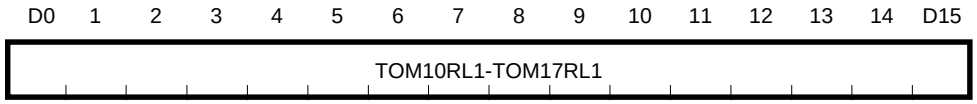
Note: These registers must always be accessed in halfwords.

The TOM0 Reload 1 Registers are used to reload the TOM0 Counter Registers (TOM00CT through TOM07CT) with data.
The following shows the timing at which the content of the reload 1 register is loaded into the corresponding counter.

- When the counter underflows after reaching the count value set by reload 0 register in PWM output or single-shot PWM output mode

Simply because data is written to the reload 1 register does not mean that the counter is loaded with the data.

■ **TOM1_0 Reload 1 Register (TOM10RL1)** **<Address: H'0080 0D94>**
■ **TOM1_1 Reload 1 Register (TOM11RL1)** **<Address: H'0080 0D9C>**
■ **TOM1_2 Reload 1 Register (TOM12RL1)** **<Address: H'0080 0DA4>**
■ **TOM1_3 Reload 1 Register (TOM13RL1)** **<Address: H'0080 0DAC>**
■ **TOM1_4 Reload 1 Register (TOM14RL1)** **<Address: H'0080 0DB4>**
■ **TOM1_5 Reload 1 Register (TOM15RL1)** **<Address: H'0080 0DBC>**
■ **TOM1_6 Reload 1 Register (TOM16RL1)** **<Address: H'0080 0DC4>**
■ **TOM1_7 Reload 1 Register (TOM17RL1)** **<Address: H'0080 0DCC>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	TOM10RL1-TOM17RL1	16-bit reload register value	○	○

Note: These registers must always be accessed in halfwords.

The TOM1 Reload 1 Registers are used to reload the TOM1 Counter Registers (TOM10CT through TOM17CT) with data.
The following shows the timing at which the content of the reload 1 register is loaded into the corresponding counter.

- When the counter underflows after reaching the count value set by reload 0 register in PWM output or single-shot PWM output mode

Simply because data is written to the reload 1 register does not mean that the counter is loaded with the data.

10.6.10 TOM Enable Protect Registers

■ **TOM0 Enable Protect Register (TOM0PRO)** **<Address: H'0080 0CDD>**

D8	9	10	11	12	13	14	D15
TOM00PRO	TOM01PRO	TOM02PRO	TOM03PRO	TOM04PRO	TOM05PRO	TOM06PRO	TOM07PRO

<When reset: H'00>

D	Bit Name	Function	R	W
8	TOM00PRO (TOM0_0 enable protect)	0: Enables rewriting 1: Disables rewriting	○	○
9	TOM01PRO (TOM0_1 enable protect)			
10	TOM02PRO (TOM0_2 enable protect)			
11	TOM03PRO (TOM0_3 enable protect)			
12	TOM04PRO (TOM0_4 enable protect)			
13	TOM05PRO (TOM0_5 enable protect)			
14	TOM06PRO (TOM0_6 enable protect)			
15	TOM07PRO (TOM0_7 enable protect)			

The TOM0 Enable Protect Register is used to control rewriting of the TOM0 count enable bit described in the next page by disabling or enabling the rewrite.

■ **TOM1 Enable Protect Register (TOM1PRO)** **<Address: H'0080 0DDD>**

D8	9	10	11	12	13	14	D15
TOM10PRO	TOM11PRO	TOM12PRO	TOM13PRO	TOM14PRO	TOM15PRO	TOM16PRO	TOM17PRO

<When reset: H'00>

D	Bit Name	Function	R	W
8	TOM10PRO (TOM1_0 enable protect)	0: Enables rewriting 1: Disables rewriting	○	○
9	TOM11PRO (TOM1_1 enable protect)			
10	TOM12PRO (TOM1_2 enable protect)			
11	TOM13PRO (TOM1_3 enable protect)			
12	TOM14PRO (TOM1_4 enable protect)			
13	TOM15PRO (TOM1_5 enable protect)			
14	TOM16PRO (TOM1_6 enable protect)			
15	TOM17PRO (TOM1_7 enable protect)			

The TOM1 Enable Protect Register is used to control rewriting of the TOM1 count enable bit described in the next page by disabling or enabling the rewrite.

10.6.11 TOM Count Enable Registers

■ **TOM0 Count Enable Register (TOM0CEN)** **<Address: H'0080 0CDF>**

D8	9	10	11	12	13	14	D15
TOM00CEN	TOM01CEN	TOM02CEN	TOM03CEN	TOM04CEN	TOM05CEN	TOM06CEN	TOM07CEN

<When reset: H'00>

D	Bit Name	Function	R	W
8	TOM00CEN (TOM0_0 count enable)	0: Stops counting 1: Enables counting	○	○
9	TOM01CEN (TOM0_1 count enable)			
10	TOM02CEN (TOM0_2 count enable)			
11	TOM03CEN (TOM0_3 count enable)			
12	TOM04CEN (TOM0_4 count enable)			
13	TOM05CEN (TOM0_5 count enable)			
14	TOM06CEN (TOM0_6 count enable)			
15	TOM07CEN (TOM0_7 count enable)			

The TOM0 Count Enable Register controls operation of the TOM0 counters. To enable any counter in software, enable the corresponding TOM0 Enable Protect Register for write and set the count enable bit to 1.
To stop the counter, enable the TOM0 Enable Protect Register for write and set the count enable bit to 0.

In single-shot output and single-shot PWM output modes, the count enable bit is automatically reset to 0 when the counter stops after reaching the minimum count. Therefore, the TOM0 Count Enable Register when read serves as a status register, indicating the operating status of the counter (operating or idle).

■ **TOM1 Count Enable Register (TOM1CEN)** **<Address: H'0080 0DDF>**

D8	9	10	11	12	13	14	D15
TOM10CEN	TOM11CEN	TOM12CEN	TOM13CEN	TOM14CEN	TOM15CEN	TOM16CEN	TOM17CEN

<When reset: H'00>

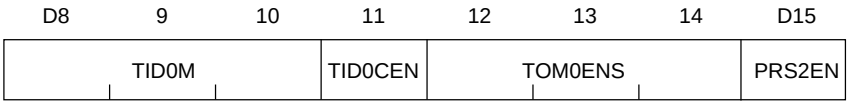
D	Bit Name	Function	R	W
8	TOM10CEN (TOM1_0 count enable)	0: Stops counting 1: Enables counting	○	○
9	TOM11CEN (TOM1_1 count enable)			
10	TOM12CEN (TOM1_2 count enable)			
11	TOM13CEN (TOM1_3 count enable)			
12	TOM14CEN (TOM1_4 count enable)			
13	TOM15CEN (TOM1_5 count enable)			
14	TOM16CEN (TOM1_6 count enable)			
15	TOM17CEN (TOM1_7 count enable)			

The TOM1 Count Enable Register controls operation of the TOM1 counters. To enable any counter in software, enable the corresponding TOM1 Enable Protect Register for write and set the count enable bit to 1.
To stop the counter, enable the TOM1 Enable Protect Register for write and set the count enable bit to 0.

In single-shot output and single-shot PWM output modes, the count enable bit is automatically reset to 0 when the counter stops after reaching the minimum count. Therefore, the TOM1 Count Enable Register when read serves as a status register, indicating the operating status of the counter (operating or idle).

10.6.12 TID Control & Prescaler Enable Registers

■ **TID0 Control & Prescaler 2 Enable Register (TID0PRS2EN)** **<Address: H'0080 0CD1>**



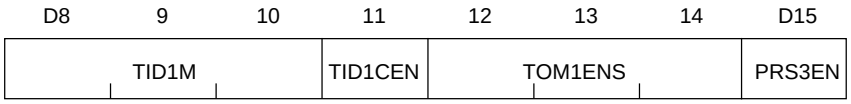
<When reset: H'00>

D	Bit Name	Function	R	W
8-10	TID0M (Selects TID0 operation mode)	X0X: Fixed period count mode X10: Multiply-by-4 event count mode 011: Event count mode 111: Up/down event count mode	○	○
11	TID0CEN (TID0 count enable)	0: Stops counting 1: Starts counting	○	○
12-14	TOM0ENS (Selects TOM0_0-7 enable source)	X0X: Disables event enable X10: TID0 output 011: TOM0_7 output 110: TID1 or TOM1_7 output 111: External input TIN18	○	○
15	PRS2EN (Prescaler 2 enable)	0: Stops counting 1: Starts counting	○	○

Note: Operation mode can only be set or changed when the counter is inactive.

The TID0 Control & Prescaler 2 Enable Register is used to select TID0 operation mode (fixed period count, event count, multiply-by-4 event count, or up/down event count mode), as well as select one of TOM0_0-7 timer enable sources and control startup of Prescaler 2.

■ **TID1 Control & Prescaler 3 Enable Register (TID1PRS3EN)** **<Address: H'0080 0DD1>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-10	TID1M (Selects TID1 operation mode)	X0X: Fixed period count mode X10: Multiply-by-4 event count mode 011: Event count mode 111: Up/down event count mode	○	○
11	TID1CEN (TID1 count enable)	0: Stops counting 1: Starts counting	○	○
12-14	TOM1ENS (Selects TOM1_0-7 enable source)	X0X: Disables event enable X10: TID1 output 011: TOM1_7 output 110: TID0 or TOM0_7 output 111: External input TIN19	○	○
15	PRS3EN (Prescaler 3 enable)	0: Stops counting 1: Starts counting	○	○

Note: Operation mode can only be set or changed when the counter is inactive.

The TID1 Control & Prescaler 3 Enable Register is used to select TID1 operation mode (fixed period count, event count, multiply-by-4 event count, or up/down event count mode), as well as select one of TOM1_0-7 timer enable sources and control startup of Prescaler 3.

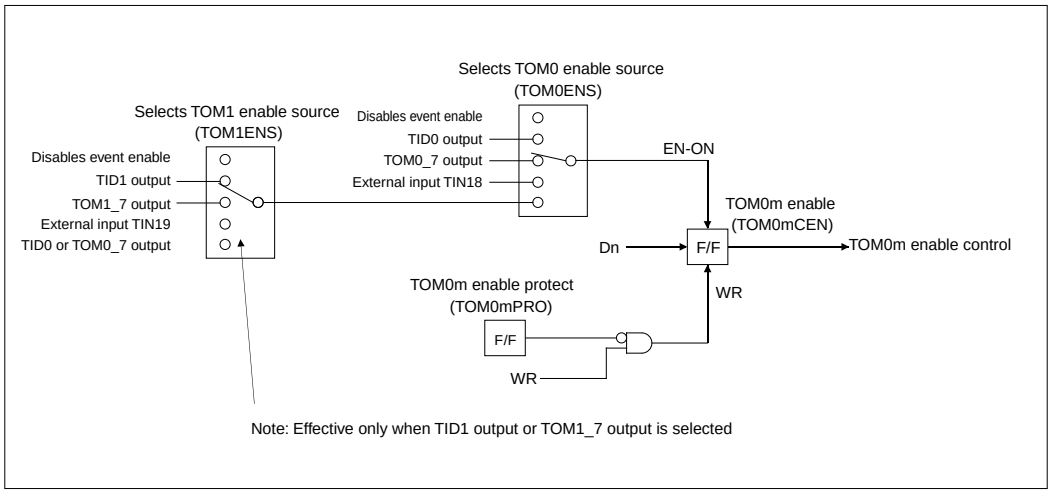


Figure 10.6.7 Configuration of the TOM0 Enable Circuit

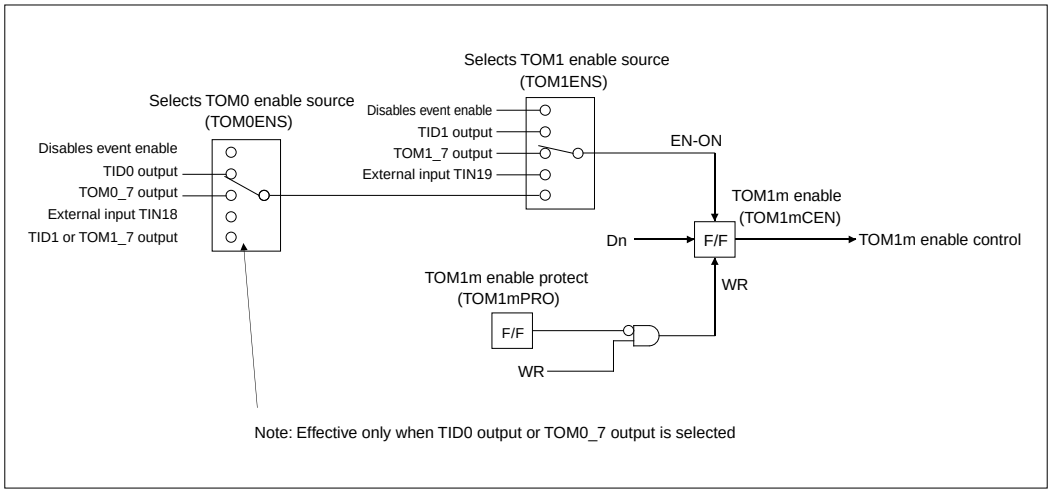


Figure 10.6.8 Configuration of the TOM1 Enable Circuit

10.6.13 Operation of TOM in PWM Output Mode

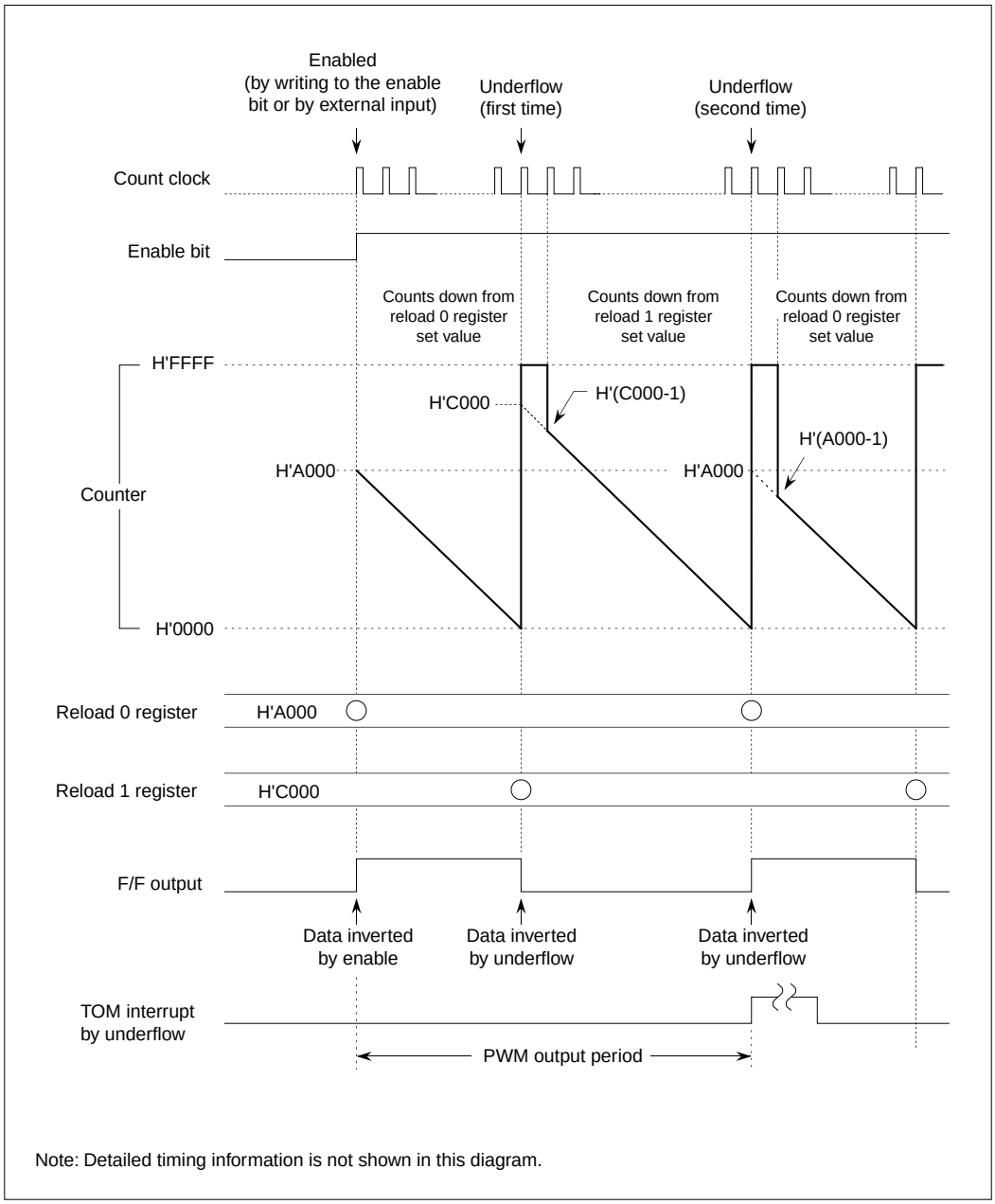
(1) Outline of PWM Output Mode for TOM

In PWM output mode, the TOM uses two reload registers to generate a waveform with any duty cycle.

When the timer is enabled after setting the initial value in the reload 0 and the reload 1 registers, the counter is loaded with the value of the reload 0 register synchronously with the count clock, from which it starts counting down. The first time the counter underflows after reaching the minimum count, it is loaded with the content of the reload 1 register. Thereafter, the counter is alternately reloaded by the reload 0 and the reload 1 registers each time it underflows. The reload 0 register set value + 1 and the reload 1 register set value + 1 respectively are the valid count values.
To stop the timer, write to the enable bit to disable counting, and the timer immediately stops (not synchronized to the PWM output period).

The F/F output waveform in PWM output mode is inverted (F/F output changes level from low to high or vice versa) when the counter starts counting and each time it underflows.
An interrupt and a DMA transfer request can be generated the second, the fourth, and other even-numbered times the counter underflows after being enabled.

PWM output mode for TOM does not have a correction function.



Note: Detailed timing information is not shown in this diagram.

Figure 10.6.9 Typical Operation of TOM in PWM Output Mode

(2) Updating of TOM reload registers in PWM mode

In PWM output mode, when the timer is idle, the reload 0 and reload 1 registers are updated simultaneously when data are written to the registers. However, when the timer is operating, the reload 1 register is updated by updating the reload 0 register. The values obtained by reading the reload 0 and reload 1 registers are always the data written to the registers.

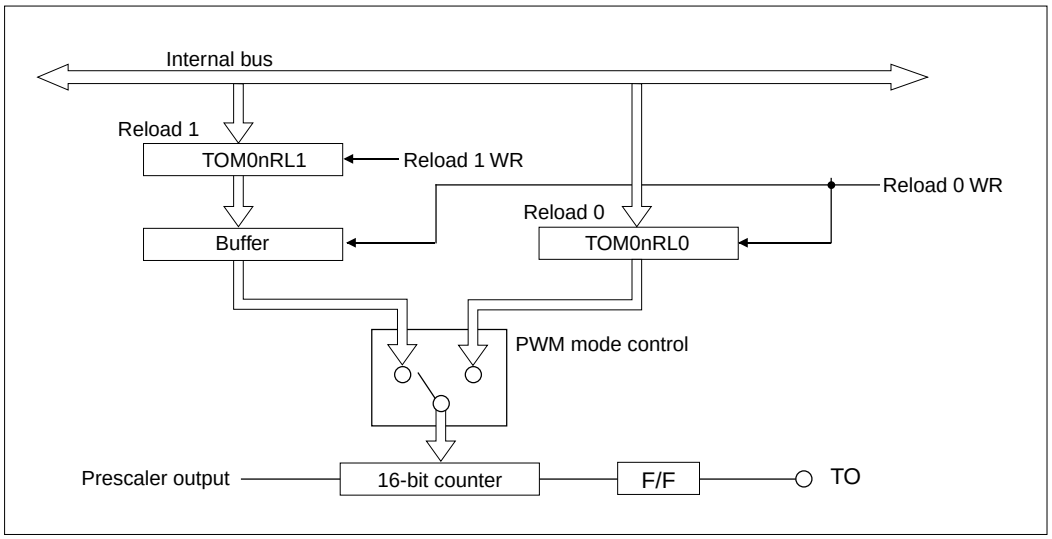


Figure 10.6.10 PWM Circuit Diagram

When the reload 0 and reload 1 registers need to be rewritten while the timer is operating, rewrite the reload 1 register first and then the reload 0 register. As a result, the reload 0 and reload 1 registers both are updated synchronously with PWM period.
Normally, this can be accomplished at a time by accessing the 32-bit address space wordwise beginning with the reload 1 register address. (The reload 1 and then the reload 0 registers are automatically written to in succession.)

If the reload registers are updated in reverse by updating the reload 0 and then the reload 1 registers, only the reload 0 register is updated. The values obtained by reading the reload 0 and reload 1 registers are always the data written to the registers, and not the reload values actually being used.

While rewriting the PWM period, if the PWM period expires before writing to the reload 0 register, the PWM period is not immediately updated and the updating is reflected in the next period.

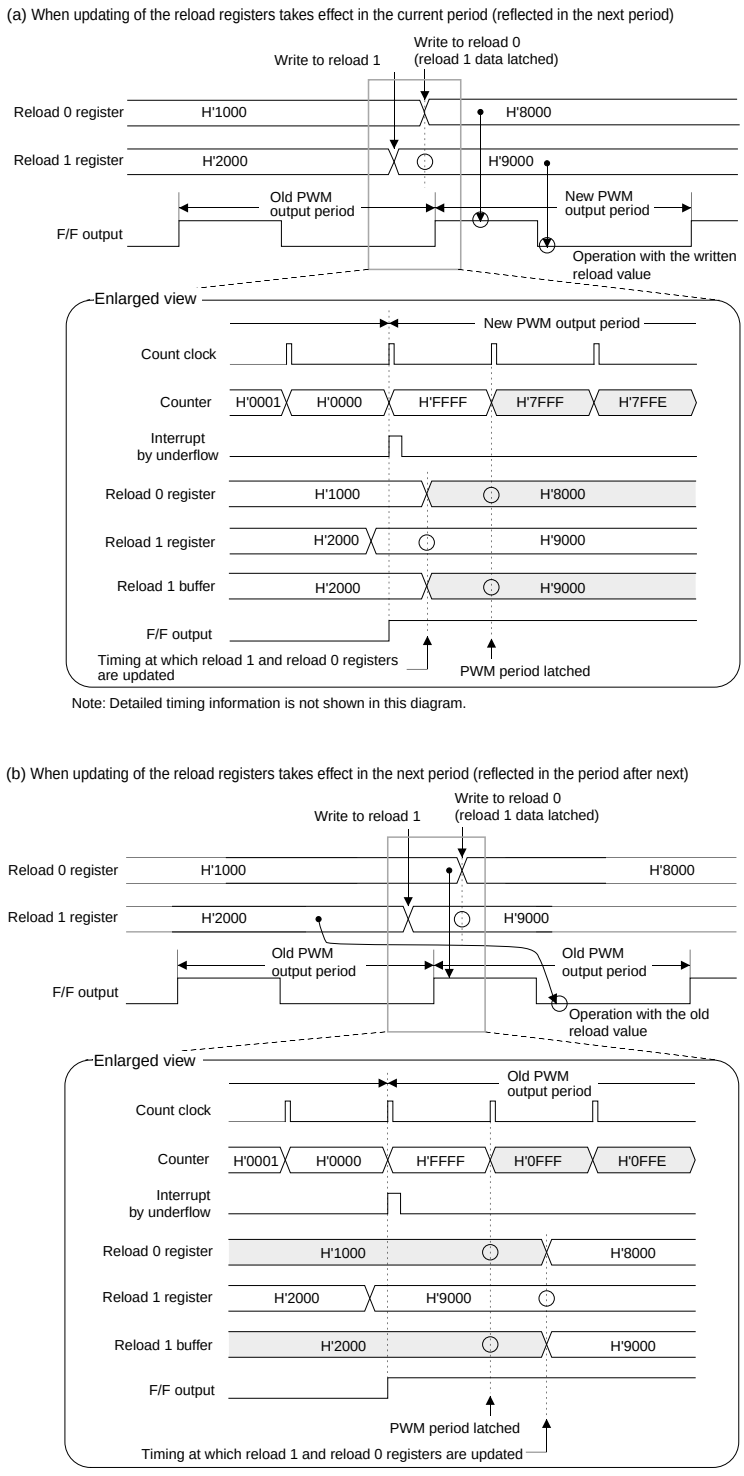


Figure 10.6.11 Updating Reload 0 and Reload 1 Registers in PWM Output Mode

10.6.14 Operation of TOM in Single-shot Output Mode (without Correction Function)

(1) Outline of single-shot output mode for TOM

In single-shot output mode, the TOM generates a pulse in duration of the reload 0 register set value + 1 only once and then stops.

When the timer is enabled after setting the reload 0 register, the counter is loaded with the content of the reload 0 register synchronously with the count clock, from which it starts counting. The counter counts down and when it underflows after reaching the minimum count, it stops.

The F/F output waveform in single-shot output mode is inverted (F/F output changes level from low to high or vice versa) when the counter starts counting and when it underflows. In this way, the TOM generates a single-shot pulse in duration of the reload 0 register set value + 1 only once. An interrupt and a DMA transfer request can be generated when the counter underflows.

The count value is the reload 0 register set value + 1.

(2) Precautions on using TOM in single-shot output mode

The following describes precautions to be observed when using the TOM in single-shot output mode.

- If the counter stops upon underflowing while it is enabled by external input in the same clock cycle, the former has priority so that the counter stops.
- If the counter stops upon underflowing while it is enabled for counting by writing to the enable bit in the same clock cycle, the latter has priority so that the counter is enabled for counting.
- If the counter is enabled by external input while it is disabled against counting by writing to the enable bit in the same clock cycle, the latter has priority so that the counter is disabled against counting.
- Because the internal circuit operation is synchronized to the count clock (prescaler output), a prescaler equivalent delay time exists before the F/F starts operation after being enabled.

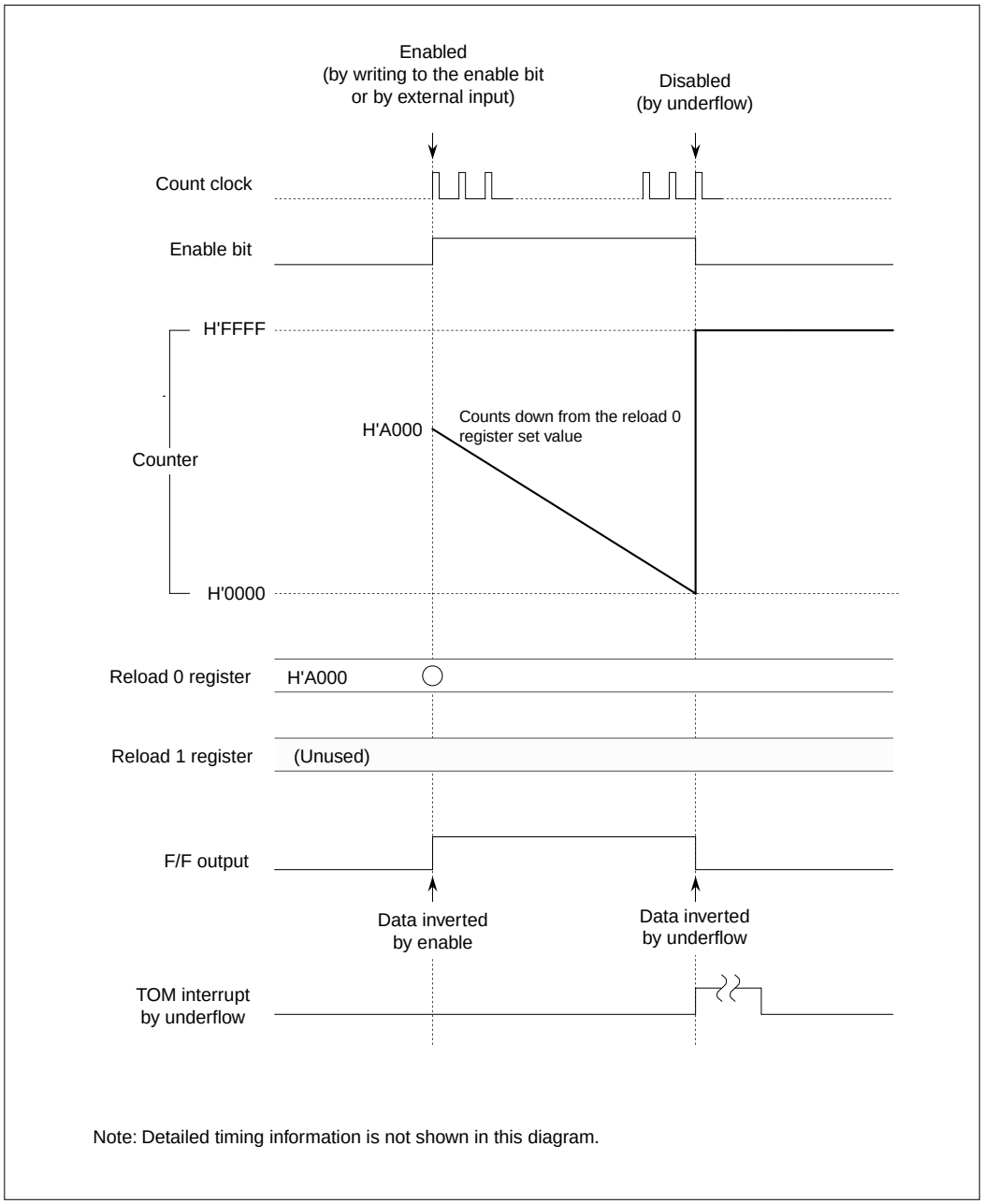


Figure 10.6.12 Typical Operation of TOM in Single-shot Output Mode (without Correction Function)

10.6.15 Operation of TOM in Single-shot PWM Output Mode (without Correction Function)

(1) Outline of single-shot PWM output mode for TOM

In single-shot PWM output mode, the TOM uses two reload registers to generate a waveform with any duty cycle only once.

When the timer is enabled after setting the initial value in the reload 0 and the reload 1 registers, the counter is loaded with the value of the reload 0 register synchronously with the count clock, from which it starts counting down. The first time the counter underflows after reaching the minimum count, it is loaded with the content of the reload 1 register. When the counter underflows second time, it stops counting. The reload 0 register set value + 1 and the reload 1 register set value + 1 respectively are the valid count values.
To stop the timer in software, write to the enable bit to disable counting, and the timer immediately stops (not synchronized to the PWM output period).

The F/F output waveform in single-shot PWM output mode is inverted (F/F output changes level from low to high or vice versa) each time the counter underflows. (Unlike in PWM output mode, the F/F output waveform is not inverted when the counter is enabled.)
An interrupt and a DMA transfer request can be generated the second time the counter underflows after being enabled.

Single-shot PWM output mode for TOM does not have a correction function.

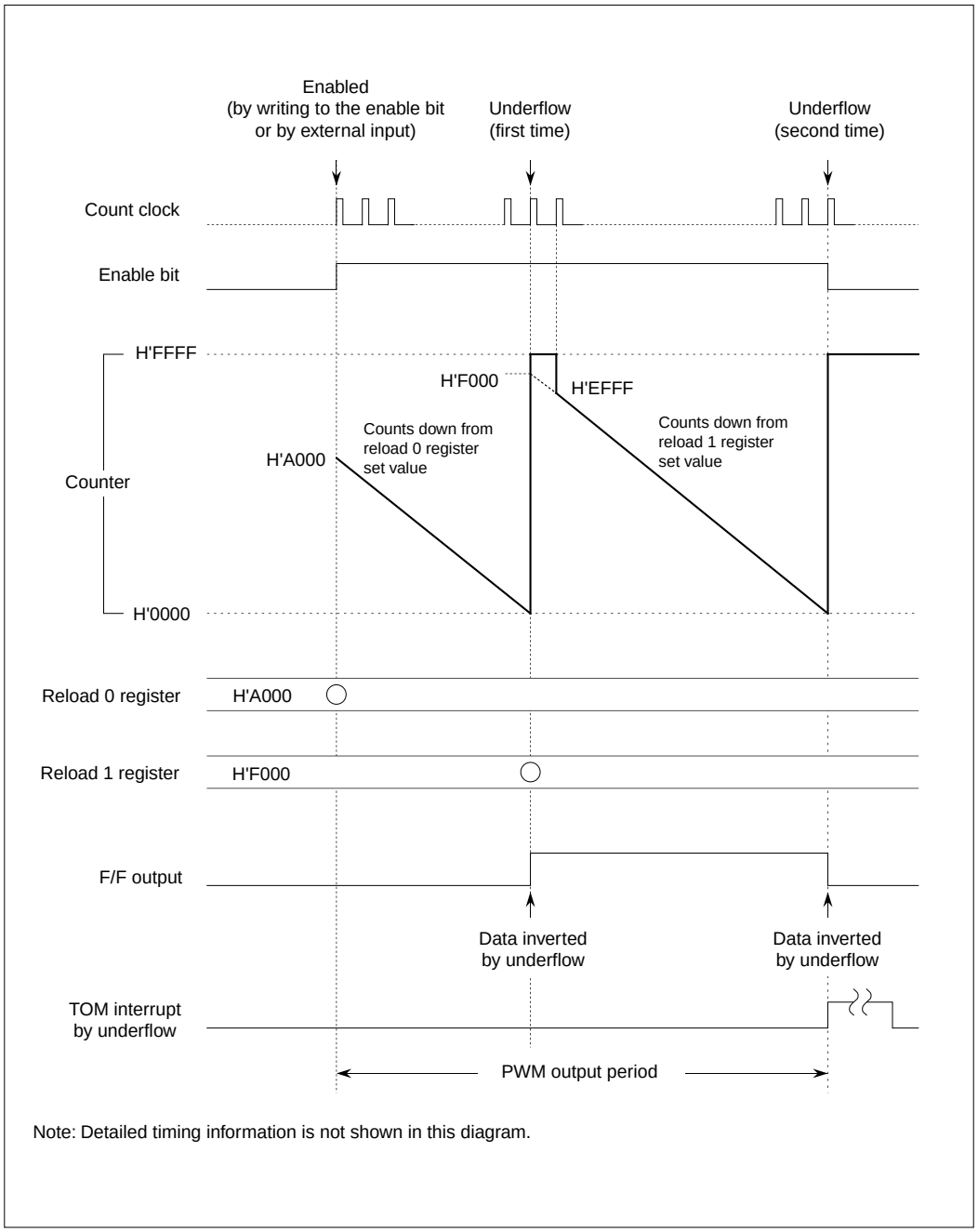


Figure 10.6.13 Typical Operation of TOM in Single-shot PWM Output Mode (without Correction Function)

10.6.16 Operation of TOM in Successive Output Mode (without Correction Function)

(1) Outline of successive output mode for TOM

In successive output mode, the counter counts down from its set value and upon underflowing, it is loaded with the value of the reload 0 register. Therefore, this operation is repeated each time the counter underflows, thereby successively generating a pulse in duration of the reload 0 register set value + 1.

When the timer is enabled after setting the counter and the reload 0 register, the counter starts counting down from its set value synchronously with the count clock.
When the counter underflows after reaching the minimum count, it is loaded with the content of the reload 0 register and starts counting again. This is repeated each time the counter underflows. To stop the counter, write to the enable bit in software to disable counting.

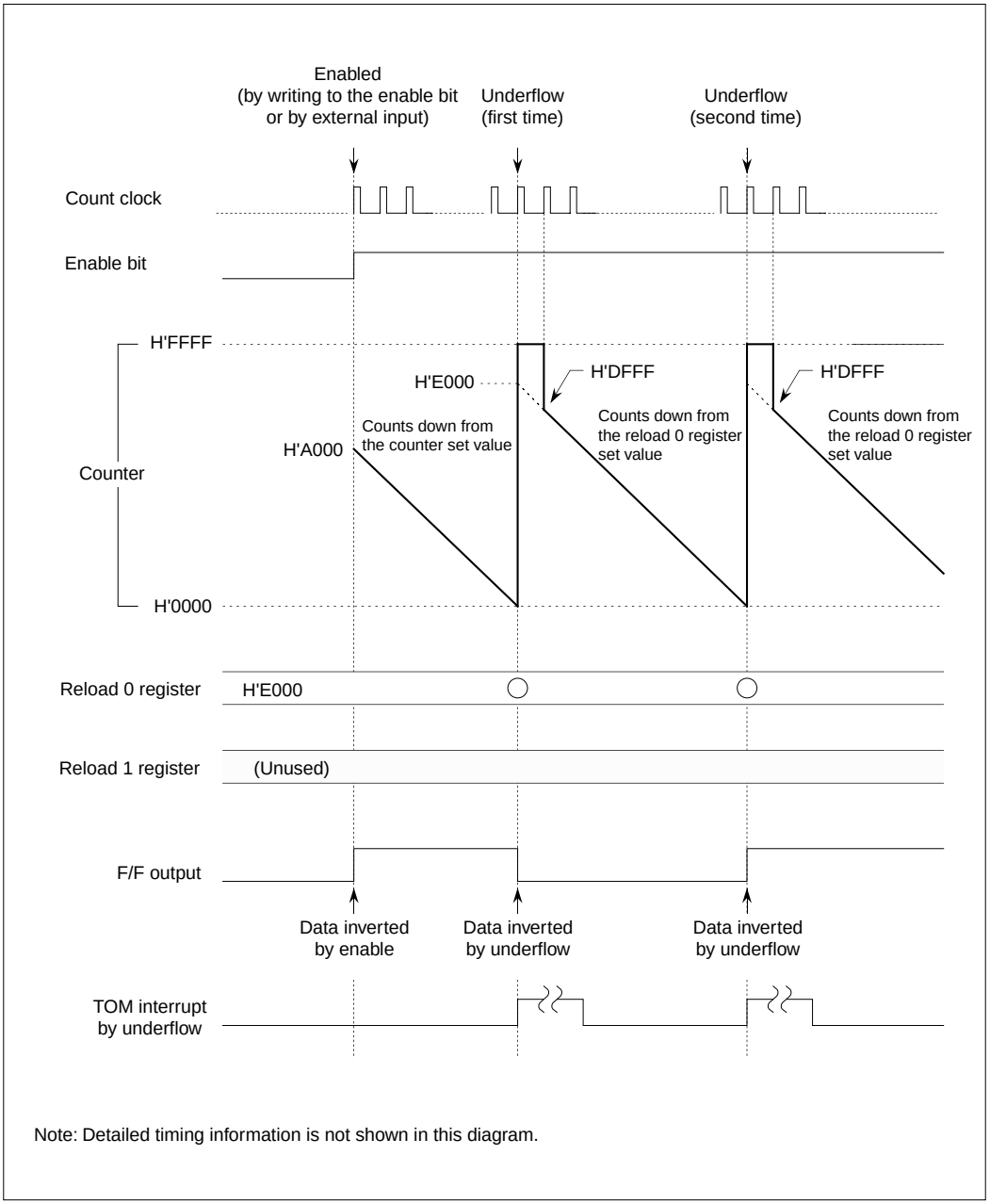
The F/F output waveform in successive output mode is inverted (F/F output changes level from low to high or vice versa) when the counter starts counting and when it underflows, thereby successively generating a pulse waveform until the count stops.
An interrupt and a DMA transfer request can be generated each time the counter underflows.

The counter set value + 1 and the reload 0 register set value + 1 are the valid count values.

(2) Precautions on using TOM in successive output mode

The following describes precautions to be observed when using the TOM in successive output mode.

- If the counter is enabled by external input while it is disabled against counting by writing to the enable bit in the same clock cycle, the latter has priority so that the counter is disabled against counting.
- When the counter is read immediately after being reloaded pursuant to underflowing, it temporarily shows the value H'FFFF which, however, immediately changes to the "reload value - 1" in the next clock cycle.
- Because the internal circuit operation is synchronized to the count clock (prescaler output), a prescaler equivalent delay time exists before the F/F starts operation after being enabled.



Note: Detailed timing information is not shown in this diagram.

Figure 10.6.14 Typical Operation of TOM in Successive Output Mode (without Correction Function)

10.6.17 TOM Output Disable Function

The TOM has the function to disable PWM outputs from TOM0_0-TOM0_5 and TOM1_0-TOM1_5 timers. The circuit configuration of this PWM output disable function is shown in Figure 10.6.15. There are following three methods to disable PWM outputs.

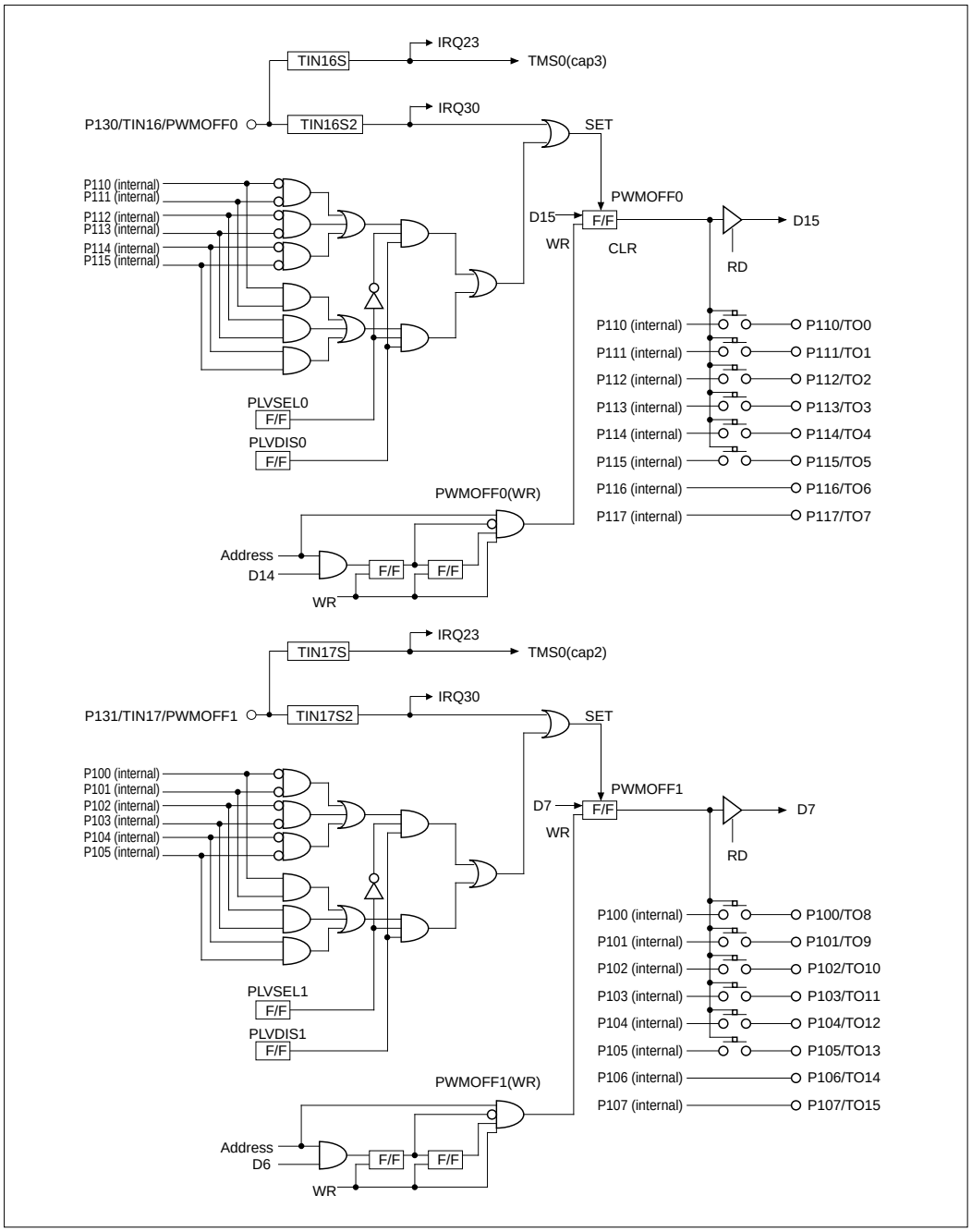


Figure 10.6.15 Circuit Configuration of PWM Output Disable Function

(1) Disabling PWM outputs with the signal entered from the external pin (TIN16 or TIN17)

The input signal on the external pin TIN16 may be used to disable PWM outputs of TOM0_0-TOM0_5 timers from being output to the corresponding ports P110-P115. Similarly, the input signal on the external pin TIN17 may be used to disable PWM outputs of TOM1_0-TOM1_5 timers from being output to the corresponding ports P100-P105.
To disable PWM outputs using the input signal on the external pin TIN16 or TIN17, the TIN Input Processing Control Register 4 (TINCR4) must be set up following the procedure described below.

[Setting up the TINCR4 Register]

- To disable PWM outputs using TIN17
  ① Set the TINCR4 Register D0 bit to 1.
  ② Subsequently after writing in ① above, set the D0 bit to 0 and the D1-D3 bits to "000," "001," "010," "011," "10X," or "11X."

Note: If a write cycle for any other area occurs between ① and ②, the value set in the D1-D3 bits has no effect.

- To disable PWM outputs using TIN16
  ① Set the TINCR4 Register D4 bit to 1.
  ② Subsequently after writing in ① above, set the D4 bit to 0 and the D5-D7 bits to "000," "001," "010," "011," "10X," or "11X."

Note: If a write cycle for any other area occurs between ① and ②, the value set in the D5-D7 bits has no effect.

(2) Disabling PWM outputs with the pin level on ports P100-P105 or P110-P115

The pin level on ports P100-P105 (high or low) may be used to disable PWM outputs of TOM1_0-TOM1_5 timers from being output to the corresponding ports P100-P105. Similarly, the pin level on ports P110-P115 (high or low) may be used to disable PWM outputs of TOM0_0-TOM0_5 timers from being output to the corresponding ports P110-P115.
To disable PWM outputs using the pin level on said ports, the Port P100-P105 Output Disable Control Register (PLVCNT1) or Port P110-P115 Output Disable Control Register (PLVCNT0) must be set up following the procedure described below.

- To disable PWM outputs using the port P100-P105 level
  ① Set the PLVCNT1 Register D6 bit (PLVSEL1) to 1 or 0 to select the port level (high or low) at which to disable PWM outputs.
  ② Set the PLVCNT1 Register D7 bit (PLVDIS1) to 1 (to validate output disable setting).

- To disable PWM outputs using the port P110-P115 level
  ① Set the PLVCNT0 Register D14 bit (PLVSEL0) to 1 or 0 to select the port level (high or low) at which to disable PWM outputs.
  ② Set the PLVCNT0 Register D15 bit (PLVDIS0) to 1 (to validate output disable setting).

(3) Disabling PWM outputs with the Port Output Disable Register (PWMOFF0 or PWMOFF1)

The Port P110-P115 Output Disable Register (PWMOFF0) may be used to disable PWM outputs of TOM0_0-TOM0_5 timers from being output to the corresponding ports P110-P115. Similarly, the Port P100-P105 Output Disable Register (PWMOFF1) may be used to disable PWM outputs of TOM1_0-TOM1_5 timers from being output to the corresponding ports P100-P105.
To disable PWM outputs using the Port P110-P115 Output Disable Register (PWMOFF0) or Port P100-P105 Output Disable Register (PWMOFF1), the respective registers must be set up following the procedure described below.

- To disable PWM outputs using the Port P110-P115 Output Disable Register (PWMOFF0)
  ① Set the PWMOFF0 Register D14 bit to 1.
  ② Subsequently after writing in ① above, set the D14 to 0 and the D15 bit to 1, respectively.

Note: If a write cycle for any other area occurs between ① and ②, the value set in the D15 bit has no effect.

- To disable PWM outputs using the Port P100-P105 Output Disable Register (PWMOFF1)
  ① Set the PWMOFF1 Register D6 bit to 1.
  ② Subsequently after writing in ① above, set the D6 to 0 and the D7 bit to 1, respectively.

Note: If a write cycle for any other area occurs between ① and ②, the value set in the D7 bit has no effect.

10.6.18 Example for Using the TOM in Motor Control Applications

The microcomputer contains two blocks of TOM timers which are designed to reduce software load during motor control. This section explains an example for using TOM0 in motor control applications.
The three-phase motor control waveforms start TOM0 according to the TOM0 startup timing at a fixed period of 20 kHz generated by TID0. By using the TOM0's internal single-shot PWM function, the output waveform can easily be configured because the waveform data only needs to be stored at the necessary rewrite timing. The H and L transistor short-circuiting prevention time can be accomplished by changing the TOM0 setup time in software. Up to 8-phase motors can be controlled by a combined use of TID and TOM.

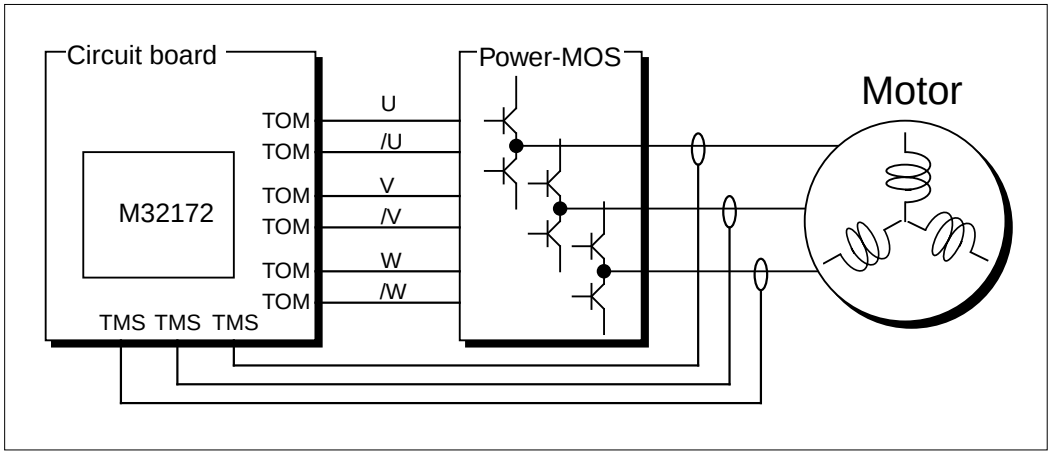


Figure 10.6.16 System Configuration Diagram

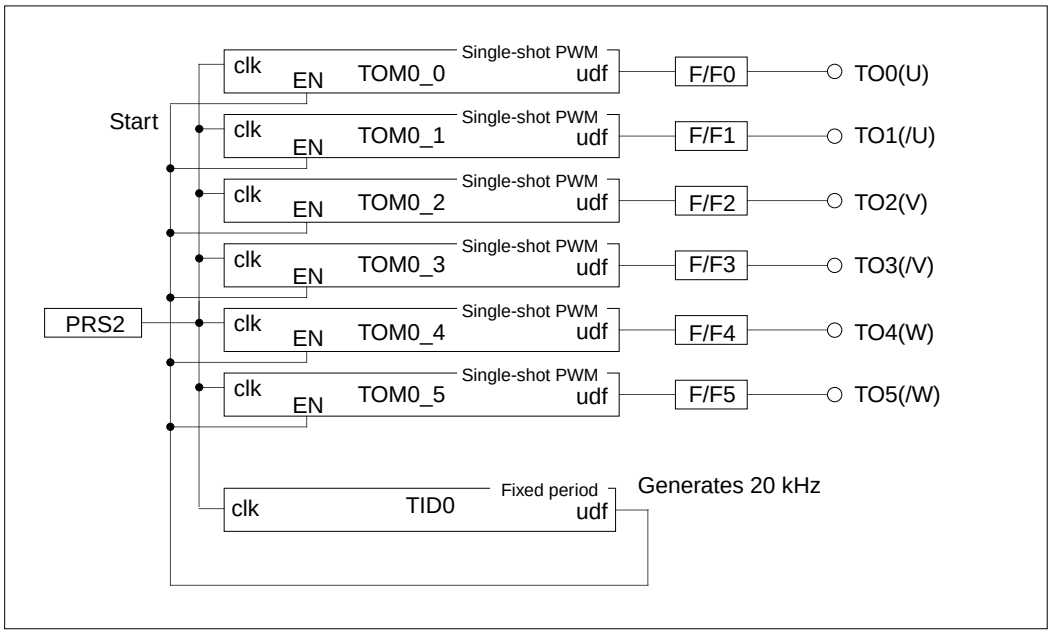


Figure 10.6.17 Interconnecting Timers for Three-phase Motor Control

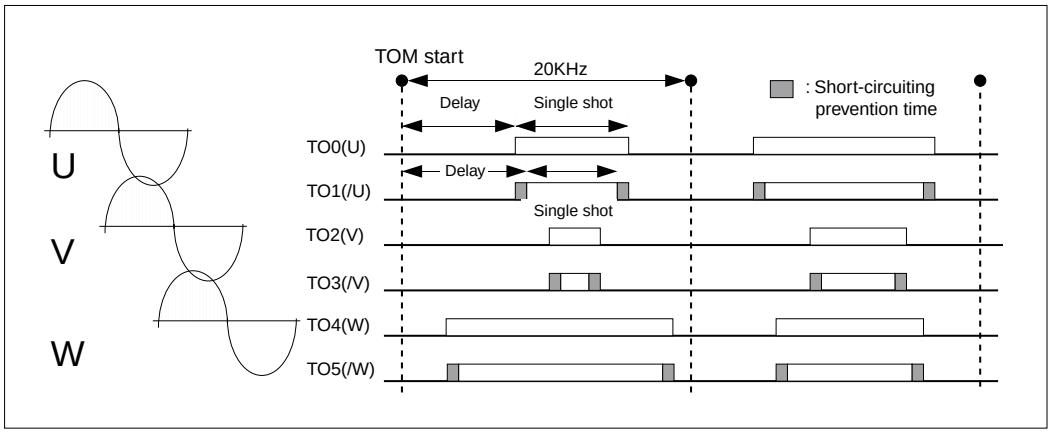


Figure 10.6.18 Conceptual Diagram of Motor Control

*** This is a blank page ***

CHAPTER 11

A-D CONVERTERS

11.1 Outline of the A-D Converters

The 32172/32173 contains two 10-bit resolution A-D converters based on successive approximation method (A-D0 and A-D1 Converters). The A-D0 Converter has eight channels of dedicated analog input pins (AD0IN0-AD0IN7) and eight other channels of analog input pins which are shared with input/output ports or internal peripheral I/O input/output pins (AD0IN8-AD0IN15), for a total of 16 channels. The A-D1 Converter has four channels of dedicated analog input pins (AD1IN0-AD1IN3) and 12 other channels of analog input pins which are shared with input/output ports or internal peripheral I/O input/output pins (AD1IN4-AD1IN15), for a total of 16 channels. For both converters, the A-D converted value can be read out in either 8 bits or 10 bits.

The A-D converters have the following conversion modes and operation modes.

(1) Conversion modes

- A-D conversion mode: This is an ordinary mode, in which analog input voltages are A-D converted.
- Comparator mode (Note): In this mode, the set comparison voltage and the analog input voltage are compared to determine which is larger or smaller than the other (single mode only).

(2) Operation modes

- Single mode: The analog input voltage on one channel is A-D converted once or comparated (note).
- Scan mode: Analog input voltages on multiple channels are sequentially A-D converted.

(3) Types of scan modes

- Single-shot scan mode: Scan operation is performed only one cycle.
- Continuous scan mode: Scan operation is performed repeatedly until halted.

(4) Special operation modes

- Forcibly execute single mode during scan mode operation:
  Single mode conversion is forcibly executed during scan operation.
- Start scan mode after executing single mode:
  Scan operation is started from single mode in succession.
- Restart conversion:
  A-D convert operation under way is restarted in single or scan mode.

The A-D conversion and comparate speed can be selected from four speeds available: low speed (normal or twice normal) or high speed (normal or twice normal). Also, an A-D conversion interrupt request or DMA transfer request can be generated when the A-D conversion, comparate, or single-shot scan is finished, or each time one cycle of continuous scan mode is finished.

Note: The comparison performed internally by the A-D converter which is of the successive approximation type and its operation in comparator mode where the A-D converter is used as a comparator are different. In this manual, therefore, operation in comparator mode is referred to as "comparate" in order to discriminate it against ordinary comparison.

The A-D converters are outlined in Table 11.1.1. A block diagram of the A-D converters are shown in Figures 11.1.1 and 11.1.2, respectively.

Table 11.1.1 Outline of the A-D Converters (1/2)

Item	Content		
Analog input	16 channels x 2 A-D0 Converter : 8 channels of dedicated analog input pins 8 channels of analog input pins shared with input/output ports or peripheral I/O pins A-D1 Converter : 4 channels of dedicated analog input pins 12 channels of analog input pins shared with input/output ports or peripheral I/O pins		
A-D conversion method	Successive approximation method		
Resolution	10 bits (Conversion results read out in 8 or 10 bits)		
Absolute accuracy (Note 1) (Conditions: Ta = -40 ~ +125°C, AVCC0 = VREF0 = 5.12V)	Low speed mode	Normal	±2LSB
		Double speed	±2LSB
	High speed mode	Normal	±3LSB
		Double speed	±3LSB
Conversion mode	A-D conversion mode, comparator mode		
Operation mode	Single mode, scan mode		
Scan mode	Single-shot scan mode, continuous scan mode		
Conversion start trigger	Started in software By setting A-D conversion start bit to 1		
	Started in hardware A-D0 Converter : Input on external pin TIN16 Underflow of TOM0_6 timer Enable event on TOM0_0-7 (Note 2) Completion of A-D1 conversion A-D1 Converter : Input on external pin TIN16 Underflow of TOM0_6 timer Underflow of TOM1_6 timer Completion of A-D0 conversion		

Note 1: The rated value of conversion accuracy here is that of the microcomputer's own as a single unit which can be exhibited when the microcomputer is used in an environment where it may not be affected by the power supply wiring or noise on the board.

Note 2: There are following sources of enable event on TOM0_0-7 (for details, see Chapter 10, "Input/output Timers"):

- TID0 overflow/underflow
- TOM0_7 underflow
- Input on external pin TIN18
- TID1 overflow/underflow or TOM1_7 underflow

Table 11.1.2 Outline of the A-D Converters (2/2)

Item	Content			
Conversion speed f(BCLK): Internal peripheral clock operating frequency (Note)	During single mode	Low speed mode	Normal	299 × 1/f(BCLK)
			Double speed	173 × 1/f(BCLK)
		High speed mode	Normal	131 × 1/f(BCLK)
			Double speed	89 × 1/f(BCLK)
	During comparator mode	Low speed mode	Normal	47 × 1/f(BCLK)
			Double speed	29 × 1/f(BCLK)
		High speed mode	Normal	23 × 1/f(BCLK)
			Double speed	17 × 1/f(BCLK)
Interrupt request generation	When A-D conversion, comparate, single-shot scan, or one cycle of function continuous scan mode is finished			
DMA transfer request generation	When A-D conversion, comparate, single-shot scan, or one cycle of function continuous scan mode is finished			

Note: When the input clock (XIN) = 10 MHz, this is f(BCLK) = 20 MHz

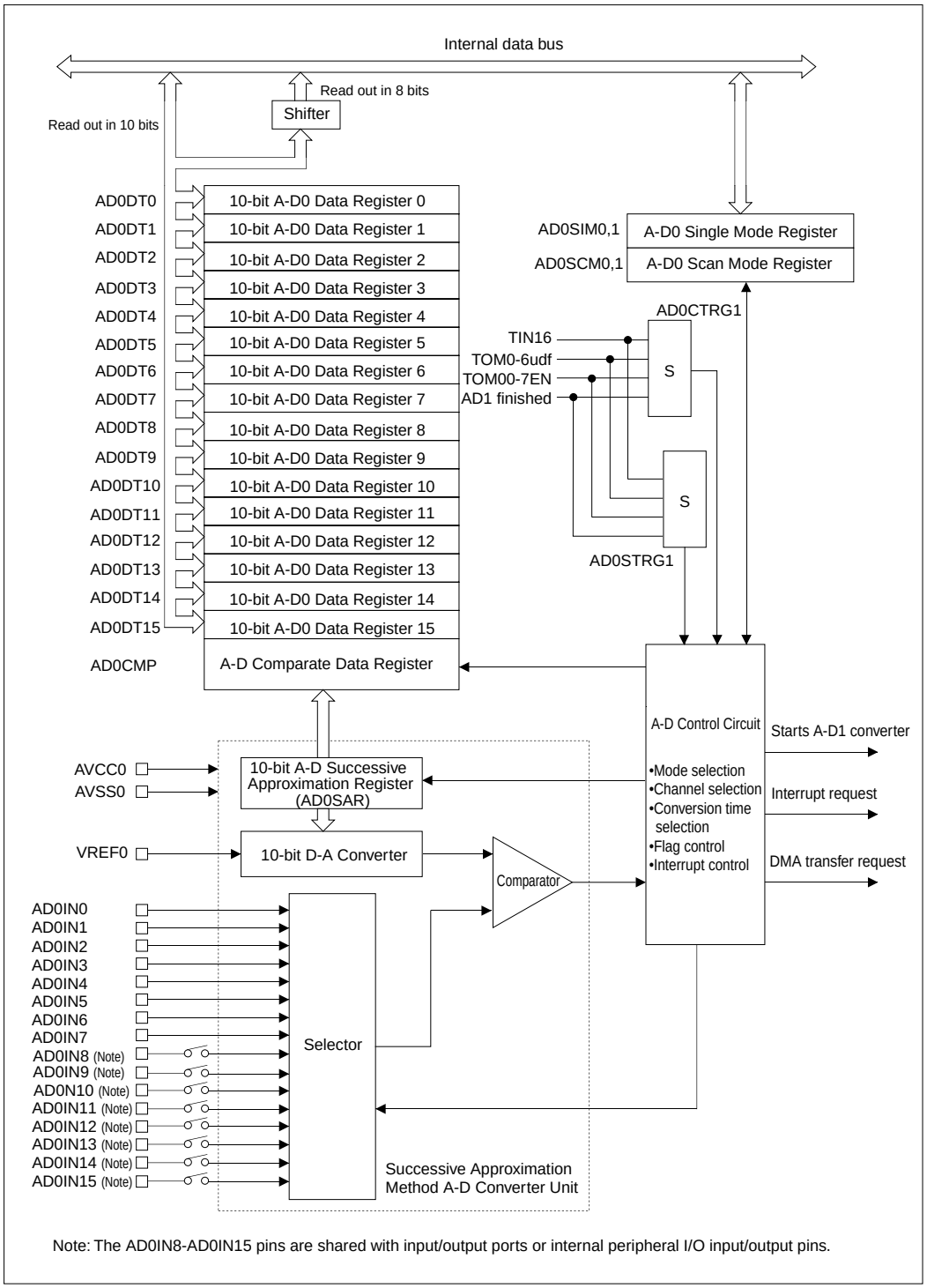
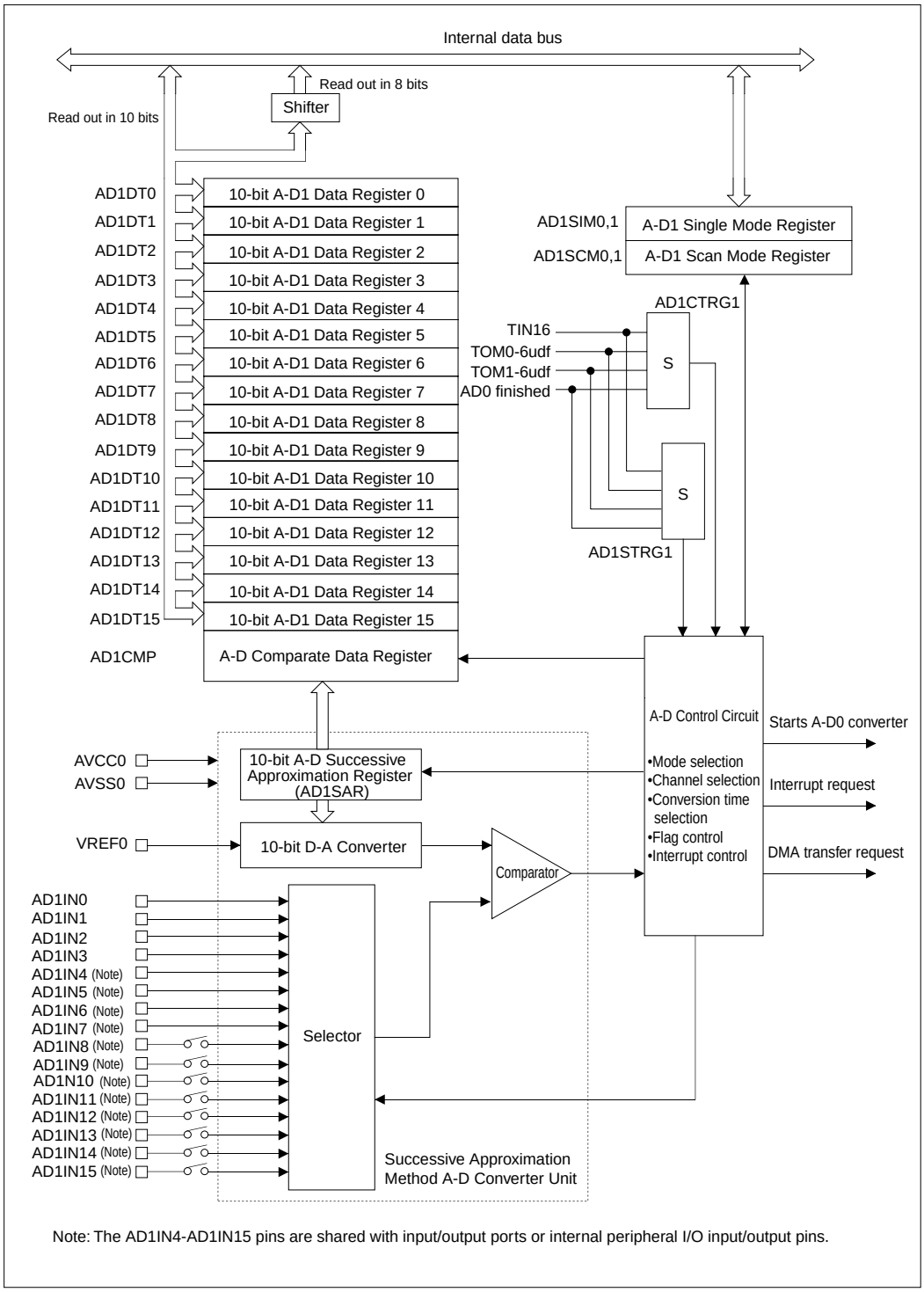


Figure 11.1.1 Block Diagram of the A-D0 Converter



Note: The AD1IN4-AD1IN15 pins are shared with input/output ports or internal peripheral I/O input/output pins.

Figure 11.1.2 Block Diagram of the A-D1 Converter

11.1.1 Conversion Modes

The A-D converters have two conversion modes: "A-D conversion mode" and "comparator mode."

(1) A-D conversion mode

In A-D conversion mode, the A-D converter converts the analog input voltage on a specified channel from analog to digital quantities.

When operating in single mode, the A-D converter performs conversion on a channel selected with the Single Mode Register 1 analog input select bits (Note).
When operating in scan mode, the A-D converter performs conversion on channels selected with Scan Mode Register 1 according to settings of Scan Mode Register 0.
The conversion result is stored in the 10-bit A-D Data Register provided for each channel converted. Also, an 8-bit conversion result can be read out from the 8-bit A-D Data Register.

An A-D conversion interrupt request or DMA transfer request can be generated when A-D conversion is finished during single mode or one cycle of scan loop is finished during scan mode.

Note: If the analog input pins AD0IN8-AD0IN15 or AD1IN6-AD1IN15 are selected with Single Mode Register 1, always be sure to set the AD Digital Input Control Register (AD0CHCON or AD1CHCON).

(2) Comparator mode

In comparator mode, the A-D converter "comparates" the analog input voltage on a specified channel with the Successive Approximation Register value and returns a flag for the result (relative magnitude of the compared values).

The Single Mode Register 1 analog input pin select bits (Note) are used to specify the channel to comparate. The flag for the comparate result (1 or 0) is set in the A-D Comparate Data Register bit corresponding to the selected channel.

An A-D conversion interrupt request or DMA transfer request can be generated when comparate operation is finished.

Note: If the analog input pins AD0IN8-AD0IN15 or AD1IN6-AD1IN15 are selected with Single Mode Register 1, always be sure to set the AD Digital Input Control Register (AD0CHCON or AD1CHCON).

11.1.2 Operation Modes

The A-D converters have two operation modes: "single mode" and "scan mode."

(1) Single mode

In single mode, the A-D converter A-D converts or comparates the analog input voltage on one selected channel only once. An A-D conversion interrupt request or DMA transfer request can be generated when A-D conversion is finished.

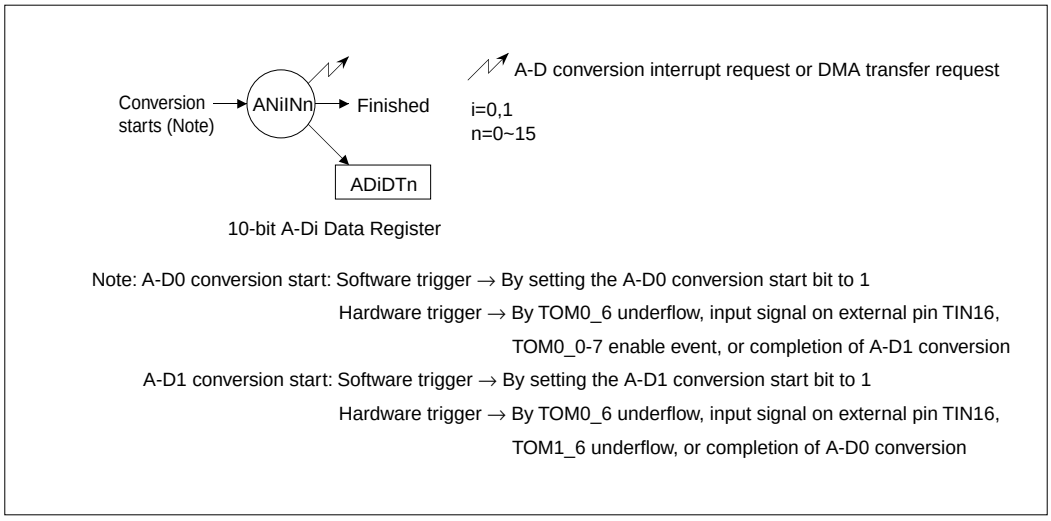


Figure 11.1.3 Single Mode Operation (A-D Conversion)

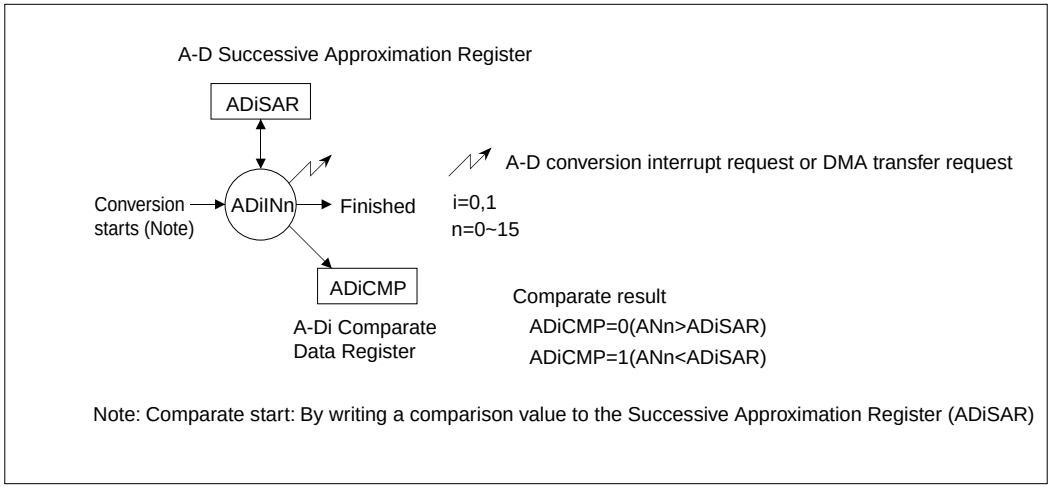


Figure 11.1.4 Single Mode Operation (Comparate)

(2) Scan mode

In scan mode, the A-D converter sequentially A-D converts analog input voltages on specified channels (channel 0-15) selected with Scan Mode Register 1 scan loop select bits, beginning with the channel ADiIN0 (i = 0, 1).

This mode further consists of "single-shot scan mode" in which A-D conversion is finished by performing one cycle of scan operation, and "continuous scan mode" in which scan operation is continued until it is halted by setting the Scan Mode Register A-D conversion stop bit to 1.

Scan Mode Register 0 is used to select between these two scan modes. Also, the Scan Mode Register 1 scan loop select bits are used to select the channels to scan. The channels selected with the scan loop select bits (channels 0-15) beginning with the channel ADiIN0 comprise the scan loop, and this group of channels are sequentially scanned beginning with the channel ADiIN0.

An A-D conversion interrupt request or DMA transfer request can be generated when one cycle of scan operation is finished.

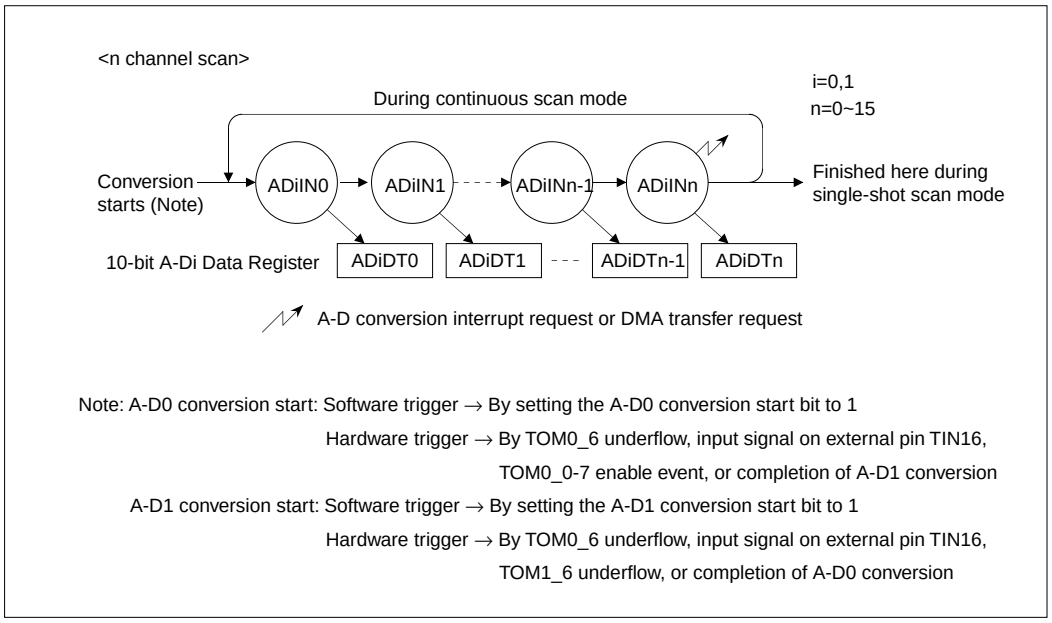


Figure 11.1.5 A-D Convert Operation in Scan Mode

Table 11.1.2 A-D Conversion Result Storage Registers during Scan Mode

Channels selected with Scan Mode Register 1	Channels to be converted in single-shot scan mode	Channels to be converted in continuous scan mode	A-D conversion result storage register
B'0000 : 0	ADiIN0	ADiIN0	10-bit A-Di Data Register 0
(ADiIN0)	Finishes	ADiIN0	10-bit A-Di Data Register 0
		⋮ (Repeated until forcibly terminated) ⋮	
B'0001 : 1	ADiIN0	ADiIN0	10-bit A-Di Data Register 0
(ADiIN1)	ADiIN1	ADiIN1	10-bit A-Di Data Register 1
	Finishes	ADiIN0	10-bit A-Di Data Register 0
		⋮ (Repeated until forcibly terminated) ⋮	
B'0010 : 2	ADiIN0	ADiIN0	10-bit A-Di Data Register 0
(ADiIN2)	ADiIN1	ADiIN1	10-bit A-Di Data Register 1
	ADiIN2	ADiIN2	10-bit A-Di Data Register 2
	Finishes	ADiIN0	10-bit A-Di Data Register 0
		⋮ (Repeated until forcibly terminated) ⋮	
B'0011 : 3	ADiIN0	ADiIN0	10-bit A-Di Data Register 0
(ADiIN3)	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
	Finishes	ADiIN0	10-bit A-Di Data Register 0
		⋮ (Repeated until forcibly terminated) ⋮	
B'XXXX : n	ADiIN0	ADiIN0	10-bit A-Di Data Register 0
(ADiINn)	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
	Finishes	ADiIN0	10-bit A-Di Data Register 0
		⋮ (Repeated until forcibly terminated) ⋮	

(i = 0, 1)

11.1.3 Special Operation Modes

(1) Forcibly execute single mode during scan mode operation

In this special operation mode, the A-D converter forcibly executes single mode conversion (A-D conversion or comparate) on a specified channel while operating in scan mode. The conversion result is stored in the 10-bit A-D Data Register of the selected channel in the case of A-D conversion mode, or in the 10-bit A-D Comparate Data Register of the selected channel in the case of comparator mode. When A-D conversion or comparate on the specified channel finishes, scan mode A-D conversion restarts beginning with the channel that has been canceled during scan.

To start single mode conversion in software during scan mode operation, select software trigger using the Single Mode Register 0's A-D conversion start trigger select bit and for A-D conversion, set the said register's A-D conversion start bit to 1. For comparate mode, write the value to be compared into the A-D Successive Approximation Register (AD0SAR or AD1SAR) while operating in scan mode.

To start single mode conversion in hardware during scan mode operation, select hardware trigger using Single Mode Register 0's A-D conversion start trigger select bit and enter the hardware trigger selected with said register (TOM0_6 underflow, input on external pin TIN16, TOM0_0-7 enable event, or completion of A-D1 conversion for the A-D0 Converter; TOM0_6 underflow, input on external pin TIN16, TOM1_6 underflow, or completion of A-D0 conversion for the A-D1 Converter).

An A-D conversion interrupt request or DMA transfer request can be generated when conversion on the selected channel is finished or when one cycle of scan operation is finished.

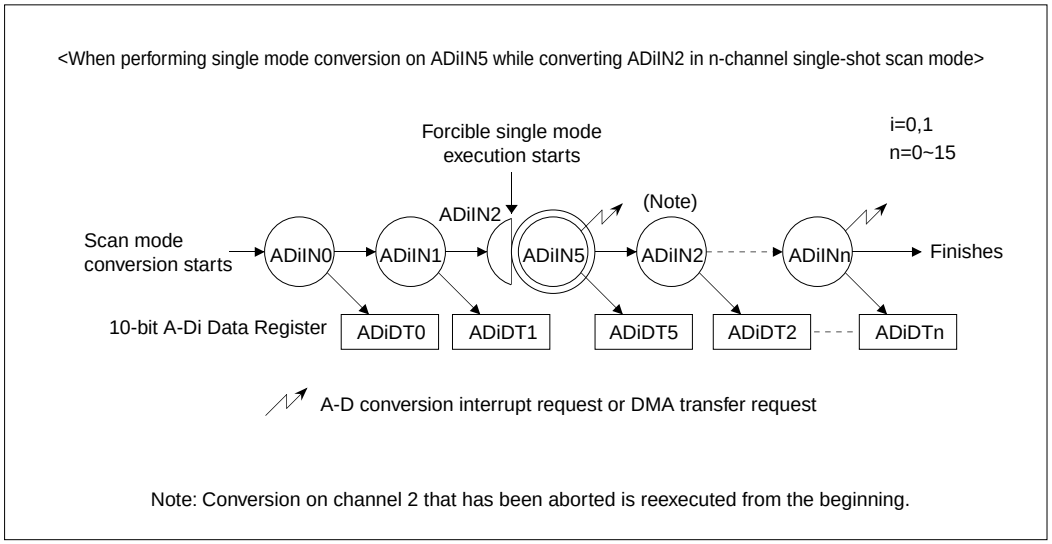


Figure 11.1.6 Forcibly Executing Single Mode during Scan Mode Operation

(2) Start scan mode after executing single mode

In this special operation mode, the A-D converter starts scan operation from single mode conversion (A-D conversion or comparate) in succession.

To start in software, select software trigger using the Scan Mode Register 0's A-D conversion start trigger select bit and set the said register's A-D conversion start bit to 1 while performing single mode conversion.

To start in hardware, select hardware trigger using Scan Mode Register 0's A-D conversion start trigger select bit and enter the hardware trigger selected with said register (TOM0_6 underflow, input on external pin TIN16, TOM0_0-7 enable event, or completion of A-D1 conversion for the A-D0 Converter; TOM0_6 underflow, input on external pin TIN16, TOM1_6 underflow, or completion of A-D0 conversion for the A-D1 Converter) while performing single mode conversion.

When a hardware trigger (TOM0_6 underflow, input on external pin TIN16, TOM0_0-7 enable event, or completion of A-D1 conversion for the A-D0 Converter; TOM0_6 underflow, input on external pin TIN16, TOM1_6 underflow, or completion of A-D0 conversion for the A-D1 Converter) is entered after selecting it with the A-D conversion start trigger select bits of both Single Mode Register 0 and Scan Mode Register 0, single mode conversion is performed first and then after executing it, scan mode conversion is performed in succession.

An A-D conversion interrupt request or DMA transfer request can be generated when single mode conversion on the selected channel is finished or when one cycle of scan operation is finished.

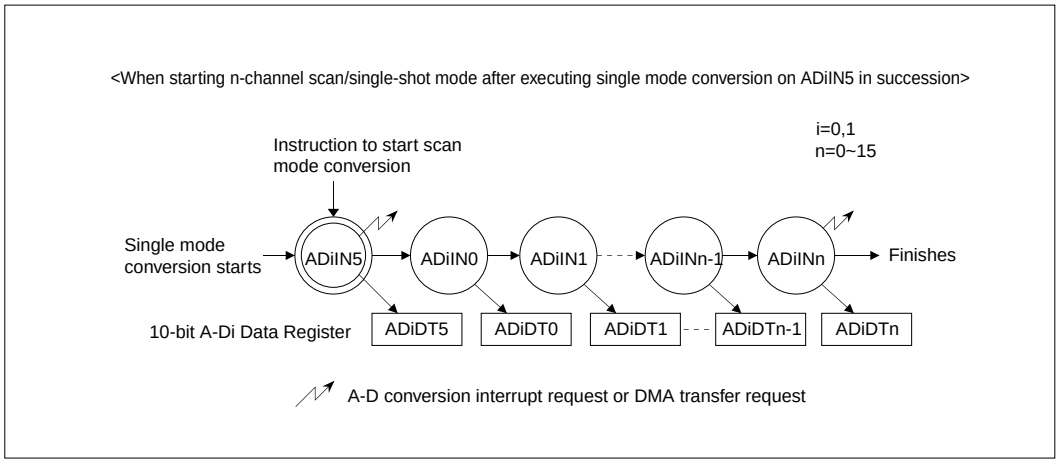


Figure 11.1.7 Starting Scan Mode after Executing Single Mode

(3) Restart conversion

In this special operation mode, the A-D converter reexecutes single mode or scan mode operation from the beginning after stopping it while in progress.

In the case of single mode, the operation being executed is reexecuted by setting Single Mode Register 0's A-D conversion start bit to 1 again or by entering a hardware trigger signal (TOM0_6 underflow, input on external pin TIN16, TOM0_0-7 enable event, or completion of A-D1 conversion for the A-D0 Converter; TOM0_6 underflow, input on external pin TIN16, TOM1_6 underflow, or completion of A-D0 conversion for the A-D1 Converter) while A-D conversion or comparate is under way.

For scan mode, the channel being converted is canceled and A-D conversion is restarted from channel 0 by setting Scan Mode Register 0's A-D conversion start bit to 1 again or by entering a hardware trigger signal (TOM0_6 underflow, input on external pin TIN16, TOM0_0-7 enable event, or completion of A-D1 conversion for the A-D0 Converter; TOM0_6 underflow, input on external pin TIN16, TOM1_6 underflow, or completion of A-D0 conversion for the A-D1 Converter) while scan operation is under way.

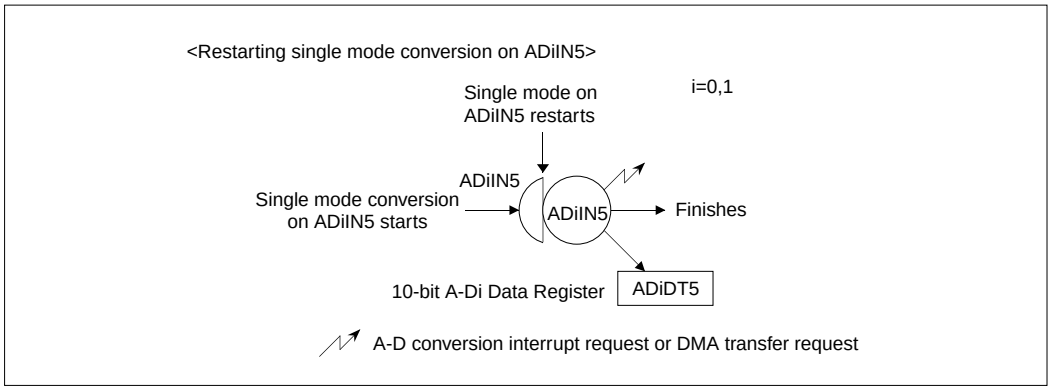


Figure 11.1.8 Restarting Conversion during Single Mode Operation

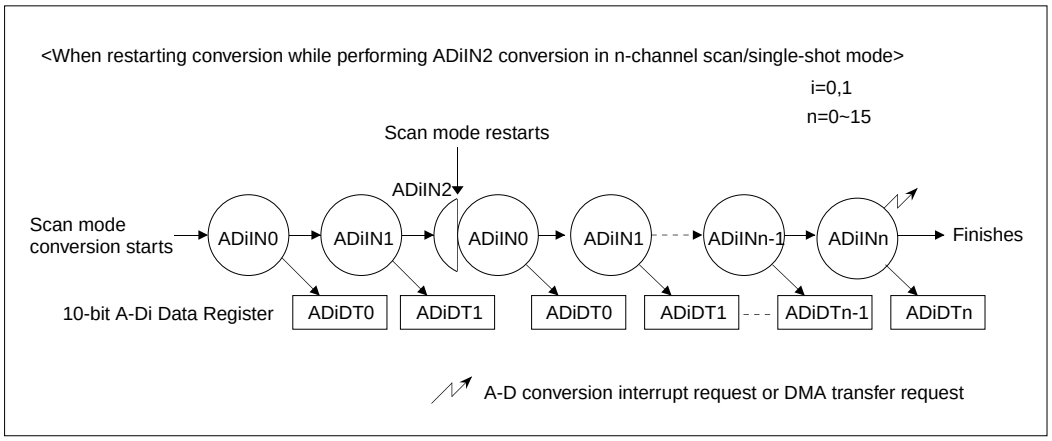


Figure 11.1.9 Restarting Conversion during Scan Mode Operation

11.1.4 Interrupt and DMA Transfer Requests by A-D Converters

The A-D converter can generate an A-D conversion interrupt request or DMA transfer request when A-D conversion, comparate, or single-shot scan is finished or each time one cycle of continuous scan mode is finished.
To select between A-D conversion interrupt and DMA transfer requests to generate, use Single Mode Register 0 and Scan Mode Register 0.

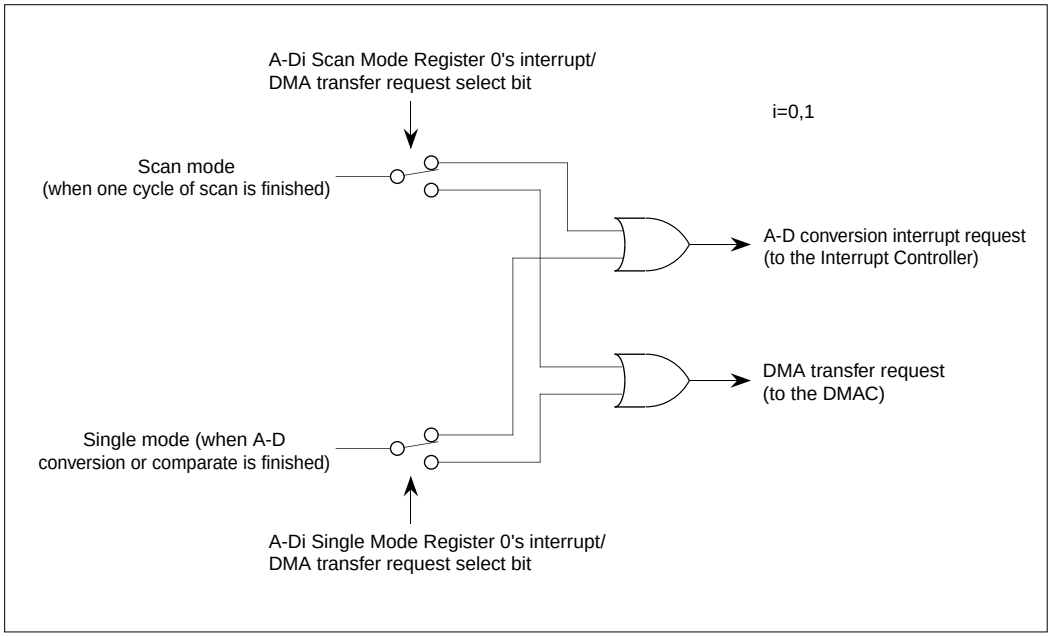


Figure 11.1.10 Selecting between A-D Conversion Interrupt and DMA Transfer Requests

11.2 A-D Converter Related Registers

An A-D converter related register map is shown below.

Address	+0 address D0 – D7	+1 address D8 – D15
H'0080 0080	A-D0 Single Mode Register 0 (AD0SIM0)	A-D0 Single Mode Register 1 (AD0SIM1)
H'0080 0082		
H'0080 0084	A-D0 Scan Mode Register 0 (AD0SCM0)	A-D0 Scan Mode Register 1 (AD0SCM1)
H'0080 0086		A-D0 Conversion Speed Control Register (AD0CVSCR)
H'0080 0088	A-D0 Successive Approximation Register(AD0SAR)	
H'0080 008A		
H'0080 008C	A-D0 Comparate Data Register(AD0CMP)	
H'0080 008E	A-D0 Digital Input Control Register(AD0CHCON)	
H'0080 0090	10-bit A-D0 Data Register0(AD0DT0)	
H'0080 0092	10-bit A-D0 Data Register1(AD0DT1)	
H'0080 0094	10-bit A-D0 Data Register2(AD0DT2)	
H'0080 0096	10-bit A-D0 Data Register3(AD0DT3)	
H'0080 0098	10-bit A-D0 Data Register4(AD0DT4)	
H'0080 009A	10-bit A-D0 Data Register5(AD0DT5)	
H'0080 009C	10-bit A-D0 Data Register6(AD0DT6)	
H'0080 009E	10-bit A-D0 Data Register7(AD0DT7)	
H'0080 00A0	10-bit A-D0 Data Register8(AD0DT8)	
H'0080 00A2	10-bit A-D0 Data Register9(AD0DT9)	
H'0080 00A4	10-bit A-D0 Data Register10(AD0DT10)	
H'0080 00A6	10-bit A-D0 Data Register11(AD0DT11)	
H'0080 00A8	10-bit A-D0 Data Register12(AD0DT12)	
H'0080 00AA	10-bit A-D0 Data Register13(AD0DT13)	
H'0080 00AC	10-bit A-D0 Data Register14(AD0DT14)	
H'0080 00AE	10-bit A-D0 Data Register15(AD0DT15)	

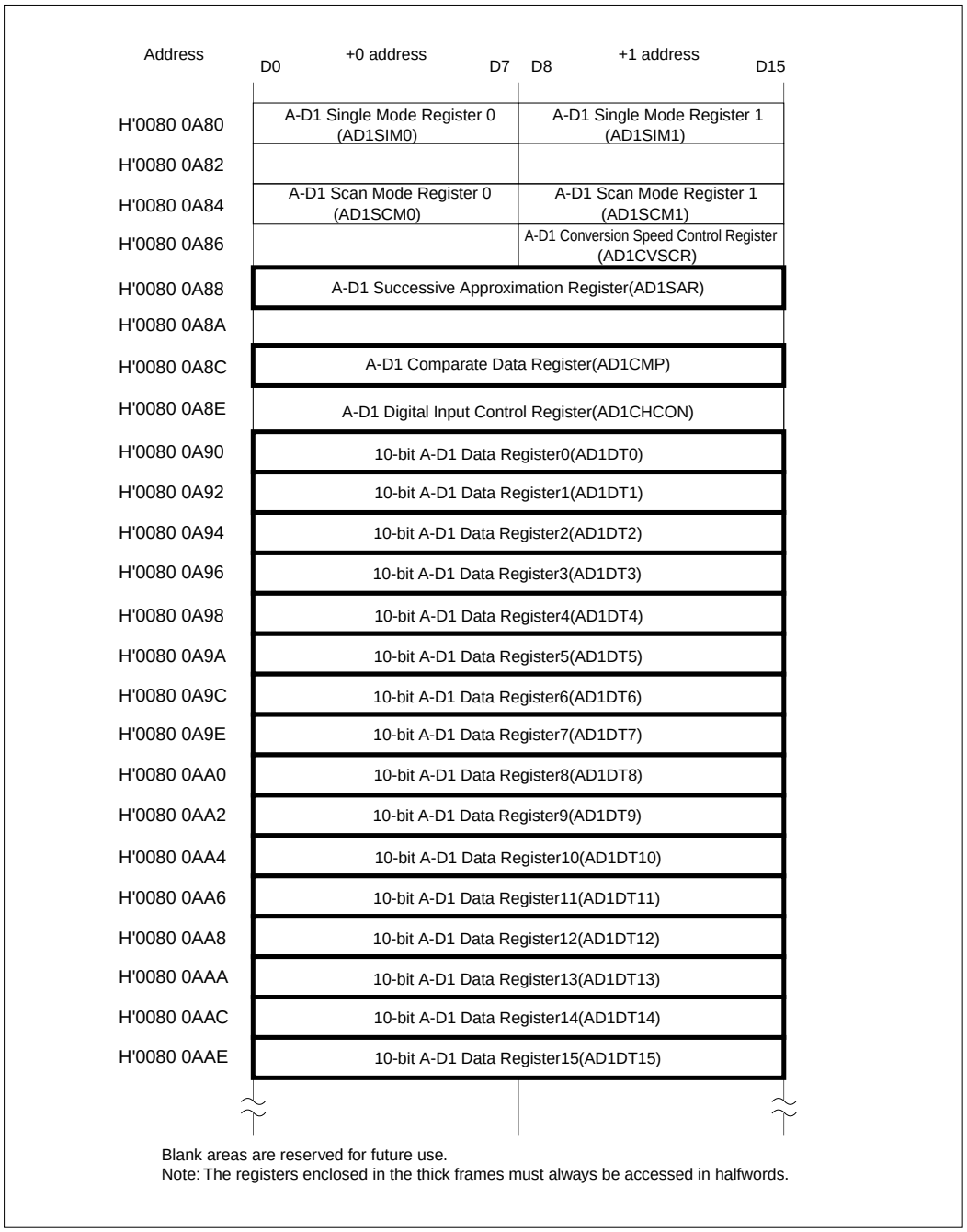
Blank areas are reserved for future use.
Note: The registers enclosed in the thick frames must always be accessed in halfwords.

Figure 11.2.1 A-D Converter Related Register Map (1/4)

Address	D0 +0 address D7	D8 +1 address D15
H'0080 00D0		8-bit A-D0 Data Register0 (AD08DT0)
H'0080 00D2		8-bit A-D0 Data Register1 (AD08DT1)
H'0080 00D4		8-bit A-D0 Data Register2 (AD08DT2)
H'0080 00D6		8-bit A-D0 Data Register3 (AD08DT3)
H'0080 00D8		8-bit A-D0 Data Register4 (AD08DT4)
H'0080 00DA		8-bit A-D0 Data Register5 (AD08DT5)
H'0080 00DC		8-bit A-D0 Data Register6 (AD08DT6)
H'0080 00DE		8-bit A-D0 Data Register7 (AD08DT7)
H'0080 00E0		8-bit A-D0 Data Register8 (AD08DT8)
H'0080 00E2		8-bit A-D0 Data Register9 (AD08DT9)
H'0080 00E4		8-bit A-D0 Data Register10 (AD08DT10)
H'0080 00E6		8-bit A-D0 Data Register11 (AD08DT11)
H'0080 00E8		8-bit A-D0 Data Register12 (AD08DT12)
H'0080 00EA		8-bit A-D0 Data Register13 (AD08DT13)
H'0080 00EC		8-bit A-D0 Data Register14 (AD08DT14)
H'0080 00EE		8-bit A-D0 Data Register15 (AD08DT15)

Blank areas are reserved for future use.

Figure 11.2.2 A-D Converter Related Register Map (2/4)



Blank areas are reserved for future use.
Note: The registers enclosed in the thick frames must always be accessed in halfwords.

Figure 11.2.3 A-D Converter Related Register Map (3/4)

Address	D0 +0 address D7	D8 +1 address D15
H'0080 0AD0		8-bit A-D1 Data Register0 (AD18DT0)
H'0080 0AD2		8-bit A-D1 Data Register1 (AD18DT1)
H'0080 0AD4		8-bit A-D1 Data Register2 (AD18DT2)
H'0080 0AD6		8-bit A-D1 Data Register3 (AD18DT3)
H'0080 0AD8		8-bit A-D1 Data Register4 (AD18DT4)
H'0080 0ADA		8-bit A-D1 Data Register5 (AD18DT5)
H'0080 0ADC		8-bit A-D1 Data Register6 (AD18DT6)
H'0080 0ADE		8-bit A-D1 Data Register7 (AD18DT7)
H'0080 0AE0		8-bit A-D1 Data Register8 (AD18DT8)
H'0080 0AE2		8-bit A-D1 Data Register9 (AD18DT9)
H'0080 0AE4		8-bit A-D1 Data Register10 (AD18DT10)
H'0080 0AE6		8-bit A-D1 Data Register11 (AD18DT11)
H'0080 0AE8		8-bit A-D1 Data Register12 (AD18DT12)
H'0080 0AEA		8-bit A-D1 Data Register13 (AD18DT13)
H'0080 0AEC		8-bit A-D1 Data Register14 (AD18DT14)
H'0080 0AEE		8-bit A-D1 Data Register15 (AD18DT15)

Blank areas are reserved for future use.

Figure 11.2.4 A-D Converter Related Register Map (4/4)

11.2.1 A-D Single Mode Registers 0

■ **A-D0 Single Mode Register 0 (AD0SIM0)** **<Address: H'0080 0080>**

D0	1	2	3	4	5	6	D7
AD0STRG1		AD0STRG0	AD0SSEL	AD0SREQ	AD0SCMP	AD0SSTP	AD0SSTT

<When reset: H'04>

D	Bit Name	Function	R	W
0	AD0STRG1 (Note 1) (A-D0 hardware trigger select 1)	Selects A-D0 hardware trigger with D0 and D2 bits D0 D2 0 0: TOM0_6 underflow 0 1: TIN16 input 1 0: TOM0_0-7 enable event (Note 2) 1 1: A-D1 conversion finished	○	○
1	No functions assigned		0	–
2	AD0STRG0 (Note 1) (A-D0 hardware trigger select 0)	Selects A-D0 hardware trigger with D0 and D2 bits (see the column for D0 bit)	○	○
3	AD0SSEL (Select A-D0 conversion start trigger)	0: Software trigger 1: Hardware trigger	○	○
4	AD0SREQ (Select interrupt request/DMA request)	0: A-D0 interrupt request 1: DMA transfer request	○	○
5	AD0SCMP (A-D0 conversion/comparate complete)	0: A-D0 conversion/comparate in progress 1: A-D0 conversion/comparate completed	○	–
6	AD0SSTP (Stop A-D0 conversion)	0: No operation 1: Stops A-D0 conversion	0	○
7	AD0SSTT (Start A-D0 conversion)	0: No operation 1: Starts A-D0 conversion	0	○

Note 1: To select a hardware trigger, use the D0 bit (A-D0 hardware trigger select 1) and D2 bit (A-D0 hardware trigger select 0).

Note 2: This refers to TID0 overflow/underflow, TOM0_7 underflow, input on external pin TIN18, TID1 overflow/underflow, or TOM1_7 underflow. (For details, see Chapter 10, "Input/output Timers.")

A-D0 Single Mode Register 0 is used to control operation of the A-D0 Converter during single mode (including special mode "Forcibly execute single mode during scan mode operation".)

■ **A-D1 Single Mode Register 0 (AD1SIM0)** **<Address: H'0080 0A80>**

D0	1	2	3	4	5	6	D7
AD1STRG1		AD1STRG0	AD1SSEL	AD1SREQ	AD1SCMP	AD1SSTP	AD1SSTT

<When reset: H'04>

D	Bit Name	Function	R	W
0	AD1STRG1 (Note 1) (A-D1 hardware trigger select 1)	Selects A-D1 hardware trigger with D0 and D2 bits D0 D2 0 0: TOM0_6 underflow 0 1: TIN16 input 1 0: TOM1_6 underflow 1 1: A-D0 conversion finished	○	○
1	No functions assigned		0	–
2	AD1STRG0 (Note) (A-D1 hardware trigger select 0)	Selects A-D1 hardware trigger with D0 and D2 bits (see the column for D0 bit)	○	○
3	AD1SSEL (Select A-D1 conversion start trigger)	0: Software trigger 1: Hardware trigger	○	○
4	AD1SREQ (Interrupt request)	0: A-D1 interrupt request 1: DMA transfer request	○	○
5	AD1SCMP (A-D1 conversion/comparate complete)	0: A-D1 conversion/comparate in progress 1: A-D1 conversion/comparate completed	○	–
6	AD1SSTP (Stop A-D1 conversion)	0: No operation 1: Stops A-D1 conversion	0	○
7	AD1SSTT (Start A-D1 conversion)	0: No operation 1: Starts A-D1 conversion	0	○

Note: To select a hardware trigger, use the D0 bit (A-D1 hardware trigger select 1) and D2 bit (A-D1 hardware trigger select 0).

A-D1 Single Mode Register 0 is used to control operation of the A-D1 Converter during single mode (including special mode "Forcibly execute single mode during scan mode operation".)

(1) ADnSTRG1 and ADnSTRG0 (A-Dn hardware trigger select) bits (D0, D2)

When starting A-D conversion of the A-Dn converter in hardware, these bits select the cause for which to start conversion (TOM0_6 underflow, input on external pin TIN16, TOM0_0-7 enable event, or completion of A-D1 conversion for A-D0; TOM0_6 underflow, input on external pin TIN16, TOM1_6 underflow, or completion of A-D0 conversion for A-D1). If software trigger is selected with the ADnSSEL (A-Dn conversion start trigger select) bit, the contents of these bits are ignored.

(2) ADnSSEL (A-Dn conversion start trigger select) bit (D3)

This bit selects whether to apply the A-Dn conversion start trigger in software or in hardware during single mode.
When software trigger is selected, A-D conversion is started by setting the ADnSSTT (A-Dn conversion start) bit to 1. When hardware trigger is selected, A-D conversion is started by the cause of conversion selected with the ADnSTRG0 (hardware trigger select 0) and ADnSTRG1 (hardware trigger select 1) bits.

(3) ADnSREQ (A-Dn interrupt request/DMA transfer request select) bit (D4)

This bit selects whether to generate an A-Dn conversion interrupt request or a DMA transfer request at completion of A-Dn converter operation in single mode (A-D conversion or comparate). When using neither interrupt nor DMA transfer, select A-Dn conversion interrupt request and mask it with the ICU's A-Dn Converter Interrupt Control Register, or select DMA transfer and disable the DMA transfer to be started at completion of A-Dn conversion with the DMAn Channel Control Register.

(4) ADnSCMP (A-Dn conversion/comparate complete) bit (D5)

This is a read-only bit, and is 1 when reset. This bit is 0 when the A-Dn converter is operating in single mode (A-D conversion or comparate) and set to 1 when the operation is completed.
It also is set to 1 when A-D convert or comparate operation is forcibly terminated by setting the ADnSSTT (A-Dn conversion stop) bit to 1 during A-D conversion or comparate.

(5) ADnSSTP (A-Dn conversion stop) bit (D6)

The A-Dn converter operating in single mode (A-D conversion or comparate) can be stopped by setting this bit to 1 while the operation is in progress. Manipulation of this bit is ignored while the converter when in single mode remains idle or is operating in scan mode.
Operation is stopped immediately after writing to this bit, and when the content of the "A-Dn Successive Approximation Register" is read after stopping the operation, it shows an intermediate value that was in the middle of conversion. (No transfers to the A-Dn Data Register are performed.)

If the A-Dn conversion start and A-Dn conversion stop bits are set to 1 simultaneously, the latter has priority so that the A-Dn conversion is stopped.

If this bit is set to 1 while single mode operation in special mode "Forcibly execute single mode during scan mode operation" is under way, only single mode conversion stops and scan mode operation restarts.

(6) ADnSSTT (A-Dn conversion start) bit (D7)

A-D conversion of the A-Dn converter is started by setting this bit to 1 while software trigger has been selected with the ADnSSEL (A-Dn conversion start trigger select) bit.

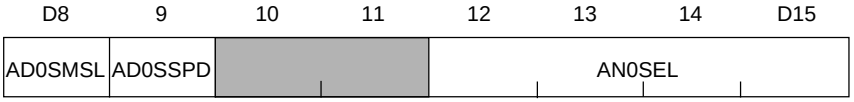
If the A-Dn conversion start and A-Dn conversion stop bits are set to 1 simultaneously, the latter has priority so that the A-Dn conversion is stopped.

When this bit is set to 1 during single mode conversion, special operation mode "Restart conversion" is assumed, so that conversion in single mode restarts.

When this bit is set to 1 during A-D conversion in scan mode, special operation mode "Forcibly execute single mode during scan mode operation" is assumed, so that the channel being converted in scan mode is canceled and single mode conversion is performed. When single mode conversion finishes, A-D conversion in scan mode restarts from the canceled channel.

11.2.2 A-D Single Mode Registers 1

■ **A-D0 Single Mode Register 1 (AD0SIM1)** **<Address: H'0080 0081>**



<When reset: H'00>

D	Bit Name	Function	R	W
8	AD0SMSL (Select A-D0 conversion mode)	0: A-D0 conversion mode 1: Comparator mode	○	○
9	AD0SSPD (Note 1) (Select A-D0 conversion speed)	0: Normal 1: Double speed	○	○
10,11	No functions assigned		0	△
12-15	AN0SEL (Note 2) (Select analog input pin)	0000: Selects AD0IN0 0001: Selects AD0IN1 0010: Selects AD0IN2 0011: Selects AD0IN3 0100: Selects AD0IN4 0101: Selects AD0IN5 0110: Selects AD0IN6 0111: Selects AD0IN7 1000: Selects AD0IN8 1001: Selects AD0IN9 1010: Selects AD0IN10 1011: Selects AD0IN11 1100: Selects AD0IN12 1101: Selects AD0IN13 1110: Selects AD0IN14 1111: Selects AD0IN15	○	○

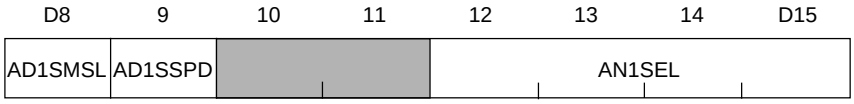
W=△: Writing 0 only is effective. Writing 1 to these bits is unaccepted, the device operation cannot be guaranteed.

Note 1: Because the A-D0 conversion speed is determined by a combined use of this AD0SSPD bit and the A-D0 Conversion Speed Control Register AD0CVSD bit, make sure the AD0SSPD and AD0CVSD bits both are set.

Note 2: When AD0IN8-AD0IN15 are selected, the A-D0 Digital Input Control Register needs to be set.

A-D0 Single Mode Register 1 is used to control operation of the A-D0 Converter during single mode (including special mode "Forcibly execute single mode during scan mode operation".)

■ **A-D1 Single Mode Register 1 (AD1SIM1)** **<Address: H'0080 0A81>**



<When reset: H'00>

D	Bit Name	Function	R	W
8	AD1SMSL (Select A-D1 conversion mode)	0: A-D1 conversion mode 1: Comparator mode	○	○
9	AD1SSPD (Note 1) (Select A-D1 conversion speed)	0: Normal 1: Double speed	○	○
10,11	No functions assigned		0	△
12-15	AN1SEL (Note 2) (Select analog input pin)	0000: Selects AD1IN0 0001: Selects AD1IN1 0010: Selects AD1IN2 0011: Selects AD1IN3 0100: Selects AD1IN4 0101: Selects AD1IN5 0110: Selects AD1IN6 0111: Selects AD1IN7 1000: Selects AD1IN8 1001: Selects AD1IN9 1010: Selects AD1IN10 1011: Selects AD1IN11 1100: Selects AD1IN12 1101: Selects AD1IN13 1110: Selects AD1IN14 1111: Selects AD1IN15	○	○

W=△: Writing 0 only is effective. Writing 1 to these bits is unaccepted, the device operation cannot be guaranteed.

Note 1: Because the A-D1 conversion speed during single mode is determined by a combined use of this AD1SSPD bit and the A-D1 Conversion Speed Control Register AD1CVSD bit, make sure the AD1SSPD and AD1CVSD bits both are set.

Note 2: When AD1IN4-AD1IN15 are selected, the A-D1 Digital Input Control Register needs to be set.

A-D1 Single Mode Register 1 is used to control operation of the A-D1 Converter during single mode (including special mode "Forcibly execute single mode during scan mode operation".)

(1) ADnSMSL (A-Dn conversion mode select) bit (D8)

This bit selects A-D conversion mode when the A-Dn converter is operating in single mode. Setting this bit to 0 selects A-D conversion mode; setting this bit to 1 selects comparator mode.

(2) ADnSSPD (A-Dn conversion speed select) bit (D9)

This bit selects the A-D conversion speed when the A-Dn converter is operating in single mode. Setting this bit to 0 selects normal speed; setting this bit to 1 selects double speed.

Note: Because the A-Dn conversion speed during single mode is determined by a combined use of this ADnSSPD bit and the A-Dn Conversion Speed Control Register ADnCVSD bit, make sure the ADnSSPD and ADnCVSD bits both are set.

(3) ANnSEL (analog input pin select) bits (D12-D15)

These bits select an analog input pin when the A-Dn converter is operating in single mode. The channel selected with this bit is the target channel on which A-D conversion or comparate operation is performed. When read, this bit shows the value that has been written to it.

11.2.3 A-D Scan Mode Registers 0

■ A-D0 Scan Mode Register 0 (AD0SCM0) **<Address: H'0080 0084>**

D0	1	2	3	4	5	6	D7
AD0CTRG1	AD0CMSL	AD0CTRG0	AD0CSEL	AD0CREQ	AD0CCMP	AD0CSTP	AD0CSTT

<When reset: H'04>

D	Bit Name	Function	R	W
0	AD0CTRG1 (Note 1) (A-D0 hardware trigger select 1)	Selects A-D0 hardware trigger with D0 and D2 bits D0 D2 0 0: TOM0_6 underflow 0 1: TIN16 input 1 0: TOM0_0-7 enable event (Note 2) 1 1: A-D1 conversion finished	○	○
1	AD0CMSL (Select A-D0 scan mode)	0: Single-shot mode 1: Continuous mode	○	○
2	AD0CTRG0 (Note 1) (A-D0 hardware trigger select 0)	Selects A-D0 hardware trigger with D0 and D2 bits (see the column for D0 bit)	○	○
3	AD0CSEL (Select A-D0 conversion start trigger)	0: Software trigger 1: Hardware trigger	○	○
4	AD0CREQ (Select interrupt request/DMA request)	0: A-D0 interrupt request 1: DMA transfer request	○	○
5	AD0CCMP (A-D0 conversion complete)	0: A-D0 conversion in progress 1: A-D0 conversion completed	○	–
6	AD0CSTP (Stop A-D0 conversion)	0: No operation 1: Stops A-D0 conversion	0	○
7	AD0CSTT (Start A-D0 conversion)	0: No operation 1: Starts A-D0 conversion	0	○

Note 1: To select a hardware trigger, use the D0 bit (A-D0 hardware trigger select 1) and D2 bit (A-D0 hardware trigger select 0).

Note 2: This refers to TID0 overflow/underflow, TOM0_7 underflow, input on external pin TIN18, TID1 overflow/underflow, or TOM1_7 underflow. (For details, see Chapter 10, "Input/output Timers.")

A-D0 Scan Mode Register 0 is used to control operation of the A-D0 Converter during scan mode.

■ **A-D1 Scan Mode Register 0 (AD1SCM0)** **<Address: H'0080 0A84>**

D0	1	2	3	4	5	6	D7
AD1CTRG1	AD1CMSL	AD1CTRG0	AD1CSEL	AD1CREQ	AD1CCMP	AD1CSTP	AD1CSTT

<When reset: H'04>

D	Bit Name	Function	R	W
0	AD1CTRG1 (Note 1) (A-D1 hardware trigger select 1)	Selects A-D1 hardware trigger with D0 and D2 bits D0 D2 0 0: TOM0_6 underflow 0 1: TIN16 input 1 0: TOM1_6 underflow 1 1: A-D0 conversion finished	○	○
1	AD1CMSL (Select A-D1 scan mode)	0: Single-shot mode 1: Continuous mode	○	○
2	AD1CTRG0 (Note) (A-D1 hardware trigger select 0)	Selects A-D1 hardware trigger with D0 and D2 bits (see the column for D0 bit)	○	○
3	AD1CSEL (Select A-D1 conversion start trigger)	0: Software trigger 1: Hardware trigger	○	○
4	AD1CREQ (interrupt request)	0: A-D1 interrupt request 1: DMA transfer request	○	○
5	AD1CCMP (A-D1 conversion complete)	0: A-D1 conversion in progress 1: A-D1 conversion completed	○	–
6	AD1CSTP (Stop A-D1 conversion)	0: No operation 1: Stops A-D1 conversion	0	○
7	AD1CSTT (Start A-D1 conversion)	0: No operation 1: Starts A-D1 conversion	0	○

Note: To select a hardware trigger, use the D0 bit (A-D0 hardware trigger select 1) and D2 bit (A-D0 hardware trigger select 0).

A-D1 Scan Mode Register 0 is used to control operation of the A-D1 Converter during scan mode.

(1) ADnCTRG1 and ADnCTRG0 (A-Dn hardware trigger select) bits (D0, D2)

When starting A-D conversion of the A-Dn converter in hardware, these bits select the cause for which to start conversion (TOM0_6 underflow, input on external pin TIN16, TOM0_0-7 enable event, or completion of A-D1 conversion for A-D0; TOM0_6 underflow, input on external pin TIN16, TOM1_6 underflow, or completion of A-D0 conversion for A-D1). If software trigger is selected with the ADnCSEL (A-Dn conversion start trigger select) bit, the contents of these bits are ignored.

(2) ADnCMSL (A-Dn scan mode select) bit (D1)

This bit selects the A-Dn converter scan mode between single-shot scan and continuous scan modes.

Setting this bit to 0 selects single-shot scan mode, so that A-D conversion of channels selected with the ANnSCAN (scan loop select) bits are performed sequentially, beginning with the channel ADnIN0. When A-D conversion on all selected channels is completed, the convert operation stops.

Setting this bit to 1 selects continuous scan mode, so that when operation in single-shot scan mode finishes, A-D conversion is performed from the first channel again. This is repeated until stopped by setting the ADnCSTP (A-Dn conversion stop) bit to 1.

(3) ADnCSEL (A-Dn conversion start trigger select) bit (D3)

This bit selects whether to apply the A-D conversion start trigger in software or in hardware during scan mode of the A-Dn converter.
When software trigger is selected, A-D conversion is started by setting the ADnCSTT (A-Dn conversion start) bit to 1. When hardware trigger is selected, A-D conversion is started by the cause of conversion selected with the ADnCTRG0 (hardware trigger select 0) and ADnCTRG1 (hardware trigger select 1) bits.

(4) ADnCREQ (A-Dn interrupt/DMA transfer request select) bit (D4)

This bit selects whether to generate an A-Dn conversion interrupt request or a DMA transfer request when one cycle of scan mode operation is finished. When using neither interrupt nor DMA transfer, select A-Dn conversion interrupt request and mask it with the ICU's A-Dn Converter Interrupt Control Register, or select DMA transfer and disable the DMA transfer to be started at completion of A-Dn conversion with the DMAn Channel Control Register.

(5) ADnCCMP (A-Dn conversion complete) bit (D5)

This is a read-only bit, and is 1 when reset. This bit is 0 when scan mode conversion of the A-Dn converter is in progress and set to 1 when single-shot scan mode is completed or when continuous scan mode is stopped by setting the ADnCSTT (A-Dn conversion stop) bit to 1.

(6) ADnCSTP (A-Dn conversion stop) bit (D6)

Scan mode A-D conversion with the A-Dn converter can be stopped by setting this bit to 1 while operation is under way. This bit is effective for only scan mode operation, and does not affect single mode operation when both single and scan modes are activated in special operation mode.
Operation is stopped immediately after writing to this bit and A-C conversion on the channel which is in the middle of conversion is aborted, with no data transferred to the A-D Data Register.

If the A-Dn conversion start and A-Dn conversion stop bits are set to 1 simultaneously, the latter has priority so that the A-Dn conversion is stopped.

(7) ADnCSTT (A-Dn conversion start) bit (D7)

This bit is used to start scan mode of the A-Dn converter in software. Only when software trigger has been selected with the ADnCSEL (A-Dn conversion start trigger select) bit, A-D conversion can be started by setting this bit to 1.

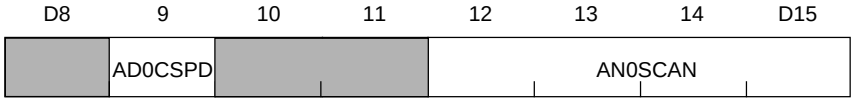
If the A-Dn conversion start and A-Dn conversion stop bits are set to 1 simultaneously, the latter has priority so that the A-Dn conversion is stopped.

When this bit is set to 1 during scan mode conversion again, special operation mode "Restart conversion" is assumed, so that scan operation restarts according to the contents set by Scan Mode Register 0 and Scan Mode Register 1.

When this bit is set to 1 during single mode A-D conversion, special operation mode "Start scan mode after executing single mode" is assumed, so that scan mode operation starts in succession after single mode finishes.

11.2.4 A-D Scan Mode Registers 1

■ **A-D0 Scan Mode Register 1 (AD0SCM1)** **<Address: H'0080 0085>**



<When reset: H'00>

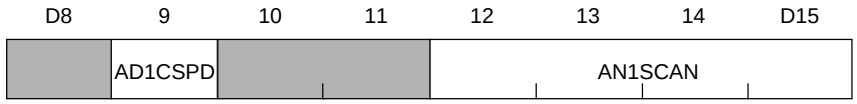
D	Bit Name	Function	R	W
8	No functions assigned		0	–
9	AD0CSPD (Note 1) (Select A-D0 conversion speed)	0: Normal 1: Double speed	○	○
10,11	No functions assigned		0	–
12-15	AN0SCAN (Note 2) (Select A-D0 scan loop)	<When writing> B'0000-B'1111 (Channel 0 through channel 15) <When read during conversion> 0000: AD0IN0 being converted 0001: AD0IN1 being converted 0010: AD0IN2 being converted 0011: AD0IN3 being converted 0100: AD0IN4 being converted 0101: AD0IN5 being converted 0110: AD0IN6 being converted 0111: AD0IN7 being converted 1000: AD0IN8 being converted 1001: AD0IN9 being converted 1010: AD0IN10 being converted 1011: AD0IN11 being converted 1100: AD0IN12 being converted 1101: AD0IN13 being converted 1110: AD0IN14 being converted 1111: AD0IN15 being converted	○	○

Note 1: Because the A-D0 conversion speed during scan mode is determined by a combined use of this AD0CSPD bit and the A-D0 Conversion Speed Control Register AD0CVSD bit, make sure the AD0CSPD and AD0CVSD bits both are set.

Note 2: If the scan loop includes any channels AD0IN8-AD0IN15, the AD0 Digital Input Control Register needs to be set.

A-D0 Scan Mode Register 1 is used to control operation of the A-D0 Converter during scan mode.

■ **A-D1 Scan Mode Register 1 (AD1SCM1)** **<Address: H'0080 0A85>**



<When reset: H'00>

D	Bit Name	Function	R	W
8	No functions assigned		0	–
9	AD1CSPD (Note 1) (Select A-D1 conversion speed)	0: Normal 1: Double speed	○	○
10,11	No functions assigned		0	–
12-15	AN1SCAN (Note 2) (Select A-D1 scan loop)	<When writing> B'0000-B'1111 (Channel 0 through channel 15) <When read during conversion> 0000: AD1IN0 being converted 0001: AD1IN1 being converted 0010: AD1IN2 being converted 0011: AD1IN3 being converted 0100: AD1IN4 being converted 0101: AD1IN5 being converted 0110: AD1IN6 being converted 0111: AD1IN7 being converted 1000: AD1IN8 being converted 1001: AD1IN9 being converted 1010: AD1IN10 being converted 1011: AD1IN11 being converted 1100: AD1IN12 being converted 1101: AD1IN13 being converted 1110: AD1IN14 being converted 1111: AD1IN15 being converted	○	○

Note 1: Because the A-D1 conversion speed during scan mode is determined by a combined use of this AD1CSPD bit and the A-D1 Conversion Speed Control Register AD1CVSD bit, make sure the AD1CSPD and AD1CVSD bits both are set.

Note 2: If the scan loop includes any channels AD1IN4-AD1IN15, the AD1 Digital Input Control Register needs to be set.

A-D1 Scan Mode Register 1 is used to control operation of the A-D1 Converter during scan mode.

(1) ADnCSPD (A-Dn conversion speed select) bit (D9)

This bit selects the A-D conversion speed when the A-Dn converter is operating in scan mode. Setting this bit to 0 selects normal speed; setting this bit to 1 selects double speed.

Note: Because the A-Dn conversion speed during scan mode is determined by a combined use of this ADnCSPD bit and the A-Dn Conversion Speed Control Register ADnCVSD bit, make sure the ADnCSPD and ADnCVSD bits both are set.

(2) ANnSCAN (A-Dn scan loop select) bits (D12-D15)

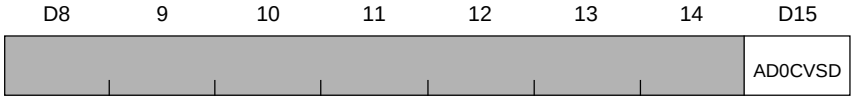
These bits select a range of channels by channel number beginning with the channel ADnIN0 which are to be operated on during scan mode operation with the A-Dn converter.
(For details about the scan loop configuration by specified channel numbers, see Table 11.1.2.)

When the ANnSCAN (A-Dn scan loop select) bits are read out during scan operation, they serve as status bits indicating the channel being converted.

When read during single mode, these bits always show the value "B'0000."
If these bits are read when the A-D conversion is stopped by setting the Scan Mode Register 0 ADnCSTP (A-Dn conversion stop) bit to 1, they show the value of the channel that has been canceled of A-D convert operation.
Also, when read during single mode conversion in special operation mode "Forcibly execute single mode during scan mode operation," they show the value of the channel that has been canceled of A-D convert operation in the middle of scan.

11.2.5 A-D Conversion Speed Control Registers

■ A-D0 Conversion Speed Control Register (AD0CVSCR) **<Address: H'0080 0087>**

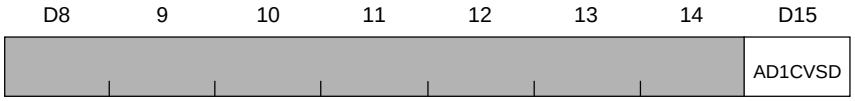


<When reset: H'00>

D	Bit Name	Function	R	W
8-14	No functions assigned		0	–
15	AD0CVSD (Note) (Control A-D0 conversion seed)	0: Low speed mode 1: High speed mode	○	○

Note: The A-D0 conversion speed is determined by a combined use of this AD0CVSD bit and the A-D0 Single Mode Register 1 AD0SSPD bit when operating in single mode, or a combined use of this AD0CVSD bit and the A-D0 Scan Mode Register 1 AD0CSPD bit when operating in scan mode. Be sure to set the AD0CVSD and AD0SSPD bits during single mode, or the AD0CVSD and AD0CSPD bits during scan mode.

■ A-D1 Conversion Speed Control Register (AD1CVSCR) **<Address: H'0080 0A87>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-14	No functions assigned		0	–
15	AD1CVSD (Note) (Control A-D1 conversion seed)	0: Low speed mode 1: High speed mode	○	○

Note: The A-D1 conversion speed is determined by a combined use of this AD1CVSD bit and the A-D1 Single Mode Register 1 AD1SSPD bit when operating in single mode, or a combined use of this AD1CVSD bit and the A-D1 Scan Mode Register 1 AD1CSPD bit when operating in scan mode. Be sure to set the AD1CVSD and AD1SSPD bits during single mode, or the AD1CVSD and AD1CSPD bits during scan mode.

(1) ADnCSPD (A-Dn conversion speed select) bit (D15)

This bit controls the A-D conversion speed of the A-Dn converter during single or scan mode. Setting this bit to 0 selects low speed mode; setting this bit to 1 selects high speed mode.

Note: The A-Dn conversion speed is determined by a combined use of this ADnCVSD bit and the A-Dn Single Mode Register 1 ADnSSPD bit when operating in single mode, or a combined use of this ADnCVSD bit and the A-Dn Scan Mode Register 1 ADnCSPD bit when operating in scan mode. Be sure to set the ADnCVSD and ADnSSPD bits during single mode, or the ADnCVSD and ADnCSPD bits during scan mode.

11.2.6 A-D Digital Input Control Registers

■ A-D0 Digital Input Control Register (AD0CHCON) **<Address: H'0080 008E>**

D0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	D15
								AD0CH8CN	AD0CH9CN	AD0CH10CN	AD0CH11CN	AD0CH12CN	AD0CH13CN	AD0CH14CN	AD0CH15CN

<When reset: H'00>

D	Bit Name	Function	R	W
0-7	No functions assigned (Note 1)		○	○
8-15	AD0CH8CN-AD0CH15CN (AD0 channel 8-15 input enable bit)	0: Disables input to AD0CHnCN 1: Enables input to AD0CHnCN	○	○

Note 1: The D0-D7 bits are blank bits. When read, they show the values written to them.
Note 2: The D8-D15 bits correspond one for one to channels 8 through channel 15.

The analog inputs AD0IN8-AD0IN15 to the A-D0 Converter may be used to monitor input ports or internal peripheral I/O pins. When using the analog input pins AD0IN8-AD0IN15, set any bit D8-D15 to 1 that corresponds to the analog input pin used.
However, the analog inputs AD0IN8-AD0IN15 do not have the conversion accuracy comparable to that of the dedicated analog input pins AD0IN0-AD0IN7.

■ A-D1 Digital Input Control Register (AD1CHCON) **<Address: H'0080 0A8E>**

D0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	D15
						AD1CH6CN	AD1CH7CN	AD1CH8CN	AD1CH9CN	AD1CH10CN	AD1CH11CN	AD1CH12CN	AD1CH13CN	AD1CH14CN	AD1CH15CN

<When reset: H'00>

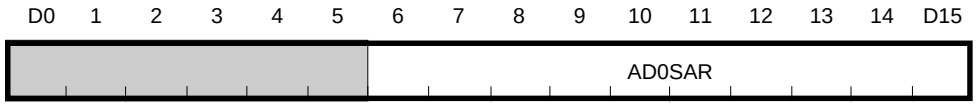
D	Bit Name	Function	R	W
0-5	No functions assigned (Note 2)		○	○
6-15	AD1CH8CN-AD1CH15CN (AD1 channel 6-15 input enable bit)	0: Disables input to AD1CHnCN 1: Enables input to AD1CHnCN	○	○

Note 1: The D0-D5 bits are blank bits. When read, they show the values written to them.
Note 2: The D6-D15 bits correspond one for one to channels 6 through channel 15.

The analog inputs AD1IN4-AD1IN15 to the A-D1 Converter may be used to monitor input ports or internal peripheral I/O pins. When using the analog input pins AD1IN6-AD1IN15, set any bit D6-D15 to 1 that corresponds to the analog input pin used.
However, the analog inputs AD1IN4-AD1IN15 do not have the conversion accuracy comparable to that of the dedicated analog input pins AD1IN0-AD1IN3.

11.2.7 A-D Successive Approximation Registers

■ **A-D0 Successive Approximation Register (AD0SAR)** **<Address: H'0080 0088>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6-15	AD0SAR (A-D0 successive approximation value/comparison value)	• A-D0 successive approximation value (A-D conversion mode) • Comparison value (comparator mode)	○	○

Note: This register must always be accessed in halfwords.

The A-D0 Successive Approximation Register (AD0SAR) is used to read the conversion result of the A-D0 Converter when operating in A-D conversion mode or write a comparison value when operating in comparator mode.

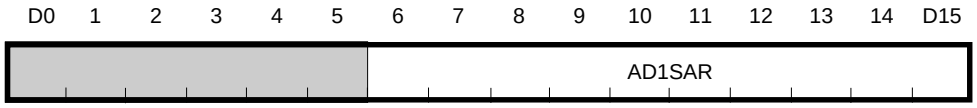
In A-D conversion mode, the A-D conversion is performed based on the successive approximation method. In this method, the reference voltage VREF and analog input voltages are compared bitwise sequentially beginning with the most significant bit, and the result is set in the A-D0 Successive Approximation Register (AD0SAR)'s each corresponding bit (D6-D15). When the A-D conversion finishes, the value of this register is transferred to the 10-bit A-D0 Data Register (AD0DTn) corresponding to the converted channel. When this register is read in the middle of A-D conversion, it shows an intermediate result of conversion.

In comparator mode, this register has a comparison value (the comparison voltage to "comparate") written in it. Comparate operation with the analog input pins set with Single Mode Register 1 is started upon writing to this register. When the comparate operation finishes, the result is stored in the A-D0 Comparate Data Register (AD0CMP).

The comparison value written to the A-D0 Successive Approximation Register (AD0SAR) during comparator mode is calculated by the equation below.

$$\text{Comparison value} = \text{H' 3FF} \times \frac{\text{Comparison voltage to comparate [V]}}{\text{VREF0 input voltage [V]}}$$

■ **A-D1 Successive Approximation Register (AD1SAR)** **<Address: H'0080 0A88>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6-15	AD1SAR (A-D1 successive approximation value/comparison value)	• A-D successive approximation value (A-D conversion mode) • Comparison value (comparator mode)	○	○

Note: This register must always be accessed in halfwords.

The A-D1 Successive Approximation Register (AD1SAR) is used to read the conversion result of the A-D1 Converter when operating in A-D conversion mode or write a comparison value when operating in comparator mode.

In A-D conversion mode, the A-D conversion is performed based on the successive approximation method. In this method, the reference voltage VREF and analog input voltages are compared bitwise sequentially beginning with the most significant bit, and the result is set in the A-D1 Successive Approximation Register (AD1SAR)'s each corresponding bit (D6-D15). When the A-D conversion finishes, the value of this register is transferred to the 10-bit A-D1 Data Register (AD1DTn) corresponding to the converted channel. When this register is read in the middle of A-D conversion, it shows an intermediate result of conversion.

In comparator mode, this register has a comparison value (the comparison voltage to "comparate") written in it. Comparate operation with the analog input pins set with Single Mode Register 1 is started upon writing to this register. When the comparate operation finishes, the result is stored in the A-D1 Comparate Data Register (AD1CMP).

The comparison value written to the A-D1 Successive Approximation Register (AD1SAR) during comparator mode is calculated by the equation below.

$$\text{Comparison value} = \text{H' 3FF} \times \frac{\text{Comparison voltage to comparate [V]}}{\text{VREF1 input voltage [V]}}$$

11.2.8 A-D Comparate Data Registers

■ **A-D0 Comparate Data Register (AD0CMP)** **<Address: H'0080 008C>**

D0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	D15
AD0 CMP0	AD0 CMP1	AD0 CMP2	AD0 CMP3	AD0 CMP4	AD0 CMP5	AD0 CMP6	AD0 CMP7	AD0 CMP8	AD0 CMP9	AD0 CMP10	AD0 CMP11	AD0 CMP12	AD0 CMP13	AD0 CMP14	AD0 CMP15

<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	AD0CMP0 AD0CMP15 (A-D0 comparate result flag)	0: Analog input voltage > comparison voltage 1: Analog input voltage < comparison voltage	○	–

Note 1: This register must always be accessed in halfwords.
Note 2: During comparator mode, each bit of this register corresponds to channel 0 through channel 15.

When comparator mode is selected with A-D0 Single Mode Register 1 AD0SMSL (A-D0 conversion mode select) bit, the selected analog input value and the value written to the A-D0 Successive Approximation Register are compared and the result is stored in this register's corresponding bit.

The result flag is 0 when the analog input voltage > comparison voltage, or 1 when the analog input voltage < comparison voltage.

■ **A-D1 Comparate Data Register (AD1CMP)** **<Address: H'0080 0A8C>**

D0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	D15
AD1 CMP0	AD1 CMP1	AD1 CMP2	AD1 CMP3	AD1 CMP4	AD1 CMP5	AD1 CMP6	AD1 CMP7	AD1 CMP8	AD1 CMP9	AD1 CMP10	AD1 CMP11	AD1 CMP12	AD1 CMP13	AD1 CMP14	AD1 CMP15

<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	AD1CMP0 AD1CMP15 (A-D1 comparate result flag)	0: Analog input voltage > comparison voltage 1: Analog input voltage < comparison voltage	○	–

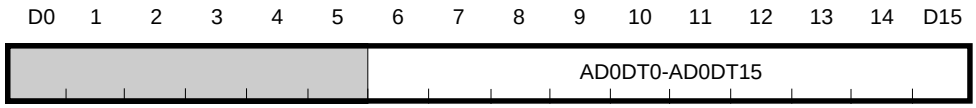
Note 1: This register must always be accessed in halfwords.
Note 2: During comparator mode, each bit of this register corresponds to channel 0 through channel 15.

When comparator mode is selected with A-D1 Single Mode Register 1 AD1SMSL (A-D1 conversion mode select) bit, the selected analog input value and the value written to the A-D1 Successive Approximation Register are compared and the result is stored in this register's corresponding bit.

The result flag is 0 when the analog input voltage > comparison voltage, or 1 when the analog input voltage < comparison voltage.

11.2.9 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>**



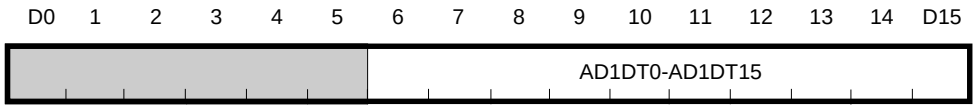
<When reset: indeterminate>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6-15	AD0DT0-AD0DT15 (A-D0 data)	A-D conversion result	○	–

Note: These registers must always be accessed in halfwords.

In A-D0 Converter single mode, the result of A-D conversion is stored in the 10-bit A-D0 Data Register of the corresponding channel.
In single-shot or continuous scan mode, each time A-D conversion on a channel is finished, the content of the A-D0 Successive Approximation Register is transferred to the 10-bit A-D Data Register of the corresponding channel.
Each 10-bit A-D Data Register holds the last conversion result until they receive the next conversion result transferred, so that the register content can be read out anytime.

■ **10-bit A-D1 Data Register 0 (AD1DT0)** **<Address: H'0080 0A90>**
■ **10-bit A-D1 Data Register 1 (AD1DT1)** **<Address: H'0080 0A92>**
■ **10-bit A-D1 Data Register 2 (AD1DT2)** **<Address: H'0080 0A94>**
■ **10-bit A-D1 Data Register 3 (AD1DT3)** **<Address: H'0080 0A96>**
■ **10-bit A-D1 Data Register 4 (AD1DT4)** **<Address: H'0080 0A98>**
■ **10-bit A-D1 Data Register 5 (AD1DT5)** **<Address: H'0080 0A9A>**
■ **10-bit A-D1 Data Register 6 (AD1DT6)** **<Address: H'0080 0A9C>**
■ **10-bit A-D1 Data Register 7 (AD1DT7)** **<Address: H'0080 0A9E>**
■ **10-bit A-D1 Data Register 8 (AD1DT8)** **<Address: H'0080 0AA0>**
■ **10-bit A-D1 Data Register 9 (AD1DT9)** **<Address: H'0080 0AA2>**
■ **10-bit A-D1 Data Register 10 (AD1DT10)** **<Address: H'0080 0AA4>**
■ **10-bit A-D1 Data Register 11 (AD1DT11)** **<Address: H'0080 0AA6>**
■ **10-bit A-D1 Data Register 12 (AD1DT12)** **<Address: H'0080 0AA8>**
■ **10-bit A-D1 Data Register 13 (AD1DT13)** **<Address: H'0080 0AAA>**
■ **10-bit A-D1 Data Register 14 (AD1DT14)** **<Address: H'0080 0AAC>**
■ **10-bit A-D1 Data Register 15 (AD1DT15)** **<Address: H'0080 0AAE>**



<When reset: indeterminate>

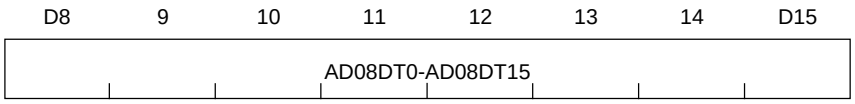
D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6-15	AD1DT0-AD1DT15 (A-D1 data)	A-D conversion result	○	–

Note: These registers must always be accessed in halfwords.

In A-D1 Converter single mode, the result of A-D conversion is stored in the 10-bit A-D1 Data Register of the corresponding channel.
In single-shot or continuous scan mode, each time A-D conversion on a channel is finished, the content of the A-D1 Successive Approximation Register is transferred to the 10-bit A-D Data Register of the corresponding channel.
Each 10-bit A-D Data Register holds the last conversion result until they receive the next conversion result transferred, so that the register content can be read out anytime.

11.2.10 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>**



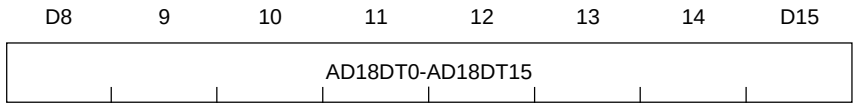
<When reset: indeterminate>

D	Bit Name	Function	R	W
8-15	AD08DT0-AD08DT15 (8-bit A-D0 data)	8-bit A-D conversion result	○	–

This A-D data register has the A-D0 Converter's 8-bit conversion result stored in it.

In A-D0 Converter single mode, the result of A-D conversion is stored in the 8-bit A-D0 Data Register of the corresponding channel.
In single-shot or continuous scan mode, each time A-D conversion on a channel is finished, the content of the A-D0 Successive Approximation Register is transferred to the 8-bit A-D0 Data Register of the corresponding channel.
Each 8-bit A-D0 Data Register holds the last conversion result until they receive the next conversion result transferred, so that the register content can be read out anytime.

■ **8-bit A-D1 Data Register 0 (AD18DT0)** **<Address: H'0080 0AD1>**
■ **8-bit A-D1 Data Register 1 (AD18DT1)** **<Address: H'0080 0AD3>**
■ **8-bit A-D1 Data Register 2 (AD18DT2)** **<Address: H'0080 0AD5>**
■ **8-bit A-D1 Data Register 3 (AD18DT3)** **<Address: H'0080 0AD7>**
■ **8-bit A-D1 Data Register 4 (AD18DT4)** **<Address: H'0080 0AD9>**
■ **8-bit A-D1 Data Register 5 (AD18DT5)** **<Address: H'0080 0ADB>**
■ **8-bit A-D1 Data Register 6 (AD18DT6)** **<Address: H'0080 0ADD>**
■ **8-bit A-D1 Data Register 7 (AD18DT7)** **<Address: H'0080 0ADF>**
■ **8-bit A-D1 Data Register 8 (AD18DT8)** **<Address: H'0080 0AE1>**
■ **8-bit A-D1 Data Register 9 (AD18DT9)** **<Address: H'0080 0AE3>**
■ **8-bit A-D1 Data Register 10 (AD18DT10)** **<Address: H'0080 0AE5>**
■ **8-bit A-D1 Data Register 11 (AD18DT11)** **<Address: H'0080 0AE7>**
■ **8-bit A-D1 Data Register 12 (AD18DT12)** **<Address: H'0080 0AE9>**
■ **8-bit A-D1 Data Register 13 (AD18DT13)** **<Address: H'0080 0AEB>**
■ **8-bit A-D1 Data Register 14 (AD18DT14)** **<Address: H'0080 0AED>**
■ **8-bit A-D1 Data Register 15 (AD18DT15)** **<Address: H'0080 0AEF>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
8-15	AD18DT0-AD18DT15 (8-bit A-D0 data)	8-bit A-D1 conversion result	○	–

This A-D data register has the A-D1 Converter's 8-bit conversion result stored in it.

In A-D1 Converter single mode, the result of A-D conversion is stored in the 8-bit A-D1 Data Register of the corresponding channel.
In single-shot or continuous scan mode, each time A-D conversion on a channel is finished, the content of the A-D1 Successive Approximation Register is transferred to the 8-bit A-D1 Data Register of the corresponding channel.
Each 8-bit A-D1 Data Register holds the last conversion result until they receive the next conversion result transferred, so that the register content can be read out anytime.

11.3 Functional Description of the A-D Converters

11.3.1 How to Find Analog Input Voltages

The A-D converters each use a 10-bit successive approximation method. Therefore, to find the actual analog input voltages from the digital quantities obtained as a result of A-D conversion, use the equation below.

$$\text{Analog input voltage [V]} = \frac{\text{A-D conversion result x VREF input voltage [V]}}{1024}$$

The A-D converters each are comprised of 10 bits, so they provide a resolution of 1,024. Because the voltage applied to the VREF pin constitutes the reference voltage for the A-D converter, connect an accurate and stable, constant-voltage power supply to the VREF. Also, separate the analog block power supply and ground (AVCC, AVSS) from those of the digital block and take a sufficient protective measure against noise.
For details about the accuracy of conversion, see Section 11.3.5, "Accuracy of A-D Conversion."

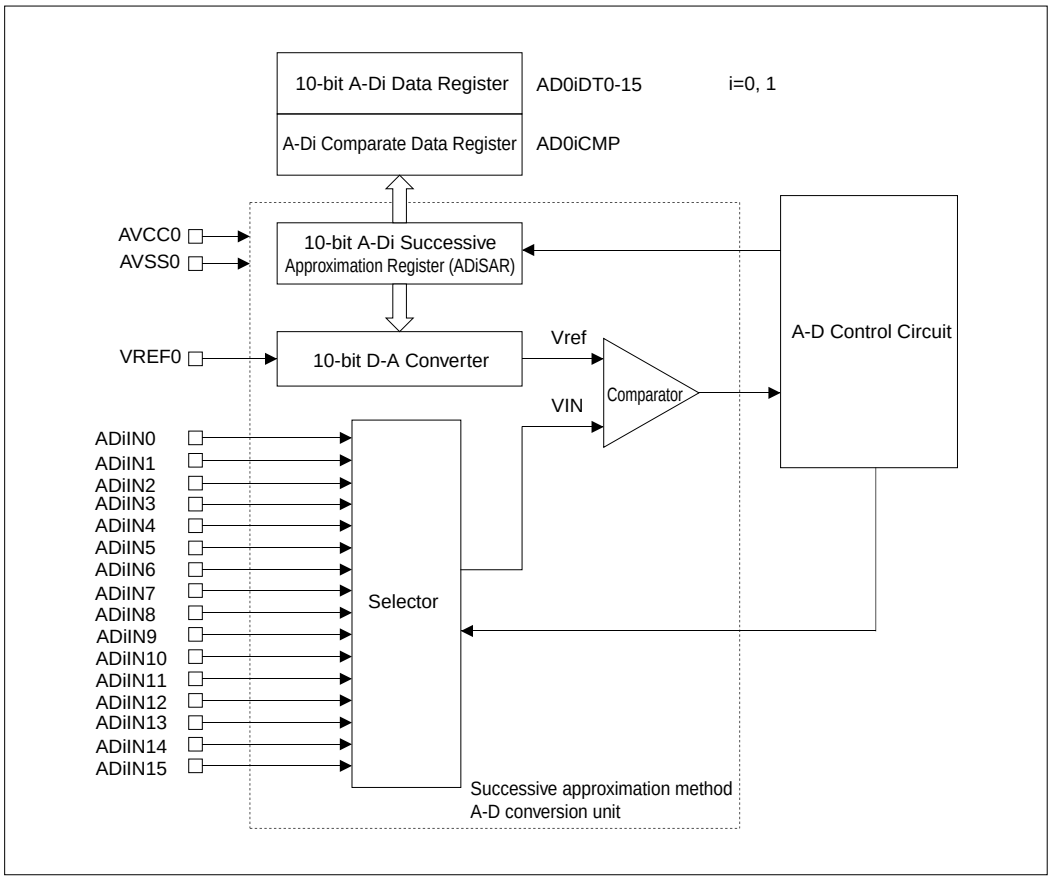


Figure 11.3.1 Block Diagram of the Successive Approximation Method A-D0 Conversion Unit

11.3.2 A-D Conversion of Successive Approximation Method

The A-D converter starts A-D convert operation as initiated by a hardware or software A-D conversion start trigger. One A-D conversion starts, the following operations are executed automatically.

(a) During single mode, clear the Single Mode Register 0 A-D conversion/comparate complete bit to 0. During scan mode, clear the Scan Mode Register 0 A-D conversion complete bit to 0.
(b) Clear the content of the A-D Successive Approximation Register to H'0000.
(c) Set the A-D Successive Approximation Register's most significant bit (D6) to 1.
(d) Enter the comparison voltage Vref (Note) from the D-A Converter into the Comparator.
(e) Compare the comparison voltage Vref and the analog input voltage VIN. Next, store the comparison result in D6.

If Vref < VIN, then D6 = 1
If Vref > VIN, then D6 = 0

(f) Execute the above operations (c) through (e) on all other bits D7-D15.
(g) When comparison on D15 finishes, the value stored in the A-D Successive Approximation Register is made the valid A-D conversion result.

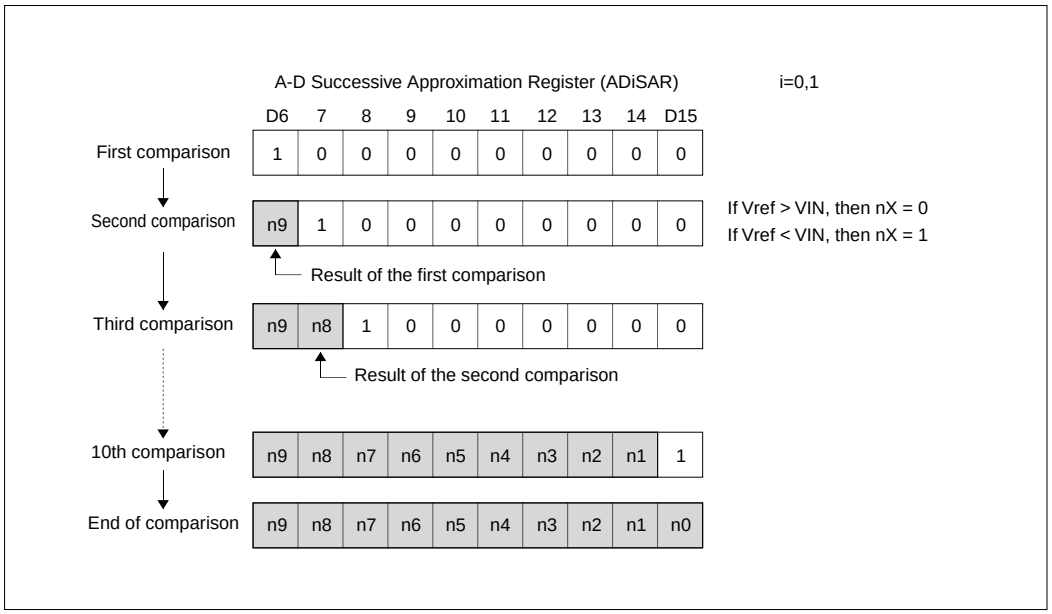


Figure 11.3.2 Change of A-D Successive Approximation Register Content during A-D Conversion

Note: The comparison voltage Vref (the voltage entered from the D-A Converter into the Comparator) is determined according to changes of the content of the A-D Successive Approximation Register. The comparison voltage Vref is calculated by the equation below.

- When the content of the A-D Successive Approximation Register = 0
  Vref [V] = 0
- When the content of the A-D Successive Approximation Register = 1 to 1023
  Vref [V] = (reference voltage VREF / 1024) x (content of the A-D Successive Approximation Register-0.5)

The comparison result is stored in the 10-bit A-D Data Registers (AD0DTn or AD1DTn) corresponding to the converted channel. Also, the 8 high-order bits of the A-D conversion result can be obtained by reading the 8-bit A-D Data Registers (AD08DTn or AD18DTn).

The following shows how A-D conversion of the successive approximation method is performed in each operation mode.

(1) For single mode

The convert operation stops when comparison on the A-D Successive Approximation Register D15 bit is completed. The content of the A-D Successive Approximation Register (A-D conversion result) is transferred to the 10-bit A-D Data Registers 0-15 of the converted channel.

(2) For single-shot scan mode

When comparison on the A-D Successive Approximation Register D15 bit for 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 (b) through (g) described above are executed for the next channel to be converted.
In single-shot scan mode, convert operation stops when A-D conversion in one specified scan loop is finished.

(3) For continuous scan mode

When comparison on the A-D Successive Approximation Register D15 bit for 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 (b) through (g) described above are executed for the next channel to be converted.
In continuous scan mode, convert operations are executed continuously until scan operation is forcibly stopped by setting the A-D conversion stop bit (Scan Mode Register 0 D6 bit) to 1.

11.3.3 Comparator Operation

When comparator mode (single mode only) is selected, the A-D converter functions as a comparator to compare the comparison voltage set in software and the analog input voltages.

Upon writing a comparison value to the Successive Approximation Register, the A-D converter starts "comparating" the analog input voltages selected with the Single Mode Register 1 analog input select bits and the value written to the Successive Approximation Register. Once comparate operation starts, the following operations are executed automatically.

(a) Clear the Single Mode Register 0 or Scan Mode Register 0 A-D conversion/comparate complete flag to 0.
(b) Enter the comparison voltage Vref (Note) from the D-A Converter into the Comparator.
(c) Compare the comparison voltage Vref and the analog input voltage VIN. Next, store the comparison result in the comparate result flag (A-D Comparate Data Register D15 bit).
    If Vref < VIN, then the comparate result flag = 0
    If Vref > VIN, then the comparate result flag = 1
(d) Stop comparate operation after storing the comparison result.

The comparison result is stored in the A-D Comparate Data Register (AD0CMP or AD1CMP)'s corresponding bits.

Note: The comparison voltage Vref (the voltage entered from the D-A Converter into the Comparator) is determined according to changes of the content of the A-D Successive Approximation Register. The comparison voltage Vref is calculated by the equation below.

- When the content of the A-D Successive Approximation Register = 0
  Vref [V] = 0
- When the content of the A-D Successive Approximation Register = 1 to 1023
  Vref [V] = (reference voltage VREF / 1024) x (content of the A-D Successive Approximation Register - 0.5)

11.3.4 Calculating the A-D Conversion Time

The A-D conversion time is expressed by the sum of a dummy cycle time and the actual execution cycle time. The terms needed for calculation of the conversion time are listed below.

(a) Start dummy time

The time from when the CPU executes an A-D conversion start instruction to when the A-D Converter starts A-D conversion

(b) A-D conversion execution cycle time

(c) Comparate execution cycle time

(d) End dummy time

The time from when the A-D Converter finished A-D conversion to when the CPU can stably read out the conversion result from the A-D Data Register.

(e) Scan interval dummy time

The time during single-shot or continuous scan mode from when the A-D Converter finishes 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 interval dummy time + execution cycle time
+ scan interval dummy time + execution cycle time
+ scan interval dummy time ... + execution cycle time)
+ end dummy time

Note: Enclosed in parentheses ( ) are the conversion time for the second and subsequent channels in scan mode.

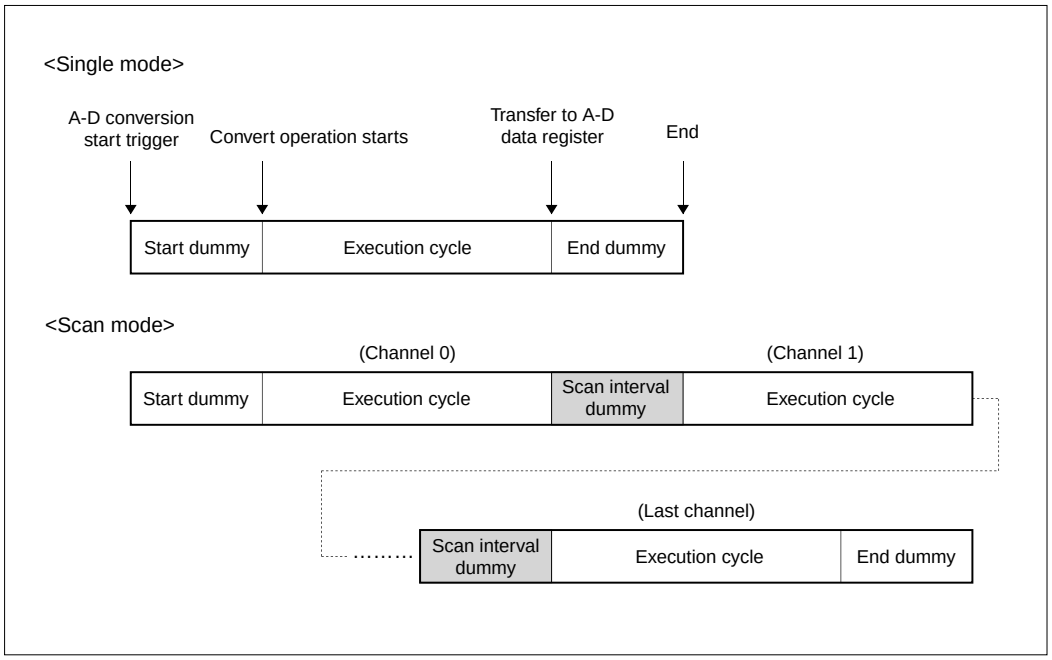


Figure 11.3.3 Conceptual Diagram of A-D Conversion Time

Table 11.3.1 Conversion Time in Terms of Clock Periods

Unit: BCLK

Transfer speed		Start dummy (Note 1)	Start dummy (Note 2)	Start dummy (Note 3)	A-D conversion execution cycle	Comparate execution cycle	End dummy	Scan interval dummy (Note 4)
Low speed	Normal	4	4	4	294	42	1	4
mode	Double speed	4	4	4	168	24	1	4
High speed	Normal	4	4	4	126	18	1	4
mode	Double speed	4	4	4	84	12	1	4

Note 1: When started by a software trigger.
Note 2: When started by a hardware trigger.
Note 3: When writing to the A-D Successive Approximation Register (comparate mode)
Note 4: In only scan operation, the execution time is added up for each channel.

Table 11.3.2 A-D Conversion Time when Started by Software Trigger (Total Time)

Conversion started by	Conversion speed	Conversion mode (Note 1)		Conversion time [BCLK]
Software trigger (Note 2)	Low speed mode: Normal	Single mode		299
		Single-shot /continuous scan mode	n channels scanned	(298 × n)+1
		Comparator mode		47
	Low speed mode: Double	Single mode		173
		Single-shot /continuous scan mode	n channels scanned	(172 × n)+1
		Comparator mode		29
	High speed mode: Normal	Single mode		131
		Single-shot /continuous scan mode	n channels scanned	(130 × n)+1
		Comparator mode		23
	High speed mode: Double	Single mode		89
		Single-shot /continmuous scan mode	n channels scanned	(88 × n)+1
		Comparator mode		17

Note 1: For single and comparator modes, this refers to the A-D conversion or comparate time needed for one channel. For single-shot and continuous scan modes, this refers to the A-D conversion time needed for one scan loop.

Note 2: This refers to the time from when a register write cycle is finished to when an A-D conversion interrupt request is generated.

Table 11.3.3 A-D Conversion Time when Started by Hardware Trigger (Total Time)

Conversion started by	Conversion speed	Conversion mode (Note 1)	Conversion time [BCLK]
Hardware trigger (Note 2)	Low speed mode: Normal	Single mode	299
		Single-shot /continuous scan mode	n channels scanned (298 × n)+1
		Comparator mode	47
	Low speed mode: Double	Single mode	173
		Single-shot /continuous scan mode	n channels scanned (172 × n)+1
		Comparator mode	29
	High speed mode: Normal	Single mode	131
		Single-shot /continuous scan mode	n channels scanned (130 × n)+1
		Comparator mode	23
	High speed mode: Double	Single mode	89
		Single-shot /continuous scan mode	n channels scanned (88 × n)+1
		Comparator mode	17

Note 1: For single and comparator modes, this refers to the A-D conversion or comparate time needed for one channel. For single-shot and continuous scan modes, this refers to the A-D conversion time needed for one scan loop.

Note 2: For the A-D0 Converter, this refers to the time from TOM0_6 underflow, input on external pin TIN16, TOM0_0-7 enable event, or completion of A-D1 conversion to when an A-D conversion interrupt request is generated. For the A-D1 Converter, this refers to the time from TOM0_6 underflow, input on external pin TIN16, TOM1_6 underflow, or completion of A-D0 conversion to when an A-D1 conversion interrupt request is generated.

11.3.5 Definition of the A-D Conversion Accuracy

The accuracy of the A-D Converter is expressed by absolute accuracy. Absolute accuracy refers to the difference, expressed in terms of LSB, between the output code actually obtained by A-D converting the analog input voltage and the output code that can be expected from an A-D converter with ideal characteristics.
The analog input voltages used during accuracy measurement are chosen to be the midpoint values of voltage width at which an A-D converter with ideal characteristics will produce the same output code. For example, when VREF0 = 5.12 V, the width of 1 LSB of a 10-bit A-D converter is 5 mV, so that the middle points of analog input voltages are chosen to be 0 mV, 5 mV, 10 mV, 15 mV, 20 mV, 25 mV, and so on.

If the absolute accuracy of an A-D converter is said to be +2 LSB, it means that if the input voltage is 25 mV, for example, the actual A-D conversion result is in the range of H'003 to H'007, whereas the output code that can be expected from an ideal A-D converter is H'005. Note that absolute accuracy includes a zero error and full-scale error.

Although when actually using the A-D Converter, the analog input voltages are in the range of AVSS0 to VREF0, excessively lowering the VREF0 voltage requires caution because resolution may be degraded. Note also that output codes for analog input voltages from VREF0 to AVCC0 are always H'3FF.

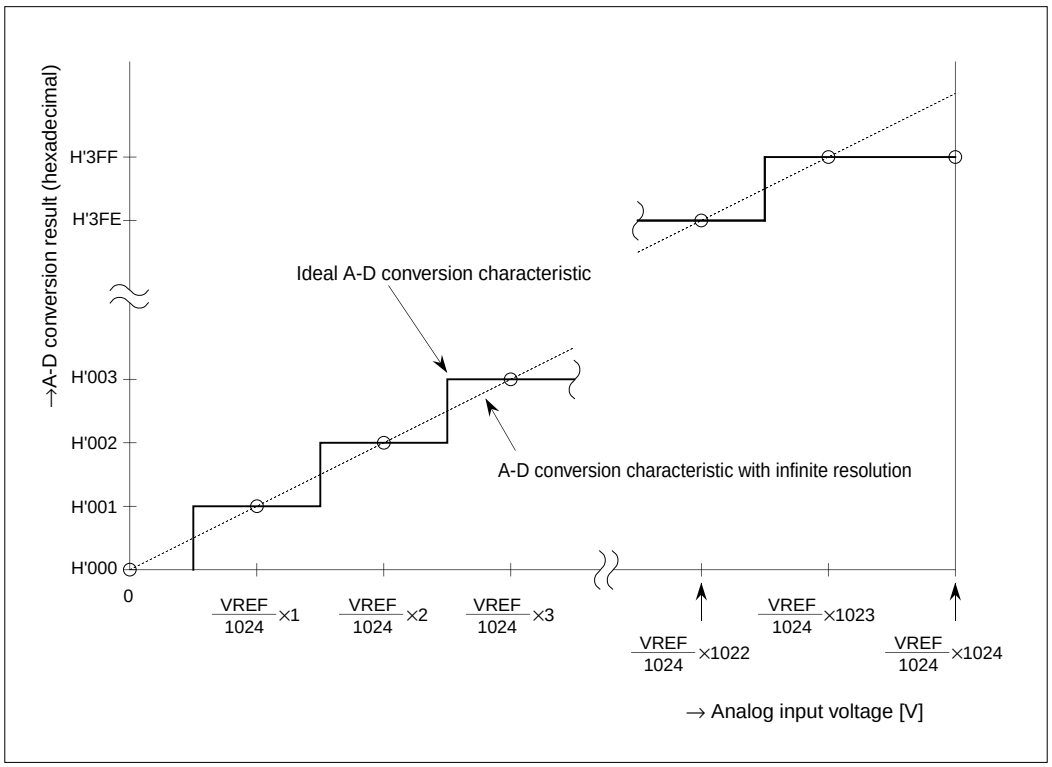


Figure 11.3.4 Ideal A-D Conversion Characteristics Relative to Analog Input Voltages of a 10-bit A-D Converter

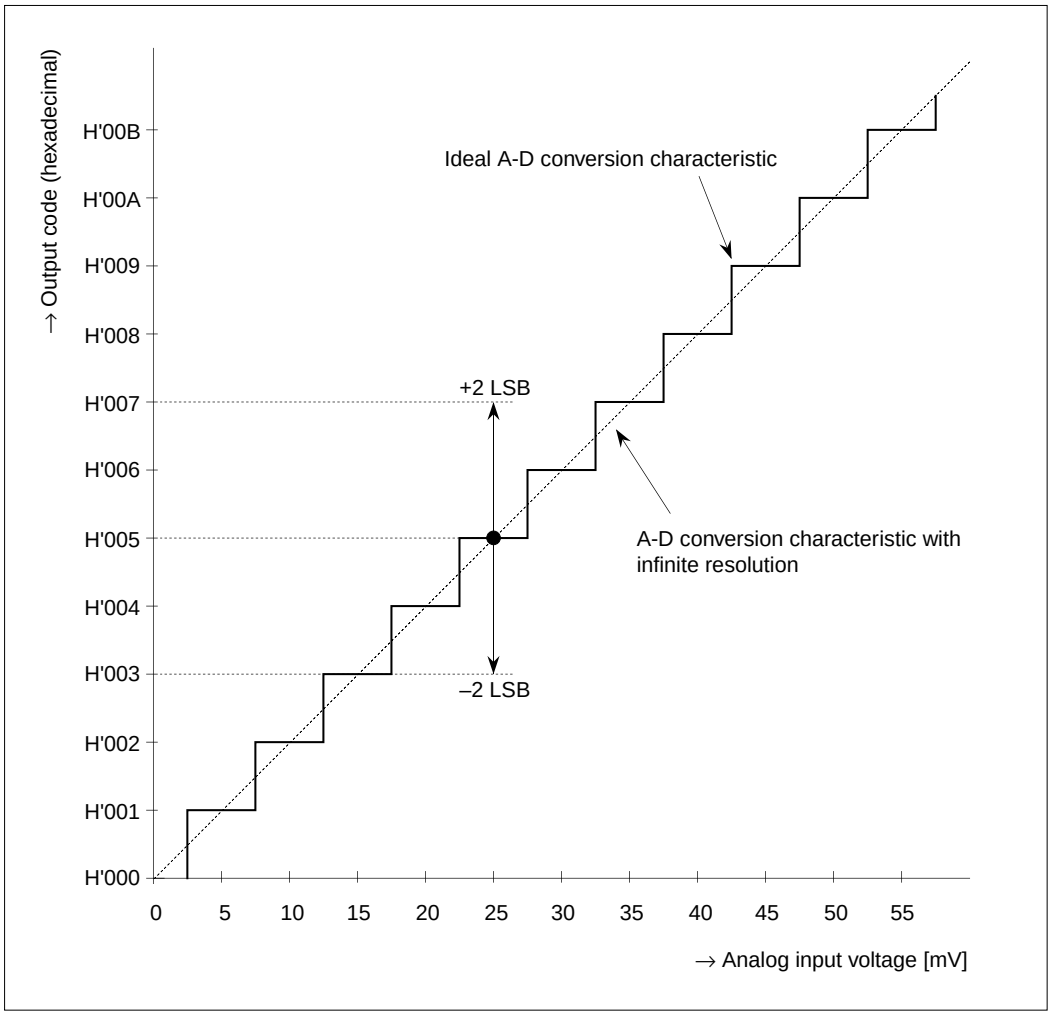


Figure 11.3.5 Absolute Accuracy of the A-D Converter

11.4 Precautions on Using the A-D Converters

- **Forcible termination during scan operation**

  If A-D conversion is forcibly stopped during scan mode operation by setting the A-D conversion stop bit (AD0CSTP or AD1CSTP) to 1 and the A-D data register for the channel in the middle of conversion is read, the data obtained by this read is the last conversion result that was transferred to the register before being forcibly stopped.

- **Modification of A-D converter related registers**

  The A-D Conversion Interrupt Control Register (except for the A-D conversion stop bit), each mode register, and A-D Successive Approximation Register can only be modified while A-D conversion is not in operation. Otherwise, restart A-D conversion after modifying the register contents. If any register is modified during A-D conversion, the conversion result cannot be guaranteed.

- **Handling of analog input signals**

  The A-D Converter described here does not contain a sample-and-hold circuit. Therefore, make sure the analog input levels are fixed during A-D conversion.

- **Timing at which to read the A-D conversion complete bit**

  If it is necessary to read the A-D conversion complete bit (Single Mode Register 0 D5 bit or Scan Mode Register 0 D5 bit) immediately after starting A-D conversion, be sure to insert a NOP instruction, etc. to adjust the timing for one cycle before reading the bit.

- **Rated value of absolute accuracy**

  The rated value of absolute accuracy is that of the microcomputer's own as a single unit which can be exhibited when the microcomputer is used in an environment where it may not be affected by the power supply wiring or noise on the board. When designing the board, pay careful attention to its layout by, for example, separating AVCC0, AVSS0, and VREF0 from other digital block power supplies, or by making analog input pins unsusceptible to influences from other digital signal lines (e.g., noise).

CHAPTER 12
SERIAL I/O

12.1 Outline of Serial I/O

The 32172/32173 contains a total of eight serial I/O channels, SIO0 through SIO7. Serial channels SIO0, SIO1, SIO4, and SIO5 can be selected between CSIO mode (clock-synchronous serial I/O) and UART mode (asynchronous serial I/O). SIO2, SIO3, SIO6, and SIO7 are used in UART mode only.

- CSIO mode (clock-synchronous serial I/O)

  The communication performed in this mode is synchronized to the transfer clock, using the same clock on both transmit and receive sides. Data are transferred in a fixed length of 8 bits.

- UART mode (asynchronous serial I/O)

  The communication performed in this mode is asynchronous. The transfer data length can be selected from 7, 8, or 9 bits.

Serial I/Os 0-7 each have transmit DMA and receive DMA transfer requests. Through a combined use with the internal DMA, they allow for fast serial communication, and help to reduce the data communication load on the CPU.

Serial I/Os are outlined below.

Table 12.1.1 Outline of Serial I/O

Item	Content
Number of channels	CSIO/UART: 4 channels (SIO0,SIO1,SIO4,SIO5) UART-only : 4 channels (SIO2,SIO3,SIO6,SIO7)
Clock	During CSIO mode : Internal clock or external clock selectable (Note 1) During UART mode : Internal clock only
Transfer mode	Transmit half-duplex, receive half-duplex, transmit/receive full-duplex
BRG count source	f(BCLK), f(BCLK)/8, f(BCLK)/32, f(BCLK)/256 (when internal peripheral clock is selected) (Note 2) f(BCLK): Internal 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 = With or without (odd/even selectable) Stop bit = 1 or 2 bits Order of transfer: LSB first (fixed)
Baud rate	CSIO mode : 152 bits/second to 2 Mbits/second (when f(BCLK) = 20 MHz) UART mode : 19 bits/second to 156 Kbits/second (when f(BCLK) = 20 MHz)
Error detection	CSIO mode : Overrun error only UART mode : Overrun, parity, and framing errors (Any error if occurs is indicated by an errorsum bit)
Fixed period clock output function	When using SIO0, SIO1, SIO4, or SIO5 in UART mode, this function outputs a divided-by-2 clock of BRG from the SCLK pin.

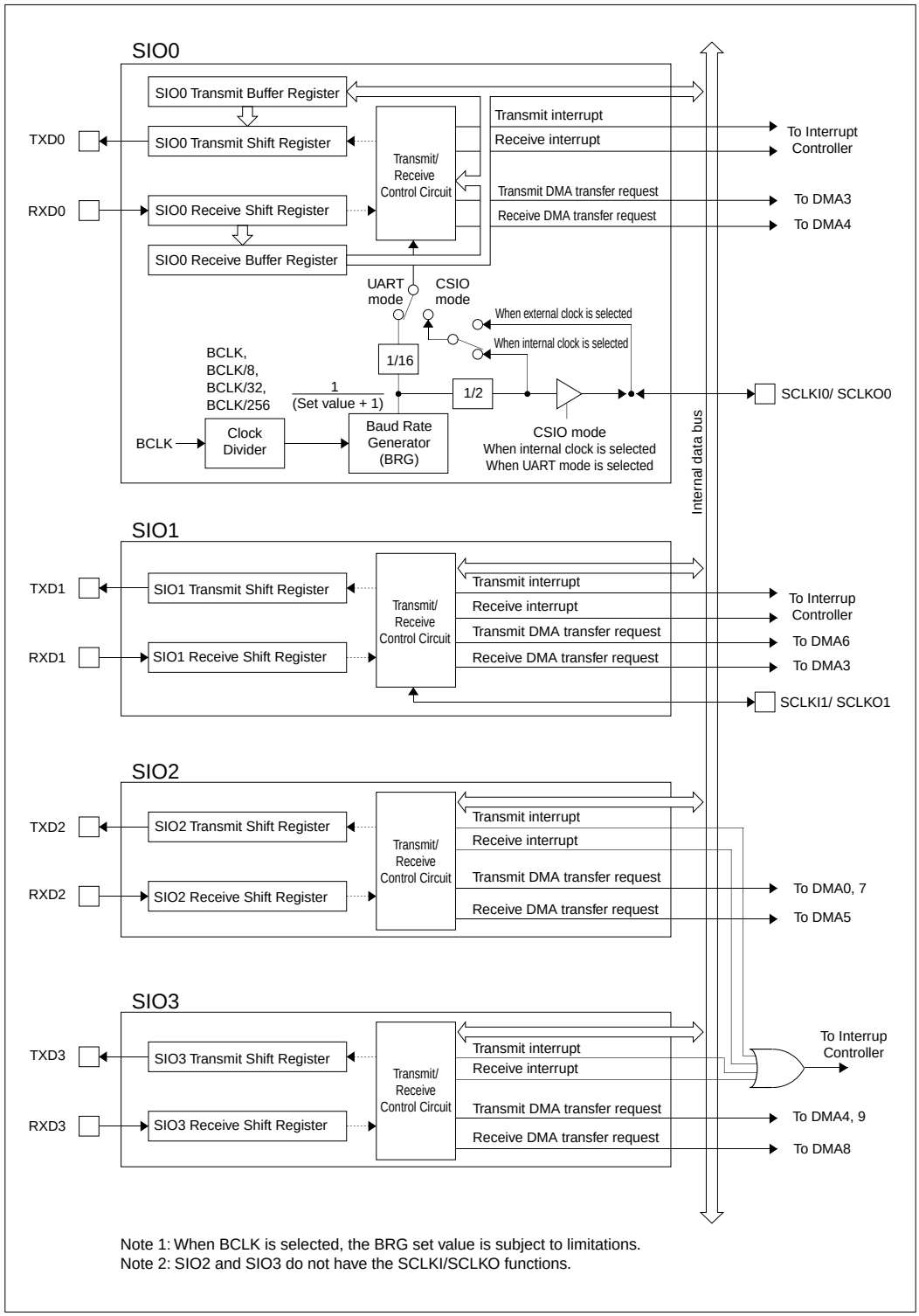
Note 1: During CSIO mode, the maximum input frequency of an external clock is f(BCLK) divided by 16.
Note 2: If f(BCLK) is selected for the BRG count source, the BRG set value is subject to limitations.

Table 12.1.2 Interrupt Request Generation Functions of Serial I/O

Serial I/O interrupt request	Interrupt source on ICU
SIO0 transmit buffer empty interrupt	SIO0 transmit interrupt
SIO0 receive complete or receive error interrupt (selectable)	SIO0 receive interrupt
SIO1 transmit buffer empty interrupt	SIO1 transmit interrupt
SIO1 receive complete or receive error interrupt (selectable)	SIO1 receive interrupt
SIO2 transmit buffer empty interrupt	SIO2,3 Transmit/Receive Interrupt (Group Interrupt)
SIO2 receive complete or receive error interrupt (selectable)	SIO2,3 Transmit/Receive Interrupt (Group Interrupt)
SIO3 transmit buffer empty interrupt	SIO2,3 Transmit/Receive Interrupt (Group Interrupt)
SIO3 receive complete or receive error interrupt (selectable)	SIO2,3 Transmit/Receive Interrupt (Group Interrupt)
SIO4 transmit buffer empty interrupt	SIO4 transmit interrupt
SIO4 receive complete or receive error interrupt (selectable)	SIO4 receive interrupt
SIO5 transmit buffer empty interrupt	SIO5 transmit interrupt
SIO5 receive complete or receive error interrupt (selectable)	SIO5 receive interrupt
SIO6 transmit buffer empty interrupt	SIO6,7 Transmit/Receive Interrupt (Group Interrupt)
SIO6 receive complete or receive error interrupt (selectable)	SIO6,7 Transmit/Receive Interrupt (Group Interrupt)
SIO7 transmit buffer empty interrupt	SIO6,7 Transmit/Receive Interrupt (Group Interrupt)
SIO7 receive complete or receive error interrupt (selectable)	SIO6,7 Transmit/Receive Interrupt (Group Interrupt)

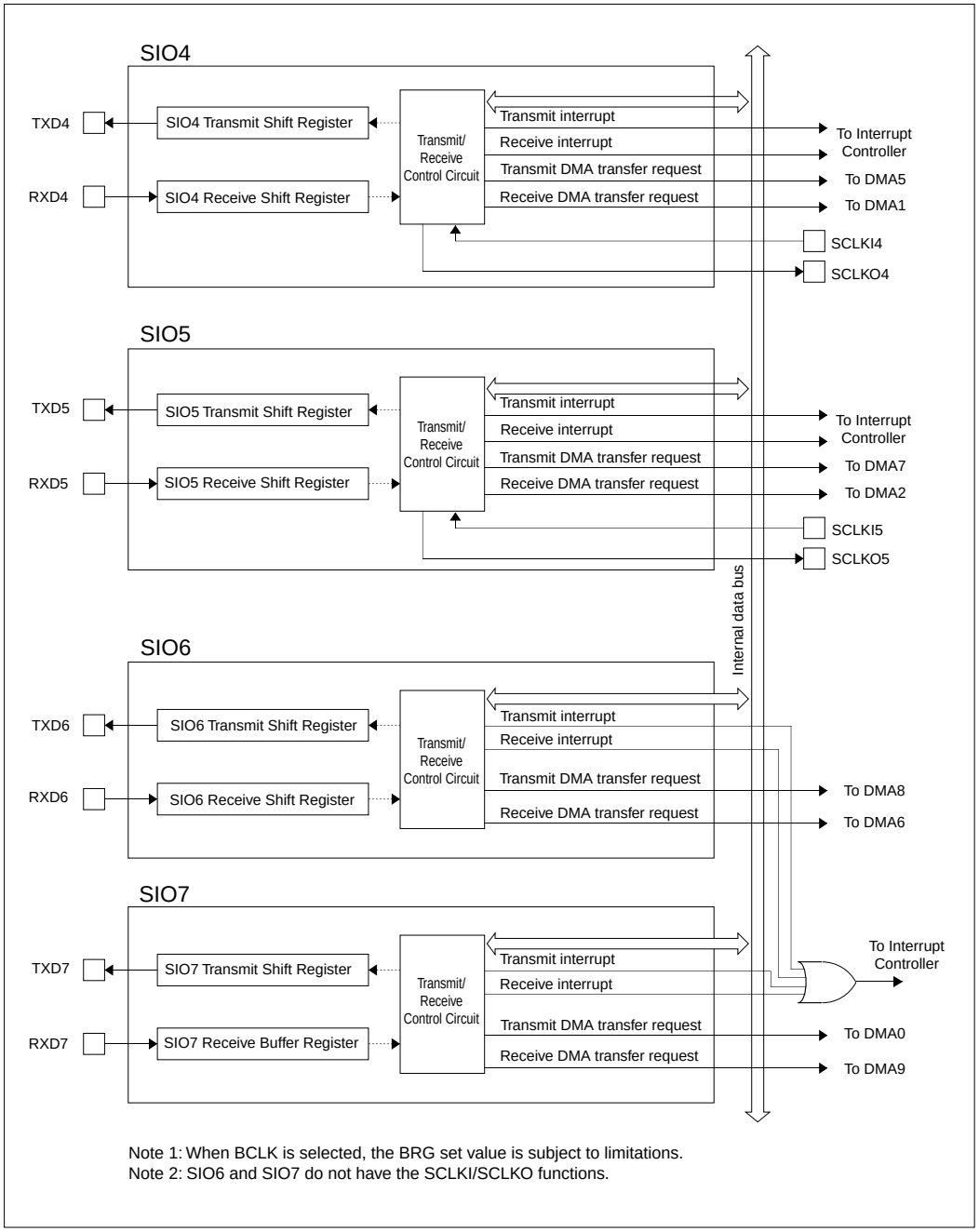
Table 12.1.3 DMA Transfer Request Generation Functions of Serial I/O

Serial I/O DMA transfer request	DMA input channel
SIO0 transmit buffer empty	Channel 3
SIO0 receive complete	Channel 4
SIO1 transmit buffer empty	Channel 6
SIO1 receive complete	Channel 3
SIO2 transmit buffer empty	Channels 0 and 7
SIO2 receive complete	Channel 5
SIO3 transmit buffer empty	Channels 4 and 9
SIO3 receive complete	Channel 8
SIO4 transmit buffer empty	Channel 5
SIO4 receive complete	Channel 1
SIO5 transmit buffer empty	Channel 7
SIO5 receive complete	Channel 2
SIO6 transmit buffer empty	Channel 8
SIO6 receive complete	Channel 6
SIO7 transmit buffer empty	Channel 0
SIO7 receive complete	Channel 9



Note 1: When BCLK is selected, the BRG set value is subject to limitations.
Note 2: SIO2 and SIO3 do not have the SCLKI/SCLKO functions.

Figure 12.1.1 Block Diagram of SIO0-SIO3



Note 1: When BCLK is selected, the BRG set value is subject to limitations.
Note 2: SIO6 and SIO7 do not have the SCLKI/SCLKO functions.

Figure 12.1.2 Block Diagram of SIO4-SIO7

12.2 Serial I/O Related Registers

A serial I/O related register map is shown below.

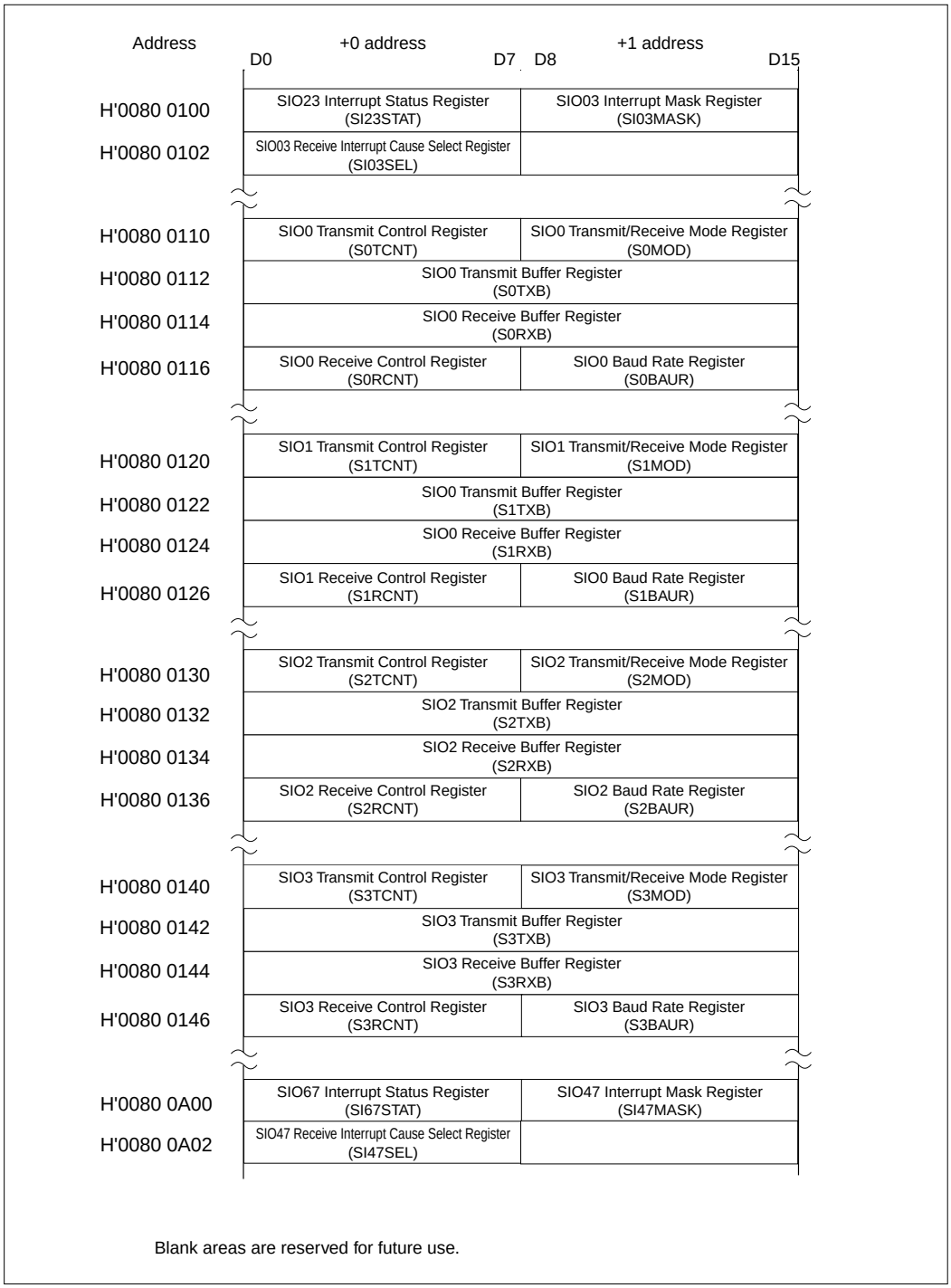


Figure 12.2.1 Serial I/O Related Register Map (1/2)

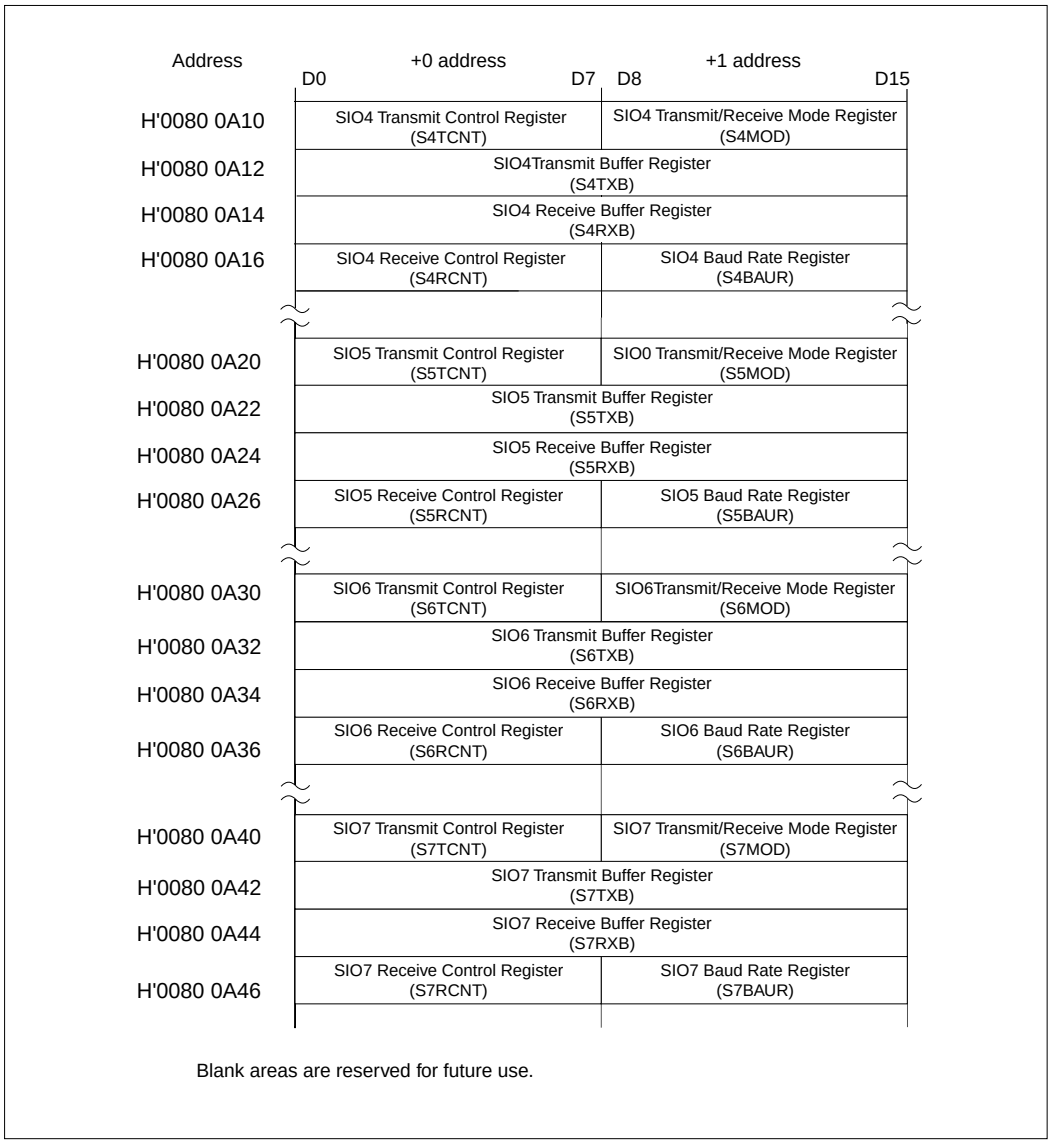


Figure 12.2.2 Serial I/O Related Register Map (2/2)

12.2.1 SIO Interrupt Related Registers

(1) Selecting interrupt sources

The interrupt signals output from each SIO to the ICU (Interrupt Controller) consist of transmit and receive interrupts. Transmit interrupts are generated when the transmit buffer is empty. Receive interrupts can be selected between receive complete and receive error interrupts by using the Receive Interrupt Source Select Register (SI03SEL or SI47SEL).

Note 1: Any interrupt signal can only be generated by enabling the TEN (Transmit Enable) or REN (Receive Enable) bit for its corresponding SIO and then enabling the interrupt with the SIO Interrupt Mask Register.
Note 2: SIO2 and SIO3 together comprise one group interrupt, so do SIO6 and SIO7.

(2) Notes on transmit interrupts

A transmit interrupt is generated by enabling its corresponding TEN (Transmit Enable) bit while the SIO Interrupt Mask Register is enabled for interrupts.

(3) Regarding DMA transfer requests from SIO

Each SIO can generate transmit DMA transfer and receive-complete DMA transfer requests. These DMA transfer requests can be generated by enabling the corresponding TEN (Transmit Enable) or TEN (Receive Enable) bit for the SIO concerned.
When using DMA transfers to communicate with the microcomputer, please be sure to set up DMA before enabling the TEN and REN bits.
If a receive error occurs, no receive-complete DMA transfer requests are generated.

- **Transmit DMA transfer requests**

  This DMA transfer request is generated when the transmit buffer is empty while the TEN bit is enabled.

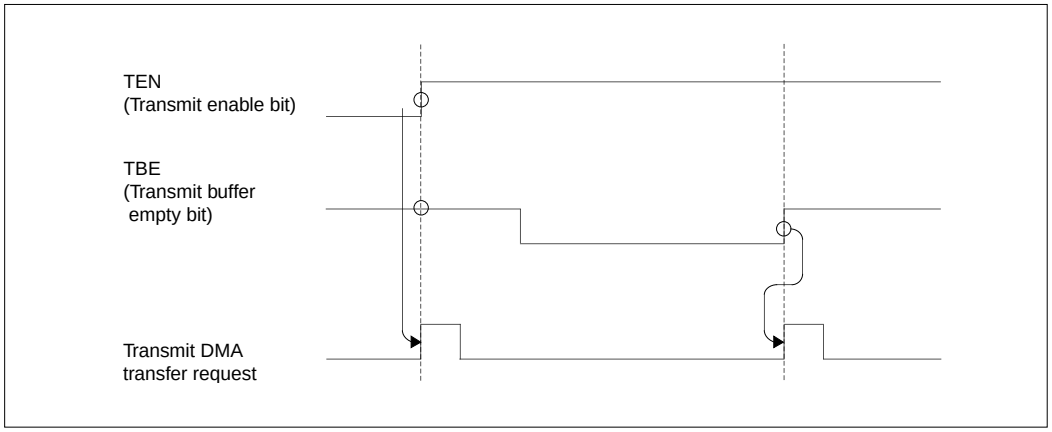
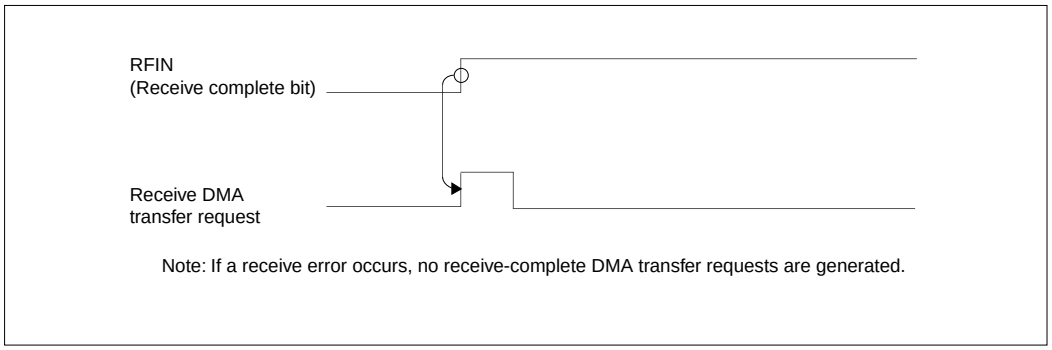


Figure 12.2.3 Transmit DMA Transfer Request

• **Receive-complete DMA transfer requests**

This DMA transfer request is generated when the receive buffer is full.

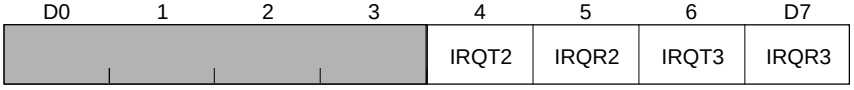


Note: If a receive error occurs, no receive-complete DMA transfer requests are generated.

Figure 12.2.4 Receive-complete DMA Transfer Request

12.2.2 SIO Interrupt Control Registers

■ **SIO23 Interrupt Status Register (SI23STAT)** **<Address: H'0080 0100>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4	IRQT2 (SIO2 transmit-complete interrupt request status bit)	0: Interrupt not requested 1: Interrupt requested	○	△
5	IRQR2 (SIO2 receive interrupt request status bit)	0: Interrupt not requested 1: Interrupt requested	○	△
6	IRQT3 (SIO3 transmit-complete interrupt request status bit)	0: Interrupt not requested 1: Interrupt requested	○	△
7	IRQR3 (SIO3 receive interrupt request status bit)	0: Interrupt not requested 1: Interrupt requested	○	△

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Transmit/receive interrupt requests from SIO2 and SIO3 are described below.

[Setting the interrupt request status bit]

The interrupt request status bit is set in hardware, and cannot be set in software.

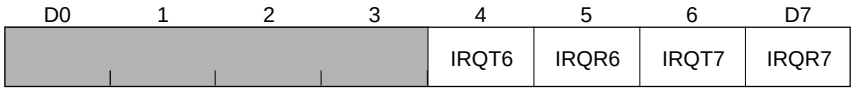
[Clearing the interrupt request status bit]

The interrupt request status 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 so that the status bit is set.

When writing to the SIO Interrupt Request Status Register, be sure to write 0 for the bits to be cleared and 1 for all other bits. Writing 1 in software does not affect any bit of this register, the bit retains the value it had before writing.

■ **SIO67 Interrupt Status Register (SI67STAT)** **<Address: H'0080 0A00>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4	IRQT6 (SIO6 transmit-complete interrupt request status bit)	0: Interrupt not requested 1: Interrupt requested	○	△
5	IRQR6 (SIO6 receive interrupt request status bit)	0: Interrupt not requested 1: Interrupt requested	○	△
6	IRQT7 (SIO7 transmit-complete interrupt request status bit)	0: Interrupt not requested 1: Interrupt requested	○	△
7	IRQR7 (SIO7 receive interrupt request status bit)	0: Interrupt not requested 1: Interrupt requested	○	△

W=△: Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Transmit/receive interrupt requests from SIO6 and SIO7 are described below.

[Setting the interrupt request status bit]

The interrupt request status bit is set in hardware, and cannot be set in software.

[Clearing the interrupt request status bit]

The interrupt request status 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 so that the status bit is set.

When writing to the SIO Interrupt Request Status Register, be sure to write 0 for the bits to be cleared and 1 for all other bits. Writing 1 in software does not affect any bit of this register, the bit retains the value it had before writing.

■ **SIO03 Interrupt Mask Register (SI03MASK)** **<Address: H'0080 0101>**

D8	9	10	11	12	13	14	D15
T0MASK	R0MASK	T1MASK	R1MASK	T2MASK	R2MASK	T3MASK	R3MASK

<When reset: H'00>

D	Bit Name	Function	R	W
8	T0MASK (SIO0 transmit interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
9	R0MASK (SIO0 receive interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
10	T1MASK (SIO1 transmit interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
11	R1MASK (SIO1 receive interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
12	T2MASK (SIO2 transmit interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
13	R2MASK (SIO2 receive interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
14	T3MASK (SIO3 transmit interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
15	R3MASK (SIO3 receive interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○

This register controls interrupt requests sent from each SIO by enabling or disabling them. An interrupt request from any SIO is enabled by setting its corresponding interrupt mask bit to 1.

■ **SIO47 Interrupt Mask Register (SI47MASK)** **<Address: H'0080 0A01>**

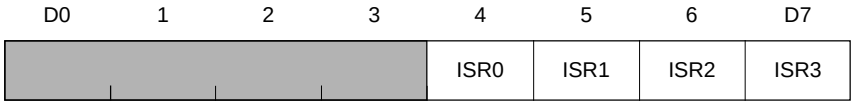
D8	9	10	11	12	13	14	D15
T4MASK	R4MASK	T5MASK	R5MASK	T6MASK	R6MASK	T7MASK	R7MASK

<When reset: H'00>

D	Bit Name	Function	R	W
8	T4MASK (SIO4 transmit interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
9	R4MASK (SIO4 receive interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
10	T5MASK (SIO5 transmit interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
11	R5MASK (SIO5 receive interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
12	T6MASK (SIO6 transmit interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
13	R6MASK (SIO6 receive interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
14	T7MASK (SIO7 transmit interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
15	R7MASK (SIO7 receive interrupt mask bit)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○

This register controls interrupt requests sent from each SIO by enabling or disabling them. An interrupt request from any SIO is enabled by setting its corresponding interrupt mask bit to 1.

■ **SIO03 Receive Interrupt Cause Select Register (SI03SEL)** **<Address: H'0080 0102>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4	ISR0	0: Receive complete interrupt	○	○
	(SIO0 receive interrupt cause select bit)	1: Receive error interrupt		
5	ISR1	0: Receive complete interrupt	○	○
	(SIO1 receive interrupt cause select bit)	1: Receive error interrupt		
6	ISR2	0: Receive complete interrupt	○	○
	(SIO2 receive interrupt cause select bit)	1: Receive error interrupt		
7	ISR3	0: Receive complete interrupt	○	○
	(SIO3 receive interrupt cause select bit)	1: Receive error interrupt		

This register selects the cause of interrupt generated at completion of receive operation.

[When set to 0]

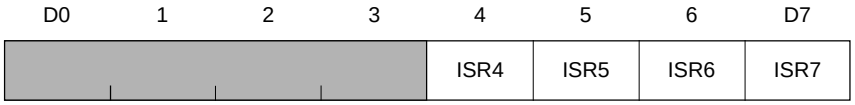
Setting this bit to 0 selects Receive complete interrupt (receive buffer full). A receive complete interrupt is generated even when an error occurred when receiving data (except for an overrun error).

[When set to 1]

Setting this bit to 1 selects Receive error interrupt. Following are detected for receive errors:

- CSIO mode : Overrun error
- UART mode : Overrun, parity, and framing errors

■ **SIO47 Receive Interrupt Cause Select Register (SI47SEL)** **<Address: H'0080 0A02>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4	ISR4	0: Receive complete interrupt	○	○
	(SIO4 receive interrupt cause select bit)	1: Receive error interrupt		
5	ISR5	0: Receive complete interrupt	○	○
	(SIO5 receive interrupt cause select bit)	1: Receive error interrupt		
6	ISR6	0: Receive complete interrupt	○	○
	(SIO6 receive interrupt cause select bit)	1: Receive error interrupt		
7	ISR7	0: Receive complete interrupt	○	○
	(SIO7 receive interrupt cause select bit)	1: Receive error interrupt		

This register selects the cause of interrupt generated at completion of receive operation.

[When set to 0]

Setting this bit to 0 selects Receive complete interrupt (receive buffer full). A receive complete interrupt is generated even when an error occurred when receiving data (except for an overrun error).

[When set to 1]

Setting this bit to 1 selects Receive error interrupt. Following are detected for receive errors:

- CSIO mode : Overrun error
- UART mode : Overrun, parity, and framing errors

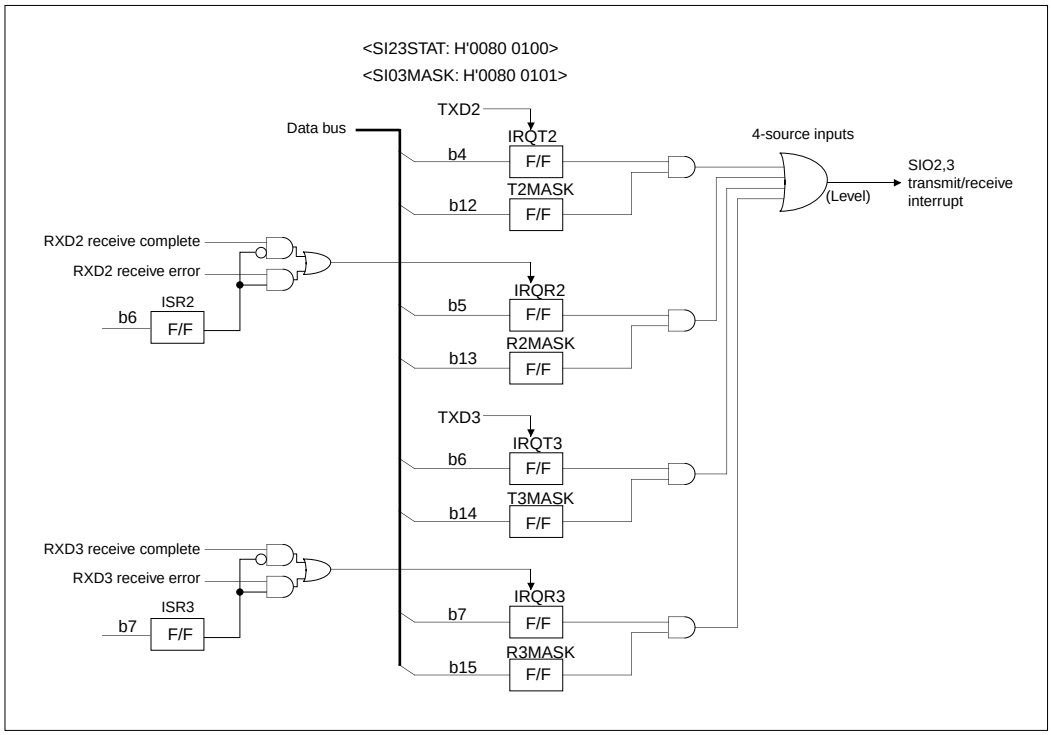


Figure 12.2.5 Block Diagram of SIO2, 3 Transmit/Receive Interrupts

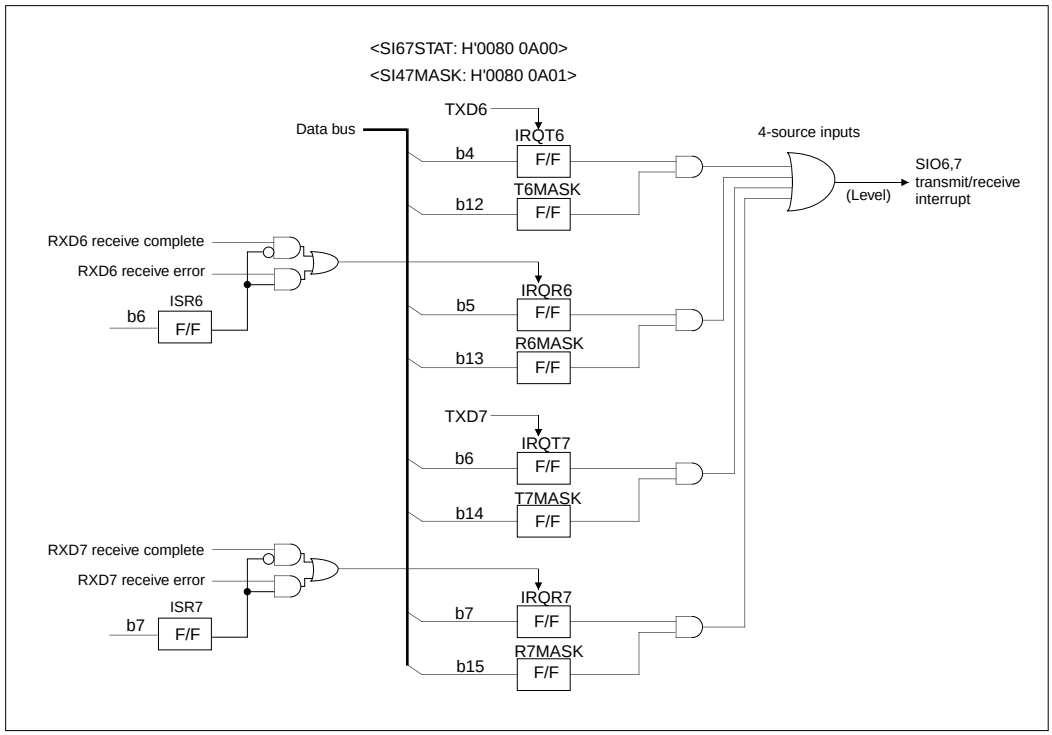


Figure 12.2.6 Block Diagram of SIO6, 7 Transmit/Receive Interrupts

12.2.3 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>**
■ **SIO4 Transmit Control Register (S4TCNT)** **<Address: H'0080 0A10>**
■ **SIO5 Transmit Control Register (S5TCNT)** **<Address: H'0080 0A20>**
■ **SIO6 Transmit Control Register (S6TCNT)** **<Address: H'0080 0A30>**
■ **SIO7 Transmit Control Register (S7TCNT)** **<Address: H'0080 0A40>**

<When reset: H'12>

D	Bit Name	Function	R	W
0,1	No functions assigned		0	–
2,3	CDIV (BRG count source select bit)	D2 D3 0 0: Selects f(BCLK) 0 1: Selects divided-by-8 clock of f(BCLK) 1 0: Selects divided-by-32 clock of f(BCLK) 1 1: Selects divided-by-256 clock of f(BCLK)	○	○
4	No functions assigned		0	–
5	TSTAT (Transmit status bit)	0: Transmission idle & no data exist in transmit buffer register 1: Transmission in progress or data exist in transmit buffer register	○	–
6	TBE (Transmit buffer empty bit)	0: Data exist in transmit buffer register 1: No data exist in transmit buffer register	○	–
7	TEN (Transmit enable bit)	0: Disables transmission 1: Enables transmission	○	○

(1) CDIV (baud rate generator count source select) bits (D2, D3)

These bits select the count source for the Baud Rate Generator (BRG).

Note: If f(BCLK) is selected for the BRG count source, the BRG must be set in such a way that the baud rate will not exceed the maximum transfer rate. For details, see the section where the BRG Register is discussed.

(2) TSTAT (transmit status) bit (D5)

[Set condition]

This bit is set to 1 by writing to the Transmit Buffer Register when transmission is enabled.

[Clear condition]

This bit is cleared to 0 when transmission remains idle (no data exists in the Transmit Shift Register) nor does data exist in the Transmit Buffer Register. It also is cleared upon clearing the transmit enable bit.

(3) TBE (transmit buffer empty) bit (D6)

[Set condition]

This bit is set to 1 when data is transferred from the Transmit Buffer Register to the Transmit Shift Register, with the Transmit Buffer Register thereby emptied. It also is set by clearing the transmit enable bit.

[Clear condition]

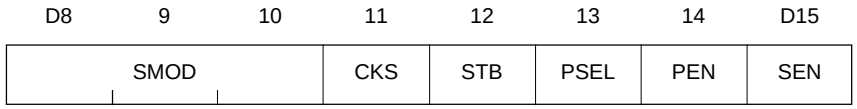
This bit is cleared to 0 by writing data to the lower byte of the Transmit Buffer Register while transmission is enabled (TEN = 1).

(4) TEN (transmit enable) bit (D7)

Setting this bit to 1 enables the SIO for transmission; clearing this bit to 0 disables the SIO against transmission.
The SIO stops sending data when this bit is cleared to 0 during transmission.

12.2.4 SIO Transmit/Receive Mode Registers

■ **SIO0 Transmit/Receive Mode Registers (S0MOD)** **<Address: H'0080 0111>**
■ **SIO1 Transmit/Receive Mode Registers (S1MOD)** **<Address: H'0080 0121>**
■ **SIO2 Transmit/Receive Mode Registers (S2MOD)** **<Address: H'0080 0131>**
■ **SIO3 Transmit/Receive Mode Registers (S3MOD)** **<Address: H'0080 0141>**
■ **SIO4 Transmit/Receive Mode Registers (S4MOD)** **<Address: H'0080 0A11>**
■ **SIO5 Transmit/Receive Mode Registers (S5MOD)** **<Address: H'0080 0A21>**
■ **SIO6 Transmit/Receive Mode Registers (S6MOD)** **<Address: H'0080 0A31>**
■ **SIO7 Transmit/Receive Mode Registers (S7MOD)** **<Address: H'0080 0A41>**



<When reset: 00>

D	Bit Name	Function	R	W
8-10	SMOD (Serial I/O mode select bit) (Note 1)	D8 D9 D10 0 0 0 : 7-bit UART 0 0 1 : 8-bit UART 0 1 X : 9-bit UART 1 X X : 8-bit clock synchronized serial I/O	○	○
11	CKS (internal/external clock select bit)	0: Internal clock 1: External clock	○	○ (Note 2)
12	STB (stop bit length select bit, UART mode only)	0: 1 stop bit 1: 2 stop bits	○	○ (Note 3)
13	PSEL (odd/even parity select bit, UART mode only)	0: Odd parity 1: Even parity	○	○ (Note 3)
14	PEN (parity enable bit, UART mode only)	0: Disables parity 1: Enables parity	○	○ (Note 3)
15	SEN (sleep select bit, UART mode only)	0: Disables sleep function 1: Enables sleep function	○	○ (Note 3)

Note 1: For SIO2 and 3, the D8 bit is fixed to 0 in hardware. The 8 bit cannot be set to 1 (to select clock synchronized serial I/O).
Note 2: When UART mode selected, this bit has no effect.
Note 3: When clock synchronized mode selected, D12-D15 have no effect.

The SIO Transmit/Receive Mode Registers consist of the bits to select serial I/O operation mode, data format, and the function to be used during communication.
The SIO Transmit/Receive Mode Registers must always be set before serial I/O starts operation. To modify the register contents after the SIO started sending or receiving, check to see that the transmit or receive operation is completed and disable transmit/receive operations (by clearing the SIO Transmit Control Register transmit enable bit and the SIO Receive Control Register receive enable bit to 0) before setting the register.

(1) SMOD (serial I/O mode select) bits (D8-D10)

These bits select serial I/O operation mode.

(2) CKS (internal/external clock select) bit (D11)

This bit is effective when CSIO mode is selected. When UART mode is selected, this bit has no effect and the SIO operates with an internal clock.

(3) STB (stop bit length select) bit (D12)

This bit is effective when in UART mode. Use this bit to select the length of the stop bit that indicates the end of the transmit data. Setting this bit to 0 selects one stop bit; setting this bit to 1 selects two stop bits.
During clock synchronized mode, the content of this bit has no effect.

(4) PSEL (odd/even parity select) bit (D13)

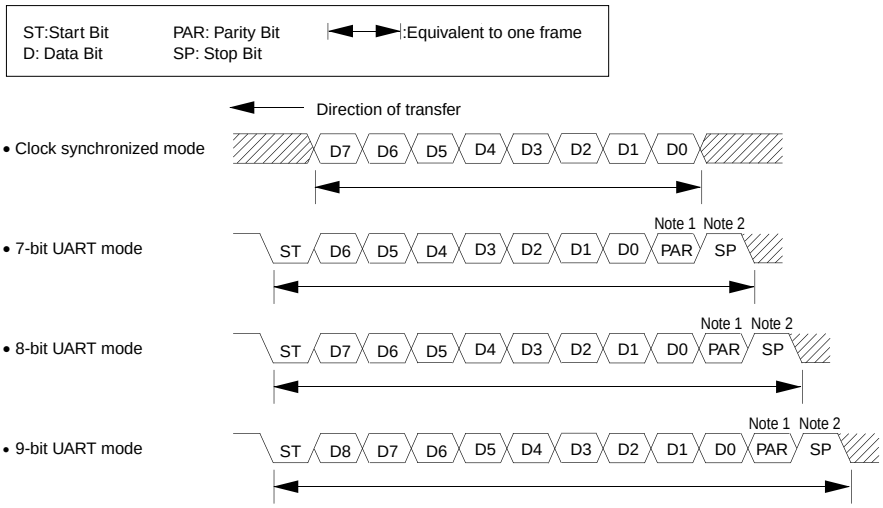
This bit is effective when in UART mode. When parity is enabled (D14 = 1), use this bit to select the parity attribute (odd or even). Setting this bit to 0 selects odd parity; setting this bit to 1 selects even parity.
When parity is disabled (D14 = 0) and when in clock synchronized mode, the content of this bit has no effect.

(5) PEN (parity enable) bit (D14)

This bit is effective when in UART mode. Setting this bit to 1 enables parity, so that a parity bit is added immediately after the data bits of transmit data. When receiving data, the received data is checked for parity.
The parity bit added to the transmit data is automatically determined to be to 0 or 1 so that the attribute of the result derived by adding the number of 1's in the data bits and the content of the parity bit matches the attribute selected with the odd/even parity select bit (D13).
Figure 12.2.4 shows an example of a data format where parity is enabled.

(6) SEN (sleep select) bit (D15)

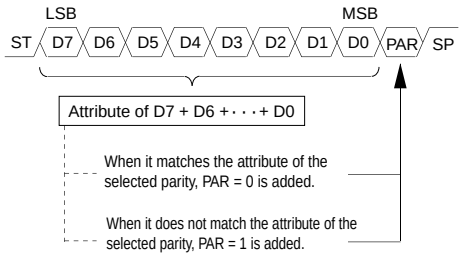
This bit is effective when in UART mode. When the sleep function is enabled by setting this bit to 1, data is latched into the UART Receive Buffer Register only when the most significant bit (MSB) of the received data = 1.



Note 1: The parity bit can be chosen to be or not to be included.
Note 2: The stop bit length can be selected between one bit or two bits.

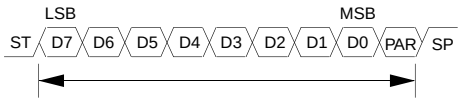
- When sending

  When the attribute (odd or even) represented by the number of 1's in the data bits matches that of the selected parity, the SIO adds a parity bit which is 0 to the transmit data. When the attribute represented by the number of 1's in the data bits does not match that of the selected parity, the SIO adds a parity bit which is 1 to the transmit data.



- When receiving

  The received data is checked for parity to see if the number of 1's in the data and parity bits matches the attribute (odd or even) of the parity (parity check).



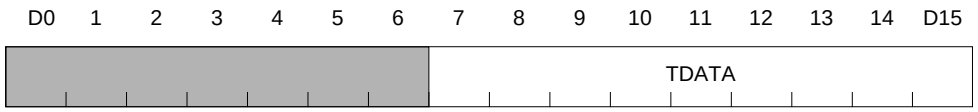
If the result of D7 + D6 + · · · + D0 + PAR does not match he attribute of the selected parity, a parity error is assumed.

Note 1: The data format shown above is an example for 8-bit UART mode.
Note 2: The data bit number (Dn) above only indicates a sequence of data, and does not represent the bit number (Dn) in the register.

Figure 12.2.7 Data Format where Parity is Enabled

12.2.5 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>**
■ **SIO4 Transmit Buffer Register (S4TXB)** **<Address: H'0080 0A12>**
■ **SIO5 Transmit Buffer Register (S5TXB)** **<Address: H'0080 0A22>**
■ **SIO6 Transmit Buffer Register (S6TXB)** **<Address: H'0080 0A32>**
■ **SIO7 Transmit Buffer Register (S7TXB)** **<Address: H'0080 0A42>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-6	No functions assigned		?	○
7-15	TDATA (Transmit data)	Set transmit data.	?	○

R = ?: Indeterminate when read

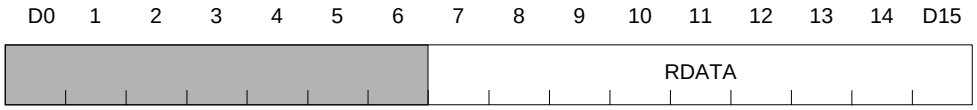
The SIO Transmit Buffer Registers are used to set transmit data. These registers are a write-only register, so that the register contents cannot be read out. When setting transmit data in this register, make sure the data is LSB-aligned, and that the data is written to D9-D15 for 7-bit data (UART mode only), D8-D15 for 8-bit data, or D7-D15 for 9-bit data (UART mode only).

Before writing transmit data to this register, be sure to enable the Transmit Control Register TEN (transmit enable) bit (by setting it to 1). Writing to this register while the TEN bit is disabled (= 0) has no effect.
When data is written to the Transmit Buffer Register while the SIO is enabled for transmission, the data is transferred from the SIO Transmit Buffer Register to the SIO Transmit Shift Register, upon which the serial I/O status transmitting the data.

Note: When sending 7 or 8-bit data, the register can be accessed bytewise.

12.2.6 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>**
■ **SIO4 Receive Buffer Register (S4RXB)** **<Address: H'0080 0A14>**
■ **SIO5 Receive Buffer Register (S5RXB)** **<Address: H'0080 0A24>**
■ **SIO6 Receive Buffer Register (S6RXB)** **<Address: H'0080 0A34>**
■ **SIO7 Receive Buffer Register (S7RXB)** **<Address: H'0080 0A44>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-6	No functions assigned		0	–
8-15	RDATA (Receive data)	Stores received data.	○	–

The SIO Receive Buffer Registers are used to store the received data. When the SIO finishes receiving data, the received data is transferred from the SIO Receive Shift Register to the SIO Receive Buffer Register. The SIO Receive Buffer Registers are a read-only register.
For 7-bit data (UART mode only), the data are set in D9-D15, with D8 and D7 always set to 0. For 8-bit data, the data are set in D8-D15, with D7 always set to 0.
If after completion of reception, the SIO finishes receiving the next data before reading out the content of the SIO Receive Buffer Register, an overrun error occurs and the subsequently received data will not be stored in the Receive Buffer Register.
To restart reception normally, clear the Receive Control Register REN (receive enable) bit to 0.

Note: When receiving 7 or 8-bit data, the register can be accessed bytewise.

12.2.7 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>**
■ **SIO4 Receive Control Register (S4RCNT)** **<Address: H'0080 0A16>**
■ **SIO5 Receive Control Register (S5RCNT)** **<Address: H'0080 0A26>**
■ **SIO6 Receive Control Register (S6RCNT)** **<Address: H'0080 0A36>**
■ **SIO7 Receive Control Register (S7RCNT)** **<Address: H'0080 0A46>**

D0	1	2	3	4	5	6	D7
	RSTAT	RFIN	REN	OVR	PTY	FLM	ERS

<When reset: H'00>

D	Bit Name	Function	R	W
0	No functions assigned		0	–
1	RSTAT (Receive status bit)	0: Not receiving (idle) 1: Receiving data	○	–
2	RFIN (Receive finished bit)	0: No data exist in the receive buffer register 1: Data exist in the receive buffer register	○	–
3	REN (Receive enable bit)	0: Disables reception 1: Enables reception	○	○
4	OVR (Overrun error bit)	0: No overrun error 1: Overrun error occurred	○	–
5	PTY (Parity error bit, UART mode only)	0: No parity error 1: Parity error occurred	○	–
6	FLM (Framing error bit, UART mode only)	0: No framing error 1: Framing error occurred	○	–
7	ERS (Errorsum bit)	0: No error 1: Error occurred	○	–

(1) RSTAT (Receive status) bit (D1)

[Set condition]

This bit is set to 1 by starting receive operation. When this bit is 1, it means that the SIO is receiving data.

[Clear condition]

This bit is cleared upon completion of receive operation or by clearing the REN (receive enable) bit to 0.

(2) RFIN (Receive finished) bit (D2)

[Set condition]

This bit is set to 1 when all data bits, after being prepared in the Receive Shift Register, are transferred from that register to the Receive Buffer Register.

[Clear condition]

This bit is cleared upon reading out the lower byte from 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 from the Receive Buffer Register. In this case, clear the REN (receive enable) bit to 0.

(3) REN (Receive enable) bit (D3)

Setting this bit to 1 enables the SIO for reception; setting this bit to 0 disables the SIO against reception while at the same initializing the receiver unit. Pursuant to this bit manipulation, the receive status flag and receive finished flag bits, as well as the overrun error, framing error, parity error, and errorsum flags all are cleared to 0.
If the receive enable bit is cleared to 0 while receiving data, the receive operation stops immediately.

(4) OVR (Overrun error) bit (D4)

[Set condition]

This bit is set to 1 when all bits of the next received data have been prepared in the Receive Shift Register while the previously received data still exists in the Receive Buffer Register. The newly received data is not transferred from the Receive Shift Register to 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. To restart reception normally, this bit needs to be cleared.

[Clear condition]

This bit can only be cleared by clearing the REN (receive enable) bit to 0.

(5) PTY (Parity error) bit (D5)

This bit is effective in only UART mode. It is fixed to 0 during CSIO mode.

[Set condition]

This bit is set to 1 when while the SIO Transmit/Receive Mode Register PEN (parity enable/disable) bit is enabled, the parity (even/odd) of the received data does not match the one selected with the said register's PSEL (parity select) bit.

[Clear condition]

This bit is cleared upon reading out the lower byte from 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 from the Receive Buffer Register. In this case, clear the REN (receive enable) bit to 0.

(6) FLM (Framing error) bit (D6)

This bit is effective in only UART mode. It is fixed to 0 during CSIO mode.

[Set condition]

This bit is set to 1 when the number of received bits does not match the one selected with the SIO Transmit/Receive Mode Register.

[Clear condition]

This bit is cleared upon reading out the lower byte from the SIO Receive Buffer Register or by clearing the SIO Receive Control Register REN (receive enable) bit.

(7) ERS (Errorsum) bit (D7)

[Set condition]

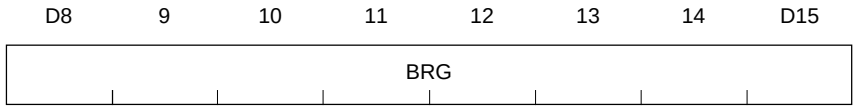
This bit is set to 1 when any error-whether an overrun, framing, or parity error-occurs before reception is completed.

[Clear condition]

For overrun errors, this bit can be cleared by clearing the REN (receive enable) bit to 0. For all other errors, this bit is cleared upon reading out the lower byte from the SIO Receive Buffer Register or by clearing the SIO Receive Control Register REN (receive enable) bit.

12.2.8 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>**
■ **SIO4 Baud Rate Register (S4BAUR)** **<Address: H'0080 0A17>**
■ **SIO5 Baud Rate Register (S5BAUR)** **<Address: H'0080 0A27>**
■ **SIO6 Baud Rate Register (S6BAUR)** **<Address: H'0080 0A37>**
■ **SIO7 Baud Rate Register (S7BAUR)** **<Address: H'0080 0A47>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
8-15	BRG (Baud rate divide value)	The baud rate count source selected with the SIO Mode Register is divided by (n +1) where n = BRG divide value set here.	○	○

BRG (Baud rate divide value) (D8-D15)

The SIO Baud Rate Registers are used to divide the baud rate count source selected with the SIO Mode Register by (n + 1) where n is the BRG value that is set with this register.
In the initial state, the BRG value is indeterminate, so always be sure to set the divide value with this register before using serial I/O. The value written to the BRG register while sending or receiving data becomes effective beginning with the next cycle after the BRG counter finished counting.

When using an internal clock in CSIO mode (i.e., producing SCLKO output signal), the internal BCLK is first divided by the clock divider and the resulting clock frequency is divided by (n + 1) where n = BRG set value and is further divided by 2 to produce the transmit/receive shift clock.

When using an external clock in CSIO mode, the BRG is not used (transmit/receive operations are synchronized to an externally sourced clock).

In UART mode, the internal BCLK is first divided by the clock divider and the resulting clock

frequency is divided by (n + 1) where n = BRG set value and is further divided by 16 to produce the transmit/receive shift clock.

When using SIO0, SIO1, SIO4, or SIO5 in UART mode, the SIO's corresponding port (P84, P87, P65, or P66) may be changed to the SCLKO pin, so that a divided-by-2 clock of BRG output is generated.

When using an internal clock (internal clock CSIO mode or UART mode) with f(BCLK) selected for the BRG count source, the BRG value set with this register requires caution. During CSIO mode, make sure the transfer speed will not exceed 2 Mbits/second; during UART mode, make sure the BRG value is equal to or greater than 7.

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 to generate the transmit/receive shift clock is selected from the internal clock f(BCLK) or external clock. The CKS (internal/external clock select) bit (SIO Transmit/Receive Mode Register D11 bit) is used to select the clock source. The equation by which to calculate the transmit/receive baud rate values differs with the selected clock source, whether internal or external.

(1) When internal clock is selected in CSIO mode

When the internal clock is selected, f(BCLK) is divided by the clock divider before being fed into 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 D2, D3 bits). The baud rate generator divides the clock divider output by (baud rate register set value + 1) and then by 2, which results in generating a transmit/receive shift clock.

When the internal clock is selected in CSIO mode, the baud rate is calculated using the equation below.

$$\text{Baud rate [bps]} = \frac{1\ (\text{BCLK})}{\text{Clock divider's divide-by value} \times (\text{baud rate register set value} + 1) \times 2}$$

Baud rate register set value = H'00 to H'FF (Note)
Clock divider's divide-by value = 1, 8, 32, or 256

Note: If the divide-by value selected for the baud rate generator count source is "1" (i.e., f(BCLK) itself), make sure the baud rate register value you set does not exceed 2 Mbps.

(2) When external clock is selected in CSIO mode

In this case, the baud rate generator is not used; instead, the input clock from the SCLKI pin serves directly as CSIO transmit/receive shift clock.
The maximum frequency of the SCLKI pin input clock is 1/16 of f(BCLK).

Baud rate [bps] = SCLKI pin input clock

12.3.2 Initial Settings for CSIO Transmission

To transmit data in CSIO mode, initialize the serial I/O following the procedure described below.

(1) Setting SIO Transmit/Receive Mode Register

- Set the register to CSIO mode
- Select the internal or an external clock

(2) Setting SIO Transmit Control Register

- Select the clock divider's divide-by ratio (when internal clock selected)

(3) Setting SIO Baud Rate Register

When the internal clock is selected, set a baud rate generator value. (Refer to Section 12.3.1, "Setting the CSIO Baud Rate.")

(4) Setting SIO Interrupt Mask Register

- Enable or disable the transmit buffer empty interrupt (SIO Interrupt Mask Register)

(5) Setting the Interrupt Controller (SIO Transmit Interrupt Control Register)

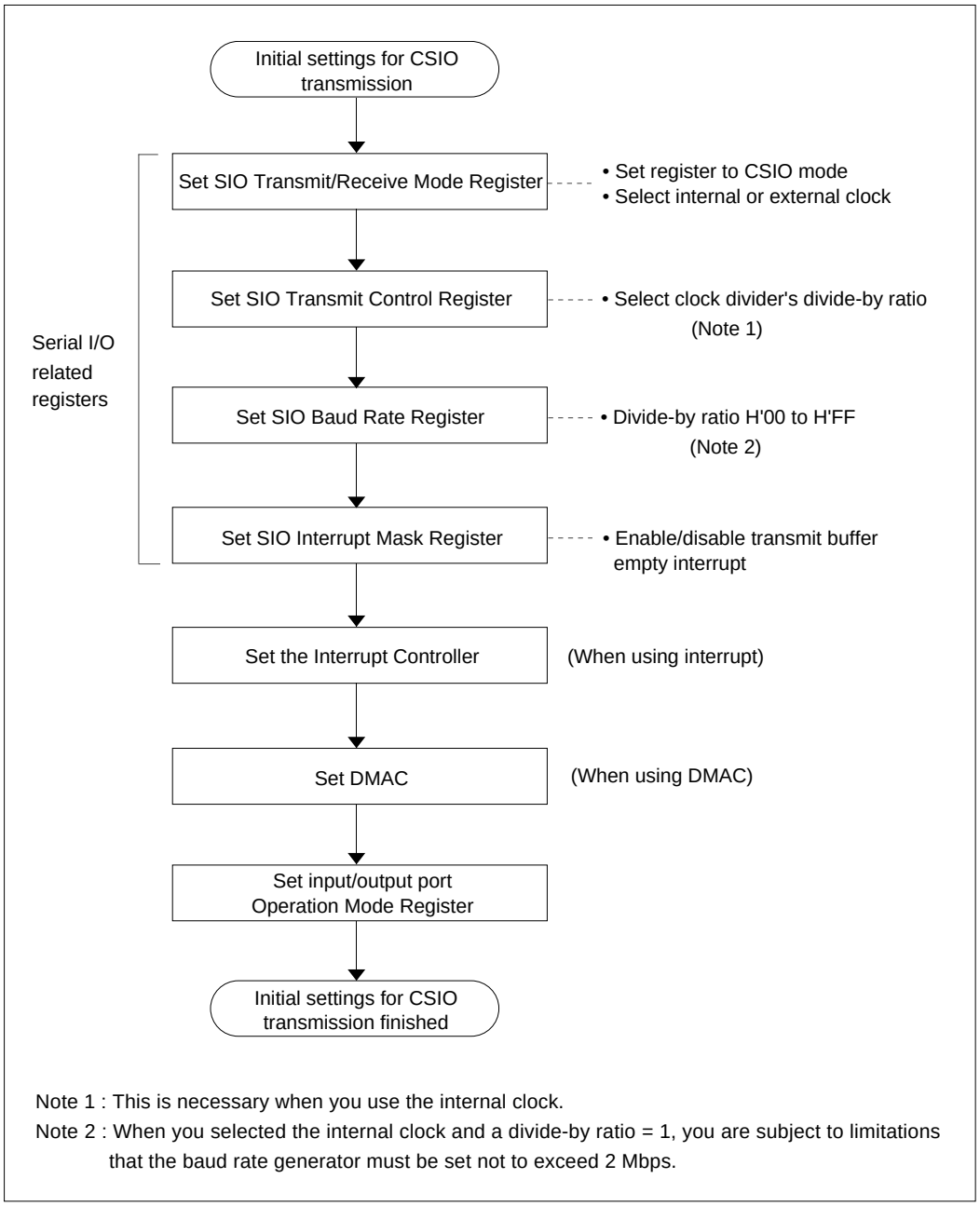
When you use a transmit buffer empty interrupt during transmission, set its priority level.

(6) Setting DMAC

When you issue DMA transfer requests to the internal DMAC when the transmit buffer is empty, set the DMAC. (Refer to Chapter 9, "DMAC.")

(7) Selecting pin functions

Because the serial I/O related pins serve dual purposes (shared with input/output ports), set pin functions. (Refer to Chapter 8, "Input/Output Ports and Pin Functions.")



Note 1 : This is necessary when you use the internal clock.
Note 2 : When you selected the internal clock and a divide-by ratio = 1, you are subject to limitations that the baud rate generator must be set not to exceed 2 Mbps.

Figure 12.3.1 Procedure for CSIO Transmit Initialization

12.3.3 Starting CSIO Transmission

When all of the following transmit conditions are met after you finished initialization, the serial I/O starts transmit operation.

(1) Transmit conditions when CSIO mode internal clock is selected

- The SIO Control Register's 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 Control Register 0's 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).
- A falling edge of transmit clock on the SCLKI pin is detected.

Note 1: While the transmit enable bit is cleared to 0, writes to the transmit buffer register are ignored. Always be sure to set the transmit enable bit to 1 before you write to the transmit buffer register.
Note 2: When the internal clock is selected, a write to the lower byte of the transmit buffer register in Note 1 above triggers a start of transmission.
Note 3: 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 I/O transmits data following the procedure 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)
- Start sending data synchronously with the shift clock beginning with the LSB.

Note: A transmit buffer empty interrupt request and/or a DMA transfer request can be generated when the transmit buffer is emptied.

12.3.4 Successive CSIO Transmission

Once data is transferred from the transmit buffer register to the transmit shift register, the next data can be written to the transmit buffer register even when transmission of the preceding data is not completed. When the next data is written to the transmit buffer before completion of the preceding data transmission, the preceding and the next data are successively transmitted.

To see if data has been transferred from the transmit buffer register to the transmit shift register, check the SIO Status Register's transmit buffer empty flag.

12.3.5 Processing at End of CSIO Transmission

When data transmission is completed, the following operation is automatically performed in hardware.

(1) When not transmitting successively

- The transmit status bit is set to 0.

(2) When transmitting successively

- When transmission of the last data in a consecutive data train is completed, the transmit status bit is set to 0.

12.3.6 Transmit Interrupt

If a transmit buffer empty interrupt has been enabled by the SIO Interrupt Mask Register, a transmit buffer empty interrupt is generated at the time data is transferred from the transmit buffer register to the transmit shift register. Also, a transmit buffer empty interrupt is generated when the TEN (transmit enable) bit is set to 1 (enabled after being disabled) while a transmit buffer empty interrupt has been enabled.
You must set the Interrupt Controller (ICU) before you can use transmit interrupts.

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 ouput to the DMAC. This transfer request is also output when the TEN (transmit enable) bit is set to 1 (enabled after being disabled).
You must set the Interrupt Controller (ICU) before you can transmit data using DMA transfers.

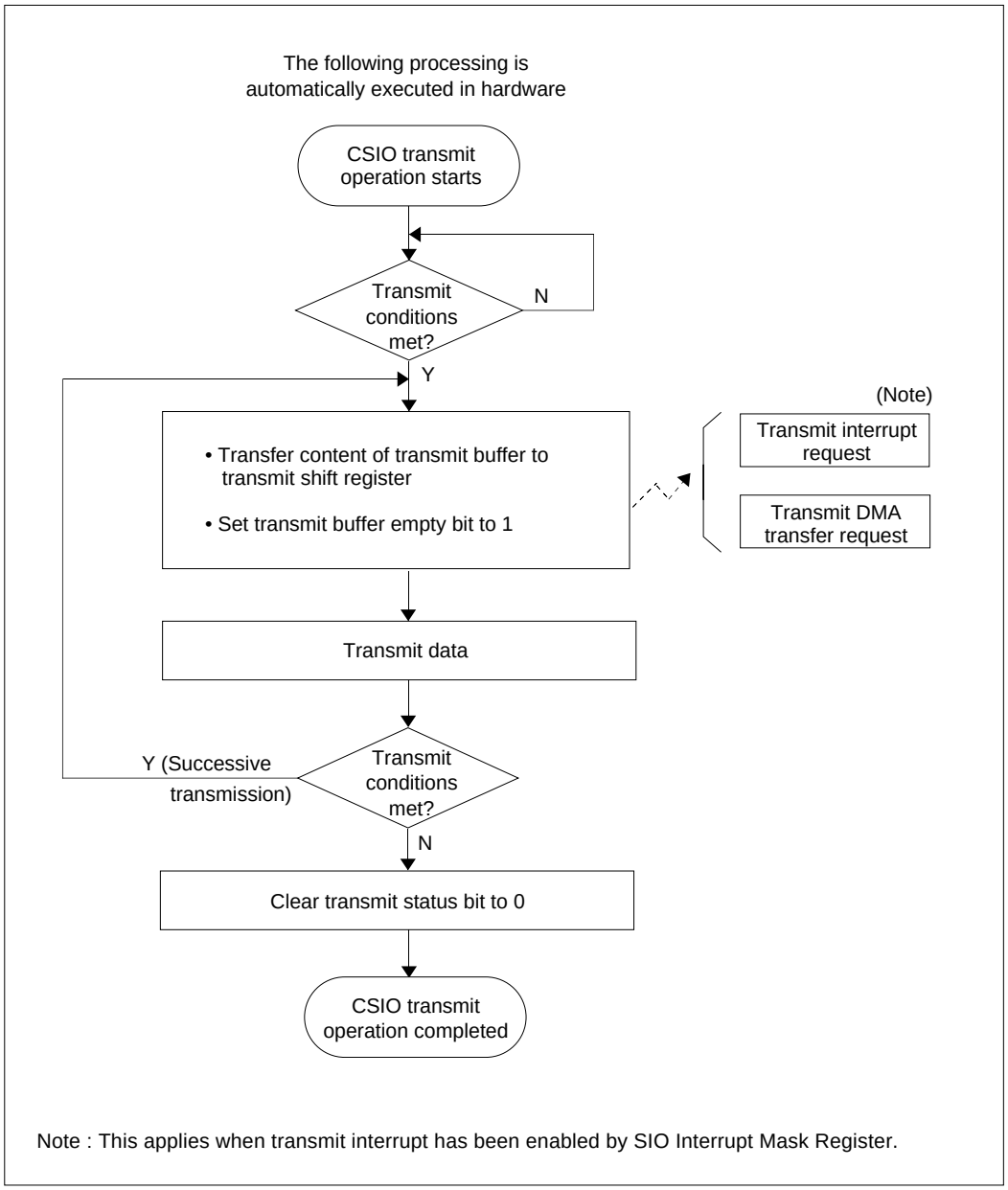
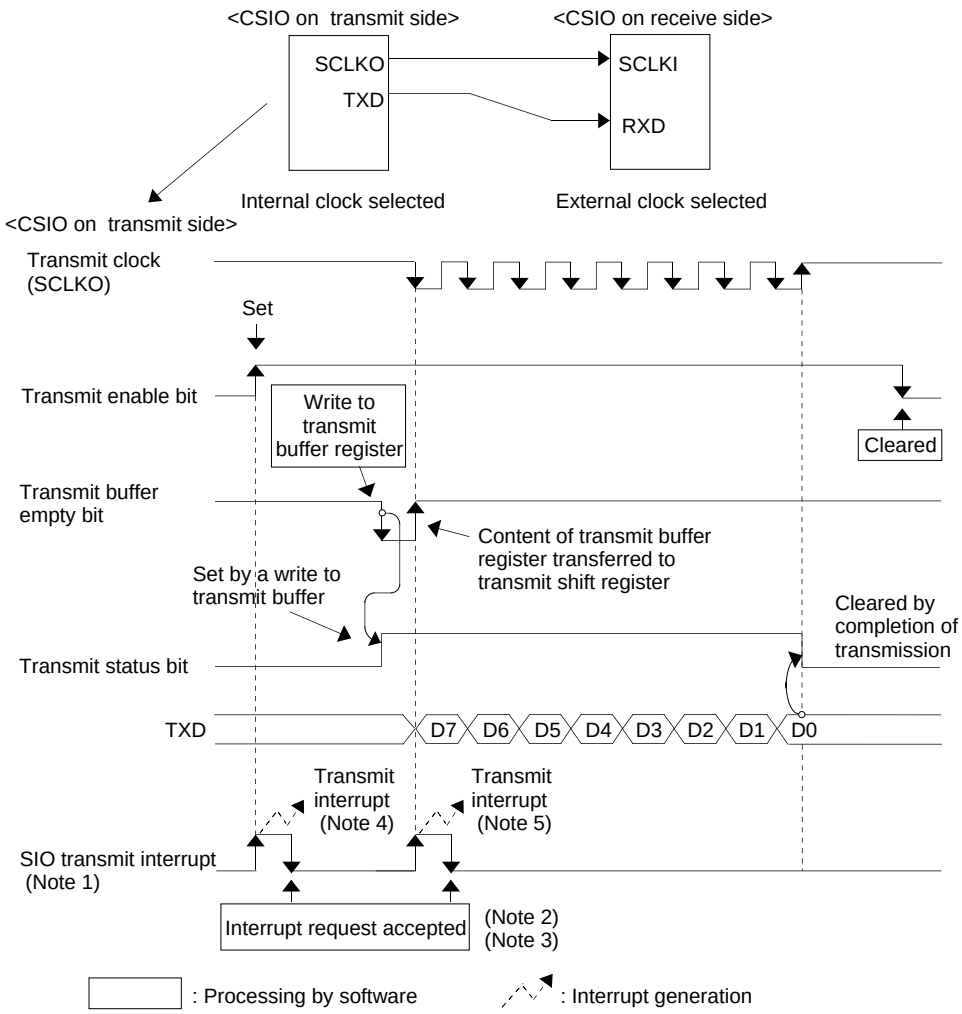


Figure 12.3.2 Transmit Operation during CSIO Mode (Hardware Processing)

12.3.8 Typical CSIO Transmit Operation

The following shows a typical transmit operation in CSIO mode.



Note 1 : Change of the Interrupt Controller "SIO Transmit Interrupt Control Register" interrupt request bit

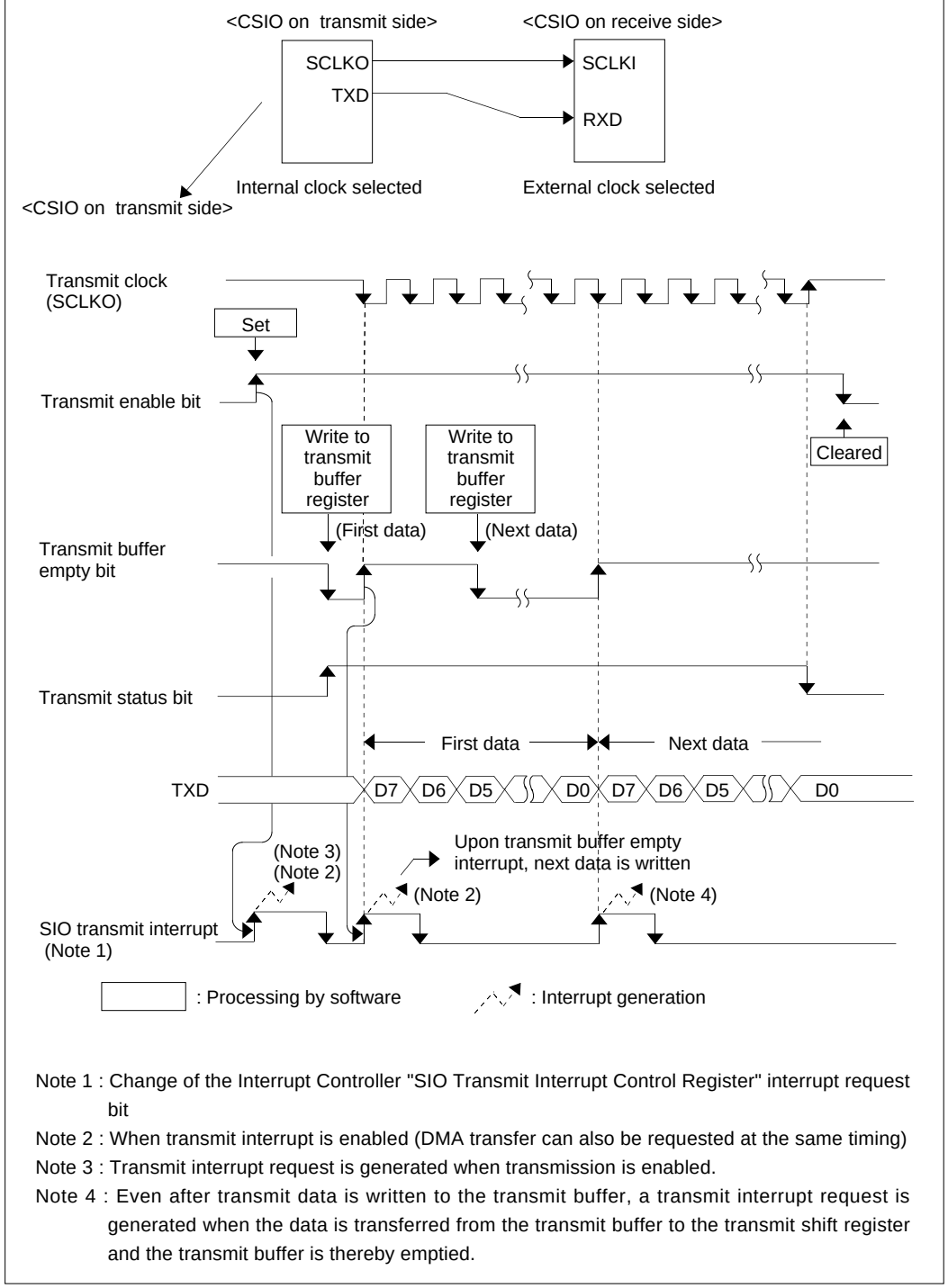
Note 2 : When transmit interrupt is enabled (DMA transfer can also be requested at the same timing)

Note 3 : The Interrupt Controller IVECT register is read or "SIO Transmit Interrupt Control Register" interrupt request bit cleared

Note 4 : Transmit interrupt request is generated when transmission is enabled.

Note 5 : 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.

Figure 12.3.3 Example of CSIO Transmission (Transmitted Only Once, with Transmit Interrupt Used)



Note 1 : Change of the Interrupt Controller "SIO Transmit Interrupt Control Register" interrupt request bit

Note 2 : When transmit interrupt is enabled (DMA transfer can also be requested at the same timing)

Note 3 : Transmit interrupt request is generated when transmission is enabled.

Note 4 : 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.

Figure 12.3.4 Example of CSIO Transmission (Successive Transmission, with Transmit Buffer Empty and Transmit Finished Interrupts Used)

12.4 Receive Operation in CSIO Mode

12.4.1 Initial Settings for CSIO Reception

To receive data in CSIO mode, initialize the serial I/O following the procedure described below. Note, however, that because the receive shift clock is derived from operation of the transmit circuit, you need to execute transmit operation even when you only want to receive data.

(1) Setting SIO Transmit/Receive Mode Register

- Set the register to CSIO mode
- Select the internal or an external clock

(2) Setting SIO Transmit Control Register

- Select the clock divider's divide-by ratio (when internal clock selected)

(3) Setting SIO Baud Rate Register

When the internal clock is selected, set a baud rate generator value. (Refer to Section 12.3.1, "Setting the CSIO Baud Rate.")

(4) Setting SIO Interrupt Mask Register

- Enable or disable the transmit buffer empty interrupt (SIO Interrupt Mask Register)
- Select the cause of receive interrupt (receive finished/error) (Cause of Receive Interrupt Select Register)

(5) Setting SIO Receive Control Register

Set the receive enable bit

(6) Setting the Interrupt Controller (SIO Transmit Interrupt Control Register)

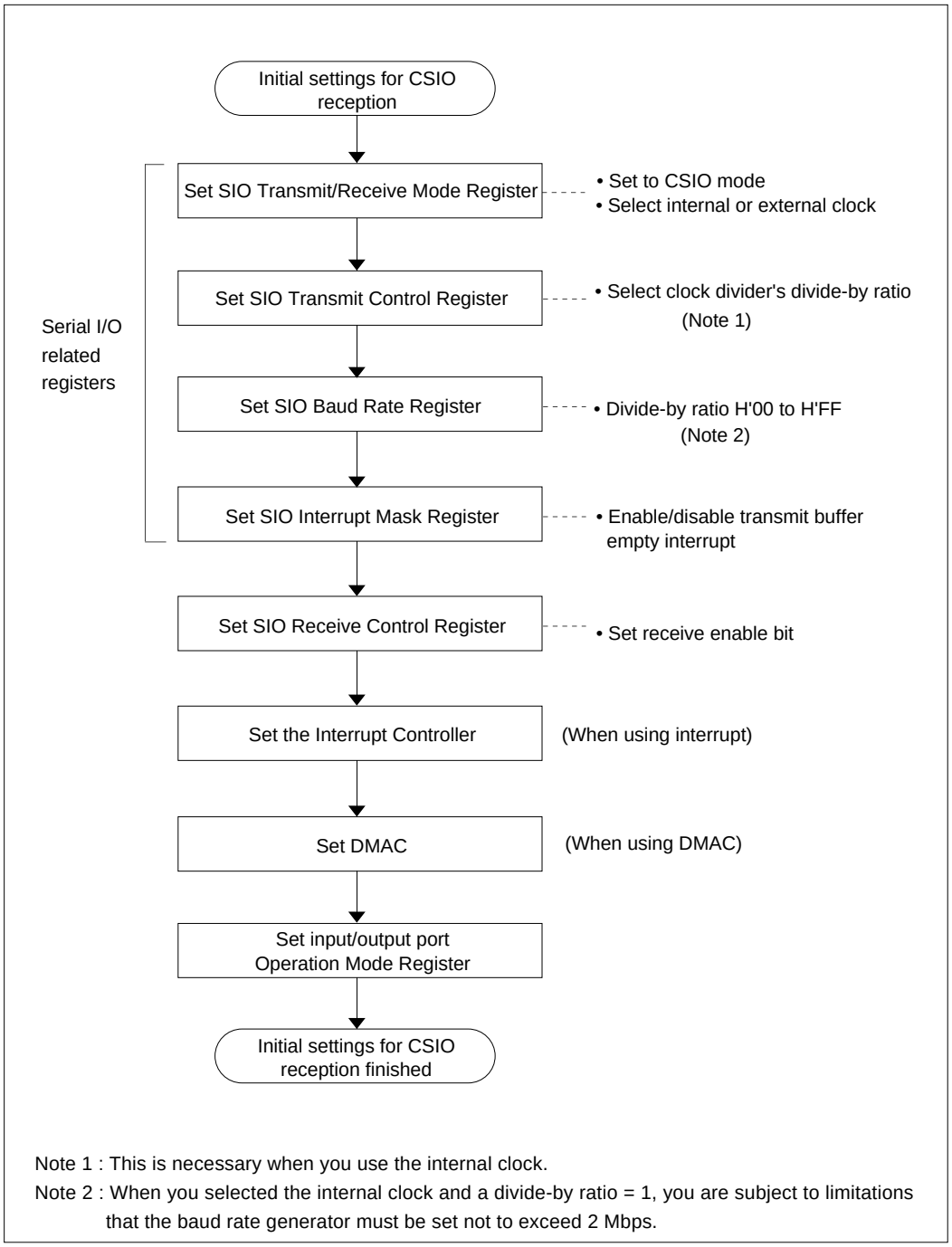
When you use a transmit interrupt or receive interrupt during transmission/reception, set its priority level.

(7) Setting the DMAC

When you generate a DMA transfer request to the internal DMAC when the transmit buffer is empty or transmission is completed, set the DMAC. (Refer to Chapter 9, "DMAC.")

(8) Selecting pin functions

Because the serial I/O related pins serve dual purposes (shared with input/output ports), set pin functions. (Refer to Chapter 8, "Input/Output Ports and Pin Functions.")



Note 1 : This is necessary when you use the internal clock.

Note 2 : When you selected the internal clock and a divide-by ratio = 1, you are subject to limitations that the baud rate generator must be set not to exceed 2 Mbps.

Figure 12.4.1 Procedure for CSIO Receive Initialization

12.4.2 Starting CSIO Reception

When all of the following receive conditions are met after you finished initialization, the serial I/O starts receive operation.

(1) Receive conditions when CSIO mode internal clock is selected

- The SIO Receive Control Register's receive enable bit is set to 1.
- Transmit conditions are met. (Refer to Section 12.3.3, "Starting CSIO Transmission.")

(2) Receive conditions when CSIO mode external clock is selected

- The SIO Receive Control Register's receive enable bit is set to 1.
- Transmit conditions are met. (Refer to 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 I/O 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 is completed, the following operation is automatically performed in hardware.

(1) When reception is completed normally

The receive-finished (receive buffer full) bit is set to 1.

Note 1: If a receive-finished (receive buffer full) interrupt has been enabled, an interrupt request is generated.
Note 2: A DMA transfer request is generated.

(2) When error occurs during reception

When an error (only overrun error in CSIO mode) occurs during reception, the overrun error bit and receive sum bit are set to 1.

Note 1: If a receive-finished interrupt has been selected (by SIO Cause of Receive Interrupt Select Register), neither a receive-finished interrupt request nor a DMA transfer request is generated.
Note 2: If a receive error interrupt has been selected (by SIO Cause of Receive Interrupt 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

When the following conditions are met at completion of data reception, data may be received successively.

- The receive enable bit is set to 1.
- Transmit conditions are met.
- No overrun error has occurred.

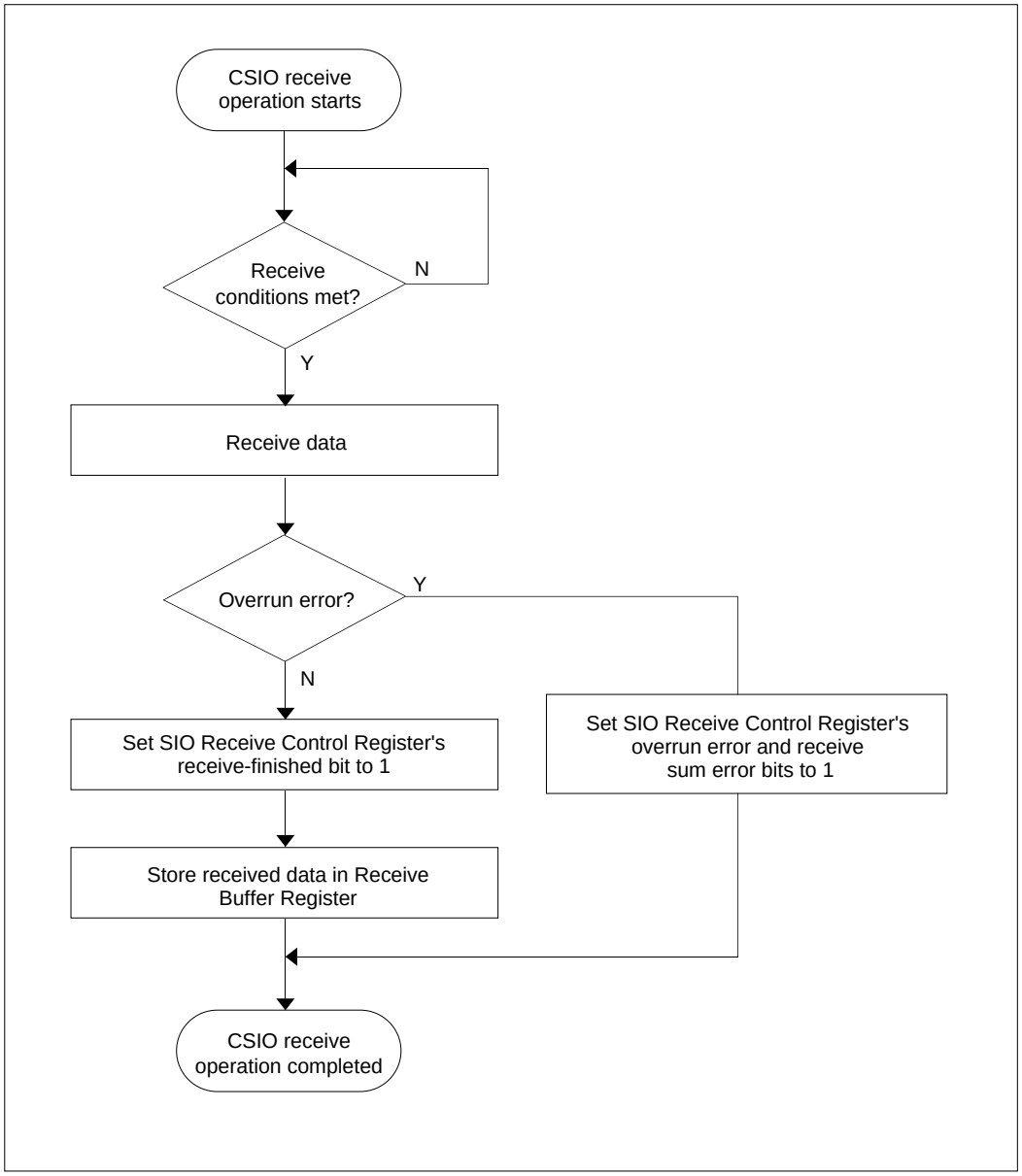


Figure 12.4.2 Receive Operation during CSIO Mode (Hardware Processing)

12.4.5 Flags Indicating the Status of CSIO Receive Operation

Following flags are available that indicate the status of receive operation in CSIO mode.

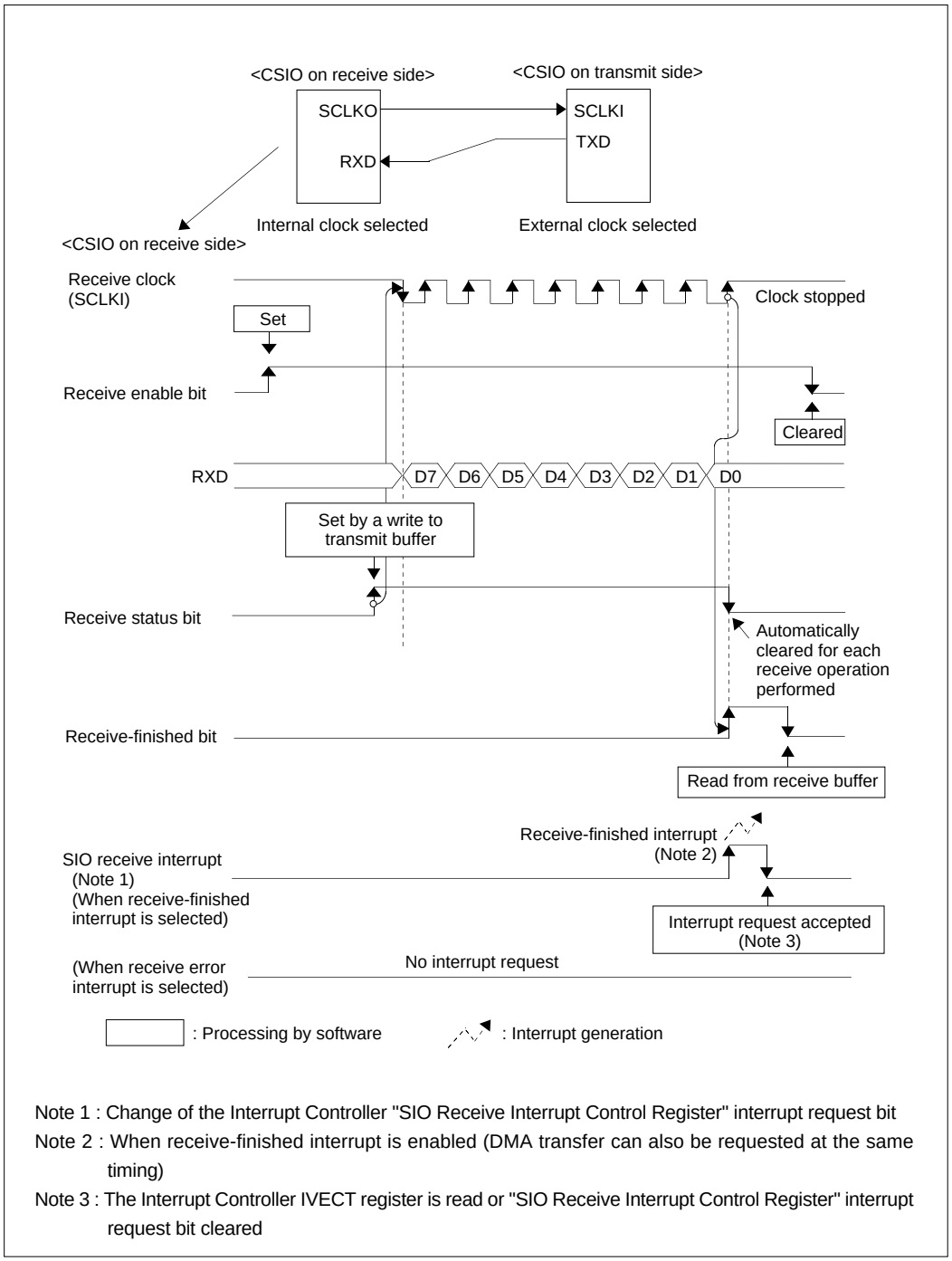
- SIO Receive Control Register receive status bit
- SIO Receive Control Register receive-finished bit
- SIO Receive Control Register receive error bit
- SIO Receive Control Register overrun error bit

After reception is completed, you may read out the content of the SIO Receive Buffer Register, but if the serial I/O finishes receiving the next data before you read, an overrun error occurs. In this case, the data received thereafter is not transferred to the SIO Receive Buffer Register. To restart reception, temporarily clear the receive enable bit to 0 and initialize the receive control block before you restart.
The said receive enable bit can be cleared, when there are no receive errors(note) encountered, by reading the lower byte from the SIO Receive Buffer Register or clearing the REN (receive enable) bit. If any receive error has occurred, it can only be cleared by clearing the REN (receive enable) bit, and cannot be cleared by reading the lower byte from the SIO Receive Buffer Register.

Note: Overrun error is the only error that can be detected during reception in CSIO mode.

12.4.6 Typical CSIO Receive Operation

The following shows a typical receive operation in CSIO mode.

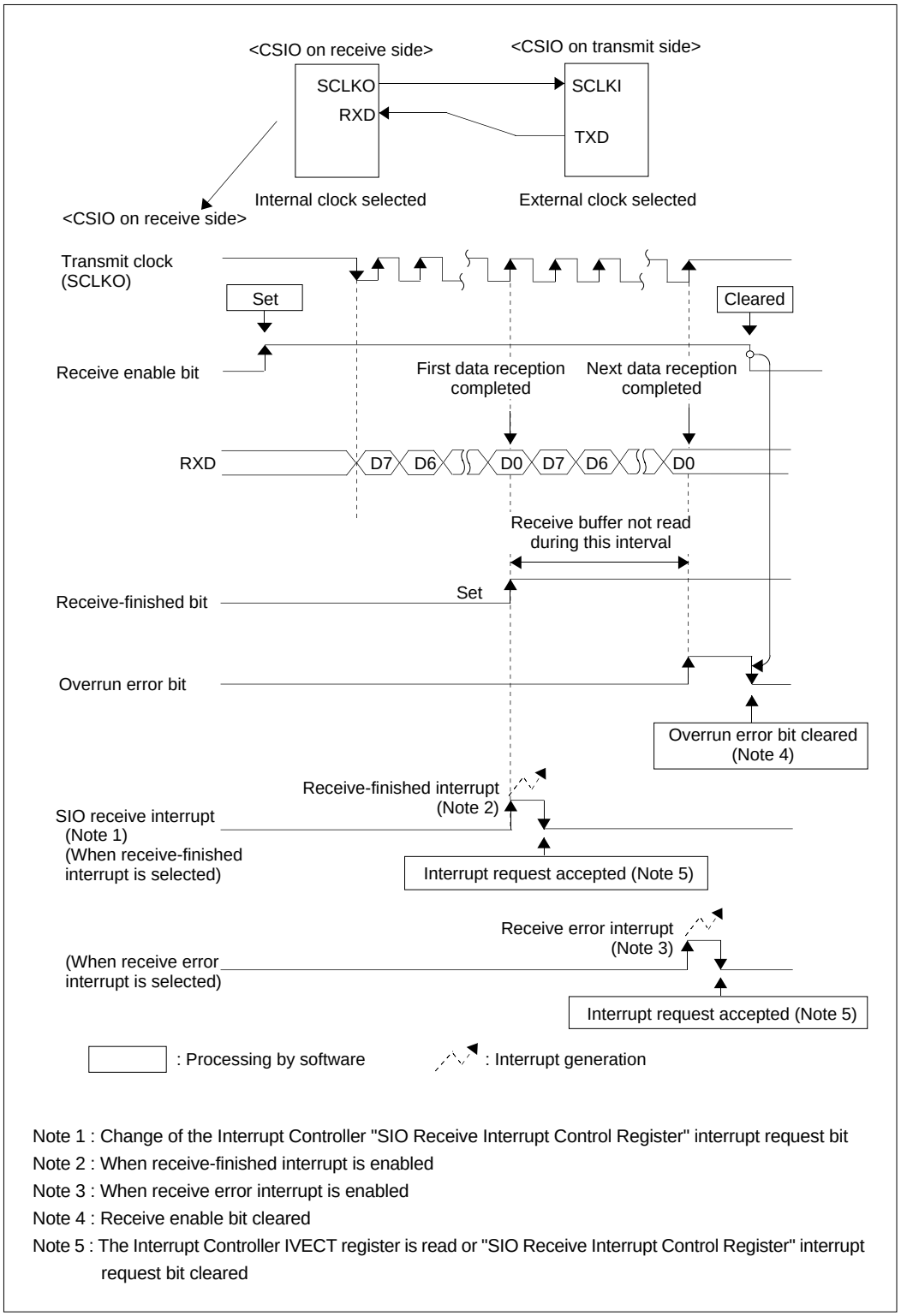


Note 1 : Change of the Interrupt Controller "SIO Receive Interrupt Control Register" interrupt request bit

Note 2 : When receive-finished interrupt is enabled (DMA transfer can also be requested at the same timing)

Note 3 : The Interrupt Controller IVECT register is read or "SIO Receive Interrupt Control Register" interrupt request bit cleared

Figure 12.4.3 Example of CSIO Reception (When Received Normally)



Note 1 : Change of the Interrupt Controller "SIO Receive Interrupt Control Register" interrupt request bit
Note 2 : When receive-finished interrupt is enabled
Note 3 : When receive error interrupt is enabled
Note 4 : Receive enable bit cleared
Note 5 : The Interrupt Controller IVECT register is read or "SIO Receive Interrupt Control Register" interrupt request bit cleared

Figure 12.4.4 Example of CSIO Reception (When Overrun Error Occurred)

12.5 Precautions 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 not operating. When transmitting or receiving data, be sure to check that transmission and/or reception under way has been completed and clear the transmit and receive enable bits before you set the registers.

- **Settings of Baud Rate (BRG) Register**

  If you selected f(BCLK) with the BRG clock source select bit, make sure the BRG register value you set does not exceed 2 Mbps.

- **About successive transmission**

  To transmit multiple data successively, set the next transmit data in the SIO Transmit Buffer Register before transmission of the preceding data is completed.

- **About reception**

  Because during CSIO mode the receive shift clock is derived from operation of the transmit circuit, you need to execute transmit operation (by sending dummy data) even when you only want to receive data. In this case, note that if the port function is set for TXD pin (by setting the operation mode register to 1), dummy data is actually output from the pin.

- **About successive reception**

  To receive multiple data successively, set data (dummy data) in the SIO Transmit Buffer Register before the transmitter starts sending data.

- **Transmit/receive operations using DMA**

  To transmit/receive data in DMA request mode, enable the DMAC to accept transfer requests (by setting the DMA Mode Register) before you start serial communication.

- **About the receive-finished bit**

  If a receive error (overrun error) occurs, the receive-finished bit cannot be cleared by reading out the receive buffer register. In this case, it can only be cleared by clearing the receive enable bit.

- **About overrun error**

  If all bits of the next receive data are received in the SIO Receive Shift Register before you read out the SIO Receive Buffer Register (an overrun error occurs), the receive data is not stored in the Receive Buffer Register and the Receive Buffer Register retains the previously received data. Thereafter, although receive operation is continued, no receive data is stored in the Receive Buffer Register (the receive status bit = 1). To restart reception normally, you need to temporarily clear the receive enable bit before you restart. This is the only way you can clear the overrun error flag.

- **About DMA transfer request generation during SIO transmission**

  If the Transmit Buffer Register becomes empty (the transmit buffer empty flag = 1) while the transmit enable bit is set to 1 (transmit enabled), an SIO transmit buffer empty DMA transfer request is generated.

- **About DMA transfer request generation during SIO reception**

  When the receive-finished bit is set to 1 (the receive buffer register full), a receive-finished DMA transfer request is generated. However, if an overrun error has occurred, this DMA transfer request is not generated.

12.6 Transmit Operation in UART Mode

12.6.1 Setting the UART Baud Rate

The baud rate (data transfer rate) during UART mode is determined by a transmit/receive shift clock. In UART mode, the source for this transmit/receive shift clock is always the internal clock regardless of how the internal/external clock select bit (SIO Transmit/Receive Mode Register bit D11) is set.

(1) Calculating the UART mode baud rate

After being divided by the clock divider, f(BCLK) is fed into 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(note) using the SIO Transmit Control Register's CDIV (baud rate generator count source select) bits (D2, D3). The Baud Rate Generator divides the clock it received from the clock divider by (baud rate register set value + 1) and further divides the resulting clock by 16 to produce a transmit/receive shift clock.

During UART mode (in which the internal clock is always used), the baud rate is calculated using the equation below.

$$\text{Baud rate [bps]} = \frac{1\ (\text{BCLK})}{\text{Clock divider's divide-by value} \times (\text{baud rate register set value} + 1) \times 16}$$

Baud rate register set value = H'00 to H'FF (Note)
Clock divider's divide-by value = 1, 8, 32, or 256

Note: If the divide-by value selected for the baud rate generator count source is "1" (i.e., f(BCLK) itself), make sure the baud rate register value you set is equal to or greater than 7.

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.

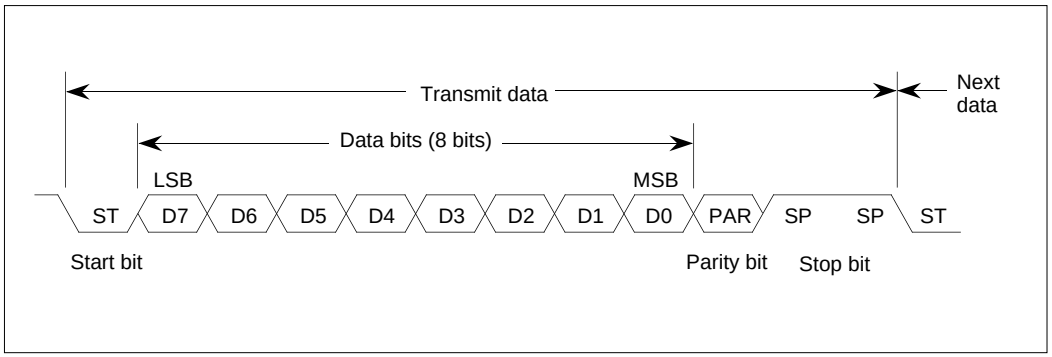
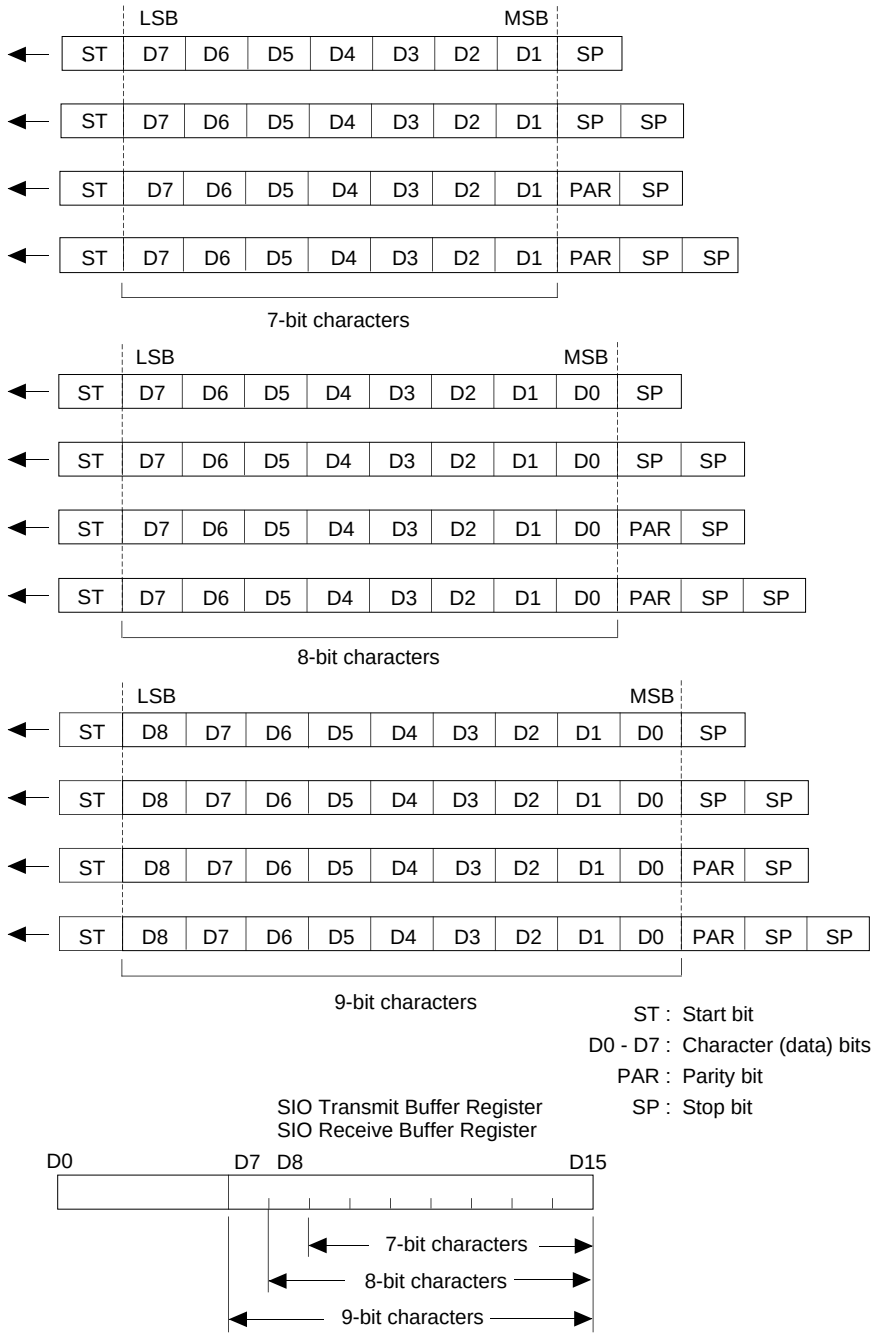


Figure 12.6.1 Example of Transmit/Receive Data Format in UART Mode

Table 12.6.1 Transfer Data in UART Mode

Bit Name	Content
ST (start bit)	Indicates the beginning of data transmission. This is a low signal of a one bit duration, which is added immediately before the transmit data.
D0-D8 (character bits)	Transmit/receive data transferred via serial I/O. In UART mode, data in 7, 8, or 9 bits can be transmitted/received.
PAR (parity bit)	Added to the transmit/receive characters. When parity is enabled, parity is automatically set in such a way that the number of 1's in characters 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, and is added immediately after characters (or if parity enabled, immediately after the parity bit). The stop bit can be chosen to be one bit or two bits long.



Note 1 : The high-order bits of the SIO Receive Buffer Register's selected character bits are fixed to 0.
Note 2 : The data bit numbers (Dn) above indicate bit numbers in a data list, and not the register bit numbers (Dn).

Figure 12.6.2 Selectable Data Formats during UART Mode

12.6.3 Initial Settings for UART Transmission

To transmit data in UART mode, initialize the serial I/O 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 stop bit length
- Set 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

- Select the clock divider's divide-by ratio.

(3) Setting SIO Baud Rate Register

Set a baud rate generator value. (Refer to Section 12.6.1, "Setting the UART Baud Rate.")

(4) Setting SIO Interrupt Mask Register

- Enable or disable SIO transmit interrupt.

(5) Setting the Interrupt Controller (SIO Transmit Interrupt Control Register)

When you use a transmit interrupt, set its priority level.

(6) Setting DMAC

When you issue DMA transfer requests to the internal DMAC when the transmit buffer is empty, set the DMAC. (Refer to Chapter 9, "DMAC.")

(7) Selecting pin functions

Because the serial I/O related pins serve dual purposes (shared with input/output ports), set pin functions. (Refer to Chapter 8, "Input/Output Ports and Pin Functions.")

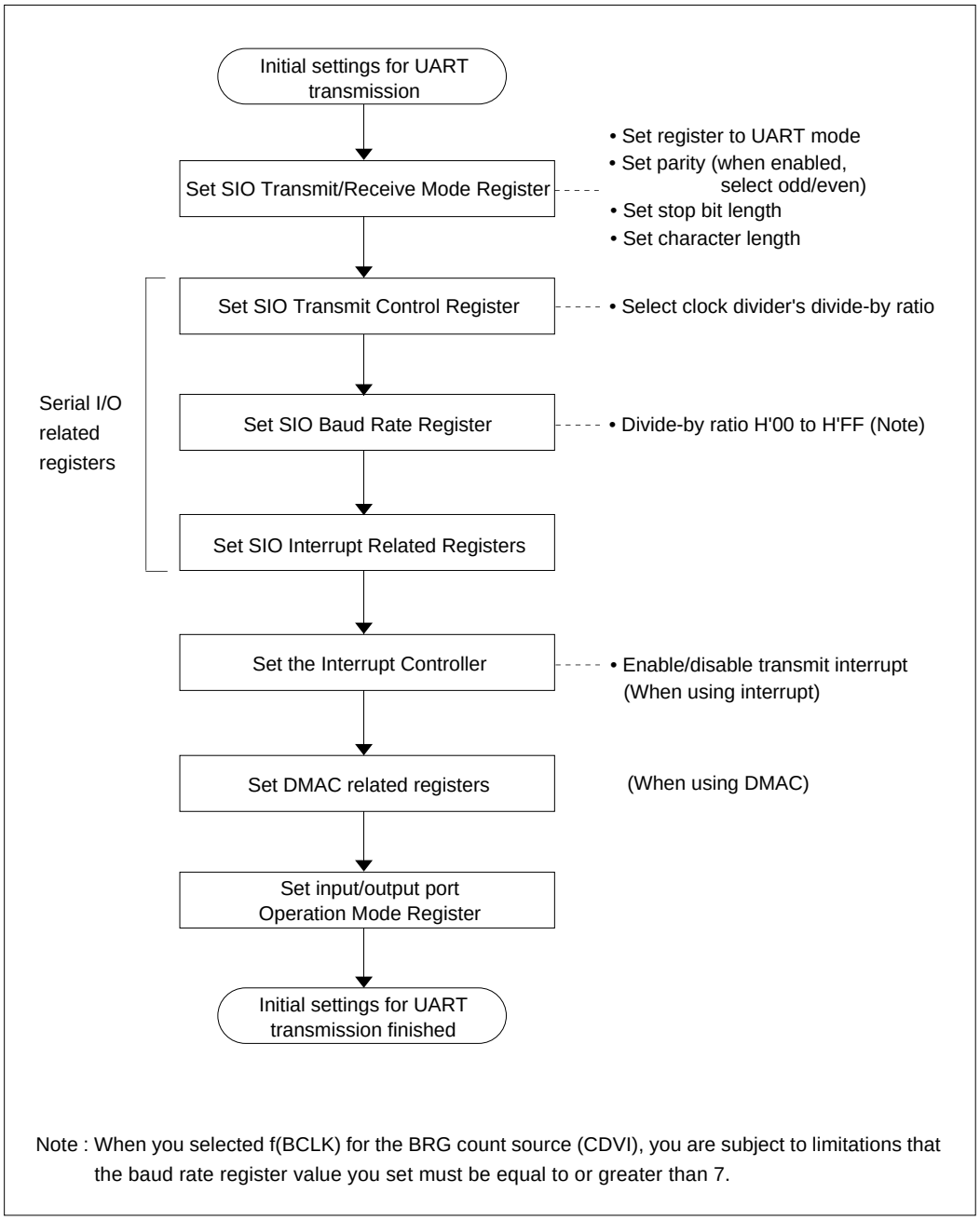


Figure 12.6.3 Procedure for UART Transmit Initialization

12.6.4 Starting UART Transmission

When all of the following transmit conditions are met after you finished initialization, the serial I/O starts transmit operation.

- The SIO Transmit Control Register's TEN (transmit enable) bit is set to 1. (Note)
- Transmit data is written to the SIO Transmit Buffer Register (transmit buffer empty bit = 0).

Note: 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 you write to the transmit buffer register.

When transmission starts, the serial I/O transmits data following the procedure 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)
- Start sending data synchronously with the shift clock beginning with the LSB.

Note: A transmit buffer empty interrupt request and/or a DMA transfer request can be generated when the transmit buffer is emptied.

12.6.5 Successive UART Transmission

Once data is transferred from the transmit buffer register to the transmit shift register, the next data can be written to the transmit buffer register even when transmission of the preceding data is not completed. When the next data is written to the transmit buffer before completion of the preceding data transmission, the preceding and the next data are successively transmitted.

To see if data has been transferred from the transmit buffer register to the transmit shift register, check the SIO Transmit Control Register's transmit buffer empty flag.

12.6.6 Processing at End of UART Transmission

When data transmission is completed, the following operation is automatically performed in hardware.

(1) When not transmitting successively

- The transmit status bit is set to 0.

(2) When transmitting successively

- When transmission of the last data in a consecutive data train is completed, the transmit status bit is set to 0.

12.6.7 Transmit Interrupt

If a transmit buffer empty interrupt has been enabled by the SIO Interrupt Mask Register, a transmit buffer empty interrupt is generated at the time data is transferred from the transmit buffer register to the transmit shift register. Also, a transmit buffer empty interrupt is generated when the TEN (transmit enable) bit is set to 1 (enabled after being disabled) while a transmit buffer empty interrupt has been enabled.
You must set the Interrupt Controller (ICU) before you can use transmit interrupts.

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 ouput to the DMAC. This transfer request is also output when the TEN (transmit enable) bit is set to 1 (enabled after being disabled).
You must set the Interrupt Controller (ICU) before you can transmit data using DMA transfers.

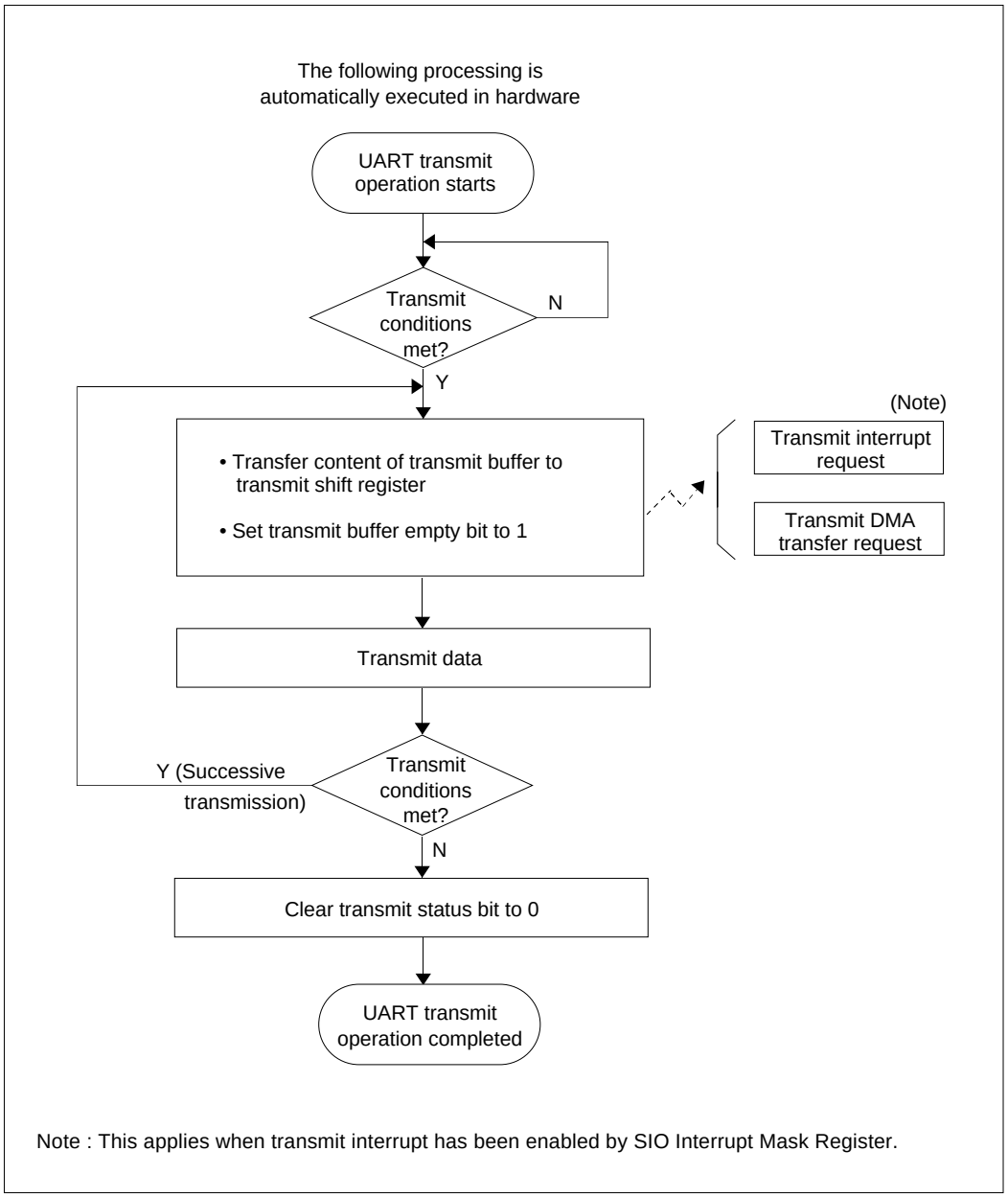
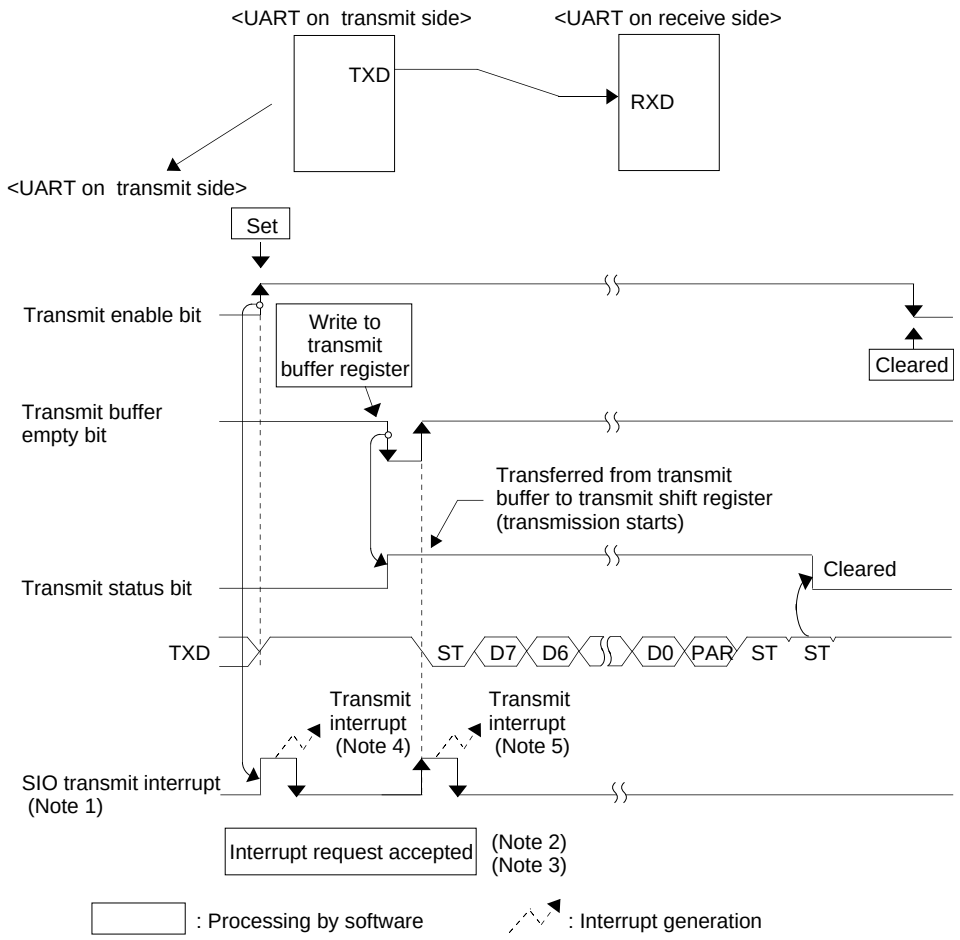


Figure 12.6.4 Transmit Operation during UART Mode (Hardware Processing)

12.6.9 Typical UART Transmit Operation

The following shows a typical transmit operation in CSIO mode.



Note 1 : Change of the Interrupt Controller "SIO Transmit Interrupt Control Register" interrupt request bit

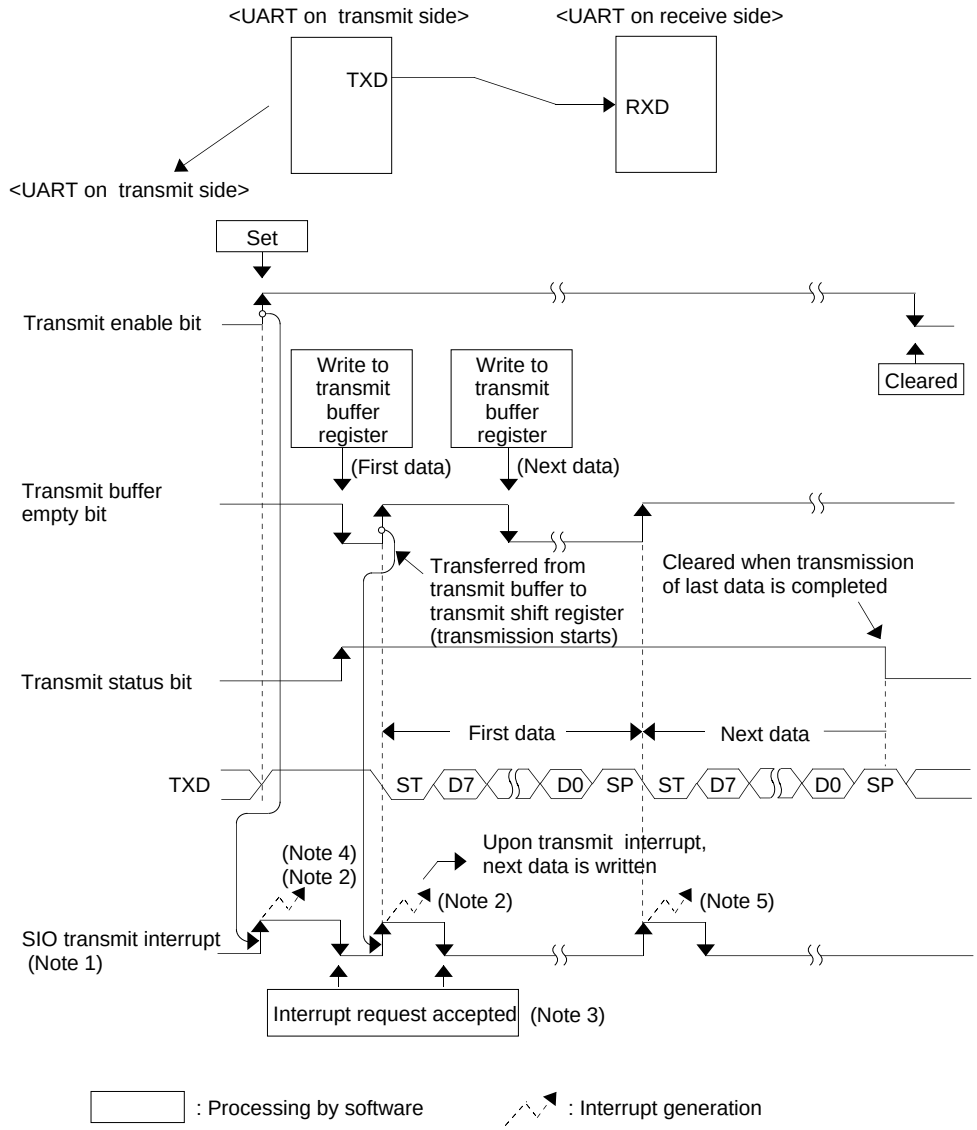
Note 2 : When transmit-finished interrupt is enabled (DMA transfer can also be requested at the same timing)

Note 3 : The Interrupt Controller IVECT register is read or "SIO Transmit Interrupt Control Register" interrupt request bit cleared

Note 4 : Transmit interrupt request is generated when transmission is enabled.

Note 5 : 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.

Figure 12.6.5 Example of UART Transmission (Transmitted Only Once, with Transmit Interrupt Used)



Note 1 : Change of the Interrupt Controller "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 timing)

Note 3 : The Interrupt Controller IVECT register is read or "SIO Transmit Interrupt Control Register" interrupt request bit cleared

Note 4 : Transmit interrupt request is generated when transmission is enabled.

Note 5 : 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.

Figure 12.6.6 Example of UART Transmission (Successive Transmission, with Transmit Interrupt Used)

12.7 Receive Operation in UART Mode

12.7.1 Initial Settings for UART Reception

To receive data in UART mode, initialize the serial I/O 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 stop bit length
- Set 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

- Select the clock divider's divide-by ratio.

(3) Setting SIO Baud Rate Register

Set a baud rate generator value. (Refer to Section 12.6.1, "Setting the UART Baud Rate.")

(4) Setting SIO interrupt related registers

- Cause of Receive Interrupt Select Register
  Select the cause of receive interrupt (receive finished/receive error)
- Interrupt Mask Register
  Enable/disable receive interrupts

(5) Setting the Interrupt Controller

When you use interrupts during reception, set its priority level.

(6) Setting DMAC

When you issue DMA transfer requests to the internal DMAC when reception is completed, set the DMAC. (Refer to Chapter 9, "DMAC.")

(7) Selecting pin functions

Because the serial I/O related pins serve dual purposes (shared with input/output ports), set pin functions. (Refer to Chapter 8, "Input/Output Ports and Pin Functions.")

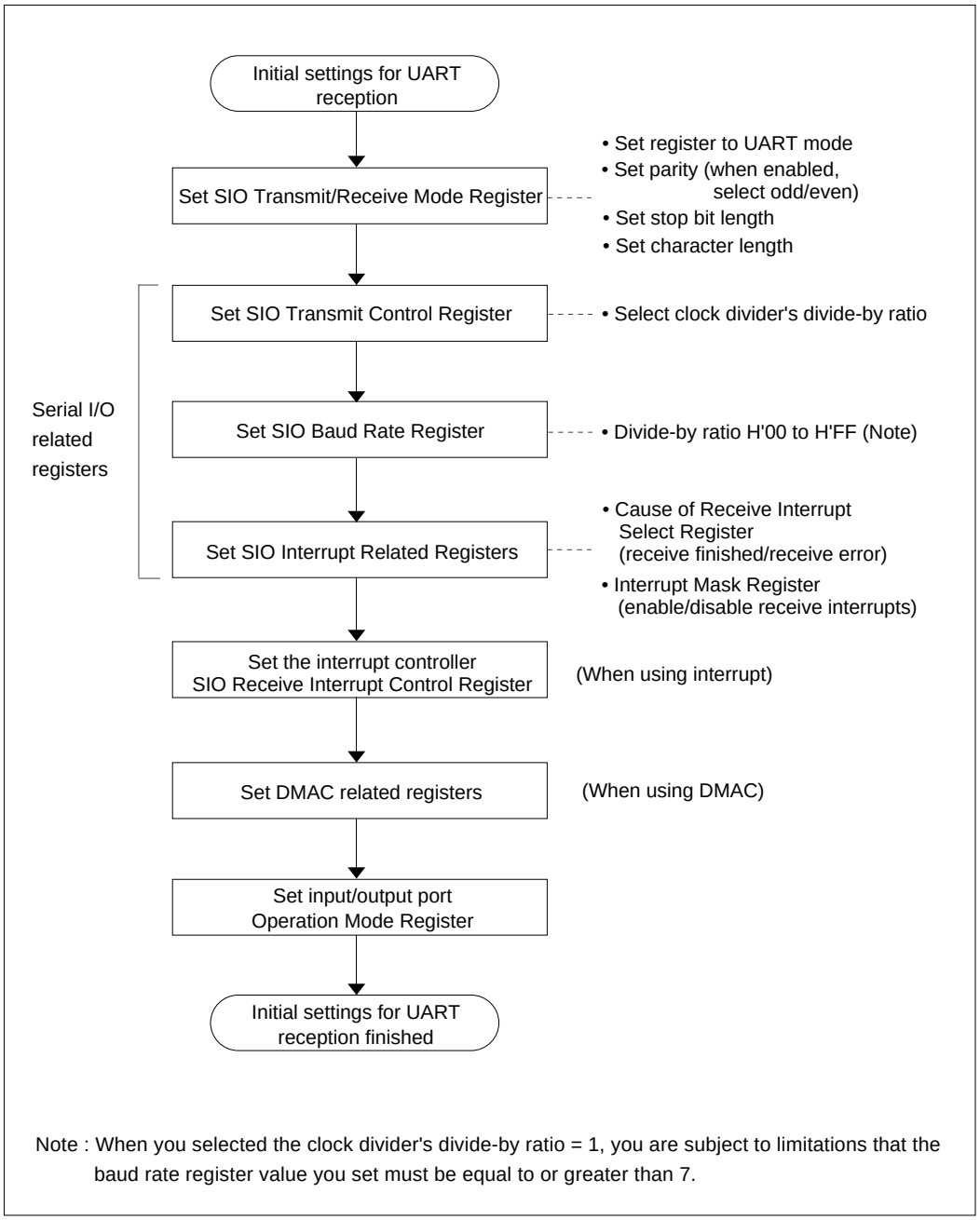


Figure 12.7.1 Procedure for UART Receive Initialization

12.7.2 Starting UART Reception

When all of the following receive conditions are met after you finished initialization, the serial I/O starts receive operation.

- The SIO Receive Control Register's 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 I/O enters UART receive operation. However, if the start bit when checked again at the first rise of the internal receive shift clock is detected high for reason of noise, etc., the serial I/O stops receive operation and waits for the start bit again.

12.7.3 Processing at End of UART Reception

When data reception is completed, the following operation is automatically performed in hardware.

(1) When reception is completed normally

The receive-finished (receive buffer full) bit is set to 1.

Note 1: If a receive-finished (receive buffer full) interrupt has been enabled, an interrupt request is generated.
Note 2: A DMA transfer request is generated.

(2) When error occurs during reception

When an error occurs during reception, the corresponding error bit (OE, FE, or PE) and the receive sum bit are set to 1.

Note 1: If a receive-finished interrupt has been selected (by SIO Cause of Receive Interrupt Select Register), a receive-finished interrupt request is generated when interrupt requests are enabled. However, if an overrun error has occurred, this interrupt is not generated.
Note 2: If a receive error interrupt has been selected (by SIO Cause of Receive Interrupt Select Register), a receive error interrupt request is generated when interrupt requests are enabled.
Note 3: No DMA transfer requests are generated.

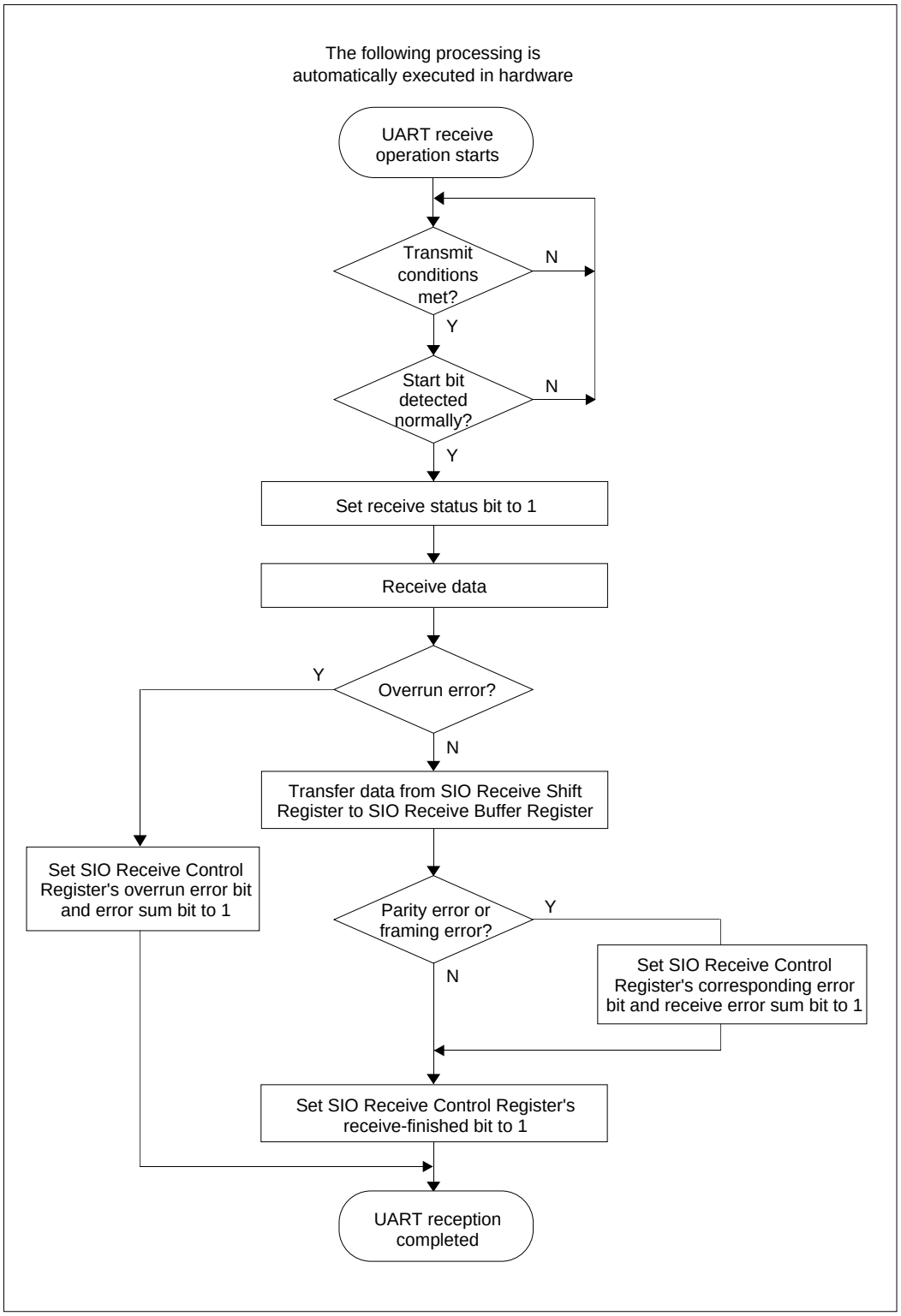
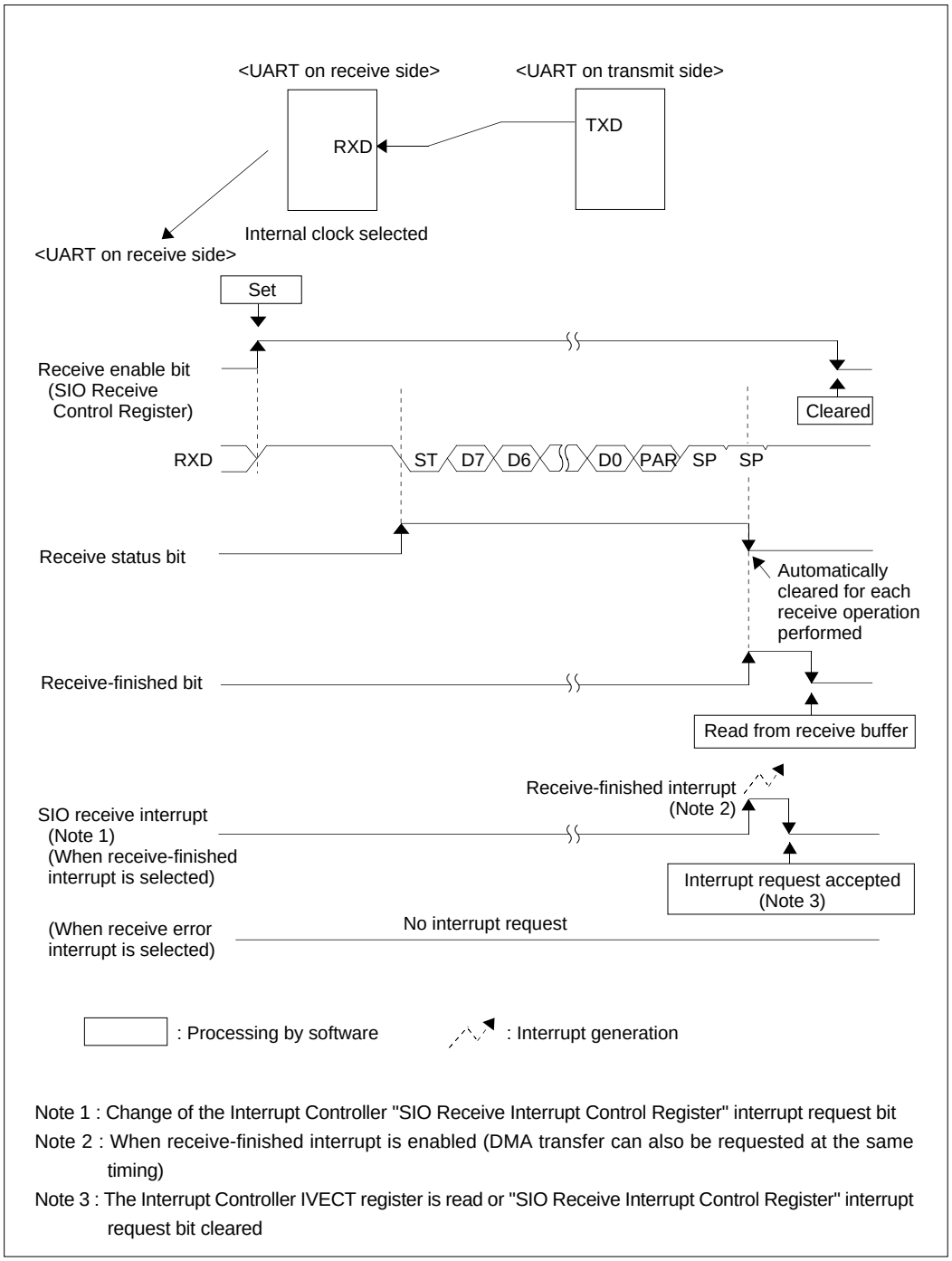


Figure 12.7.2 Receive Operation during UART Mode (Hardware Processing)

12.7.4 Typical UART Receive Operation

The following shows a typical receive operation in UART mode.

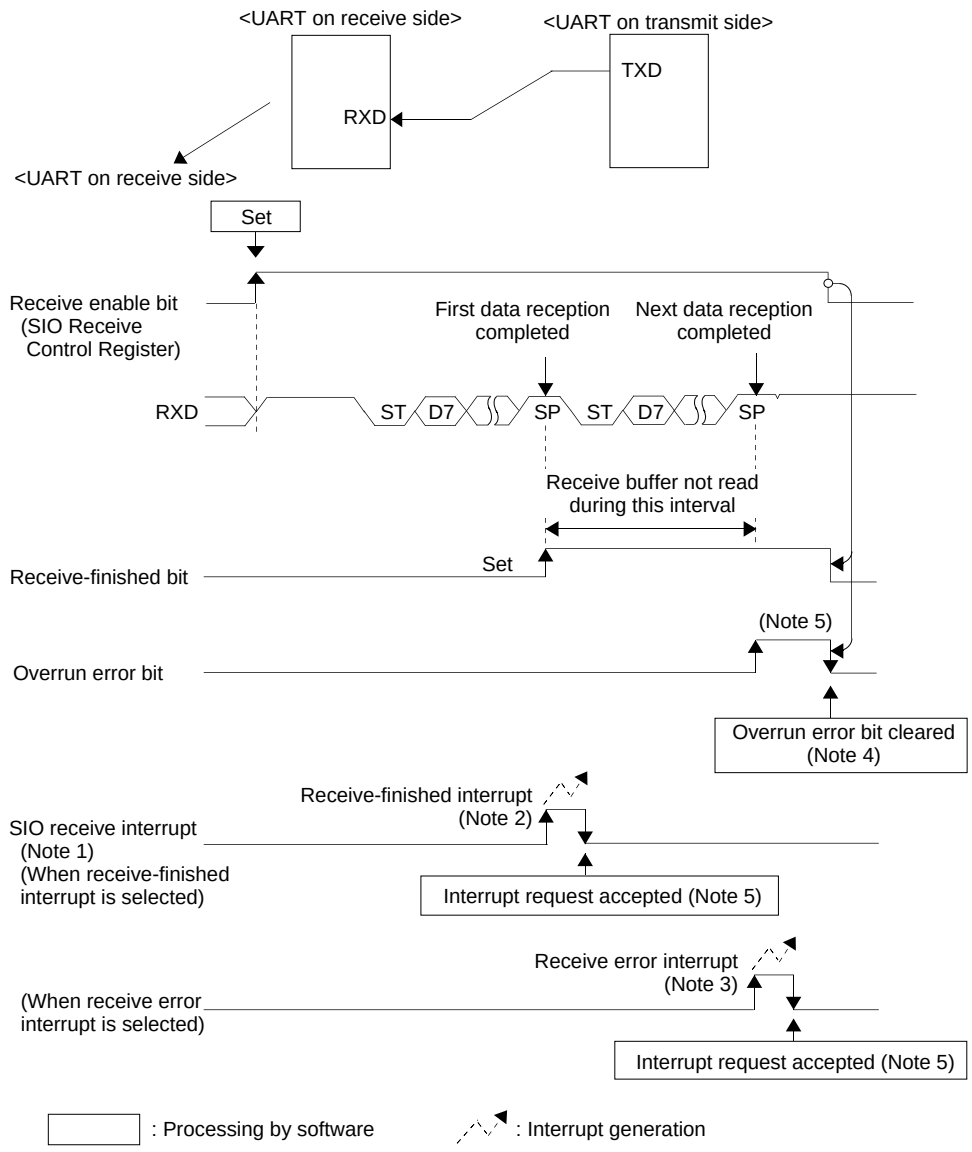


Note 1 : Change of the Interrupt Controller "SIO Receive Interrupt Control Register" interrupt request bit

Note 2 : When receive-finished interrupt is enabled (DMA transfer can also be requested at the same timing)

Note 3 : The Interrupt Controller IVECT register is read or "SIO Receive Interrupt Control Register" interrupt request bit cleared

Figure 12.7.3 Example of UART Reception (When Received Normally)



Note 1 : Change of the Interrupt Controller "SIO Receive Interrupt Control Register" interrupt request bit
Note 2 : When receive-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 IVECT register is read or "SIO Receive Interrupt Control Register" interrupt request bit cleared

Figure 12.7.4 Example of UART Reception (When Overrun Error Occurred)

12.8 Fixed Period Clock Output Function

When using SIO0, SIO1, SIO4 or SIO5 in UART mode, you can choose the relevant port (P84, P87, P65 or P66) to function as the SCLKO0, SCLKO1, SCLKO4 or SCLKO5 pin. In this way, a clock derived from BRG output by dividing it by 2 can be output from the SCLKO pin.

Note: This clock is output all the time, not just during data transfer.

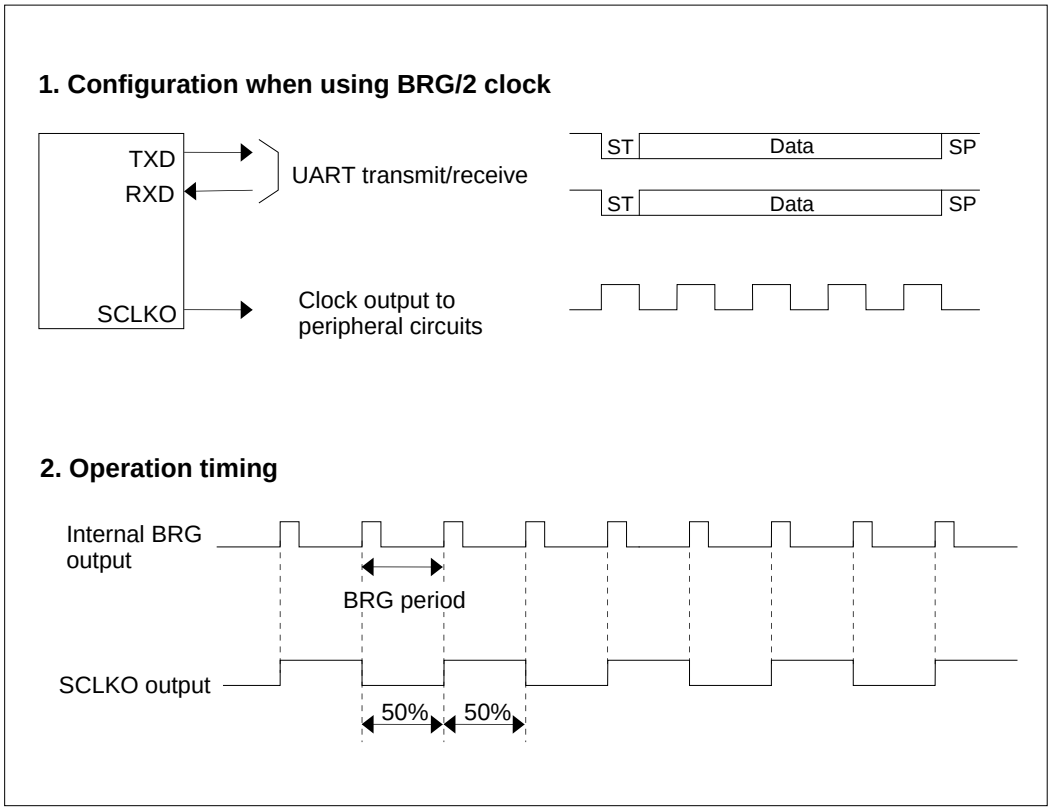


Figure 12.8.1 Example of Fixed Period Clock Output

12.9 Precautions 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 not operating. When transmitting or receiving data, be sure to check that transmission and/or reception under way has been completed and clear the transmit and receive enable bits before you set the registers.

- **Settings of Baud Rate (BRG) Register**

  If you selected f(BCLK) with the BRG clock source select bit, make sure the BRG register value you set is equal to or greater than 7.

  The value written to the SIO Baud Rate Register becomes effective beginning with the next period after the BRG counter finished counting. However, when transmit and receive operations are disabled, the register value can be changed at the same time you write to the register.

- **Transmit/receive operations using DMA**

  To transmit/receive data in DMA request mode, enable the DMAC to accept transfer requests (by setting the DMA Mode Register) before you start serial communication.

- **About overrun error**

  If all bits of the next receive data are received in the SIO Receive Shift Register before you read out the SIO Receive Buffer Register (an overrun error occurs), the receive data is not stored in the Receive Buffer Register and the Receive Buffer Register retains the previously received data. Once an overrun error occurs, no receive data is stored in the Receive Buffer Register although receive operation is continued. To restart reception normally, you need to temporarily clear the receive enable bit before you restart. This is the only way you can clear the overrun error flag.

- **Flags indicating the status of UART receive operation**

Following flags are available that indicate the status of receive operation during UART mode.

- SIO Receive Control Register receive status bit
- SIO Receive Control Register receive-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 receive-finished bit and various error bit flags are cleared varies depending on whether an overrun error has occurred or not, as described below.

[When no overrun error has occurred]

Said bits can be cleared by reading the lower byte from the receive buffer register or clearing the receive enable bit to 0.

[When an overrun error has occurred]

Said bits can only be cleared by clearing the receive enable bit to 0.

CHAPTER 13

CAN MODULES

13.1 Outline of the CAN Modules

The 32172/32173 contains two Full CAN modules (CAN0 and CAN1) compliant with CAN (Controller Area Network) Specification 2.0B active. CAN0 and CAN1 each have 16 message slots and three mask registers. Making use of these message slots and mask registers helps to reduce data processing load on the CPU.

The Full CAN modules are outlined below.

Table 13.1.1 Outline of the CAN Modules

Item	Content
Protocol	CAN Specification 2.0B active
Number of message slots	16 slots (14 global slots and two local slots) x 2
Polarity	0: Dominant 1: Recessive
Acceptance filter	Global mask: one mask x 2 Local mask: two masks x 2
Baud rate (Note)	1 Time quantum (Tq) = (BRP + 1) / CPU clock (BRP: baud rate prescaler set value) Baud rate = $\frac{1}{\text{Tq period x number of Tq's in one bit}}$ ••• Max 1 Mbps BRP : 1-255 (0: disable) Number of Tq's in one bit = Synchronization Segment + Propagation Segment + Phase Segment 1 + Phase Segment 2 Propagation Segment : 1-8Tq Phase Segment 1 : 1-8Tq Phase Segment 2 : 2-8Tq (IPT = 2)
Remote frame automatic response function	The message slot that received a remote frame automatically to send a data frame.
Time stamp function	This time stamp function is based on a 16-bit counter. The count period can be setto divide-by-1, 2, 3, or 4 with respect to the CAN bus bit period as the fundamental period.
BasicCAN mode	BasicCAN function is materialized using two local slots.
Transmit abort function	Transmit request can be canceled.
Loopback function	The data transmitted by CAN module itself is received.
Return bus-off function	Forcibly placed into error active mode after clearing the error counter.

Note: The maximum allowable error of oscillation depends on the system configuration (bus length, communication rate, CAN bus transceiver, sampling position, and bit configuration).

Table 13.1.2 Interrupt Generating Functions of the CAN Modules

CAN module interrupt source	ICU interrupt source
CAN0 transmit complete interrupt	CAN0 transmit/receive & error interrupt
CAN0 receive complete interrupt	CAN0 transmit/receive & error interrupt
CAN0 bus error interrupt	CAN0 transmit/receive & error interrupt
CAN0 error-passive interrupt	CAN0 transmit/receive & error interrupt
CAN0 bus-off interrupt	CAN0 transmit/receive & error interrupt
CAN1 transmit complete interrupt	CAN1 transmit/receive & error interrupt
CAN1 receive complete interrupt	CAN1 transmit/receive & error interrupt
CAN1 bus error interrupt	CAN1 transmit/receive & error interrupt
CAN1 error-passive interrupt	CAN1 transmit/receive & error interrupt
CAN1 bus-off interrupt	CAN1 transmit/receive & error interrupt

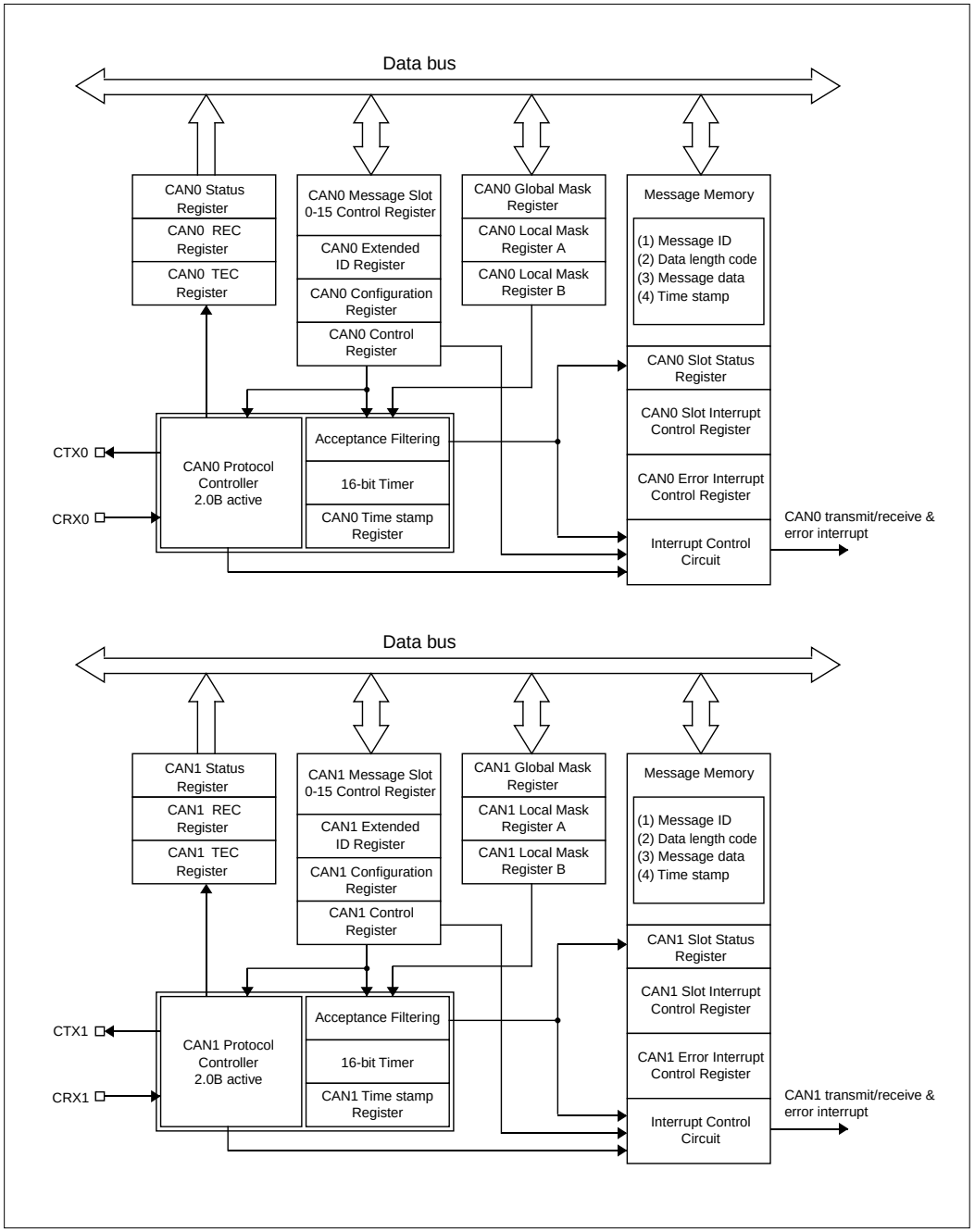


Figure 13.1.1 Block Diagram of the CAN Modules

13.2 CAN Module Related Registers

A CAN module related register map is shown below.

Address	+0 address D0	D7, D8 +1 address D15
H'0080 1000	CAN0 Control Register(CAN0CNT)	
H'0080 1002	CAN0 Status Register(CAN0STAT)	
H'0080 1004	CAN0 Extended ID Register(CAN0EXTID)	
H'0080 1006	CAN0 Configuration Register(CAN0CONF)	
H'0080 1008	CAN0 Time stamp Count Register(CAN0TSTMP)	
H'0080 100A	CAN0 Receive Error Count Register(CAN0REC)	CAN0 Transmit Error Count Register(CAN0TEC)
H'0080 100C	CAN0 Slot Interrupt Status Register(CAN0SLIST)	
H'0080 100E		
H'0080 1010	CAN0 Slot Interrupt Mask Register(CAN0SLIMK)	
H'0080 1012		
H'0080 1014	CAN0 Error Interrupt Status Register(CAN0ERIST)	CAN0 Error Interrupt Mask Register(CAN0ERIMK)
H'0080 1016	CAN0 Baud Rate Prescaler(CAN0BRP)	
H'0080 1028	CAN0 Global Mask Register Standard ID0(C0GMSKS0)	CAN0 Global Mask Register Standard ID1(C0GMSKS1)
H'0080 102A	CAN0 Global Mask Register Extended ID0(C0GMSKE0)	CAN0 Global Mask Register Extended ID1(C0GMSKE1)
H'0080 102C	CAN0 Global Mask Register Extended ID2(C0GMSKE2)	
H'0080 102E		
H'0080 1030	CAN0 Local Mask Register A Standard ID0(C0LMSKAS0)	CAN0 Local Mask Register A Standard ID1(C0LMSKAS1)
H'0080 1032	CAN0 Local Mask Register A Extended ID0(C0LMSKAE0)	CAN0 Local Mask Register A Extended ID1(C0LMSKAE1)
H'0080 1034	CAN0 Local Mask Register A Extended ID2(C0LMSKAE2)	
H'0080 1036		
H'0080 1038	CAN0 Local Mask Register B Standard ID0(C0LMSKAS0)	CAN0 Local Mask Register B Standard ID1(C0LMSKAS1)
H'0080 103A	CAN0 Local Mask Register B Extended ID0(C0LMSKAE0)	CAN0 Local Mask Register B Extended ID1(C0LMSKAE1)
H'0080 103C	CAN0 Local Mask Register B Extended ID2(C0LMSKAE2)	
H'0080 1050	CAN0 Message Slot 0 Control Register(C0MSL0CNT)	CAN0 Message Slot 1 Control Register(C0MSL1CNT)
H'0080 1052	CAN0 Message Slot 2 Control Register(C0MSL2CNT)	CAN0 Message Slot 3 Control Register(C0MSL3CNT)
H'0080 1054	CAN0 Message Slot 4 Control Register(C0MSL4CNT)	CAN0 Message Slot 5 Control Register(C0MSL5CNT)
H'0080 1056	CAN0 Message Slot 6 Control Register(C0MSL6CNT)	CAN0 Message Slot 7 Control Register(C0MSL7CNT)
H'0080 1058	CAN0 Message Slot 8 Control Register(C0MSL8CNT)	CAN0 Message Slot 9 Control Register(C0MSL9CNT)
H'0080 105A	CAN0 Message Slot 10 Control Register(C0MSL10CNT)	CAN0 Message Slot 11 Control Register(C0MSL11CNT)
H'0080 105C	CAN0 Message Slot 12 Control Register(C0MSL12CNT)	CAN0 Message Slot 13 Control Register(C0MSL13CNT)
H'0080 105E	CAN0 Message Slot 14 Control Register(C0MSL14CNT)	CAN0 Message Slot 15 Control Register(C0MSL15CNT)

Blank areas are reserved for future use.

Figure 13.2.1 CAN Module Related Register Map (1/8)

Address	+0 address D0	D7	+1 address D8	D15
H'0080 1100	CAN0 Message Slot 0 Standard ID0(C0MSL0SID0)		CAN0 Message Slot 0 Standard ID1(C0MSL0SID1)	
H'0080 1102	CAN0 Message Slot 0 Extended ID0(C0MSL0EID0)		CAN0 Message Slot 0 Extended ID1(C0MSL0EID1)	
H'0080 1104	CAN0 Message Slot 0 Extended ID2(C0MSL0EID2)		CAN0 Message Slot 0 Data Length Register(C0MSL0DLC)	
H'0080 1106	CAN0 Message Slot 0 Data 0(C0MSL0DT0)		CAN0 Message Slot 0 Data 1(C0MSL0DT1)	
H'0080 1108	CAN0 Message Slot 0 Data 2(C0MSL0DT2)		CAN0 Message Slot 0 Data 3(C0MSL0DT3)	
H'0080 110A	CAN0 Message Slot 0 Data 4(C0MSL0DT4)		CAN0 Message Slot 0 Data 5(C0MSL0DT5)	
H'0080 110C	CAN0 Message Slot 0 Data 6(C0MSL0DT6)		CAN0 Message Slot 0 Data 7(C0MSL0DT7)	
H'0080 110E	CAN0 Message Slot 0 Time stamp(C0MSL0TSP)			
H'0080 1110	CAN0 Message Slot 1 Standard ID0(C0MSL1SID0)		CAN0 Message Slot 1 Standard ID1(C0MSL1SID1)	
H'0080 1112	CAN0 Message Slot 1 Extended ID0(C0MSL1EID0)		CAN0 Message Slot 1 Extended ID1(C0MSL1EID1)	
H'0080 1114	CAN0 Message Slot 1 Extended ID2(C0MSL1EID2)		CAN0 Message Slot 1 Data Length Register(C0MSL1DLC)	
H'0080 1116	CAN0 Message Slot 1 Data 0(C0MSL1DT0)		CAN0 Message Slot 1 Data 1(C0MSL1DT1)	
H'0080 1118	CAN0 Message Slot 1 Data 2(C0MSL1DT2)		CAN0 Message Slot 1 Data 3(C0MSL1DT3)	
H'0080 111A	CAN0 Message Slot 1 Data 4(C0MSL1DT4)		CAN0 Message Slot 1 Data 5(C0MSL1DT5)	
H'0080 111C	CAN0 Message Slot 1 Data 6(C0MSL1DT6)		CAN0 Message Slot 1 Data 7(C0MSL1DT7)	
H'0080 111E	CAN0 Message Slot 1 Time stamp(C0MSL1TSP)			
H'0080 1120	CAN0 Message Slot 2 Standard ID0(C0MSL2SID0)		CAN0 Message Slot 2 Standard ID1(C0MSL2SID1)	
H'0080 1122	CAN0 Message Slot 2 Extended ID0(C0MSL2EID0)		CAN0 Message Slot 2 Extended ID1(C0MSL2EID1)	
H'0080 1124	CAN0 Message Slot 2 Extended ID2(C0MSL2EID2)		CAN0 Message Slot 2 Data Length Register(C0MSL2DLC)	
H'0080 1126	CAN0 Message Slot 2 Data 0(C0MSL2DT0)		CAN0 Message Slot 2 Data 1(C0MSL2DT1)	
H'0080 1128	CAN0 Message Slot 2 Data 2(C0MSL2DT2)		CAN0 Message Slot 2 Data 3(C0MSL2DT3)	
H'0080 112A	CAN0 Message Slot 2 Data 4(C0MSL2DT4)		CAN0 Message Slot 2 Data 5(C0MSL2DT5)	
H'0080 112C	CAN0 Message Slot 2 Data 6(C0MSL2DT6)		CAN0 Message Slot 2 Data 7(C0MSL2DT7)	
H'0080 112E	CAN0 Message Slot 2 Time stamp(C0MSL2TSP)			
H'0080 1130	CAN0 Message Slot 3 Standard ID0(C0MSL3SID0)		CAN0 Message Slot 3 Standard ID1(C0MSL3SID1)	
H'0080 1132	CAN0 Message Slot 3 Extended ID0(C0MSL3EID0)		CAN0 Message Slot 3 Extended ID1(C0MSL3EID1)	
H'0080 1134	CAN0 Message Slot 3 Extended ID2(C0MSL3EID2)		CAN0 Message Slot 3 Data Length Register(C0MSL3DLC)	
H'0080 1136	CAN0 Message Slot 3 Data 0(C0MSL3DT0)		CAN0 Message Slot 3 Data 1(C0MSL3DT1)	
H'0080 1138	CAN0 Message Slot 3 Data 2(C0MSL3DT2)		CAN0 Message Slot 3 Data 3(C0MSL3DT3)	
H'0080 113A	CAN0 Message Slot 3 Data 4(C0MSL3DT4)		CAN0 Message Slot 3 Data 5(C0MSL3DT5)	
H'0080 113C	CAN0 Message Slot 3 Data 6(C0MSL3DT6)		CAN0 Message Slot 3 Data 7(C0MSL3DT7)	
H'0080 113E	CAN0 Message Slot 3 Time stamp(C0MSL3TSP)			
H'0080 1140	CAN0 Message Slot 4 Standard ID0(C0MSL4SID0)		CAN0 Message Slot 4 Standard ID1(C0MSL4SID1)	
H'0080 1142	CAN0 Message Slot 4 Extended ID0(C0MSL4EID0)		CAN0 Message Slot 4 Extended ID1(C0MSL4EID1)	
H'0080 1144	CAN0 Message Slot 4 Extended ID2(C0MSL4EID2)		CAN0 Message Slot 4 Data Length Register(C0MSL4DLC)	
H'0080 1146	CAN0 Message Slot 4 Data 0(C0MSL4DT0)		CAN0 Message Slot 4 Data 1(C0MSL4DT1)	
H'0080 1148	CAN0 Message Slot 4 Data 2(C0MSL4DT2)		CAN0 Message Slot 4 Data 3(C0MSL4DT3)	
H'0080 114A	CAN0 Message Slot 4 Data 4(C0MSL4DT4)		CAN0 Message Slot 4 Data 5(C0MSL4DT5)	
H'0080 114C	CAN0 Message Slot 4 Data 6(C0MSL4DT6)		CAN0 Message Slot 4 Data 7(C0MSL4DT7)	
H'0080 114E	CAN0 Message Slot 4 Time stamp(C0MSL4TSP)			
H'0080 1150	CAN0 Message Slot 5 Standard ID0(C0MSL5SID0)		CAN0 Message Slot 5 Standard ID1(C0MSL5SID1)	
H'0080 1152	CAN0 Message Slot 5 Extended ID0(C0MSL5EID0)		CAN0 Message Slot 5 Extended ID1(C0MSL5EID1)	

Blank areas are reserved for future use.

Figure 13.2.2 CAN Module Related Register Map (2/8)

Address	+0 address D0 D7	+1 address D8 D15
H'0080 1154	CAN0 Message Slot 5 Extended ID2(C0MSL5EID2)	CAN0 Message Slot 5 Data Length Register(C0MSL5DLC)
H'0080 1156	CAN0 Message Slot 5 Data 0(C0MSL5DT0)	CAN0 Message Slot 5 Data 1(C0MSL5DT1)
H'0080 1158	CAN0 Message Slot 5 Data 2(C0MSL5DT2)	CAN0 Message Slot 5 Data 3(C0MSL5DT3)
H'0080 115A	CAN0 Message Slot 5 Data 4(C0MSL5DT4)	CAN0 Message Slot 5 Data 5(C0MSL5DT5)
H'0080 115C	CAN0 Message Slot 5 Data 6(C0MSL5DT6)	CAN0 Message Slot 5 Data 7(C0MSL5DT7)
H'0080 115E	CAN0 Message Slot 5 Time stamp(C0MSL5TSP)	
H'0080 1160	CAN0 Message Slot 6 Standard ID0(C0MSL6SID0)	CAN0 Message Slot 6 Standard ID1(C0MSL6SID1)
H'0080 1162	CAN0 Message Slot 6 Extended ID0(C0MSL6EID0)	CAN0 Message Slot 6 Extended ID1(C0MSL6EID1)
H'0080 1164	CAN0 Message Slot 6 Extended ID2(C0MSL6EID2)	CAN0 Message Slot 6 Data Length Register(C0MSL6DLC)
H'0080 1166	CAN0 Message Slot 6 Data 0(C0MSL6DT0)	CAN0 Message Slot 6 Data 1(C0MSL6DT1)
H'0080 1168	CAN0 Message Slot 6 Data 2(C0MSL6DT2)	CAN0 Message Slot 6 Data 3(C0MSL6DT3)
H'0080 116A	CAN0 Message Slot 6 Data 4(C0MSL6DT4)	CAN0 Message Slot 6 Data 5(C0MSL6DT5)
H'0080 116C	CAN0 Message Slot 6 Data 6(C0MSL6DT6)	CAN0 Message Slot 6 Data 7(C0MSL6DT7)
H'0080 116E	CAN0 Message Slot 6 Time stamp(C0MSL6TSP)	
H'0080 1170	CAN0 Message Slot 7 Standard ID0(C0MSL7SID0)	CAN0 Message Slot 7 Standard ID1(C0MSL7SID1)
H'0080 1172	CAN0 Message Slot 7 Extended ID0(C0MSL7EID0)	CAN0 Message Slot 7 Extended ID1(C0MSL7EID1)
H'0080 1174	CAN0 Message Slot 7 Extended ID2(C0MSL7EID2)	CAN0 Message Slot 7 Data Length Register(C0MSL7DLC)
H'0080 1176	CAN0 Message Slot 7 Data 0(C0MSL7DT0)	CAN0 Message Slot 7 Data 1(C0MSL7DT1)
H'0080 1178	CAN0 Message Slot 7 Data 2(C0MSL7DT2)	CAN0 Message Slot 7 Data 3(C0MSL7DT3)
H'0080 117A	CAN0 Message Slot 7 Data 4(C0MSL7DT4)	CAN0 Message Slot 7 Data 5(C0MSL7DT5)
H'0080 117C	CAN0 Message Slot 7 Data 6(C0MSL7DT6)	CAN0 Message Slot 7 Data 7(C0MSL7DT7)
H'0080 117E	CAN0 Message Slot 7 Time stamp(C0MSL7TSP)	
H'0080 1180	CAN0 Message Slot 8 Standard ID0(C0MSL8SID0)	CAN0 Message Slot 8 Standard ID1(C0MSL8SID1)
H'0080 1182	CAN0 Message Slot 8 Extended ID0(C0MSL8EID0)	CAN0 Message Slot 8 Extended ID1(C0MSL8EID1)
H'0080 1184	CAN0 Message Slot 8 Extended ID2(C0MSL8EID2)	CAN0 Message Slot 8 Data Length Register(C0MSL8DLC)
H'0080 1186	CAN0 Message Slot 8 Data 0(C0MSL8DT0)	CAN0 Message Slot 8 Data 1(C0MSL8DT1)
H'0080 1188	CAN0 Message Slot 8 Data 2(C0MSL8DT2)	CAN0 Message Slot 8 Data 3(C0MSL8DT3)
H'0080 118A	CAN0 Message Slot 8 Data 4(C0MSL8DT4)	CAN0 Message Slot 8 Data 5(C0MSL8DT5)
H'0080 118C	CAN0 Message Slot 8 Data 6(C0MSL8DT6)	CAN0 Message Slot 8 Data 7(C0MSL8DT7)
H'0080 118E	CAN0 Message Slot 8 Time stamp(C0MSL8TSP)	
H'0080 1190	CAN0 Message Slot 9 Standard ID0(C0MSL9SID0)	CAN0 Message Slot 9 Standard ID1(C0MSL9SID1)
H'0080 1192	CAN0 Message Slot 9 Extended ID0(C0MSL9EID0)	CAN0 Message Slot 9 Extended ID1(C0MSL9EID1)
H'0080 1194	CAN0 Message Slot 9 Extended ID2(C0MSL9EID2)	CAN0 Message Slot 9 Data Length Register(C0MSL9DLC)
H'0080 1196	CAN0 Message Slot 9 Data 0(C0MSL9DT0)	CAN0 Message Slot 9 Data 1(C0MSL9DT1)
H'0080 1198	CAN0 Message Slot 9 Data 2(C0MSL9DT2)	CAN0 Message Slot 9 Data 3(C0MSL9DT3)
H'0080 119A	CAN0 Message Slot 9 Data 4(C0MSL9DT4)	CAN0 Message Slot 9 Data 5(C0MSL9DT5)
H'0080 119C	CAN0 Message Slot 9 Data 6(C0MSL9DT6)	CAN0 Message Slot 9 Data 7(C0MSL9DT7)
H'0080 119E	CAN0 Message Slot 9 Time stamp(C0MSL9TSP)	
H'0080 11A0	CAN0 Message Slot 10 Standard ID0(C0MSL10SID0)	CAN0 Message Slot 10 Standard ID1(C0MSL10SID1)
H'0080 11A2	CAN0 Message Slot 10 Extended ID0(C0MSL10EID0)	CAN0 Message Slot 10 Extended ID1(C0MSL10EID1)
H'0080 11A4	CAN0 Message Slot 10 Extended ID2(C0MSL10EID2)	CAN0 Message Slot 10 Data Length Register(C0MSL10DLC)
H'0080 11A6	CAN0 Message Slot 10 Data 0(C0MSL10DT0)	CAN0 Message Slot 10 Data 1(C0MSL10DT1)

Blank areas are reserved for future use.

Figure 13.2.3 CAN Module Related Register Map (3/8)

Address	+0 address D0	+0 address D7	+1 address D8	+1 address D15
H'0080 11A8	CAN0 Message Slot 10 Data 2(C0MSL10DT2)		CAN0 Message Slot 10 Data 3(C0MSL10DT3)	
H'0080 11AA	CAN0 Message Slot 10 Data 4(C0MSL10DT4)		CAN0 Message Slot 10 Data 5(C0MSL10DT5)	
H'0080 11AC	CAN0 Message Slot 10 Data 6(C0MSL10DT6)		CAN0 Message Slot 10 Data 7(C0MSL10DT7)	
H'0080 11AE	CAN0 Message Slot 10 Time stamp(C0MSL10TSP)			
H'0080 11B0	CAN0 Message Slot 11 Standard ID0(C0MSL11SID0)		CAN0 Message Slot 11 Standard ID1(C0MSL11SID1)	
H'0080 11B2	CAN0 Message Slot 11 Extended ID0(C0MSL11EID0)		CAN0 Message Slot 11 Extended ID1(C0MSL11EID1)	
H'0080 11B4	CAN0 Message Slot 11 Extended ID2(C0MSL11EID2)		CAN0 Message Slot 11 Data Length Register(C0MSL11DLC)	
H'0080 11B6	CAN0 Message Slot 11 Data 0(C0MSL11DT0)		CAN0 Message Slot 11 Data 1(C0MSL11DT1)	
H'0080 11B8	CAN0 Message Slot 11 Data 2(C0MSL11DT2)		CAN0 Message Slot 11 Data 3(C0MSL11DT3)	
H'0080 11BA	CAN0 Message Slot 11 Data 4(C0MSL11DT4)		CAN0 Message Slot 11 Data 5(C0MSL11DT5)	
H'0080 11BC	CAN0 Message Slot 11 Data 6(C0MSL11DT6)		CAN0 Message Slot 11 Data 7(C0MSL11DT7)	
H'0080 11BE	CAN0 Message Slot 11 Time stamp(C0MSL11TSP)			
H'0080 11C0	CAN0 Message Slot 12 Standard ID0(C0MSL12SID0)		CAN0 Message Slot 12 Standard ID1(C0MSL12SID1)	
H'0080 11C2	CAN0 Message Slot 12 Extended ID0(C0MSL12EID0)		CAN0 Message Slot 12 Extended ID1(C0MSL12EID1)	
H'0080 11C4	CAN0 Message Slot 12 Extended ID2(C0MSL12EID2)		CAN0 Message Slot 12 Data Length Register(C0MSL12DLC)	
H'0080 11C6	CAN0 Message Slot 12 Data 0(C0MSL12DT0)		CAN0 Message Slot 12 Data 1(C0MSL12DT1)	
H'0080 11C8	CAN0 Message Slot 12 Data 2(C0MSL12DT2)		CAN0 Message Slot 12 Data 3(C0MSL12DT3)	
H'0080 11CA	CAN0 Message Slot 12 Data 4(C0MSL12DT4)		CAN0 Message Slot 12 Data 5(C0MSL12DT5)	
H'0080 11CC	CAN0 Message Slot 12 Data 6(C0MSL12DT6)		CAN0 Message Slot 12 Data 7(C0MSL12DT7)	
H'0080 11CE	CAN0 Message Slot 12 Time stamp(C0MSL12TSP)			
H'0080 11D0	CAN0 Message Slot 13 Standard ID0(C0MSL13SID0)		CAN0 Message Slot 13 Standard ID1(C0MSL13SID1)	
H'0080 11D2	CAN0 Message Slot 13 Extended ID0(C0MSL13EID0)		CAN0 Message Slot 13 Extended ID1(C0MSL13EID1)	
H'0080 11D4	CAN0 Message Slot 13 Extended ID2(C0MSL13EID2)		CAN0 Message Slot 13 Data Length Register(C0MSL13DLC)	
H'0080 11D6	CAN0 Message Slot 13 Data 0(C0MSL13DT0)		CAN0 Message Slot 13 Data 1(C0MSL13DT1)	
H'0080 11D8	CAN0 Message Slot 13 Data 2(C0MSL13DT2)		CAN0 Message Slot 13 Data 3(C0MSL13DT3)	
H'0080 11DA	CAN0 Message Slot 13 Data 4(C0MSL13DT4)		CAN0 Message Slot 13 Data 5(C0MSL13DT5)	
H'0080 11DC	CAN0 Message Slot 13 Data 6(C0MSL13DT6)		CAN0 Message Slot 13 Data 7(C0MSL13DT7)	
H'0080 11DE	CAN0 Message Slot 13 Time stamp(C0MSL13TSP)			
H'0080 11E0	CAN0 Message Slot 14 Standard ID0(C0MSL14SID0)		CAN0 Message Slot 14 Standard ID1(C0MSL14SID1)	
H'0080 11E2	CAN0 Message Slot 14 Extended ID0(C0MSL14EID0)		CAN0 Message Slot 14 Extended ID1(C0MSL14EID1)	
H'0080 11E4	CAN0 Message Slot 14 Extended ID2(C0MSL14EID2)		CAN0 Message Slot 14 Data Length Register(C0MSL14DLC)	
H'0080 11E6	CAN0 Message Slot 14 Data 0(C0MSL14DT0)		CAN0 Message Slot 14 Data 1(C0MSL14DT1)	
H'0080 11E8	CAN0 Message Slot 14 Data 2(C0MSL14DT2)		CAN0 Message Slot 14 Data 3(C0MSL14DT3)	
H'0080 11EA	CAN0 Message Slot 14 Data 4(C0MSL14DT4)		CAN0 Message Slot 14 Data 5(C0MSL14DT5)	
H'0080 11EC	CAN0 Message Slot 14 Data 6(C0MSL14DT6)		CAN0 Message Slot 14 Data 7(C0MSL14DT7)	
H'0080 11EE	CAN0 Message Slot 14 Time stamp(C0MSL14TSP)			
H'0080 11F0	CAN0 Message Slot 15 Standard ID0(C0MSL15SID0)		CAN0 Message Slot 15 Standard ID1(C0MSL15SID1)	
H'0080 11F2	CAN0 Message Slot 15 Extended ID0(C0MSL15EID0)		CAN0 Message Slot 15 Extended ID1(C0MSL15EID1)	
H'0080 11F4	CAN0 Message Slot 15 Extended ID2(C0MSL15EID2)		CAN0 Message Slot 15 Data Length Register(C0MSL15DLC)	
H'0080 11F6	CAN0 Message Slot 15 Data 0(C0MSL15DT0)		CAN0 Message Slot 15 Data 1(C0MSL15DT1)	
H'0080 11F8	CAN0 Message Slot 15 Data 2(C0MSL15DT2)		CAN0 Message Slot 15 Data 3(C0MSL15DT3)	
H'0080 11FA	CAN0 Message Slot 15 Data 4(C0MSL15DT4)		CAN0 Message Slot 15 Data 5(C0MSL15DT5)	
H'0080 11FC	CAN0 Message Slot 15 Data 6(C0MSL15DT6)		CAN0 Message Slot 15 Data 7(C0MSL15DT7)	
H'0080 11FE	CAN0 Message Slot 15 Time stamp(C0MSL15TSP)			

Blank areas are reserved for future use.

Figure 13.2.4 CAN Module Related Register Map (4/8)

Address	+0 address D0	D7 D8	+1 address D15
H'0080 1400	CAN1 Control Register(CAN1CNT)		
H'0080 1402	CAN1 Status Register(CAN1STAT)		
H'0080 1404	CAN1 Extended ID Register(CAN1EXTID)		
H'0080 1406	CAN1 Configuration Register(CAN1CONF)		
H'0080 1408	CAN1 Time stamp Count Register(CAN1TSTMP)		
H'0080 140A	CAN1 Receive Error Count Register(CAN1REC)		CAN1 Transmit Error Count Register(CAN1TEC)
H'0080 140C	CAN1 Slot Interrupt Status Register(CAN1SLIST)		
H'0080 140E			
H'0080 1410	CAN1 Slot Interrupt Mask Register(CAN1SLIMK)		
H'0080 1412			
H'0080 1414	CAN1 Error Interrupt Status Register(CAN1ERIST)		CAN1 Error Interrupt Mask Register(CAN1ERIMK)
H'0080 1416	CAN1 Baud Rate Prescaler(CAN1BRP)		
≈			
H'0080 1428	CAN1 Global Mask Register Standard ID0(C1GMSKS0)		CAN1 Global Mask Register Standard ID1(C1GMSKS1)
H'0080 142A	CAN1 Global Mask Register Extended ID0(C1GMSKE0)		CAN1 Global Mask Register Extended ID1(C1GMSKE1)
H'0080 142C	CAN1 Global Mask Register Extended ID2(C1GMSKE2)		
H'0080 142E			
H'0080 1430	CAN1 Local Mask Register A Standard ID0(C1LMSKAS0)		CAN1 Local Mask Register A Standard ID1(C1LMSKAS1)
H'0080 1432	CAN1 Local Mask Register A Extended ID0(C1LMSKAE0)		CAN1 Local Mask Register A Extended ID1(C1LMSKAE1)
H'0080 1434	CAN1 Local Mask Register A Extended ID2(C1LMSKAE2)		
H'0080 1436			
H'0080 1438	CAN1 Local Mask Register B Standard ID0(C1LMSKAS0)		CAN1 Local Mask Register B Standard ID1(C1LMSKAS1)
H'0080 143A	CAN1 Local Mask Register B Extended ID0(C1LMSKAE0)		CAN1 Local Mask Register B Extended ID1(C1LMSKAE1)
H'0080 143C	CAN1 Local Mask Register B Extended ID2(C1LMSKAE2)		
≈			
H'0080 1450	CAN1 Message Slot 0 Control Register(C1MSL0CNT)		CAN1 Message Slot 1 Control Register(C1MSL1CNT)
H'0080 1452	CAN1 Message Slot 2 Control Register(C1MSL2CNT)		CAN1 Message Slot 3 Control Register(C1MSL3CNT)
H'0080 1454	CAN1 Message Slot 4 Control Register(C1MSL4CNT)		CAN1 Message Slot 5 Control Register(C1MSL5CNT)
H'0080 1456	CAN1 Message Slot 6 Control Register(C1MSL6CNT)		CAN1 Message Slot 7 Control Register(C1MSL7CNT)
H'0080 1458	CAN1 Message Slot 8 Control Register(C1MSL8CNT)		CAN1 Message Slot 9 Control Register(C1MSL9CNT)
H'0080 145A	CAN1 Message Slot 10 Control Register(C1MSL10CNT)		CAN1 Message Slot 11 Control Register(C1MSL11CNT)
H'0080 145C	CAN1 Message Slot 12 Control Register(C1MSL12CNT)		CAN1 Message Slot 13 Control Register(C1MSL13CNT)
H'0080 145E	CAN1 Message Slot 14 Control Register(C1MSL14CNT)		CAN1 Message Slot 15 Control Register(C1MSL15CNT)
≈			

Blank areas are reserved for future use.

Figure 13.2.5 CAN Module Related Register Map (5/8)

Address	+0 address D0	+0 address D7	+1 address D8	+1 address D15
H'0080 1500	CAN1 Message Slot 0 Standard ID0(C1MSL0SID0)		CAN1 Message Slot 0 Standard ID1(C1MSL0SID1)	
H'0080 1502	CAN1 Message Slot 0 Extended ID0(C1MSL0EID0)		CAN1 Message Slot 0 Extended ID1(C1MSL0EID1)	
H'0080 1504	CAN1 Message Slot 0 Extended ID2(C1MSL0EID2)		CAN1 Message Slot 0 Data Length Register(C1MSL0DLC)	
H'0080 1506	CAN1 Message Slot 0 Data 0(C1MSL0DT0)		CAN1 Message Slot 0 Data 1(C1MSL0DT1)	
H'0080 1508	CAN1 Message Slot 0 Data 2(C1MSL0DT2)		CAN1 Message Slot 0 Data 3(C1MSL0DT3)	
H'0080 150A	CAN1 Message Slot 0 Data 4(C1MSL0DT4)		CAN1 Message Slot 0 Data 5(C1MSL0DT5)	
H'0080 150C	CAN1 Message Slot 0 Data 6(C1MSL0DT6)		CAN1 Message Slot 0 Data 7(C1MSL0DT7)	
H'0080 150E	CAN1 Message Slot 0 Time stamp(C1MSL0TSP)			
H'0080 1510	CAN1 Message Slot 1 Standard ID0(C1MSL1SID0)		CAN1 Message Slot 1 Standard ID1(C1MSL1SID1)	
H'0080 1512	CAN1 Message Slot 1 Extended ID0(C1MSL1EID0)		CAN1 Message Slot 1 Extended ID1(C1MSL1EID1)	
H'0080 1514	CAN1 Message Slot 1 Extended ID2(C1MSL1EID2)		CAN1 Message Slot 1 Data Length Register(C1MSL1DLC)	
H'0080 1516	CAN1 Message Slot 1 Data 0(C1MSL1DT0)		CAN1 Message Slot 1 Data 1(C1MSL1DT1)	
H'0080 1518	CAN1 Message Slot 1 Data 2(C1MSL1DT2)		CAN1 Message Slot 1 Data 3(C1MSL1DT3)	
H'0080 151A	CAN1 Message Slot 1 Data 4(C1MSL1DT4)		CAN1 Message Slot 1 Data 5(C1MSL1DT5)	
H'0080 151C	CAN1 Message Slot 1 Data 6(C1MSL1DT6)		CAN1 Message Slot 1 Data 7(C1MSL1DT7)	
H'0080 151E	CAN1 Message Slot 1 Time stamp(C1MSL1TSP)			
H'0080 1520	CAN1 Message Slot 2 Standard ID0(C1MSL2SID0)		CAN1 Message Slot 2 Standard ID1(C1MSL2SID1)	
H'0080 1522	CAN1 Message Slot 2 Extended ID0(C1MSL2EID0)		CAN1 Message Slot 2 Extended ID1(C1MSL2EID1)	
H'0080 1524	CAN1 Message Slot 2 Extended ID2(C1MSL2EID2)		CAN1 Message Slot 2 Data Length Register(C1MSL2DLC)	
H'0080 1526	CAN1 Message Slot 2 Data 0(C1MSL2DT0)		CAN1 Message Slot 2 Data 1(C1MSL2DT1)	
H'0080 1528	CAN1 Message Slot 2 Data 2(C1MSL2DT2)		CAN1 Message Slot 2 Data 3(C1MSL2DT3)	
H'0080 152A	CAN1 Message Slot 2 Data 4(C1MSL2DT4)		CAN1 Message Slot 2 Data 5(C1MSL2DT5)	
H'0080 152C	CAN1 Message Slot 2 Data 6(C1MSL2DT6)		CAN1 Message Slot 2 Data 7(C1MSL2DT7)	
H'0080 152E	CAN1 Message Slot 2 Time stamp(C1MSL2TSP)			
H'0080 1530	CAN1 Message Slot 3 Standard ID0(C1MSL3SID0)		CAN1 Message Slot 3 Standard ID1(C1MSL3SID1)	
H'0080 1532	CAN1 Message Slot 3 Extended ID0(C1MSL3EID0)		CAN1 Message Slot 3 Extended ID1(C1MSL3EID1)	
H'0080 1534	CAN1 Message Slot 3 Extended ID2(C1MSL3EID2)		CAN1 Message Slot 3 Data Length Register(C1MSL3DLC)	
H'0080 1536	CAN1 Message Slot 3 Data 0(C1MSL3DT0)		CAN1 Message Slot 3 Data 1(C1MSL3DT1)	
H'0080 1538	CAN1 Message Slot 3 Data 2(C1MSL3DT2)		CAN1 Message Slot 3 Data 3(C1MSL3DT3)	
H'0080 153A	CAN1 Message Slot 3 Data 4(C1MSL3DT4)		CAN1 Message Slot 3 Data 5(C1MSL3DT5)	
H'0080 153C	CAN1 Message Slot 3 Data 6(C1MSL3DT6)		CAN1 Message Slot 3 Data 7(C1MSL3DT7)	
H'0080 153E	CAN1 Message Slot 3 Time stamp(C1MSL3TSP)			
H'0080 1540	CAN1 Message Slot 4 Standard ID0(C1MSL4SID0)		CAN1 Message Slot 4 Standard ID1(C1MSL4SID1)	
H'0080 1542	CAN1 Message Slot 4 Extended ID0(C1MSL4EID0)		CAN1 Message Slot 4 Extended ID1(C1MSL4EID1)	
H'0080 1544	CAN1 Message Slot 4 Extended ID2(C1MSL4EID2)		CAN0 Message Slot 4 Data Length Register(C1MSL4DLC)	
H'0080 1546	CAN1 Message Slot 4 Data 0(C1MSL4DT0)		CAN1 Message Slot 4 Data 1(C1MSL4DT1)	
H'0080 1548	CAN1 Message Slot 4 Data 2(C1MSL4DT2)		CAN1 Message Slot 4 Data 3(C1MSL4DT3)	
H'0080 154A	CAN1 Message Slot 4 Data 4(C1MSL4DT4)		CAN1 Message Slot 4 Data 5(C1MSL4DT5)	
H'0080 154C	CAN1 Message Slot 4 Data 6(C1MSL4DT6)		CAN1 Message Slot 4 Data 7(C1MSL4DT7)	
H'0080 154E	CAN1 Message Slot 4 Time stamp(C1MSL4TSP)			
H'0080 1550	CAN1 Message Slot 5 Standard ID0(C1MSL5SID0)		CAN1 Message Slot 5 Standard ID1(C1MSL5SID1)	
H'0080 1552	CAN1 Message Slot 5 Extended ID0(C1MSL5EID0)		CAN1 Message Slot 5 Extended ID1(C1MSL5EID1)	

Blank areas are reserved for future use.

Figure 13.2.6 CAN Module Related Register Map (6/8)

Address	+0 address D0	D7	+1 address D8	D15
H'0080 1554	CAN1 Message Slot 5 Extended ID2(C1MSL5EID2)		CAN1 Message Slot 5 Data Length Register(C1MSL5DLC)	
H'0080 1556	CAN1 Message Slot 5 Data 0(C1MSL5DT0)		CAN1 Message Slot 5 Data 1(C1MSL5DT1)	
H'0080 1558	CAN1 Message Slot 5 Data 2(C1MSL5DT2)		CAN1 Message Slot 5 Data 3(C1MSL5DT3)	
H'0080 155A	CAN1 Message Slot 5 Data 4(C1MSL5DT4)		CAN1 Message Slot 5 Data 5(C1MSL5DT5)	
H'0080 155C	CAN1 Message Slot 5 Data 6(C1MSL5DT6)		CAN1 Message Slot 5 Data 7(C1MSL5DT7)	
H'0080 155E	CAN1 Message Slot 5 Time stamp(C1MSL5TSP)			
H'0080 1560	CAN1 Message Slot 6 Standard ID0(C1MSL6SID0)		CAN1 Message Slot 6 Standard ID1(C1MSL6SID1)	
H'0080 1562	CAN1 Message Slot 6 Extended ID0(C1MSL6EID0)		CAN1 Message Slot 6 Extended ID1(C1MSL6EID1)	
H'0080 1564	CAN1 Message Slot 6 Extended ID2(C1MSL6EID2)		CAN1 Message Slot 6 Data Length Register(C1MSL6DLC)	
H'0080 1566	CAN1 Message Slot 6 Data 0(C1MSL6DT0)		CAN1 Message Slot 6 Data 1(C1MSL6DT1)	
H'0080 1568	CAN1 Message Slot 6 Data 2(C1MSL6DT2)		CAN1 Message Slot 6 Data 3(C1MSL6DT3)	
H'0080 156A	CAN1 Message Slot 6 Data 4(C1MSL6DT4)		CAN1 Message Slot 6 Data 5(C1MSL6DT5)	
H'0080 156C	CAN1 Message Slot 6 Data 6(C1MSL6DT6)		CAN1 Message Slot 6 Data 7(C1MSL6DT7)	
H'0080 156E	CAN1 Message Slot 6 Time stamp(C1MSL6TSP)			
H'0080 1570	CAN1 Message Slot 7 Standard ID0(C1MSL7SID0)		CAN1 Message Slot 7 Standard ID1(C1MSL7SID1)	
H'0080 1572	CAN1 Message Slot 7 Extended ID0(C1MSL7EID0)		CAN1 Message Slot 7 Extended ID1(C1MSL7EID1)	
H'0080 1574	CAN1 Message Slot 7 Extended ID2(C1MSL7EID2)		CAN1 Message Slot 7 Data Length Register(C1MSL7DLC)	
H'0080 1576	CAN1 Message Slot 7 Data 0(C1MSL7DT0)		CAN1 Message Slot 7 Data 1(C1MSL7DT1)	
H'0080 1578	CAN1 Message Slot 7 Data 2(C1MSL7DT2)		CAN1 Message Slot 7 Data 3(C1MSL7DT3)	
H'0080 157A	CAN1 Message Slot 7 Data 4(C1MSL7DT4)		CAN1 Message Slot 7 Data 5(C1MSL7DT5)	
H'0080 157C	CAN1 Message Slot 7 Data 6(C1MSL7DT6)		CAN1 Message Slot 7 Data 7(C1MSL7DT7)	
H'0080 157E	CAN1 Message Slot 7 Time stamp(C1MSL7TSP)			
H'0080 1580	CAN1 Message Slot 8 Standard ID0(C1MSL8SID0)		CAN1 Message Slot 8 Standard ID1(C1MSL8SID1)	
H'0080 1582	CAN1 Message Slot 8 Extended ID0(C1MSL8EID0)		CAN1 Message Slot 8 Extended ID1(C1MSL8EID1)	
H'0080 1584	CAN1 Message Slot 8 Extended ID2(C1MSL8EID2)		CAN1 Message Slot 8 Data Length Register(C1MSL8DLC)	
H'0080 1586	CAN1 Message Slot 8 Data 0(C1MSL8DT0)		CAN1 Message Slot 8 Data 1(C1MSL8DT1)	
H'0080 1588	CAN1 Message Slot 8 Data 2(C1MSL8DT2)		CAN1 Message Slot 8 Data 3(C1MSL8DT3)	
H'0080 158A	CAN1 Message Slot 8 Data 4(C1MSL8DT4)		CAN1 Message Slot 8 Data 5(C1MSL8DT5)	
H'0080 158C	CAN1 Message Slot 8 Data 6(C1MSL8DT6)		CAN1 Message Slot 8 Data 7(C1MSL8DT7)	
H'0080 158E	CAN1 Message Slot 8 Time stamp(C1MSL8TSP)			
H'0080 1590	CAN1 Message Slot 9 Standard ID0(C1MSL9SID0)		CAN1 Message Slot 9 Standard ID1(C1MSL9SID1)	
H'0080 1592	CAN1 Message Slot 9 Extended ID0(C1MSL9EID0)		CAN1 Message Slot 9 Extended ID1(C1MSL9EID1)	
H'0080 1594	CAN1 Message Slot 9 Extended ID2(C1MSL9EID2)		CAN1 Message Slot 9 Data Length Register(C1MSL9DLC)	
H'0080 1596	CAN1 Message Slot 9 Data 0(C1MSL9DT0)		CAN1 Message Slot 9 Data 1(C1MSL9DT1)	
H'0080 1598	CAN1 Message Slot 9 Data 2(C1MSL9DT2)		CAN1 Message Slot 9 Data 3(C1MSL9DT3)	
H'0080 159A	CAN1 Message Slot 9 Data 4(C1MSL9DT4)		CAN1 Message Slot 9 Data 5(C1MSL9DT5)	
H'0080 159C	CAN1 Message Slot 9 Data 6(C1MSL9DT6)		CAN1 Message Slot 9 Data 7(C1MSL9DT7)	
H'0080 159E	CAN1 Message Slot 9 Time stamp(C1MSL9TSP)			
H'0080 15A0	CAN1 Message Slot 10 Standard ID0(C1MSL10SID0)		CAN1 Message Slot 10 Standard ID1(C1MSL10SID1)	
H'0080 15A2	CAN1 Message Slot 10 Extended ID0(C1MSL10EID0)		CAN1 Message Slot 10 Extended ID1(C1MSL10EID1)	
H'0080 15A4	CAN1 Message Slot 10 Extended ID2(C1MSL10EID2)		CAN1 Message Slot 10 Data Length Register(C1MSL10DLC)	
H'0080 15A6	CAN1 Message Slot 10 Data 0(C1MSL10DT0)		CAN1 Message Slot 10 Data 1(C1MSL10DT1)	

Blank areas are reserved for future use.

Figure 13.2.7 CAN Module Related Register Map (7/8)

Address	+0 address D0	D7 D8 +1 address D15
H'0080 15A8	CAN1 Message Slot 10 Data 2(C1MSL10DT2)	CAN1 Message Slot 10 Data 3(C1MSL10DT3)
H'0080 15AA	CAN1 Message Slot 10 Data 4(C1MSL10DT4)	CAN1 Message Slot 10 Data 5(C1MSL10DT5)
H'0080 15AC	CAN1 Message Slot 10 Data 6(C1MSL10DT6)	CAN1 Message Slot 10 Data 7(C1MSL10DT7)
H'0080 15AE	CAN1 Message Slot 10 Time stamp(C1MSL9TSP)	
H'0080 15B0	CAN1 Message Slot 11 Standard ID0(C1MSL11SID0)	CAN1 Message Slot 11 Standard ID1(C1MSL11SID1)
H'0080 15B2	CAN1 Message Slot 11 Extended ID0(C1MSL11EID0)	CAN1 Message Slot 11 Extended ID1(C1MSL11EID1)
H'0080 15B4	CAN1 Message Slot 11 Extended ID2(C1MSL11EID2)	CAN1 Message Slot 11 Data Length Register(C1MSL11DLC)
H'0080 15B6	CAN1 Message Slot 11 Data 0(C1MSL11DT0)	CAN1 Message Slot 11 Data 1(C1MSL11DT1)
H'0080 15B8	CAN1 Message Slot 11 Data 2(C1MSL11DT2)	CAN1 Message Slot 11 Data 3(C1MSL11DT3)
H'0080 15BA	CAN1 Message Slot 11 Data 4(C1MSL11DT4)	CAN1 Message Slot 11 Data 5(C1MSL11DT5)
H'0080 15BC	CAN1 Message Slot 11 Data 6(C1MSL11DT6)	CAN1 Message Slot 11 Data 7(C1MSL11DT7)
H'0080 15BE	CAN1 Message Slot 11 Time stamp(C1MSL11TSP)	
H'0080 15C0	CAN1 Message Slot 12 Standard ID0(C1MSL12SID0)	CAN1 Message Slot 12 Standard ID1(C1MSL12SID1)
H'0080 15C2	CAN1 Message Slot 12 Extended ID0(C1MSL12EID0)	CAN1 Message Slot 12 Extended ID1(C1MSL12EID1)
H'0080 15C4	CAN1 Message Slot 12 Extended ID2(C1MSL12EID2)	CAN1 Message Slot 12 Data Length Register(C1MSL12DLC)
H'0080 15C6	CAN1 Message Slot 12 Data 0(C1MSL12DT0)	CAN1 Message Slot 12 Data 1(C1MSL12DT1)
H'0080 15C8	CAN1 Message Slot 12 Data 2(C1MSL12DT2)	CAN1 Message Slot 12 Data 3(C1MSL12DT3)
H'0080 15CA	CAN1 Message Slot 12 Data 4(C1MSL12DT4)	CAN1 Message Slot 12 Data 5(C1MSL12DT5)
H'0080 15CC	CAN1 Message Slot 12 Data 6(C1MSL12DT6)	CAN1 Message Slot 12 Data 7(C1MSL12DT7)
H'0080 15CE	CAN1 Message Slot 12 Time stamp(C1MSL12TSP)	
H'0080 15D0	CAN1 Message Slot 13 Standard ID0(C1MSL13SID0)	CAN1 Message Slot 13 Standard ID1(C1MSL13SID1)
H'0080 15D2	CAN1 Message Slot 13 Extended ID0(C1MSL13EID0)	CAN1 Message Slot 13 Extended ID1(C1MSL13EID1)
H'0080 15D4	CAN1 Message Slot 13 Extended ID2(C1MSL13EID2)	CAN1 Message Slot 13 Data Length Register(C1MSL13DLC)
H'0080 15D6	CAN1 Message Slot 13 Data 0(C1MSL13DT0)	CAN1 Message Slot 13 Data 1(C1MSL13DT1)
H'0080 15D8	CAN1 Message Slot 13 Data 2(C1MSL13DT2)	CAN1 Message Slot 13 Data 3(C1MSL13DT3)
H'0080 15DA	CAN1 Message Slot 13 Data 4(C1MSL13DT4)	CAN1 Message Slot 13 Data 5(C1MSL13DT5)
H'0080 15DC	CAN1 Message Slot 13 Data 6(C1MSL13DT6)	CAN1 Message Slot 13 Data 7(C1MSL13DT7)
H'0080 15DE	CAN1 Message Slot 13 Time stamp(C1MSL13TSP)	
H'0080 15E0	CAN1 Message Slot 14 Standard ID0(C1MSL14SID0)	CAN1 Message Slot 14 Standard ID1(C1MSL14SID1)
H'0080 15E2	CAN1 Message Slot 14 Extended ID0(C1MSL14EID0)	CAN1 Message Slot 14 Extended ID1(C1MSL14EID1)
H'0080 15E4	CAN1 Message Slot 14 Extended ID2(C1MSL14EID2)	CAN1 Message Slot 14 Data Length Register(C1MSL14DLC)
H'0080 15E6	CAN1 Message Slot 14 Data 0(C1MSL14DT0)	CAN1 Message Slot 14 Data 1(C1MSL14DT1)
H'0080 15E8	CAN1 Message Slot 14 Data 2(C1MSL14DT2)	CAN1 Message Slot 14 Data 3(C1MSL14DT3)
H'0080 15EA	CAN1 Message Slot 14 Data 4(C1MSL14DT4)	CAN1 Message Slot 14 Data 5(C1MSL14DT5)
H'0080 15EC	CAN1 Message Slot 14 Data 6(C1MSL14DT6)	CAN1 Message Slot 14 Data 7(C1MSL14DT7)
H'0080 15EE	CAN1 Message Slot 14 Time stamp(C1MSL14TSP)	
H'0080 15F0	CAN1 Message Slot 15 Standard ID0(C1MSL15SID0)	CAN1 Message Slot 15 Standard ID1(C1MSL15SID1)
H'0080 15F2	CAN1 Message Slot 15 Extended ID0(C1MSL15EID0)	CAN1 Message Slot 15 Extended ID1(C1MSL15EID1)
H'0080 15F4	CAN1 Message Slot 15 Extended ID2(C1MSL15EID2)	CAN1 Message Slot 15 Data Length Register(C1MSL15DLC)
H'0080 15F6	CAN1 Message Slot 15 Data 0(C1MSL15DT0)	CAN1 Message Slot 15 Data 1(C1MSL15DT1)
H'0080 15F8	CAN1 Message Slot 15 Data 2(C1MSL15DT2)	CAN1 Message Slot 15 Data 3(C1MSL15DT3)
H'0080 15FA	CAN1 Message Slot 15 Data 4(C1MSL15DT4)	CAN1 Message Slot 15 Data 5(C1MSL15DT5)
H'0080 15FC	CAN1 Message Slot 15 Data 6(C1MSL15DT6)	CAN1 Message Slot 15 Data 7(C1MSL15DT7)
H'0080 15FE	CAN1 Message Slot 15 Time stamp(C1MSL15TSP)	
H'0080 3FFE		

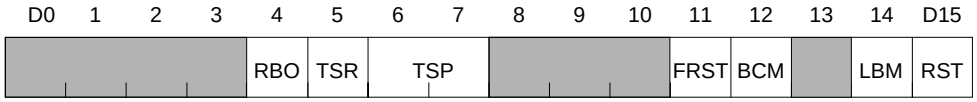
Blank areas are reserved for future use.

Figure 13.2.8 CAN Module Related Register Map (8/8)

13.2.1 CAN Control Registers

■ **CAN0 Control Register (CAN0CNT)** **<Address: H'0080 1000>**

■ **CAN1 Control Register (CAN1CNT)** **<Address: H'0080 1400>**



<When reset: H'0011>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4	RBO (Return bus off)	0 : Enables normal operation 1: Requests clearing of error counter	○	△
5	TSR (Time stamp counter reset)	0: Enables count operation 1: Initializes count (by setting H'0000)	○	△
6-7 (Time stamp prescaler)	TSP	D6 D7 0 0: Selects CAN bus bit clock 0 1: Selects divide-by-2 of CAN bus bit clock 1 0: Selects divide-by-3 of CAN bus bit clock 1 1: Selects divide-by-4 of CAN bus bit clock	○	○
8-9	No functions assigned		0	–
10	No functions assigned (Always set this bit to 0.)		0	–
11	FRST (Forcible reset)	0: Negates reset 1: Forcibly reset	○	○
12	BCM (BasicCAN mode)	0: Disables BasicCAN function 1: Enables BasicCAN mode	○	○
13	No functions assigned		0	–
14	LBM (Loopback mode)	0: Disables loopback function 1: Enables loopback function	○	○
15	RST (CAN reset)	0: Negates reset 1: Requests reset	○	○

W=△ : Only writing 1 is effective. Automatically cleared to 0 in hardware.

(1) RBO (Return bus off) bit (D4)

The receive error counter (CANREC)/transmit error counter (CANTEC) can be cleared by setting this bit to 1, thereby placing the CAN module forcibly in an error active state.
This bit is automatically cleared when an error active state is entered.

Note: After the error counters are cleared, the CAN module becomes ready to communicate when 11 consecutive recessive bits are detected on the CAN bus.

(2) TSR (Time stamp counter reset) bit (D5)

Setting this bit to 1 clears the value of the CAN Time stamp Count Register (CANTSTMP) to H'0000.
This bit is automatically cleared after the CAN Time stamp Count Register (CANTSTMP) has its value cleared to H'0000.

(3) TSP (Time stamp prescaler) bits (D6, D7)

These bits select the count clock source for the time stamp counter.

Note: Do not alter the value set with these TSP bits while the CAN module is operating (CAN Status Register CRS bit = 0).

(4) FRST (Forcible reset) bit (D11)

Setting this bit to 1 disconnects the CAN module from the CAN bus regardless of whether the CAN module is communicating or not, with its protocol control unit reset.

Note 1: For CAN communication to be performed, the FRST and RST bits must be cleared to 0.

Note 2: If the FRST bit is set to 1 during communication, outputs at the CTX0 and CTX1 pins become held high from that time on. Therefore, setting this bit to 1 while the CAN module is sending a frame may cause a CAN bus error.

Note 3: The CAN Message Slot Control Register's transmit/receive request is not cleared by setting the FRST or RST bit.

(5) BCM (BasicCAN mode) bit (D12)

The CAN module can be run in BasicCA N mode by setting this bit to 1.

- **Operation during BasicCAN mode**

  In BasicCAN mode, two local slots-slots 14 and 15 -are used as double buffers, and receive frames that are found matching to the ID by acceptance filtering are stored alternately in slots 14 and 15. Used for this acceptance filtering when slot 14 is active (next receive frame to be stored in slot 14) are the ID set for slot 14 and local mask A, and those used when slot 15 is active are the ID set for slot 15 and local mask B. Two types of frames-data frame and remote frame-can be received in this mode.
  By using the same ID and setting the same value in mask registers for the two slots, the possibility of a message-lost trouble when, for example, receiving frames which have many IDs can be reduced.

- **Procedure for entering BasicCAN mode**

  Make the following settings during initialization.

  (a) Set the IDs of slots 14 and 15 and Local Mask Registers A and B. (We recommend setting the same value)
  (b) Set the type of frame to be handled with slots 14 and 15 (standard or extended) in the CAN Extended ID Register. (We recommend setting the same type)
  (c) Set the Message Slot Control Registers for slots 14 and 15 to receive data frames.
  (d) Set the BCM bit to 1.

Note 1: Do not alter the value set with the BCM bit while the CAN module is operating (CAN Status Register CRS bit = 0).

Note 2: Slot 14 is the first slot to become active after clearing the RST bit.

Note 3: Even during BasicCAN mode, slot 0 through slot 13 can be used in the same way as when operating normally.

(6) LBM (Loopback mode) bit (D14)

When the LBM bit is set, if any receive slot exists whose ID matches that of a frame the CAN module itself transmitted, then the frame can be received.

Note 1: ACK is not returned for frames the CAN module itself transmitted.

Note 2: Do not alter the value set with the LBM bit while the CAN module is operating (CAN Status Register CRS bit = 0).

(7) RST (CAN reset) bit (D15)

When the RST bit is cleared to 0, the CAN module is connected to the CAN bus and becomes possible to communicate after detecting 11 consecutive recessive bits. Also, the CAN Time stamp Count Register thereby starts counting.
When this bit is set to 1, the protocol control unit is reset and disconnected from the CAN bus after sending frames from slots which have had transmit requests set. All frames received during this time are processed normally.

Note 1: No new transmit requests can be set from when the CAN Status Register CRS bit is set to 1 after setting the RST bit to 1 till when the protocol control unit is reset.

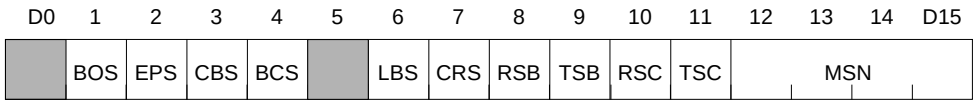
Note 2: When the protocol control unit is reset by setting the RST bit to 1, the CAN Time stamp Count Register and CAN Transmit/Receive Error Count Register are initialized to 0.

Note 3: For CAN communication to be performed, 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>**



<When reset: H'0100>

D	Bit Name	Function	R	W
0	No functions assigned		0	–
1	BOS (Bus-off status)	0: Not bus off 1: Bus-off state	○	–
2	EPS (Error passive status)	0: Not error passive 1: Error passive state	○	–
3	CBS (CAN bus error)	0: No error occurred 1: Error occurred	○	–
4	BCS (BasicCAN status)	0: Normal mode 1: BasicCAN mode	○	–
5	No functions assigned		0	–
6	LBS (Loopback status)	0: Normal mode 1: Loopback mode	○	–
7	CRS (CAN reset status)	0: Operating 1: Reset state	○	–
8	RSB (Receive status)	0: Nor receiving 1: Receiving	○	–
9	TSB (Transmit status)	0: Not transmitting 1: Transmitting	○	–
10	RSC (Receive complete status)	0: Reception not completed 1: Reception completed	○	–
11	TSC (Transmit complete status)	0: Transmission not completed 1: Transmission completed	○	–

D	Bit Name	Function	R	W
12-15 (Message slot number)	MSN	Message slot number which has had transmission/reception completed 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	○	–

(1) BOS (Bus-off status) bit (D1)

When the 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 value of the Transmit Error Count Register exceeds 255, with the CAN module in a bus-off state.

[Clear condition]

This bit is cleared when the CAN module returns from the bus-off state.

(2) EPS (Error passive status) bit (D2)

When the EPS bit = 1, it means that the CAN module is in a error-passive state.

[Set condition]

This bit is set to 1 when the value of the Transmit or Receive Error Count Register exceeds 127, with the CAN module in an error-passive state.

[Clear condition]

This bit is cleared when the CAN module returns from the error-passive state.

(3) CBS (CAN bus error) bit (D3)

[Set condition]

This bit is set to 1 when an error on the CAN bus is detected.

[Clear condition]

This bit is cleared when the CAN module finished transmitting or receiving normally.

(4) BCS (BasicCAN status) bit (D4)

When the 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.

Requirements for operation in BasicCAN mode

- The CAN Control Register BCM bit must be set to 1.
- Slots 14 and 15 must both be set for data frame reception.

[Clear condition]

This bit is cleared by clearing the BCM bit to 0.

(5) LBS (Loopback status) bit (D6)

When the 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 (D7)

When the CRS bit = 1, it means that the CAN module's protocol control unit is in a reset state.

[Set condition]

This bit is set to 1 when CAN module's protocol control unit is in a reset state.

[Clear condition]

This bit is cleared by clearing the CAN Control Register RST (CAN reset) bit to 0.

(7) RSB (Receive status) bit (D8)

[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 goes to a bus-idle state.

(8) TSB (Transmit status) bit (D9)

[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 goes to a bus-idle state.

(9) RSC (Receive complete status) bit (D10)

[Set condition]

This bit is set to 1 when the CAN module finished receiving data normally (regardless of whether there is any receive slot that satisfies receive conditions).

[Clear condition]

This bit is cleared when the CAN module finished transmitting data normally.

(10) TSC (Transmit complete status) bit (D11)

[Set condition]

This bit is set to 1 when the CAN module finished transmitting data normally.

[Clear condition]

This bit is cleared when the CAN module finished receiving data normally.

(11) MSN (Message slot number) bits (D12-D15)

When the CAN module finished transmitting data or finished storing the received data in a slot, these bits show the relevant slot number. The MSN bits cannot be cleared to 0 in software.

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>**

D0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	D15
IDE0	IDE1	IDE2	IDE3	IDE4	IDE5	IDE6	IDE7	IDE8	IDE9	IDE10	IDE11	IDE12	IDE13	IDE14	IDE15

<When reset: H'0000>

D	Bit Name	Function	R	W
0	IDE0 (Extended ID 0)	0: Standard ID format	○	○
1	IDE1 (Extended ID 1)	1: Extended ID format		
2	IDE2 (Extended ID 2)			
3	IDE3 (Extended ID 3)			
4	IDE4 (Extended ID 4)			
5	IDE5 (Extended ID 5)			
6	IDE6 (Extended ID 6)			
7	IDE7 (Extended ID 7)			
8	IDE8 (Extended ID 8)			
9	IDE9 (Extended ID 9)			
10	IDE10 (Extended ID 10)			
11	IDE11 (Extended ID 11)			
12	IDE12 (Extended ID 12)			
13	IDE13 (Extended ID 13)			
14	IDE14 (Extended ID 14)			
15	IDE15 (Extended ID 15)			

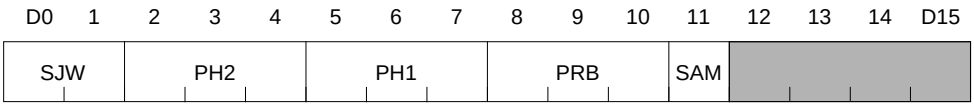
Each bit in this register selects the type of frame handled by the corresponding message slot.
Setting this bit to 0 selects the Standard ID format.
Setting this bit to 1 selects the Extended ID format.

Note: Settings of each bit of this register can only be changed when the corresponding slot dose 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>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-1	SJW	Sets reSynchronization Jump Width		
(reSynchronization Jump Width)		00 : SJW = 1Tq	○	○
		01 : SJW = 2Tq		
		10 : SJW = 3Tq		
		11 : SJW = 4Tq		
2-4	PH2	Sets Phase Segment2		
	(Phase Segment2)	000 : Settings inhibited	○	○
		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		

Note 1: Do not modify the CAN Configuration Register (CAN0CONF or CAN1CONF) while the CAN module is operating (CAN Status Register CRS bit = 0).

Note 2: The bit configuration in this register must be set to meet the conditions below.

- Number of Tq's in one bit: 8 to 25 Tq's
- SJW ≦ min(Phase Segment1,Phase Segment2)
- Phase Segment2=max(Phase Segment1,IPT) However, IPT = 2 for the M32R/E's internal CAN modules.

Note that min ( ) is the function that returns a smaller value, whereas max ( ) is the function that returns the maximum value.

<When reset: H'0000>

D	Bit Name	Function	R	W
5-7	PH1	Sets Phase Segment1		
	(Phase Segment1)	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		
8-10	PRB	Sets Propagation Segment		
	(Propagation Segment)	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		
11	SAM	0: Samples once	○	○
	(Sampling times)	1: Samples three times		
12-15	No functions assigned		0	–

(1) SJW bits (D0-D1)

These bits set the width of "reSynchronization Jump Width."

(2) PH2 bits (D2-D4)

These bits set the width of "Phase Segment2."

Note: For the M32R/E's internal CAN modules, IPT (Information Processing Time) = 2. Because the PH2 bits = 0 after reset, set the width to 2 Tq's or more before using the CAN module.

(3) PH1 bits (D5-D7)

These bits set the width of "Phase Segment1."

(4) PRB bits (D8-D10)

These bits set the width of "Propagation Segment."

(5) SAM bit (D11)

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 values sampled at three points-one sampled at the end of Phase Segment1, one sampled before 1Tq, and one sampled before 2 Tq.

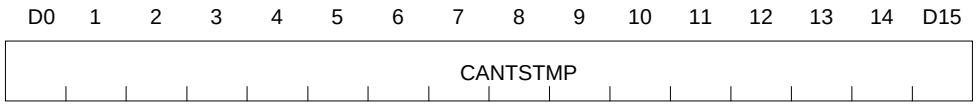
Table 13.2.1 Example of Bit Timing Settings when CPU Clock = 40 MHz

Baud rate	BRP set value	Tq period (ns)	Number of Tq's in one bit	PROP+PH1	PH2	Sampling points
1M bps	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%
	4	125	16	12	3	81%
	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%

13.2.5 CAN Time stamp Count Registers

■ **CAN0 Time stamp Count Register (CAN0TSTMP)** **<Address: H'0080 1008>**

■ **CAN1 Time stamp Count Register (CAN1TSTMP)** **<Address: H'0080 1408>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-15	CANiTSTMP	16-bit counter value	○	–

The CAN modules each contain a 16-bit counter. The count period is selected from divide-by-1, 2, 3, or 4 of the CAN bus bit period using the CAN Control Register (CANCNT)'s TSP (timestamp prescaler) bits.
The count register value is captured at completion of transmission/reception and the captured value is stored in a message slot.
The counter is made to start counting by clearing the CAN Control Register (CANCNT)'s RST bit to 0.

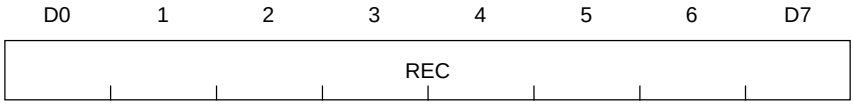
Note 1: The protocol control unit is reset and the counter is initialized to H'0000, by setting the CAN Control Register (CAN0CNT)'s RST (CAN reset) bit to 1. Also, the count register can be initialized to H'0000 while the CAN module is operating by setting the TSR (time stamp counter reset) bit to 1.

Note 2: During loopback mode, if any ID-matching slot exists, the CAN module stores the time stamp value in the corresponding slot when it finished receiving. (No time stamp value is stored this way when the CAN module finished transmitting.)

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>**



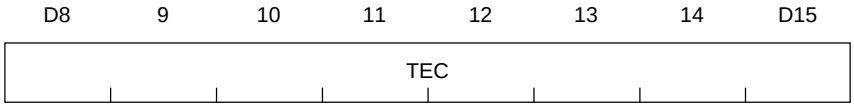
<When reset: H'00>

D	Bit Name	Function	R	W
0-7	REC (Receive error counter)	Receive error count value	○	–

When in an error-active/error-passive state, the receive error count value is stored in this register. The counter counts down during normal reception and counts up when an error occurs.
When reception is completed normally while REC ≧ 128 (error passive), REC is set to 127.
When in a bus-off state, an indeterminate value is stored in this register. The register value 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>**



<When reset: H'00>

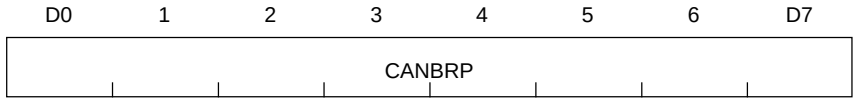
D	Bit Name	Function	R	W
8-15	TEC (Transmit error counter)	Transmit error count value	○	–

When in an error-active/error-passive state, the transmit error count value is stored in this register. The counter counts down during normal transmission and counts up when an error occurs.
When in a bus-off state, an indeterminate value is stored in this register. The register value 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>**



<When reset: H'01>

D	Bit Name	Function	R	W
0-7	BRP	Selects a baud rate prescaler value	○	○

This register is used to set the CAN module Tq period. The CAN baud rate is determined by (Tq period x number of Tq's in one bit).

Tq period = (CANBRP + 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

Note 1: Setting H'00 (divide by 1) is inhibited.

Note 2: Do not modify the CAN Baud Rate Prescaler (CAN0BRP or CAN1BRP) while the CAN module is operating (CAN Status Register CRS bit = 0).

13.2.8 CAN Interrupt Related Registers

■ **CAN0 Slot Interrupt Status Register (CAN0SLIST)** **<Address: H'0080 100C>**

■ **CAN1 Slot Interrupt Status Register (CAN1SLIST)** **<Address: H'0080 140C>**

D0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	D15
SSB0	SSB1	SSB2	SSb3	SSB4	SSB5	SSB6	SSB7	SSB8	SSB9	SSB10	SSB11	SSB12	SSB13	SSB14	SSB15

<When reset: H'0000>

D	Bit Name	Function	R	W
0	SSB0 (Slot 0 interrupt request status)	0: Interrupt not requested	○	△
1	SSB1 (Slot 1 interrupt request status)	1: Interrupt requested		
2	SSB2 (Slot 2 interrupt request status)			
3	SSB3 (Slot 3 interrupt request status)			
4	SSB4 (Slot 4 interrupt request status)			
5	SSB5 (Slot 5 interrupt request status)			
6	SSB6 (Slot 6 interrupt request status)			
7	SSB7 (Slot 7 interrupt request status)			
8	SSB8 (Slot 8 interrupt request status)			
9	SSB9 (Slot 9 interrupt request status)			
10	SSB10 (Slot 10 interrupt request status)			
11	SSB11 (Slot 11 interrupt request status)			
12	SSB12 (Slot 12 interrupt request status)			
13	SSB13 (Slot 13 interrupt request status)			
14	SSB14 (Slot 14 interrupt request status)			
15	SSB15 (Slot 15 interrupt request status)			

W=△ : Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

When using CAN interrupts, it is possible to know which slot has requested an interrupt by inspecting this register.

- **Slots set for transmission**

  The status bit is set to 1 when transmission is completed. This bit is cleared by writing 0 in software.

- **Slots set for reception**

  The status 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 Status Register, be sure to write 0 for the bits to be cleared and 1 for all other bits. Writing 1 in software does not affect any bit of this register, the bit retains the value it had before writing.

Note 1: For remote frame receive slots with the automatic answering function enabled, the status is set after receiving a remote frame and again after sending a data frame.

Note 2: For remote frame transmit slots, the status is set after having transmitted a remote frame and again after having received a data frame.

Note 3: If the status bit is set upon occurrence of an interrupt request at the same time it is cleared in software, the former has priority so that the status bit is set.

■ **CAN0 Slot Interrupt Mask Register (CAN0SLIMK)** **<Address: H'0080 1010>**

■ **CAN1 Slot Interrupt Mask Register (CAN1SLIMK)** **<Address: H'0080 1410>**

D0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	D15
IRB0	IRB1	IRB2	IRB3	IRB4	IRB5	IRB6	IRB7	IRB8	IRB9	IRB10	IRB11	IRB12	IRB13	IRB14	IRB15

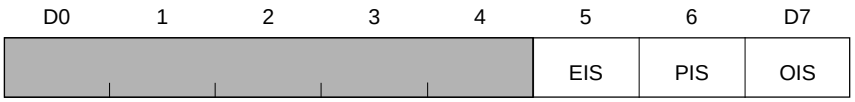
<When reset: H'0000>

D	Bit Name	Function	R	W
0	IRB0 (Slot 0 interrupt request mask)	0: Masks (disables) interrupt request	○	○
1	IRB1 (Slot 1 interrupt request mask)	1: Enables interrupt request		
2	IRB2 (Slot 2 interrupt request mask)			
3	IRB3 (Slot 3 interrupt request mask)			
4	IRB4 (Slot 4 interrupt request mask)			
5	IRB5 (Slot 5 interrupt request mask)			
6	IRB6 (Slot 6 interrupt request mask)			
7	IRB7 (Slot 7 interrupt request mask)			
8	IRB8 (Slot 8 interrupt request mask)			
9	IRB9 (Slot 9 interrupt request mask)			
10	IRB10 (Slot 10 interrupt request mask)			
11	IRB11 (Slot 11 interrupt request mask)			
12	IRB12 (Slot 12 interrupt request mask)			
13	IRB13 (Slot 13 interrupt request mask)			
14	IRB14 (Slot 14 interrupt request mask)			
15	IRB15 (Slot 15 interrupt request mask)			

This register is used to enable or disable interrupt requests generated upon completion of data transmission/reception by each slot. Setting the IRBn (n=0-15) bit for any slot to 1 enables the interrupt requests generated upon completion of data transmission/reception by that slot.
Check the CAN Slot Interrupt Status Register (CANSLIST) to see which slot has generated the interrupt request.

■ **CAN0 Error Interrupt Status Register (CAN0ERIST)** **<Address: H'0080 1014>**

■ **CAN1 Error Interrupt Status Register (CAN1ERIST)** **<Address: H'0080 1414>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-4	No functions assigned		0	–
5	EIS (CAN bus error interrupt status)	0: Interrupt not requested 1: Interrupt requested	○	△
6	PIS (Error passive interrupt status)			
7	OIS (Bus off interrupt status)			

W=△ : Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

When using CAN interrupts, checking this register will help to find the cause of interrupt if the interrupt has been generated due to an error.

(1) EIS (CAN bus error interrupt status) bit (D5)

This bit is set to 1 when a communication error is detected. This bit can be cleared by writing 0 in software.

(2) PIS (Error passive interrupt status) bit (D6)

This bit is set to 1 when the CAN module goes to an error-passive state. This bit can be cleared by writing 0 in software.

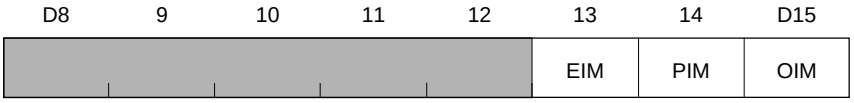
(3) OIS (Bus off interrupt status) bit (D7)

This bit is set to 1 when the CAN module goes to a bus-off state. This bit can be cleared by writing 0 in software.

When writing to the CAN Error Interrupt Status Register, be sure to write 0 for the bits to be cleared and 1 for all other bits. Writing 1 in software does not affect any bit of this register, the bit retains the value it had before writing.

■ **CAN0 Error Interrupt Mask Register (CAN0ERIMK)** **<Address: H'0080 1015>**

■ **CAN1 Error Interrupt Mask Register (CAN1ERIMK)** **<Address: H'0080 1415>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-12	No functions assigned		0	–
13	EIM (CAN bus error interrupt mask)	0: Masks (disables) interrupt request 1: Enables interrupt request	○	○
14	PIMm (Error passive interrupt mask)			
15	OIM (Bus off interrupt mask)			

(1) EIM (CAN bus error interrupt mask) bit (D5)

This bit enables or disables the interrupt request generated by occurrence of a CAN bus error. Setting this bit to 1 enables CAN bus error interrupt request.

(2) PIM (Error passive interrupt mask) bit (D6)

This bit enables or disables the interrupt request generated by transition of the CAN module status to an error-passive state. Setting this bit to 1 enables an error-passive interrupt request.

(3) OIM (Bus off interrupt mask) bit (D7)

This bit enables or disables the interrupt request generated by transition of the CAN module status to a bus-off state. Setting this bit to 1 enables a bus-off interrupt request.

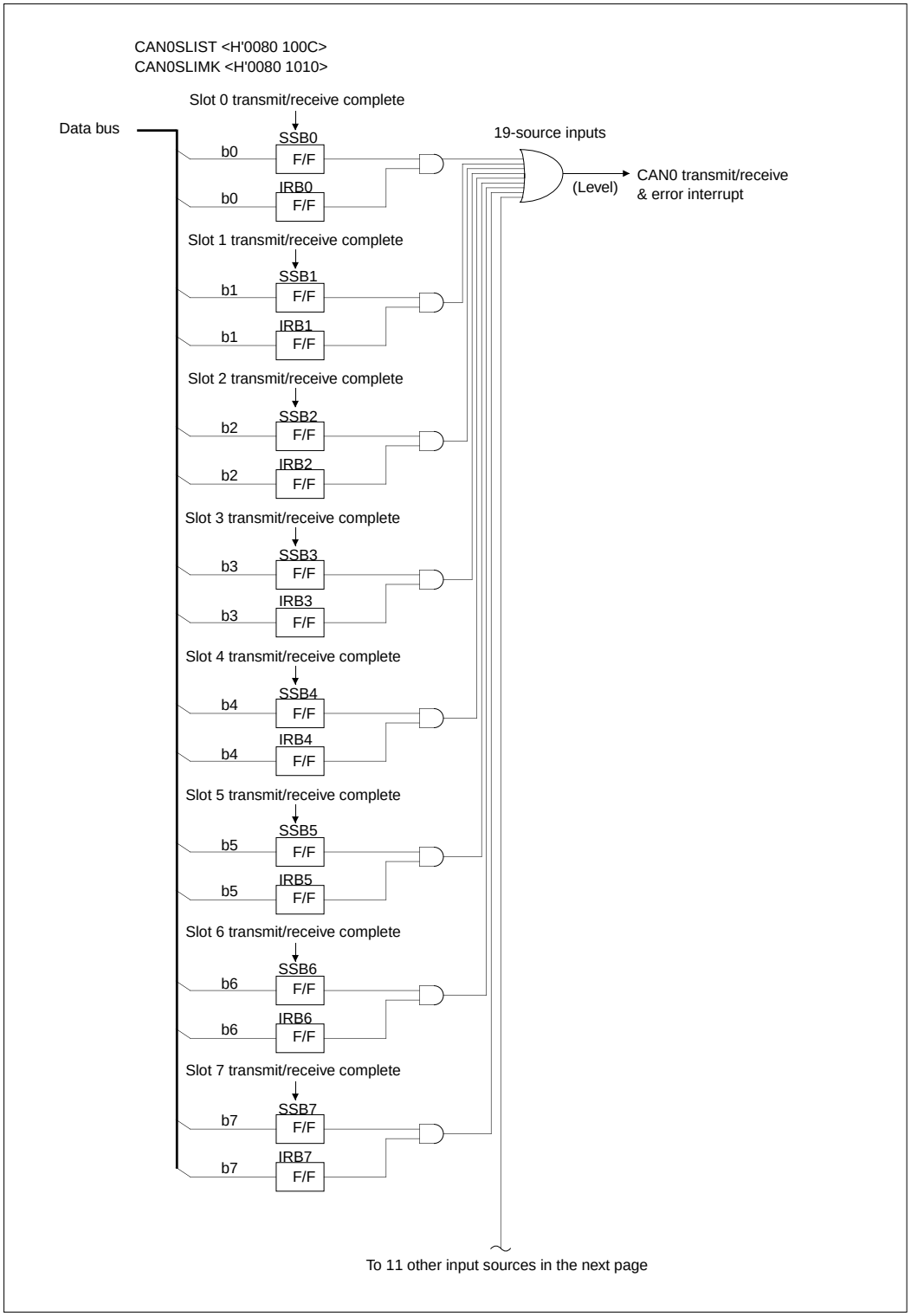


Figure 13.2.9 Block Diagram of CAN0 Transmit/Receive & Error Interrupts (1/3)

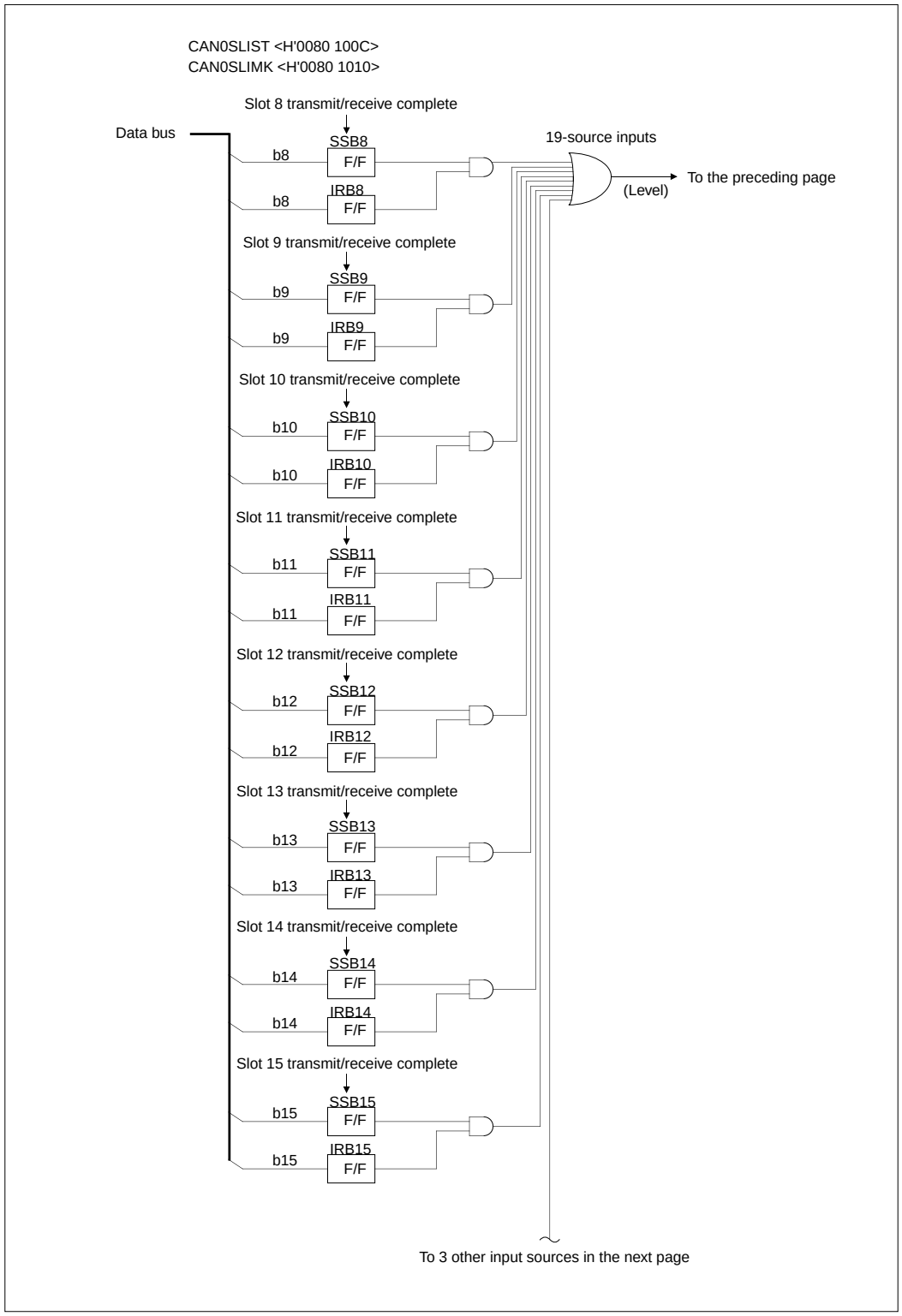


Figure 13.2.10 Block Diagram of CAN0 Transmit/Receive & Error Interrupts (2/3)

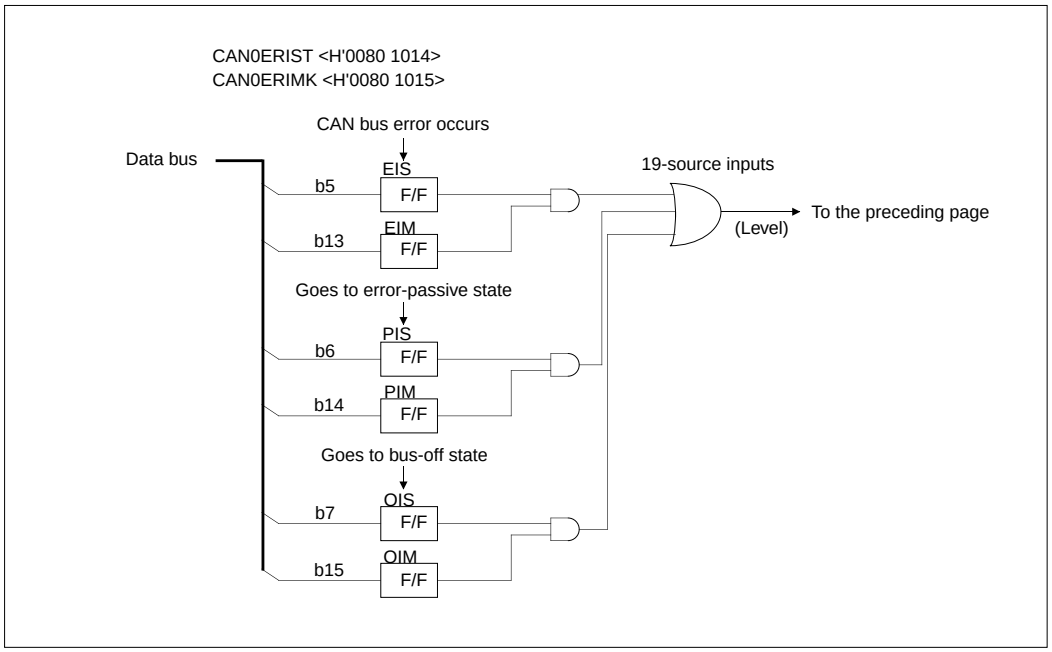


Figure 13.2.11 Block Diagram of CAN0 Transmit/Receive & Error Interrupts (3/3)

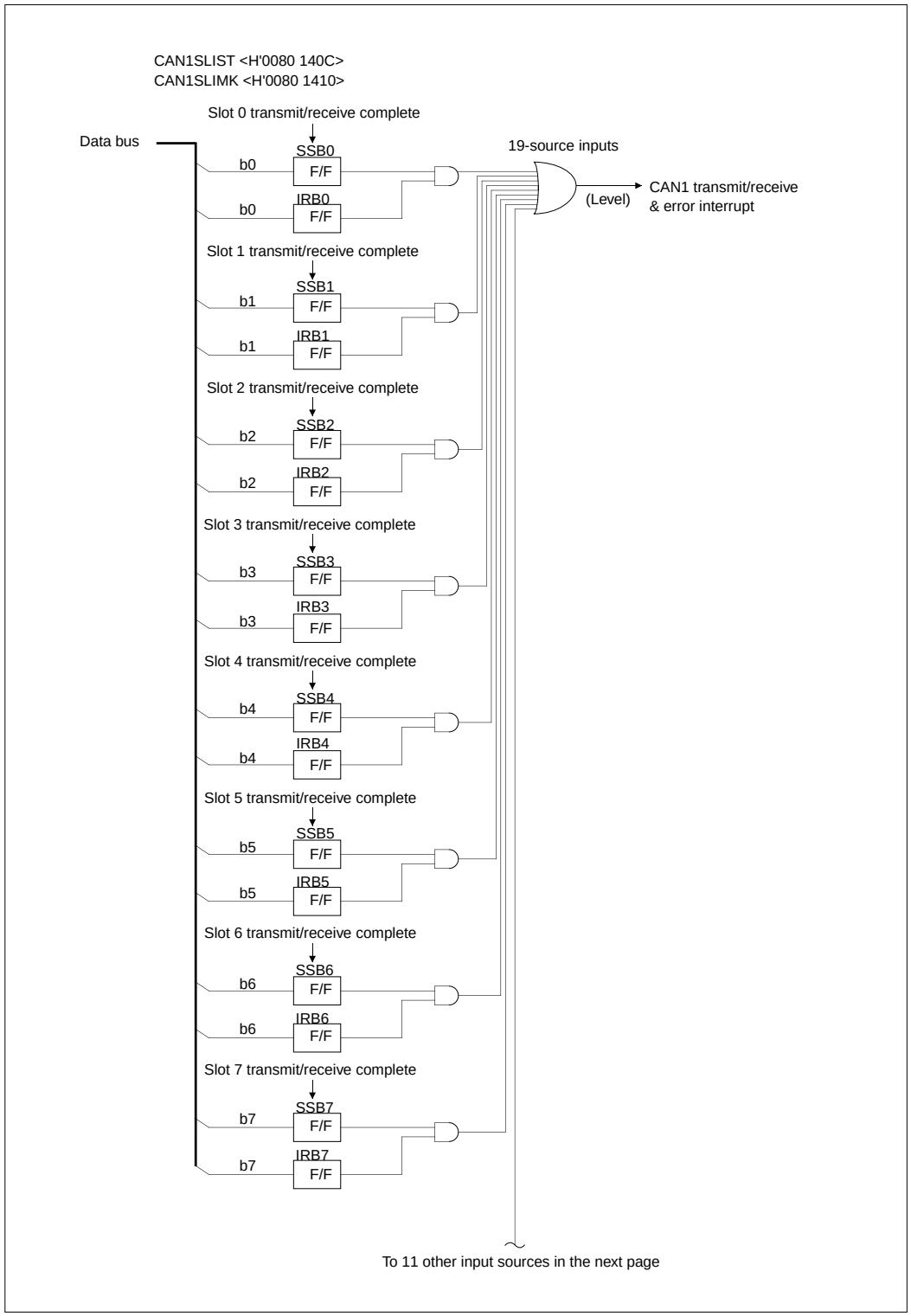


Figure 13.2.12 Block Diagram of CAN1 Transmit/Receive & Error Interrupts (1/3)

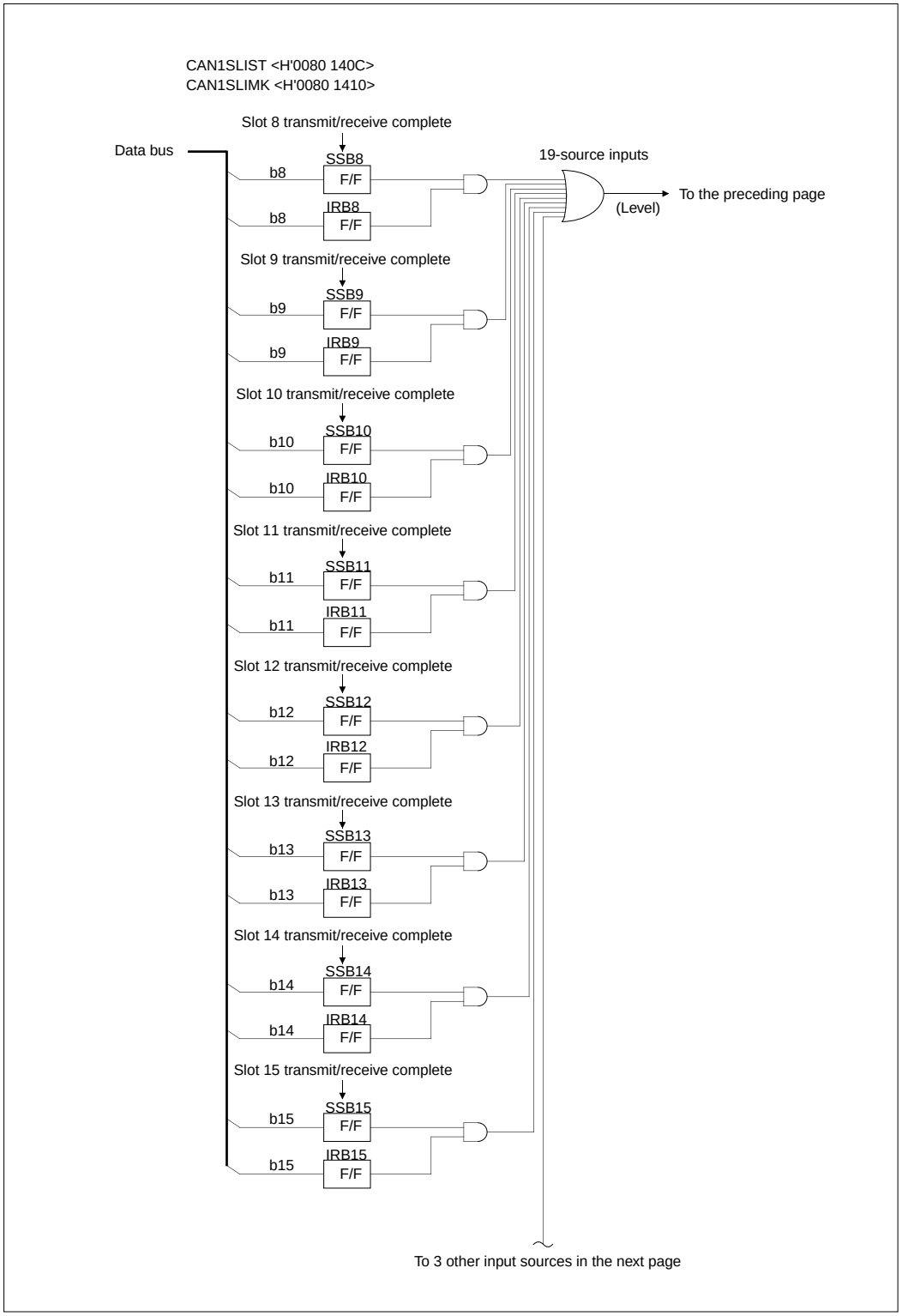


Figure 13.2.13 Block Diagram of CAN1 Transmit/Receive & Error Interrupts (2/3)

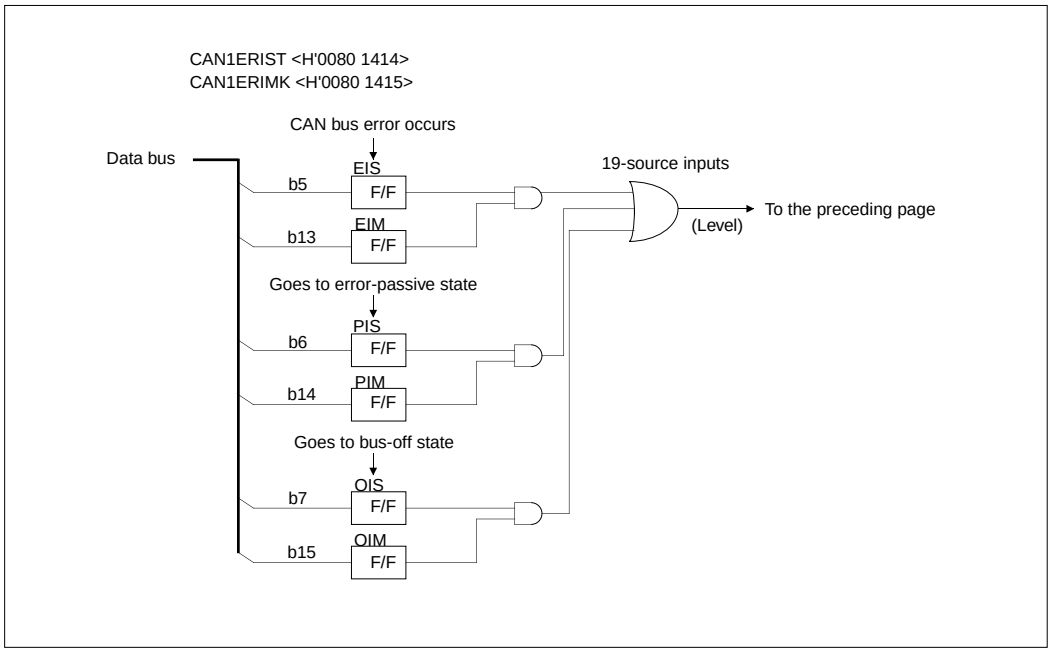


Figure 13.2.14 Block Diagram of CAN1 Transmit/Receive & Error Interrupts (3/3)

13.2.9 CAN Mask Registers

■ **CAN0 Global Mask Register Standard ID0 (C0GMSKS0)** **<Address: H'0080 1028>**

■ **CAN0 Local 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 Register A Standard ID0 (C1LMSKAS0)** **<Address: H'0080 1430>**

■ **CAN1 Local Mask Register B Standard ID0 (C1LMSKBS0)** **<Address: H'0080 1438>**

D0	1	2	3	4	5	6	D7
			SID0M	SID1M	SID2M	SID3M	SID4M

<When reset: H'00>

D	Bit Name	Function	R	W
0-2	No functions assigned		0	–
3-7	SID0M-SID4M (Standard ID0 to standard ID4)	0: ID not checked 1: ID checked	○	○

■ **CAN0 Global Mask Register Standard ID1 (C0GMSKS1)** **<Address: H'0080 1029>**

■ **CAN0 Local 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 Register A Standard ID1 (C1LMSKAS1)** **<Address: H'0080 1431>**

■ **CAN1 Local Mask Register B Standard ID1 (C1LMSKBS1)** **<Address: H'0080 1439>**

D8	9	10	11	12	13	14	D15
		SID5M	SID6M	SID7M	SID8M	SID9M	SID10M

<When reset: H'00>

D	Bit Name	Function	R	W
8-9	No functions assigned		0	–
10-15	SID5M-SID10M (Standard ID5 to standard ID10)	0: ID not checked 1: ID checked	○	○

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 the Local Mask Registers A and B respectively are used for slots 14 and 15.

- When any bit in this register is set to 0, the corresponding ID bit is masked during acceptance filtering (so that the ID is assumed to have matched).
- When any bit in this register is set to 1, the corresponding ID bit is compared with the received ID during acceptance filtering. If it matches the ID set for the message slot, the received data is stored in that slot.

Note 1: SID0M corresponds to the MSB of the standard ID.

Note 2: The Global Mask Register can only be modified when none of the slots 0-13 has receive requests set.

Note 3: The Local Mask Register A can only be modified when slot 14 does not have receive requests set.

Note 4: The Local Mask Register B can only be modified when slot 15 does not have receive requests set.

■ **CAN0 Global Mask Register Extended ID0 (C0GMSKE0)** **<Address: H'0080 102A>**

■ **CAN0 Local 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 Register A Extended ID0 (C1LMSKAE0)** **<Address: H'0080 1432>**

■ **CAN1 Local Mask Register B Extended ID0 (C1LMSKBE0)** **<Address: H'0080 143A>**

D0	1	2	3	4	5	6	D7
				EID0M	EID1M	EID2M	EID3M

<When reset: H'00>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	EID0M-EID3M (Extended ID0 to extended ID3)	0: ID not checked 1: ID checked	○	○

■ **CAN0 Global Mask Register Extended ID1 (C0GMSKE1)** **<Address: H'0080 102B>**

■ **CAN0 Local 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 Register A Extended ID1 (C1LMSKAE1)** **<Address: H'0080 1433>**

■ **CAN1 Local Mask Register B Extended ID1 (C1LMSKBE1)** **<Address: H'0080 143B>**

D8	9	10	11	12	13	14	D15
EID4M	EID5M	EID6M	EID7M	EID8M	EID9M	EID10M	EID11M

<When reset: H'00>

D	Bit Name	Function	R	W
8-15	EID4M-EID11M (Standard ID4 to standard ID11)	0: ID not checked 1: ID checked	○	○

■ **CAN0 Global Mask Register Extended ID2 (C0GMSKE2)** **<Address: H'0080 102C>**

■ **CAN0 Local 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 Register A Extended ID2 (C1LMSKAE2)** **<Address: H'0080 1434>**

■ **CAN1 Local Mask Register B Extended ID2 (C1LMSKBE2)** **<Address: H'0080 143C>**

D0	1	2	3	4	5	6	D7
		EID12M	EID13M	EID14M	EID15M	EID16M	EID17M

<When reset: H'00>

D	Bit Name	Function	R	W
0,1	No functions assigned		0	–
2-7	EID12M-EID17M (Extended ID12 to extended ID17)	0: ID not checked 1: ID checked	○	○

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 the Local Mask Registers A and B respectively are used for slots 14 and 15.

- When any bit in this register is set to 0, the corresponding ID bit is masked during acceptance filtering (so that the ID is assumed to have matched).
- When any bit in this register is set to 1, the corresponding ID bit is compared with the received ID during acceptance filtering. If it matches the ID set for the message slot, the received data is stored in that slot.

Note 1: EID0M corresponds to the MSB of the extended ID.

Note 2: The Global Mask Register can only be modified when none of the slots 0-13 has receive requests set.

Note 3: The Local Mask Register A can only be modified when slot 14 does not have receive requests set.

Note 4: The Local Mask Register B can only be modified when slot 15 does not have receive requests set.

13.2.10 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 Slot 12 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>

D0 (D8)	1	2	3	4	5	6	D7 (D15)
TR	RR	RM	RL	RA	ML	TRSTAT	TRFIN

<When reset: H'00>

D	Bit Name	Function	R	W
0	TR (Transmit request)	0: Does not use the message slot as a transmit slot 1: Uses the message slot as a transmit slot	○	○
1	RR (Receive request)	0: Does not use the message slot as a receive slot 1: Uses the message slot as a receive slot	○	○
2	RM (Remote)	0: Transmit/receives data frame 1: Transmit/receives remote frame	○	○
3	RL (Automatic answering disable)	0: Enables automatic answering for remote frame 1: Disables automatic answering for remote frame	○	○
4	RA (Remote active)	BasicCAN mode 0: Data frame receive (status) 1: Remote frame receive (status) Normal mode 0: Data frame 1: Remote frame	○	–
5	ML (Message lost)	0: No messages lost 1: Messages lost	○	△
6	TRSTAT (Transmit/receive status)	For transmit slots 0: Not transmitting (idle) 1: Transmit request accepted For receive slots 0: Not receiving (idle) 1: Storing the received data	○	–
7	TRFIN (Transmit/receive finished)	For transmit slots 0: Not transmitted yet 1: Finished transmitting For receive slots 0: Not received yet 1: Finished receiving	○	△

W=△ : Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

Note 1: Do not write to this register while the CAN module is in a reset state (CANCNT register FRST or RST bit is set). If a transmit/receive request is written to the register while in this state, the CAN module may perform transmission or reception at unexpected timing.

Note 2: If there is a data/remote frame transmit request for multiple slots, the slot with the smallest slot number has priority and a frame is transmitted from that slot. If there is a data/remote frame receive request for multiple slots, the slot with the smallest slot number among those that satisfy receive conditions has priority and a frame is received in that slot.

(1) TR (Transmit request) bit (D0)

Set this bit to 1 when using the message slot as a transmit slot.
Set this bit to 0 when using the message slot as a data frame or remote frame receive slot.

(2) RR (Receive request) bit (D1)

Set this bit to 1 when using the message slot as a receive slot.
Set this bit to 0 when using the message slot as a data frame or remote frame transmit slot.
If both the TR (Transmit request) and RR (Receive request) bits are set to 1, device operation is instable.

(3) RM (Remote) bit (D2)

Set this bit to 1 when using the message slot to handle remote frames.
The message slot may be set to handle remote frames in following two ways.

- **Set for remote frame transmission**

  The data stored in the message slot is transmitted as a remote frame. When transmission is finished, the slot automatically changes to a data frame receive slot.
  However, if a data frame is received before the CAN module finished sending the remote frame, the received data is stored in the message slot and the remote frame is not transmitted.

- **Set for remote frame reception**

  The message slot receives a remote frame. The processing performed after reception depends on how the RL (Automatic answering disable) bit is set.

(4) RL (Automatic answering disable) bit (D3)

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.
Setting this bit to 0 enables automatic answering so that after receiving a remote frame, the message slot automatically changes to a transmit slot and the data stored in it is transmitted as a data frame.
When this bit is set to 1, the message slot becomes inactive after receiving a remote frame.

Note: This bit must always be set to 0 unless the message slot is set for remote frame reception.

(5) RA (Remote active) bit (D4)

This bit functions differently between slots 0-13 and slots 14 and 15.

- **Slots 0-13**

  For slots which have been set for remote frame transmission (reception), this bit is set to 1. Then when remote frame transmission (or reception) is completed, the bit is cleared to 0.

- **Slots 14 and 15**

  For these slots, the bit functions differently depending on how the CAN Control Register BCM (BasicCAN mode) bit is set.

  When BCM = 0 (normal mode) : If the slot has been set to transmit (or receive) a remote frame, this bit is set to 1.

  When BCM = 1 (BasicCAN mode) : This bit shows which type of frame has been received. When in BasicCAN mode, either slot 14 or 15 stores the received data regardless of whether it is a data frame or a remote frame.

  When RA = 0, it means that the frame stored in the slot is a data frame;

  when RA = 1, it means that the frame stored in the slot is a remote frame.

(6) ML (Message lost) bit (D5)

This bit is effective for receive slots. It is set to 1 when the received data stored in the message slot is overwritten by receive operation before being read out.
This bit is cleared by writing 0 in software.

(7) TRSTAT (Transmit/receive status) bit (D6)

This bit indicates that transmission or reception is in progress and the CAN module is accessing the message slot. This bit is set to 1 when the message slot is being accessed or set to 0 when not accessed.

- **For transmit slots**

  This bit is set to 1 when a transmit request for the message slot has been accepted, and is cleared to 0 when arbitration fails, a CAN bus error occurs, or transmission finishes.

- **For receive slots**

  This bit is set to 1 when reception is in progress and the CAN module is storing the received data in the message slot. Note that the value read out from the message slot while this bit = 1 is indeterminate.

(8) TRFIN (Transmit/receive finished) bit (D7)

This bit indicates that the CAN module has finished transmitting or receiving data.

- **For transmit slots**

  This bit is set to 1 when the CAN module finishes sending data from the message slot. This bit is cleared by writing 0 in software. However, this bit cannot be cleared while the TRSTAT (transmit/receive status) bit remains 1.

- **For receive slots**

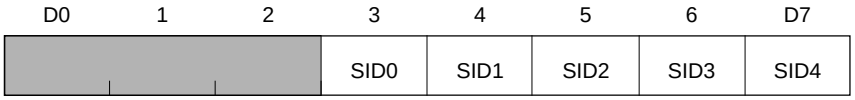
  This bit is set to 1 when the CAN module finished receiving the data normally that is to be stored in the message slot. This bit is cleared by writing 0 in software. However, this bit cannot be cleared while the TRSTAT (transmit/receive status) bit remains 1.

Note: Before reading the received data from the message slot, be sure to clear the TRFIN (transmit/receive finished) bit. If the TRFIN (transmit/receive finished) bit remains set after readout, it means that new received data was stored in the message slot while being read out, and that the read data contains an indeterminate value. In such a case, discard the read data and clear the TRFIN (transmit/receive finished) bit before performing read operation again.

13.2.11 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>



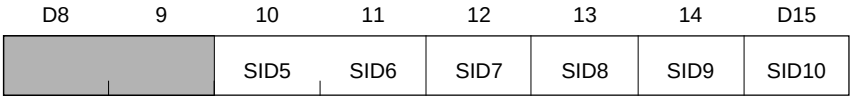
<When reset: indeterminate>

D	Bit Name	Function	R	W
0-2	No functions assigned		0	–
3-7	SID0-SID4 (Standard ID0 to standard ID4)	Standard ID0 to standard ID4	○	○

These registers comprise a transmit frame/receive frame memory space.

■ 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>



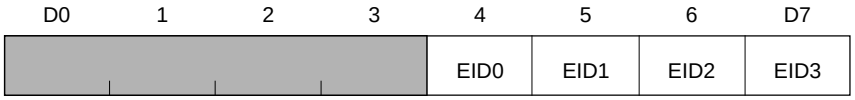
<When reset: indeterminate>

D	Bit Name	Function	R	W
8,9	No functions assigned		0	–
10-15	SID5-SID10 (Standard ID5 to standard ID10)	Standard ID5 to standard ID10	○	○

These registers comprise a transmit frame/receive frame memory space.

■ 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-7	EID0-EID3 (Extended ID0 to extended ID3)	Extended ID0 to extended ID3	○	○

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots whose frame type has been chosen to be the standard ID format, the values written to their EID bits when storing the received data are indeterminate.

■ 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>

D8	9	10	11	12	13	14	D15
EID4	EID5	EID6	EID7	EID8	EID9	EID10	EID11

<When reset: indeterminate>

D	Bit Name	Function	R	W
8-15	EID4-EID11 (Extended ID4 to extended ID11)	Extended ID4 to extended ID11	○	○

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots whose frame type has been chosen to be the standard ID format, the values written to their EID bits when storing the received data are indeterminate.

■ 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>

D0	1	2	3	4	5	6	D7
		EID12	EID13	EID14	EID15	EID16	EID17

<When reset: indeterminate>

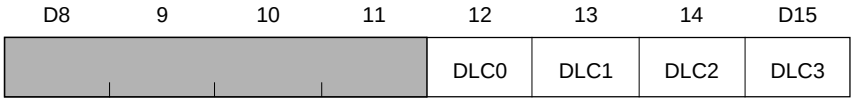
D	Bit Name	Function	R	W
0,1	No functions assigned		0	–
2-7	EID12-EID17 (Extended ID12 to extended ID17)	Extended ID12 to extended ID17	○	○

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots whose frame type has been chosen to be the standard ID format, the values written to their EID bits when storing the received data are indeterminate.

■ 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>



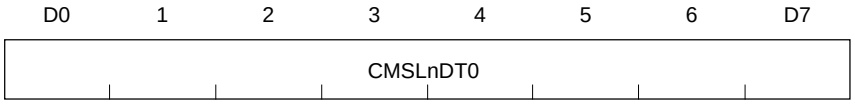
<When reset: indeterminate>

D	Bit Name	Function	R	W
8-11	No functions assigned		0	–
12-15	DLC0-DLC3 (Set data length)	0 0 0 0 : 0 byte 0 0 0 1 : 1 byte 0 0 1 0 : 2 byte 0 0 1 1 : 3 byte 0 1 0 0 : 4 byte 0 1 0 1 : 5 byte 0 1 1 0 : 6 byte 0 1 1 1 : 7 byte 1 × × × : 8 byte	○	○

These registers comprise a transmit frame/receive frame memory space. When transmitting, these registers are used to set the transmit data length. When receiving, the receive DLC is stored in these registers.

■ 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-7	CMSLnDT0	Message slot n data 0	○	○

n = 0-15

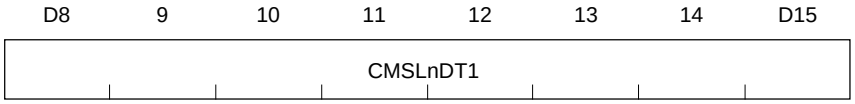
These registers comprise a transmit frame/receive frame memory space.

Note 1: For receive slots, if the data length (DLC value) = 0 when storing a data frame, an indeterminate value is written to the register.

Note 2: The first byte in the CAN frame data field corresponds to message slot n data 0. The data is transmitted and 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
8-15	CMSLnDT1	Message slot n data 1	○	○

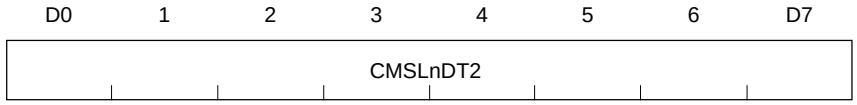
n = 0-15

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots, if the data length (DLC value) = 1 or less when storing a data frame, an indeterminate value is written to the register.

■ 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-7	CMSLnDT2	Message slot n data 2	○	○

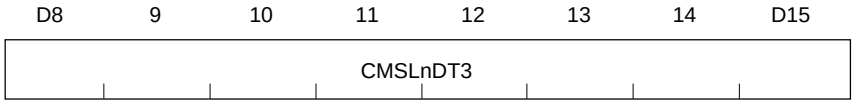
n = 0-15

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots, if the data length (DLC value) = 2 or less when storing a data frame, an indeterminate value is written to the register.

■ 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
8-15	CMSLnDT3	Message slot n data 3	○	○

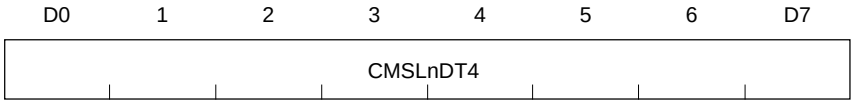
n = 0-15

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots, if the data length (DLC value) = 3 or less when storing a data frame, an indeterminate value is written to the register.

■ 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-7	CMSLnDT4	Message slot n data 4	○	○

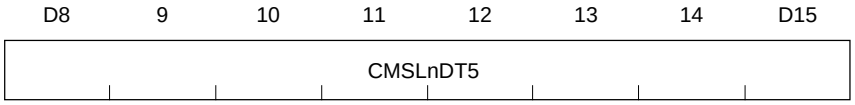
n = 0-15

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots, if the data length (DLC value) = 4 or less when storing a data frame, an indeterminate value is written to the register.

■ 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
8-15	CMSLnDT5	Message slot n data 5	○	○

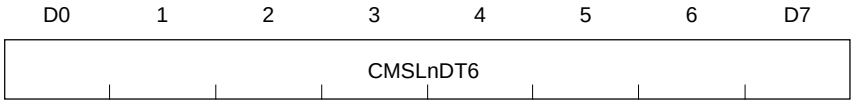
n = 0-15

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots, if the data length (DLC value) = 5 or less when storing a data frame, an indeterminate value is written to the register.

■ 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-7	CMSLnDT6	Message slot n data 6	○	○

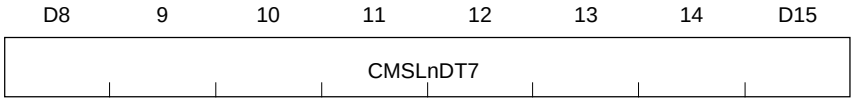
n = 0-15

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots, if the data length (DLC value) = 6 or less when storing a data frame, an indeterminate value is written to the register.

■ 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>



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-7	CMSLnDT7	Message slot n data 7	○	○

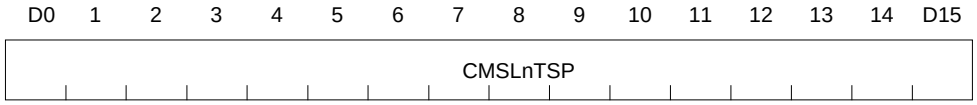
n = 0-15

These registers comprise a transmit frame/receive frame memory space.

Note: For receive slots, if the data length (DLC value) = 7 or less when storing a data frame, an indeterminate value is written to the register.

■ CAN0 Message Slot 0 Time stamp (C0MSL0TSP) <Address: H'0080 110E>
■ CAN0 Message Slot 1 Time stamp (C0MSL1TSP) <Address: H'0080 111E>
■ CAN0 Message Slot 2 Time stamp (C0MSL2TSP) <Address: H'0080 112E>
■ CAN0 Message Slot 3 Time stamp (C0MSL3TSP) <Address: H'0080 113E>
■ CAN0 Message Slot 4 Time stamp (C0MSL4TSP) <Address: H'0080 114E>
■ CAN0 Message Slot 5 Time stamp (C0MSL5TSP) <Address: H'0080 115E>
■ CAN0 Message Slot 6 Time stamp (C0MSL6TSP) <Address: H'0080 116E>
■ CAN0 Message Slot 7 Time stamp (C0MSL7TSP) <Address: H'0080 117E>
■ CAN0 Message Slot 8 Time stamp (C0MSL8TSP) <Address: H'0080 118E>
■ CAN0 Message Slot 9 Time stamp (C0MSL9TSP) <Address: H'0080 119E>
■ CAN0 Message Slot 10 Time stamp (C0MSL10TSP) <Address: H'0080 11AE>
■ CAN0 Message Slot 11 Time stamp (C0MSL11TSP) <Address: H'0080 11BE>
■ CAN0 Message Slot 12 Time stamp (C0MSL12TSP) <Address: H'0080 11CE>
■ CAN0 Message Slot 13 Time stamp (C0MSL13TSP) <Address: H'0080 11DE>
■ CAN0 Message Slot 14 Time stamp (C0MSL14TSP) <Address: H'0080 11EE>
■ CAN0 Message Slot 15 Time stamp (C0MSL15TSP) <Address: H'0080 11FE>

■ CAN1 Message Slot 0 Time stamp (C1MSL0TSP) <Address: H'0080 150E>
■ CAN1 Message Slot 1 Time stamp (C1MSL1TSP) <Address: H'0080 151E>
■ CAN1 Message Slot 2 Time stamp (C1MSL2TSP) <Address: H'0080 152E>
■ CAN1 Message Slot 3 Time stamp (C1MSL3TSP) <Address: H'0080 153E>
■ CAN1 Message Slot 4 Time stamp (C1MSL4TSP) <Address: H'0080 154E>
■ CAN1 Message Slot 5 Time stamp (C1MSL5TSP) <Address: H'0080 155E>
■ CAN1 Message Slot 6 Time stamp (C1MSL6TSP) <Address: H'0080 156E>
■ CAN1 Message Slot 7 Time stamp (C1MSL7TSP) <Address: H'0080 157E>
■ CAN1 Message Slot 8 Time stamp (C1MSL8TSP) <Address: H'0080 158E>
■ CAN1 Message Slot 9 Time stamp (C1MSL9TSP) <Address: H'0080 159E>
■ CAN1 Message Slot 10 Time stamp (C1MSL10TSP) <Address: H'0080 15AE>
■ CAN1 Message Slot 11 Time stamp (C1MSL11TSP) <Address: H'0080 15BE>
■ CAN1 Message Slot 12 Time stamp (C1MSL12TSP) <Address: H'0080 15CE>
■ CAN1 Message Slot 13 Time stamp (C1MSL13TSP) <Address: H'0080 15DE>
■ CAN1 Message Slot 14 Time stamp (C1MSL14TSP) <Address: H'0080 15EE>
■ CAN1 Message Slot 15 Time stamp (C1MSL15TSP) <Address: H'0080 15FE>



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-15	CMSLnTSP	Message slot n time stamp	○	○

n = 0-15

These registers comprise a transmit frame/receive frame memory space. When transmission/ reception is completed, the value of the CAN Time stamp Count Register is stored in this register.

13.3 CAN Protocol

13.3.1 CAN Protocol Frames

Following four types of frames are handled by CAN protocol:

(1) Data frame
(2) Remote frame
(3) Error frame
(4) Overload frame

Each frame is separated by an interframe space.

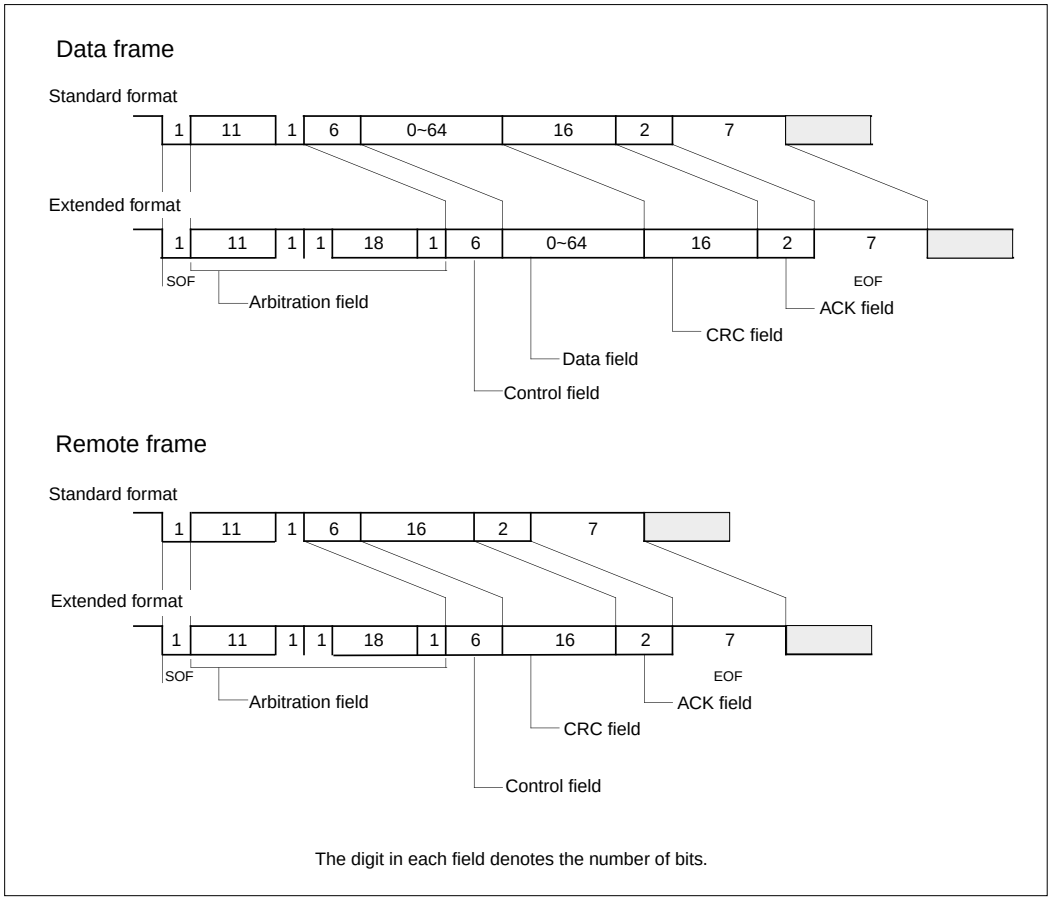


Figure 13.3.1 CAN Protocol Frames (1)

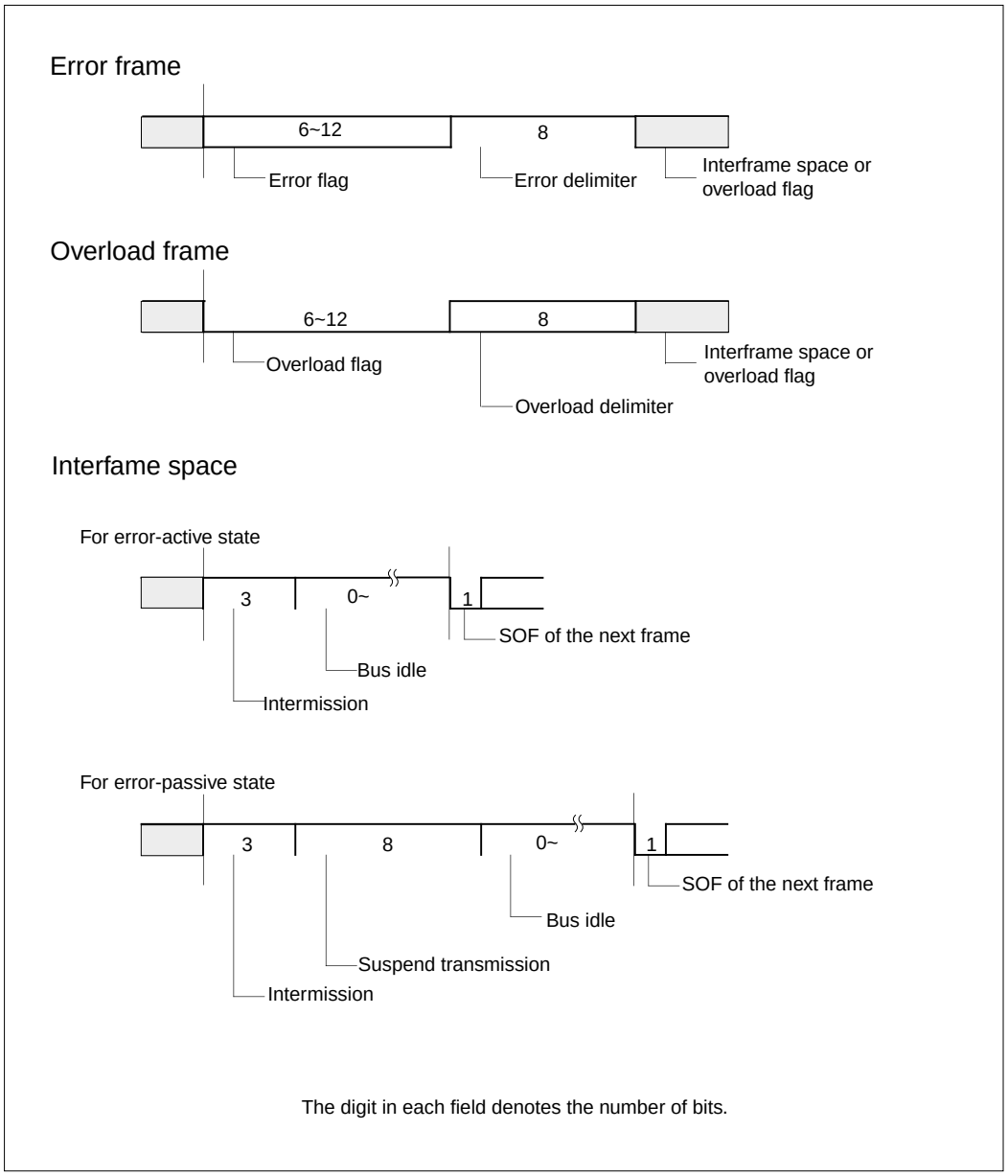


Figure 13.3.2 CAN Protocol Frames (2)

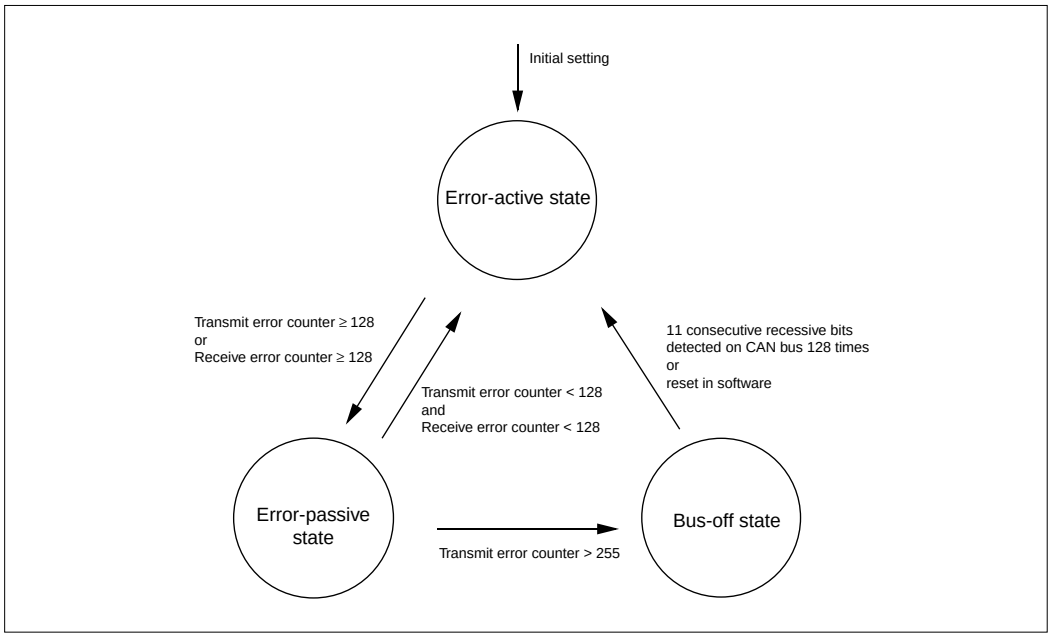


Figure 13.3.3 CAN Controller Error Status

The CAN Controller assumes one of the following three error states depending on the Transmit Error Counter and Receive Error Counter values.

(1) Error-active state

- This state is assumed when almost no errors have occurred.
- When an error is detected, the CAN Controller sends an active-error flag.
- The CAN Controller is in this state immediately after initialization.

(2) Error-passive state

- This state is assumed when many errors have occurred.
- When an error is detected, the CAN Controller sends a passive-error flag.

(3) Bus-off state

- This state is assumed when a large number of errors have occurred.
- The CAN module cannot communicate with any other node until it returns to an active-error state.

Status of Unit Error	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		–

13.4 Initialization of the CAN Module

13.4.1 Initializing the CAN Module

Before performing communication, set up the CAN module following the procedure described below.

(1) Selecting pin functions

The CAN transmit data output pins (CTX0 and CTX1) are shared with input/output ports performing dual functions. Therefore, select the function of these pins. (See Chapter 8, "Input/ Output Ports and Pin Functions.")

(2) Setting the Interrupt Controller (ICU)

When using CAN module interrupts, set their interrupt priority level with the Interrupt Controller.

(3) Setting the CAN Error Interrupt Mask and CAN Slot Interrupt Mask Registers

When using CAN bus-error, CAN error-passive, or CAN bus-off interrupts or CAN slot interrupt, enable the desired interrupt requests by setting the corresponding mask bits to 1.

(4) Setting the bit timing and sampling times

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**

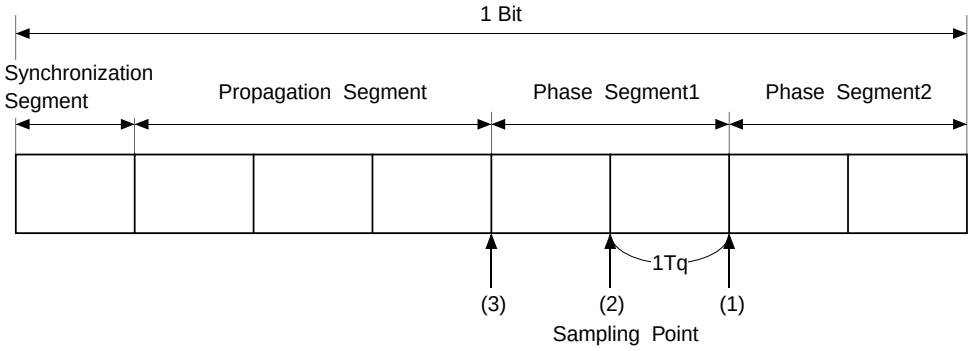
Determine the period Tq that constitutes the basis of bit timing, as well as the configuration of Propagation Segment, Phase Segment1, and Phase Segment2, and reSynchronization Jump Width.

Tq = (CANBRP + 1) / CPU clock

The baud rate is determined depending on the number of Tq's that comprise one bit. The baud rate is calculated by the equation below

$$\text{Baud rate (bps)} = \frac{1}{\text{Tq period x number of Tq's in one bit}}$$

Number of Tq's in one bit = Synchronization Segment +
Propagation Segment +
Phase Segment 1 +
Phase Segment 2



- This diagram shows bit timing when one bit is comprised of 8 Tq's.
- When chosen to sample once, the value sampled at Sampling Point (1) is assumed to be the value of the bit.
- When chosen to sample three times, the CAN bus values sampled at Sampling Points (1), (2), and (3) are resolved by majority logic to determine the value of the bit.

Figure 13.4.1 Example of Bit Timing

② **Setting the number of sampling times**

Select the number of times the CAN bus is sampled. This can be chosen to "sample once" or "sample three times."

- When chosen to sample once, the value sampled at the end of Phase Segment1 is assumed to be the value of the bit.
- When chosen to sample three times, the bus is sampled at three points-the first sampling point and 1 Tq and 2 Tq's before that, and the sampled values are resolved by majority logic to determine the value of the bit.

(5) Setting the ID Mask Registers

Set values in the ID mask registers used in acceptance filtering of the received message (Global Mask Register and Local Mask Registers A and B).

(6) Settings when operating in BasicCAN mode

- Set the CAN Extended ID Register IDE14 and IDE15 bits. (Setting the same value for both is recommended.)
- Set the ID in message slots 14 and 15.
- Set the Message Control Registers 14 and 15 to receive a data frame (H'40).

(7) Setting CAN module operation mode

Using the CAN Control Register (CANCNT), select operation mode of the CAN module (BasicCAN or loopback mode) and the clock source for the time stamp counter.

(8) Releasing the CAN module from reset

After initial settings in (1) through (7) above, clear the CAN Control Register (CANCNT)'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.

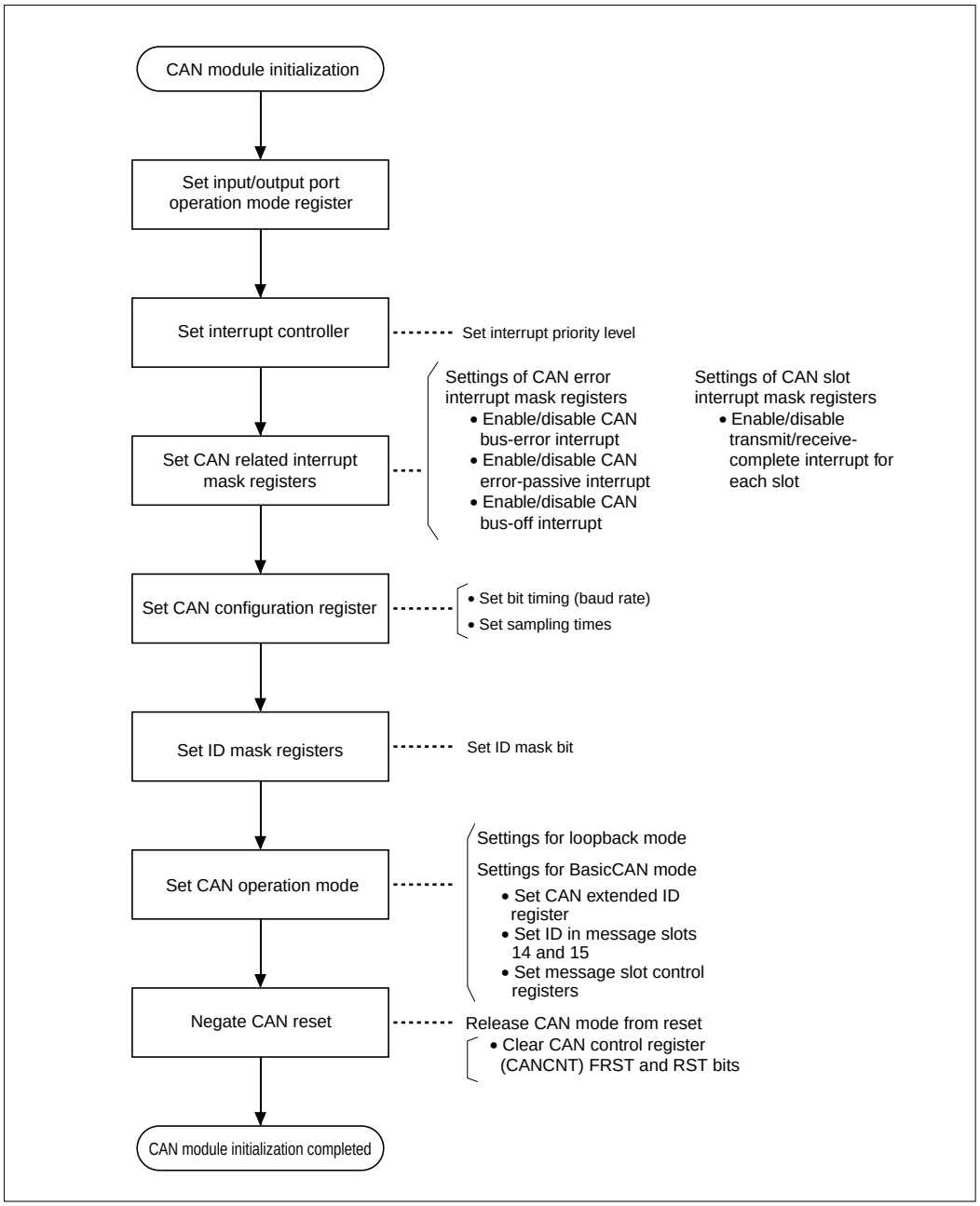


Figure 13.4.2 CAN Module Initialization

13.4.2 CAN Timing

The CAN modules incorporated in the M32R/E sample the asynchronous input signal on the CRX pin with a Tq clock period that is the base clock. The sampled signal is assumed to be the CAN bus value.

Operation timing is shown below.

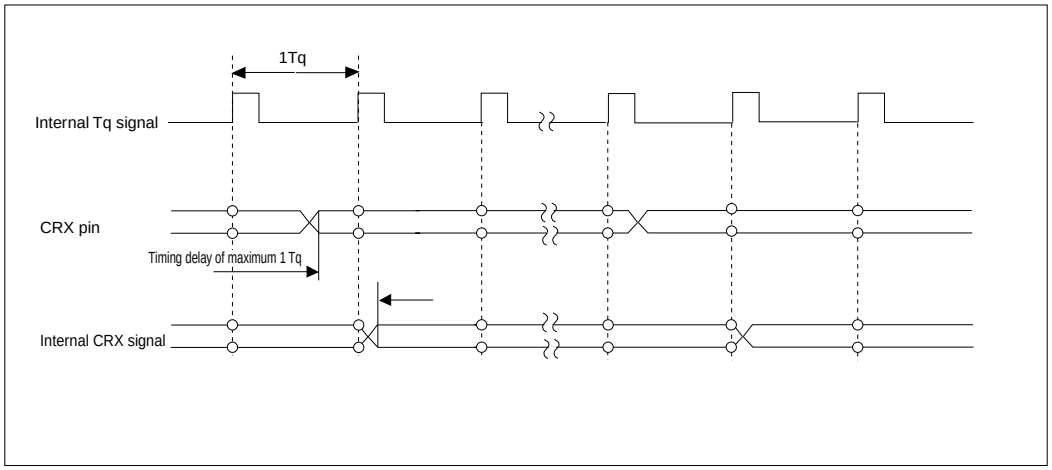


Figure 13.4.3 Operation Timing

13.5 Transmitting Data Frames

13.5.1 Data Frame Transmission Procedure

The following shows the procedure for transmitting a data frame.

(1) Initializing the CAN Message Slot Control Register

Initialize the CAN Message Slot Control Register for the message slot from which to transmit by writing H'00 to the register.

(2) Verifying that transmission is idle

Read the initialized CAN Message Slot Control Register and inspect the TRSTAT (transmit/ receive status) bit to see that transmission 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 extended ID register's corresponding bit to 0 when the data needs to be transmitted as a standard frame or 1 when the data needs to be transmitted as an extension frame.

(5) Setting the CAN Message Slot Control Register

Set the CAN Message Slot Control Register TR (transmit request) bit to 1 by writing H'80(Note) to the register.

Note: When transmitting a data frame, always be sure to write H'80 to this register.

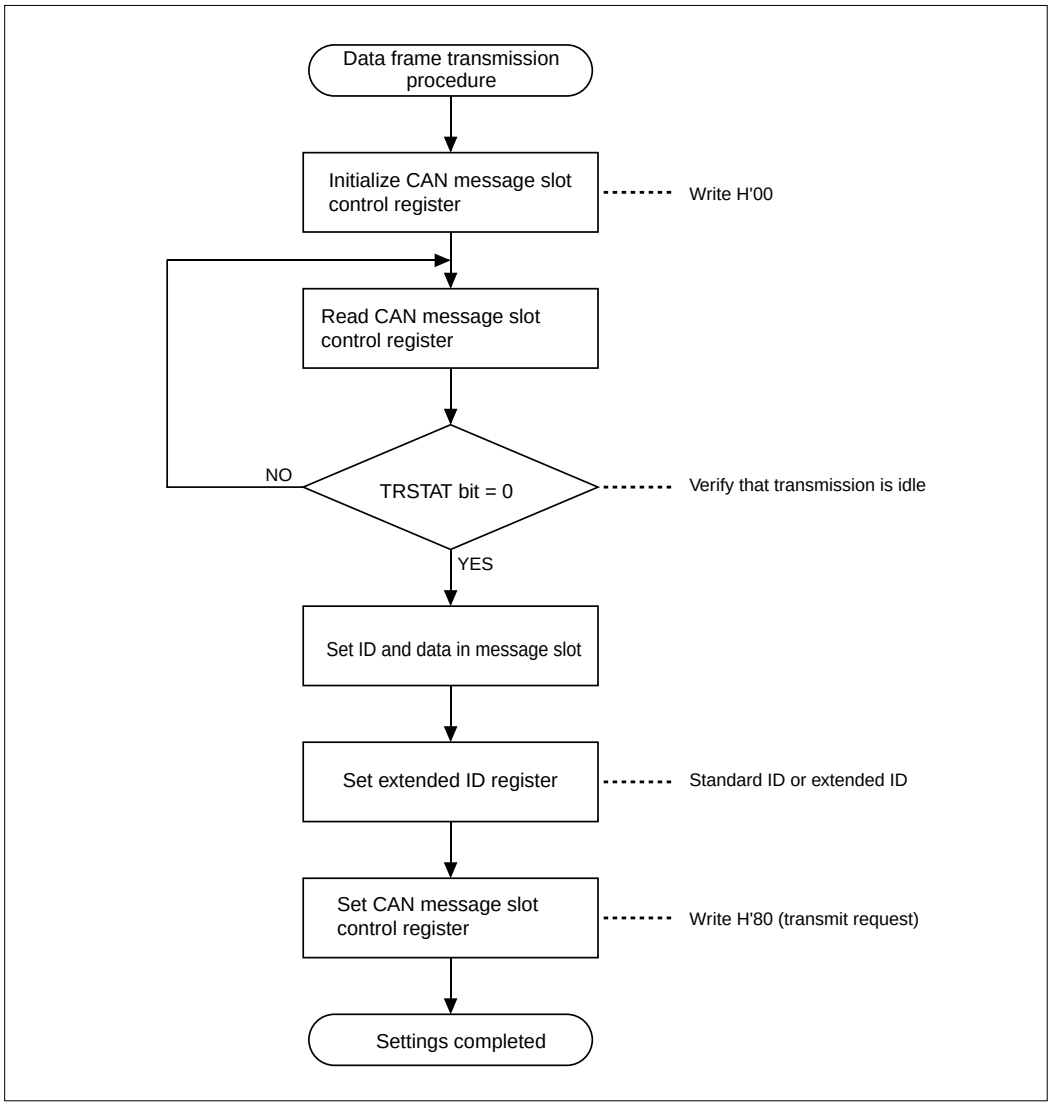


Figure 13.5.1 Procedure for Transmitting a Data Frame

13.5.2 Data Frame Transmit Operation

The following describes how data frame transmit operation is performed. All these operations are automatically performed in hardware.

(1) Selecting a transmit frame

The CAN module checks slots for which there is a transmit request (including remote frame transmit slots) every intermission to determine the frame to transmit.
If multiple transmit slots exist, frames are transmitted beginning with the slot that has the smallest slot number.

(2) Transmitting a data frame

When the transmit slot is determined, the CAN module starts transmitting the data frame from the slot after setting its corresponding CAN Message Slot Control Register's TRSTAT (transmit/ receive status) bit to 1.

(3) If failed in CAN bus arbitration or a CAN bus error occurs

If the CAN module loses in CAN bus arbitration or encounters a CAN bus error in the middle of transmission, it clears the CAN Message Slot Control Register's TRSTAT (transmit/receive status) bit to 0.
If the transmit abort function for the slot has been enabled, the transmit abort request is accepted and the message slot is freed, allowing for write to the slot.

(4) Completing data frame transmission

When the CAN module finishes transmitting the data frame, the CAN Message Slot Control Register's TRFIN (transmit/receive finished) bit and the CAN Slot Interrupt Status Register are set to 1. Also, the time stamp count value at completion of transmission is written to the CAN Message Slot Time stamp Register (CMSLnTSP), thereby completing the transmit operation.
If CAN slot interrupt has been enabled, an interrupt is generated upon completion of the transmit operation.
The slot which has had transmission completed becomes inactive and remains idle until new settings are made for it in software.

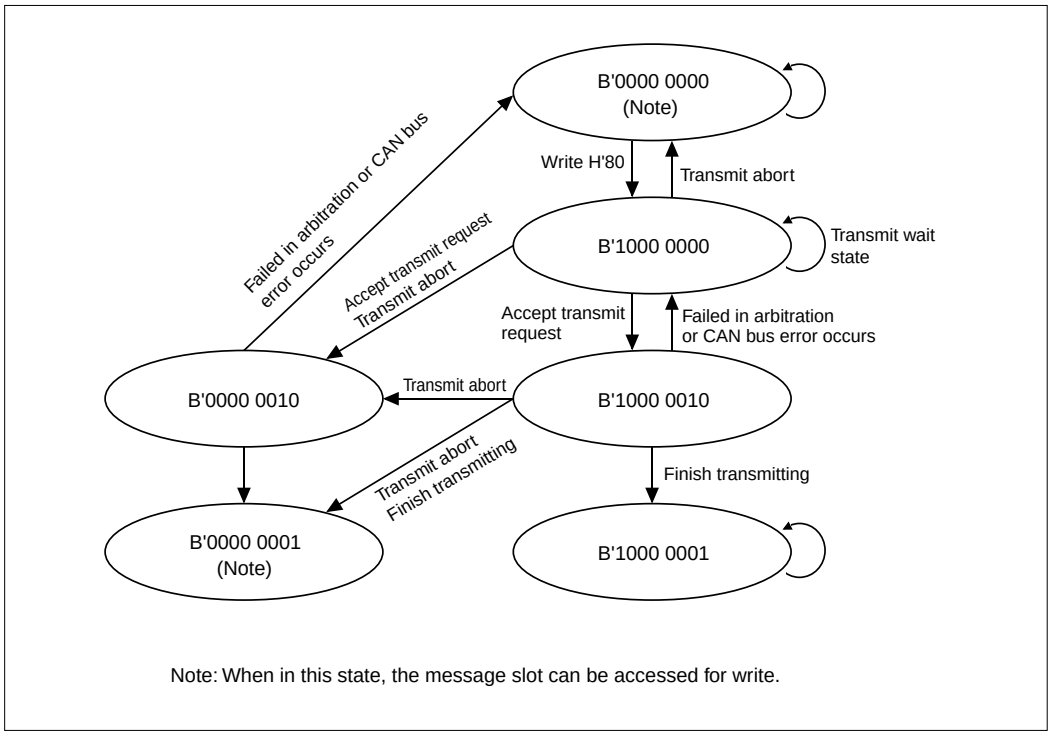


Figure 13.5.2 Operation of the CAN Message Slot Control Register during Data Frame Transmission

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 whose transmit request is to be canceled.
When transmit abort is accepted, the CAN module clears the CAN Message Slot Control Register TRSTAT (transmit/receive status) bit to 0, with the message slot thereby allowed for write.
The following lists 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 the CAN module loses in bus arbitration

13.6 Receiving Data Frames

13.6.1 Data Frame Reception Procedure

The following describes the procedure for receiving a data frame.

(1) Initializing the CAN Message Slot Control Register

Initialize the CAN Message Slot Control Register for the message slot in which to receive by writing H'00 to the register.

(2) Verifying that reception is idle

Read the initialized CAN Message Slot Control Register and inspect 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 ID

Set the ID desired to be received in the message slot.

(4) Setting the extended ID register

Set the extended ID register's corresponding bit to 0 when it is desired to receive a standard frame or 1 when it is desired to receive an extension frame.

(5) Setting the CAN Message Slot Control Register

Set the CAN Message Slot Control Register RR (receive request) bit to 1 by writing H'40 to the register.

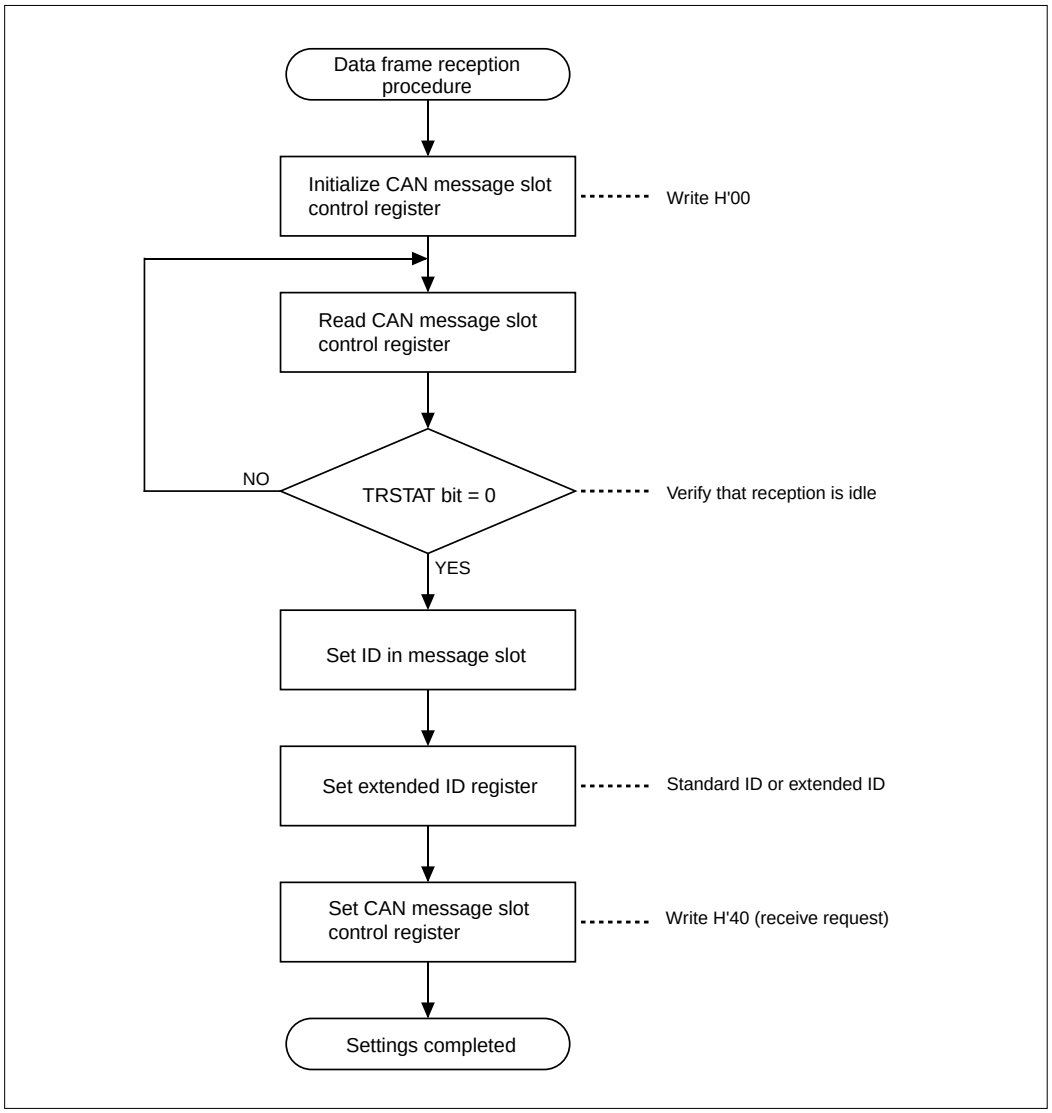


Figure 13.6.1 Procedure for Receiving a Data Frame

13.6.2 Data Frame Receive Operation

The following describes how data frame receive operation is performed. All these operations are automatically performed in hardware.

(1) Acceptance filtering

When data reception is completed, the CAN module searches for slots sequentially from slot 0 (up to slot 15) that meet the receive conditions and are therefore eligible for the received message.
The following shows the receive conditions for the slots that have been set to receive a data frame.

[Conditions]

- The received frame is a data frame.
- The received ID and the slot ID are the same, with the ID mask register bits set to 0 being ignored as "Don't care."
- The standard and extended frame types are the same.

Note: When in BasicCAN mode, slots 14 and 15, even when set to receive a data frame, can receive a remote frame too.

(2) When the receive conditions are met

When the receive conditions described in (1) above are met, the CAN module sets the CAN Message Slot Control Register's TRSTAT (transmit/receive status) and TRFIN (transmit/receive finished) bits to 1 and at the same time writes the received data to the message slot. If the TRFIN (transmit/receive finished) bit is already 1 at this time, the ML (message lost) bit also is set to 1, indicating that the message slot has been overwritten. The message slot has all of its ID and DLC fields overwritten, with an indeterminate value written to its unused area (e.g., the extended ID field when set for standard frame or an unused data field).
In addition, the time stamp count value at the time of message reception is written along with the received data to the CAN Message Slot Time stamp Register (CMSLnTSP). When the CAN module finishes writing to the message slot, it sets the CAN slot interrupt status bit to 1. If the slot has been enabled for interrupt, it generates an interrupt and waits for the next reception.

(3) When the receive conditions are not met

The CAN module discards the received frame and goes to the next transmit or receive operation. It does not write to the message slot.

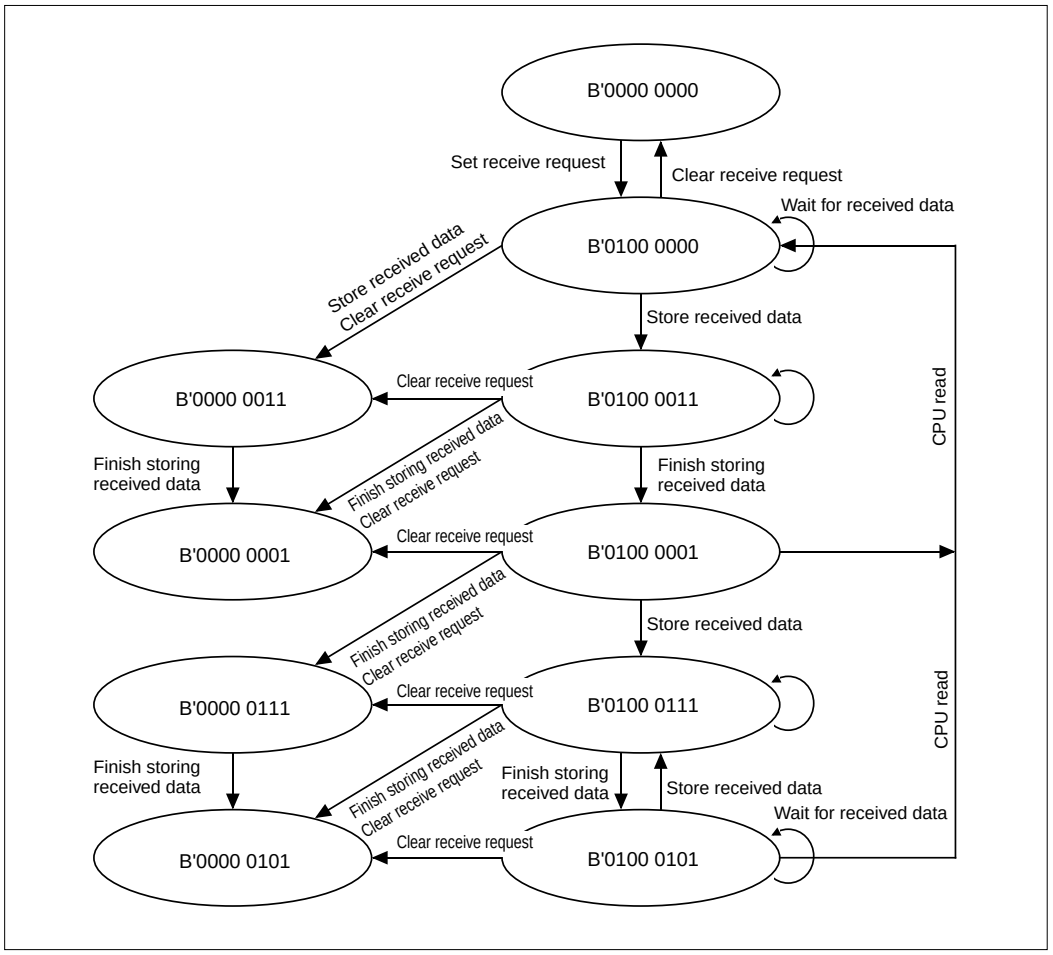


Figure 13.6.2 Operation of the CAN Message Slot Control Register during Data Frame Reception

13.6.3 Reading Out a Received Data Frame

The following shows the procedure for reading a received data frame from the slot.

(1) Clearing the TRFIN (transmit/receive finished) bit

Clear the CAN Message Slot Control Register (CMSLnCNT)'s TRFIN (transmit/receive finished) bit to 0 by writing H'4E, H'40, or H'00 to the register. The table below describes the slot operation after this write.

Value written to CMSLnCNT	Slot operation after writing
H'4E	Operates as a data frame receive slot. Whether the slot has been overwritten can be verified by inspecting the ML bit.
H'40	Operates as a data frame receive slot. Whether the slot has been overwritten can be verified by inspecting the ML bit.
H'00	Stops transmit/receive operation.

Note 1: If the slot needs to be checked for message lost by inspecting the ML bit, clear the TRFIN bit by writing H'4E to the register.

Note 2: If the TRFIN bit is cleared by writing H'4E, H'40, or H'00 to the register, it is possible that new data will be stored in the slot while reading the received data from it.

(2) Reading out from the message slot

Read the received data from the message slot.

(3) Checking the TRFIN (transmit/receive finished) bit

Read the CAN Message Slot Control Register to check the TRFIN (transmit/receive finished) bit.

① When the TRFIN (transmit/receive finished) bit = 1
It means that new data has been stored in the slot while reading out from it in (2) above. In this case, because the data read out in (2) contains an indeterminate value, redo from step (1), clearing the TRFIN (transmit/receive finished) bit.

② When the TRFIN (transmit/receive finished) bit = 0
It means that the received data has been read out normally.

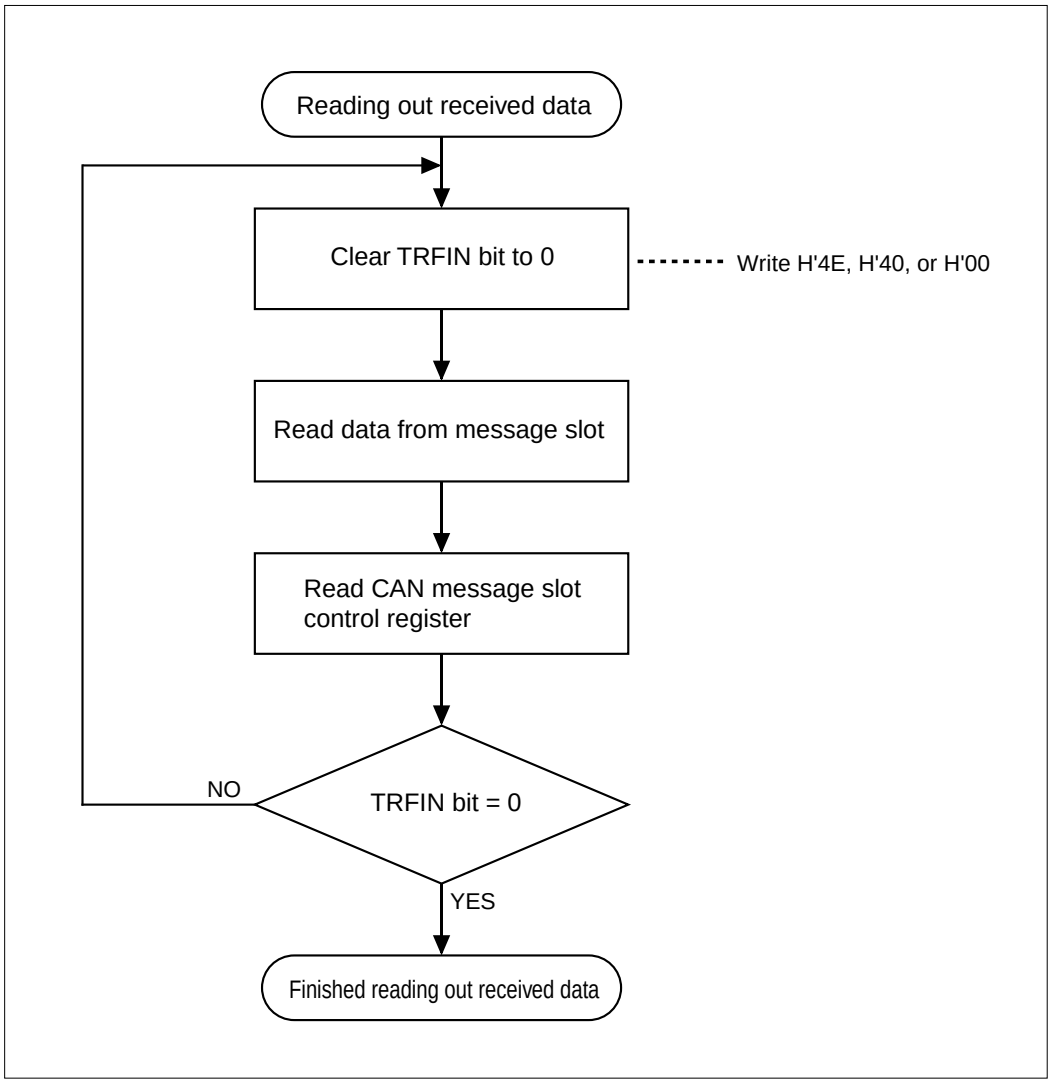


Figure 13.6.3 Procedure for Reading Out Received Data

13.7 Transmitting Remote Frames

13.7.1 Remote Frame Transmission Procedure

The following shows the procedure for transmitting a remote frame.

(1) Initializing the CAN Message Slot Control Register

Initialize the CAN Message Slot Control Register for the message slot from which to transmit by writing H'00 to the register.

(2) Verifying that transmission is idle

Read the initialized CAN Message Slot Control Register and inspect the TRSTAT (transmit/ receive status) bit to see that transmission 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 transmit ID

Set the ID to be transmitted in the message slot.

(4) Setting the extended ID register

Set the extended ID register's corresponding bit to 0 when it needs to be transmitted as a standard frame or 1 when it needs to be transmitted as an extension frame.

(5) Setting the CAN Message Slot Control Register

Set the CAN Message Slot Control Register TR (transmit request) and RM (remote) bits to 1 by writing H'A0 to the register.

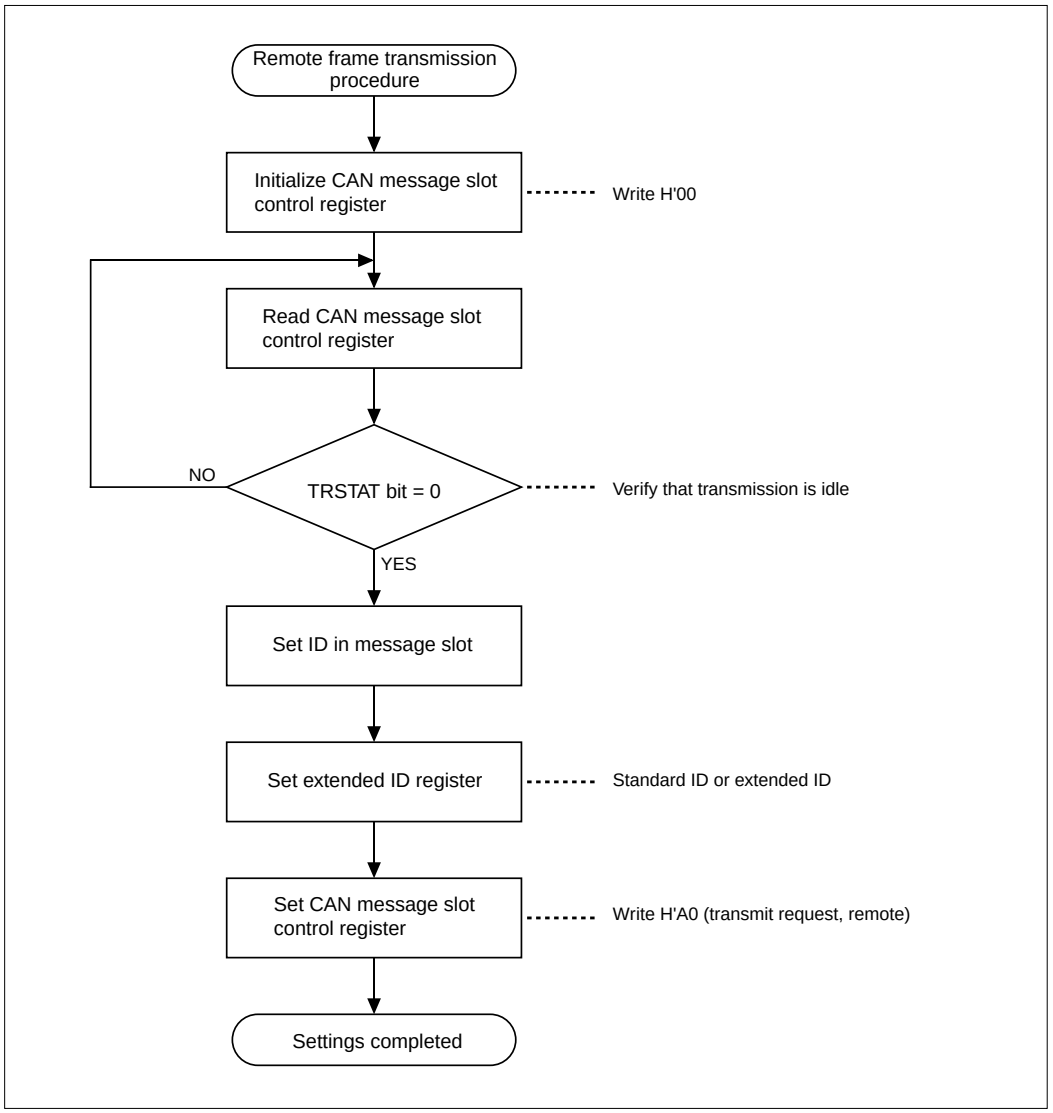


Figure 13.7.1 Procedure for Transmitting a Remote Frame

13.7.2 Remote Frame Transmit Operation

The following describes how remote frame transmit operation is performed. All these operations are automatically performed in hardware.

(1) Setting the RA (remote active) bit

The RA (remote active) bit that indicates that the slot is selected to handle a remote frame is set to 1 at the same time the data H'A0 (transmit request, remote) is written to the CAN Message Slot Control Register.

(2) Selecting a transmit frame

The CAN module checks slots for which there is a transmit request (including data frame transmit slots) every intermission to determine the frame to transmit.
If multiple transmit slots exist, frames are transmitted beginning with the slot that has the smallest slot number.

(3) Transmitting a remote frame

When the transmit slot is determined, the CAN module starts transmitting the remote frame from the slot after setting its corresponding CAN Message Slot Control Register's TRSTAT (transmit/ receive status) bit to 1.

(4) If failed in CAN bus arbitration or a CAN bus error occurs

If the CAN module loses in CAN bus arbitration or encounters a CAN bus error in the middle of transmission, it clears the CAN Message Slot Control Register's TRSTAT (transmit/receive status) bit to 0.
If the transmit abort function for the slot has been enabled, the transmit abort request is accepted and the message slot is freed, allowing for write to the slot.

(5) Completing remote frame transmission

When the CAN module finishes transmitting the remote frame, the time stamp count value at completion of transmission is stored in the CAN Message Slot Time stamp Register (CMSLnTSP) and the CAN Message Slot Control Register's RA (remote active) bit is cleared to 0.
Also, the CAN slot interrupt status bit is set to 1 by completion of transmission, while the CAN Message Slot Control Register's TRFIN (transmit/receive finished) bit remains intact (this bit is not set to 1).
If CAN slot interrupt has been enabled, an interrupt is generated upon completion of the transmit operation.

(6) Data frame reception

When remote frame transmission is completed, the slot automatically becomes to function as a data frame receive slot.

(7) Acceptance filtering

When data reception is completed, the CAN module searches for slots sequentially from slot 0 (up to slot 15) that meet the receive conditions and are therefore eligible for the received message. The following shows the receive conditions for the slots that have been set to receive a data frame.

[Conditions]

- The received frame is a data frame.
- The received ID and the slot ID are the same, with the ID mask register bits set to 0 being ignored as "Don't care."
- The standard and extended frame types are the same.

Note: When in BasicCAN mode, slots 14 and 15 cannot be used as transmit slots.

(8) When the receive conditions are met

When the receive conditions described in (7) above are met, the CAN module sets the CAN Message Slot Control Register's TRSTAT (transmit/receive status) and TRFIN (transmit/receive finished) bits to 1 and at the same time writes the received data to the message slot. If the TRFIN (transmit/receive finished) bit is already 1 at this time, the ML (message lost) bit also is set to 1, indicating that the message slot has been overwritten. The message slot has all of its ID and DLC fields overwritten, with an indeterminate value written to its unused area (e.g., the extended ID field when set for standard frame or an unused data field).
In addition, the time stamp count value at the time of message reception is written along with the received data to the CAN Message Slot Time stamp Register (CMSLnTSP). When the CAN module finishes writing to the message slot, it sets the CAN slot interrupt status bit to 1. If the slot has been enabled for interrupt, it generates an interrupt and waits for the next frame to receive.

Note: If the CAN module receives a data frame before transmitting the remote frame, it stores the received data frame and does not transmit the remote frame.

(9) When the receive conditions are not met

The CAN module discards the received frame and goes to the next transmit or receive operation. It does not write to the message slot.

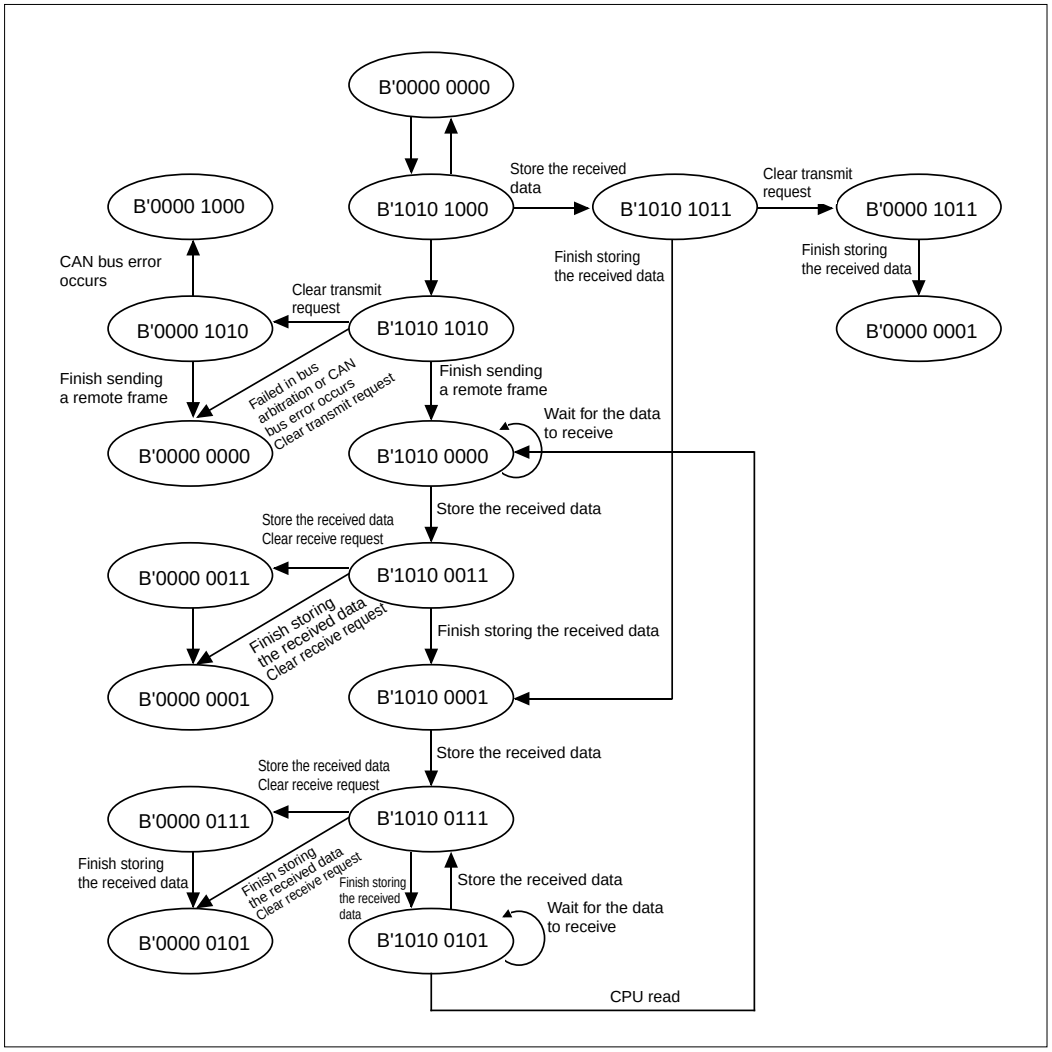


Figure 13.7.2 Operation of the CAN Message Slot Control Register during Remote Frame Transmission

13.7.3 Reading Out a Received Data Frame When Set for Remote Frame Transmission

The following shows the procedure for reading a received data frame from the slot when it is set for remote frame transmission.

(1) Clearing the TRFIN (transmit/receive finished) bit

Clear the CAN Message Slot Control Register (CMSLnCNT)'s TRFIN (transmit/receive finished) bit to 0 by writing H'AE or H'00 to the register. The table below describes the slot operation after this write.

Value written to CMSLnCNT	Slot operation after writing
H'AE	Operates as a data frame receive slot. Whether the slot has been overwritten can be verified by inspecting the ML bit.
H'00	Stops transmit/receive operation.

Note 1: If the slot needs to be checked for message lost by inspecting the ML bit, clear the TRFIN bit by writing H'AE to the register.

Note 2: If the TRFIN bit is cleared by writing H'AE or H'00 to the register, it is possible that new data will be stored in the slot while reading the received data from it.

Note 3: The received data frame cannot be read out if the TRFIN bit is cleared by writing H'A0 to the register. In this case, the slot is set to transmit a remote frame.

(2) Reading out from the message slot

Read the received data from the message slot.

(3) Checking the TRFIN (transmit/receive finished) bit

Read the CAN Message Slot Control Register to check the TRFIN (transmit/receive finished) bit.

① When the TRFIN (transmit/receive finished) bit = 1
It means that new data has been stored in the slot while reading out from it in (2) above. In this case, because the data read out in (2) contains an indeterminate value, redo from step (1), clearing the TRFIN (transmit/receive finished) bit.

② When the TRFIN (transmit/receive finished) bit = 0
It means that the received data has been read out normally.

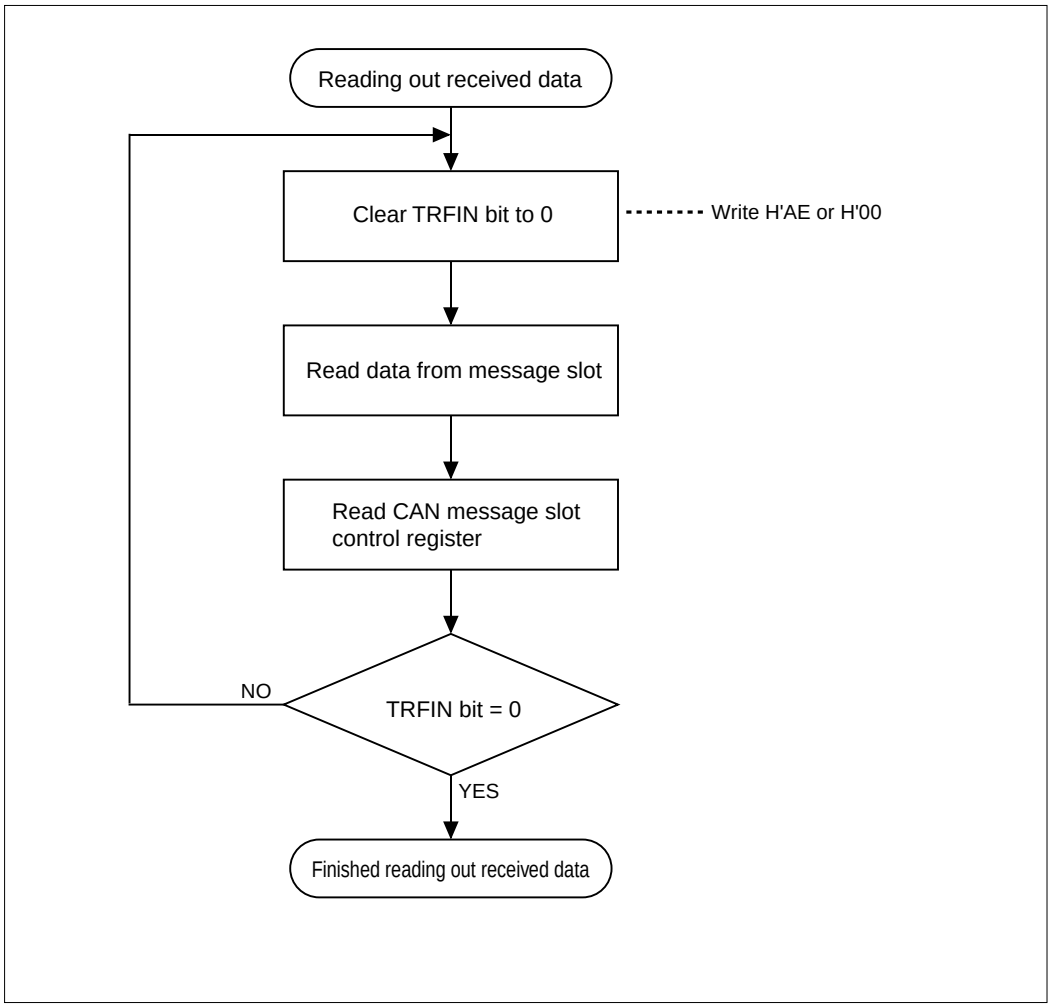


Figure 13.7.3 Procedure for Reading Out Received Data When Set for Remote Frame Transmission

13.8 Receiving Remote Frames

13.8.1 Remote Frame Reception Procedure

The following describes the procedure for receiving a remote frame.

(1) Initializing the CAN Message Slot Control Register

Initialize the CAN Message Slot Control Register for the message slot in which to receive by writing H'00 to the register.

(2) Verifying that reception is idle

Read the initialized CAN Message Slot Control Register and inspect 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 ID

Set the ID desired to be received in the message slot.

(4) Setting the extended ID register

Set the extended ID register's corresponding bit to 0 when it is desired to receive a standard frame or 1 when it is desired to receive an extension frame.

(5) Setting the CAN Message Slot Control Register

① When remote frame reception needs to be automatically answered (data frame transmission)
Set the CAN Message Slot Control Register RR (receive request) and RM (remote) bits to 1 by writing H'60 to the register.

② When remote frame reception does not need to be automatically answered
Set the CAN Message Slot Control Register RR (receive request), RM (remote), and RL (automatic answering enable) bits to 1 by writing H'70 to the register.

Note: When in BasicCAN mode, although slots 14 and 15 can receive a remote frame, they cannot be set to automatically answer the received remote frame.

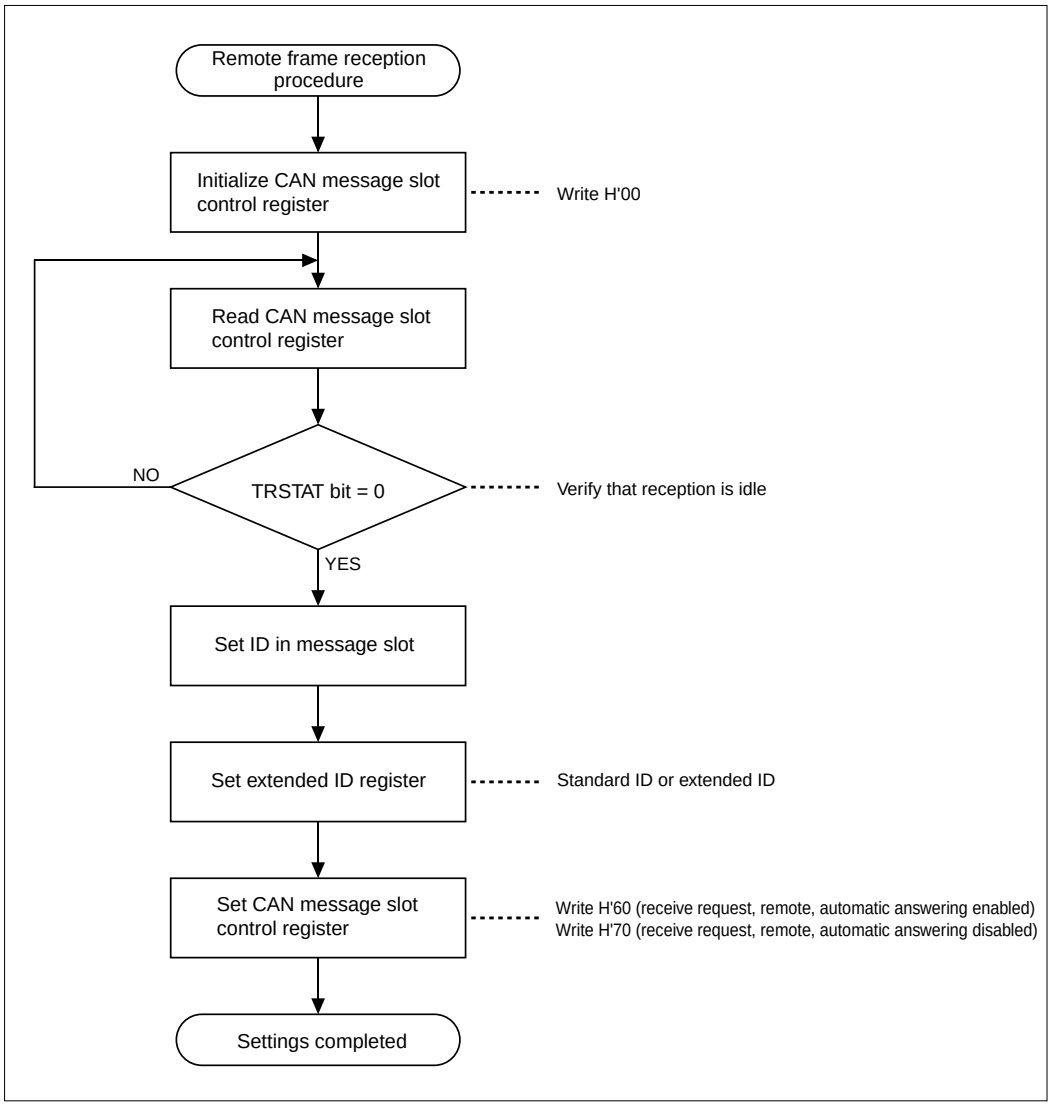


Figure 13.8.1 Procedure for Receiving a Remote Frame

13.8.2 Remote Frame Receive Operation

The following describes how remote frame receive operation is performed. All these operations are automatically performed in hardware.

(1) Setting the RA (remote active) bit

The RA (remote active) bit that indicates that the slot is selected to handle a remote frame is set to 1 at the same time the data H'60 (receive request, remote) or H'70 (receive request, remote, automatic answering disabled) is written to the CAN Message Slot Control Register.

(2) Acceptance filtering

When data reception is completed, the CAN module searches for slots sequentially from slot 0 (up to slot 15) that meet the receive conditions and are therefore eligible for the received message.
The following shows the receive conditions for the slots that have been set to receive a remote frame.

[Conditions]

- The received frame is a remote frame.
- The received ID and the slot ID are the same, with the ID mask register bits set to 0 being ignored as "Don't care."
- The standard and the extended frames are of the same type.

(3) When the receive conditions are met

When the receive conditions described in (2) above are met, the CAN module sets the CAN Message Slot Control Register's TRSTAT (transmit/receive status) and TRFIN (transmit/receive finished) bits to 1 and at the same time writes the received data to the message slot.
In addition, the time stamp count value at the time of message reception is written along with the received data to the CAN Message Slot Time stamp Register (CMSLnTSP). When the CAN module finishes writing to the message slot, it sets the CAN slot interrupt status bit to 1. If the slot has been enabled for interrupt, it generates an interrupt and waits for the next reception.

Note 1: The message slot has the ID field and the DLC value written to it.
Note 2: When the standard format is selected, the slot has an indeterminate value written to its extended ID area.
Note 3: The data field is not accessed for write.
Note 4: The RA and TRFIN bits are cleared to 0 after writing to the received remote frame data to the slot.

(4) When the receive conditions are not met

The CAN module discards the received data and waits for the next frame to receive. It does not write to the message slot.

(5) Operation after receiving a remote frame

The operation to be performed for the slot after receiving a remote frame differs depending on whether automatic answering is enabled or not.

① When automatic answering is disabled

The slot which finished receiving data becomes inactive and remains idle (neither transmitting nor receiving) until new settings are made for it in software.

② When automatic answering is enabled

After receiving a remote frame, the slot automatically changes to a data frame transmit slot, performing the transmit processing described below. In this case, the data to be transmitted conforms to the ID and DLC of the received remote frame.

- **Selecting a transmit frame**

  The CAN module checks slots for which there is a transmit request (including remote frame transmit slots) every intermission to determine the frame to transmit. If multiple transmit slots exist, frames are transmitted beginning with the slot that has the smallest slot number.

- **Transmitting a data frame**

  When the transmit slot is determined, the CAN module starts transmitting the data frame from the slot after setting its corresponding CAN Message Slot Control Register's TRSTAT (transmit/receive status) bit to 1.

- **If failed in CAN bus arbitration or a CAN bus error occurs**

  If the CAN module loses in CAN bus arbitration or encounters a CAN bus error in the middle of transmission, it clears the CAN Message Slot Control Register's TRSTAT (transmit/receive status) bit to 0.
  If the transmit abort function for the slot has been enabled, the transmit abort request is accepted and the message slot is freed, allowing for write to the slot.

- **Completing data frame transmission**

  When the CAN module finishes transmitting the data frame, the CAN Message Slot Control Register's TRFIN (transmit/receive finished) bit and the CAN Slot Interrupt Status Register are set to 1. Also, the time stamp count value at completion of transmission is written to the CAN Message Slot Time stamp Register (CMSLnTSP), thereby completing the transmit operation.
  If CAN slot interrupt has been enabled, an interrupt is generated upon completion of the transmit operation.
  The slot which has had transmission completed becomes inactive and remains idle until new settings are made for it in software.

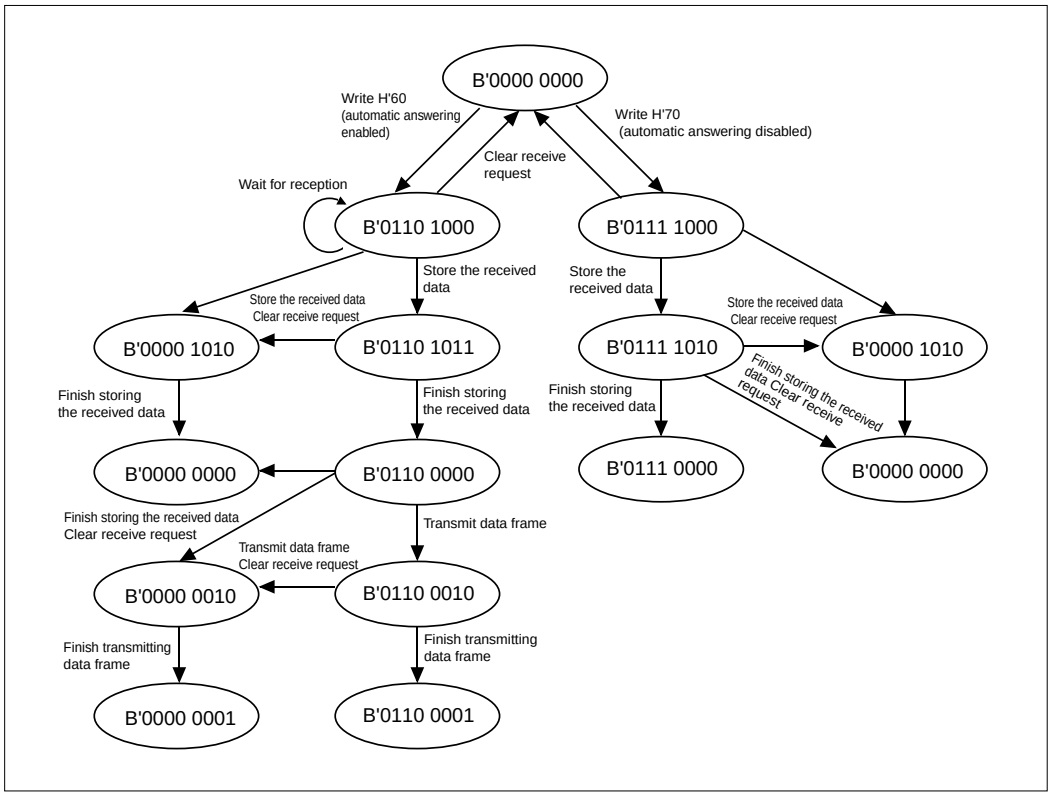


Figure 13.8.2 Operation of the CAN Message Slot Control Register during Remote Frame Reception

CHAPTER 14

REAL-TIME DEBUGGER (RTD)

14.1 Outline of the Real-Time Debugger (RTD)

The Real-Time Debugger (RTD) is a serial I/O through which to read or write to the internal RAM's entire area using commands from outside the microprocessor. Because data transfers between the RTD and internal RAM are performed using an internal dedicated bus independently of the M32R CPU, operation can be controlled without having the stop the M32R CPU.

Table 14.1.1 Outline of the Real-Time Debugger (RTD)

Item	Content
Transfer method	Clock-synchronized serial I/O
Generation of transfer clock	Generated by external host
RAM access area	Entire area of 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 lines (RTDTXD, RTDRXD, RTDACK, RTDCLK)
Number of commands	Following five functions • Monitors continuously • Outputs real-time RAM contents • Forcibly rewrites RAM contents (with verify) • Recovers from runaway • Requests RTD interrupt

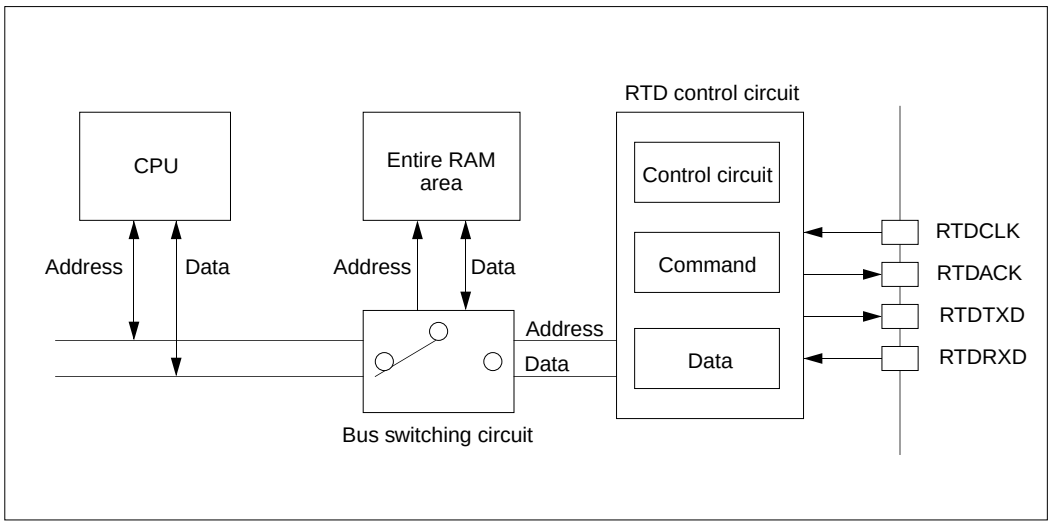


Figure 14.1.1 Block Diagram of the Real-Time Debugger (RTD)

14.2 Pin Function of the RTD

Pin functions of the RTD are shown below.

Table 14.2.1 Pin Function of the RTD

Pin Name	Type	Function
RTDTXD	Output	RTD serial data output
RTDRXD	Input	RTD serial data input
RTDACK	Output	Outputs a low-level pulse synchronously with the beginning clock edge of the output data word. The width of the low-level pulse thus output indicates the type of instruction/data that the RTD 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 frorunaway)command
RTDCLK	Input	RTD transfer clock input

14.3 Functional Description of the RTD

14.3.1 Outline of RTD Operation

Operation of the RTD is specified by a command entered from devices external to the chip. A command is specified in bits 16-19(note 1) of the RTD receive data.

Table 14.3.1 RTD Commands

RTD Receive Data				Command Mnemonic	RTD Function
b19	b18	b17	b16		
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 forcibly rewrite (with verify)
1	1	1	1	RCV (ReCoVer)	Recover from runaway (Note 2, Note 3)
0	0	0	1	System reserved (use inhibited)	

↑ (Note 1)

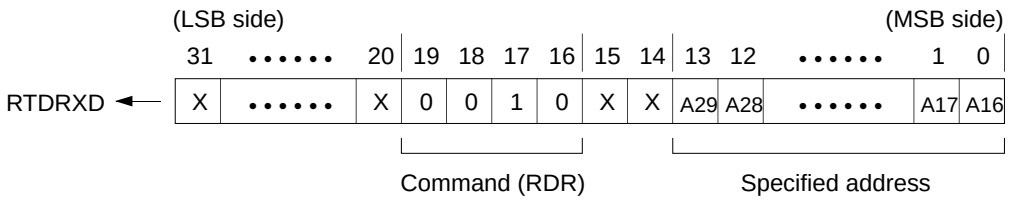
Note 1: Bit 19 of RTD receive data is not actually stored in the command register and except for the RCV command, is handled as "Don't Care" bit. (Bits 16-18 are effective for the command specified.)

Note 2: The RCV command must always be transmitted twice in succession.

Note 3: For the RCV command, all bits, not just bits 16-19, (i.e., bits 0-15 and bits 20-31) must be set to 1.

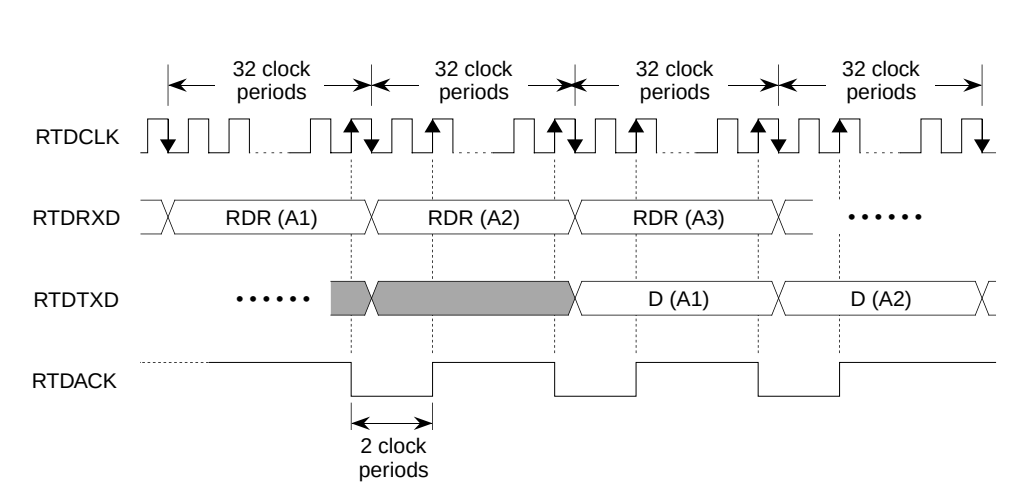
14.3.2 Operation of RDR (Real-time RAM Content Output)

When the RDR (real-time RAM content output) command is issued, the RTD is made possible 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 no transfers are being performed between the CPU and internal RAM, no extra load is levied on the CPU.
The address to be read from the internal RAM can only be specified on 32-bit word boundaries. (The two low-order address bits specified by a command are ignored.) Note also that data are read out in units of 32 bits as transferred from the internal RAM to an external device.



Note: X = Don't Care (However, if issued immediately after the RCV command, bits 20-31 must all be set to 1.)

Figure 14.3.1 RDR Command Data Format



Note: (An) = Specified address
D (An) = Data at specified address (An)

Figure 14.3.2 Operation of the RDR Command

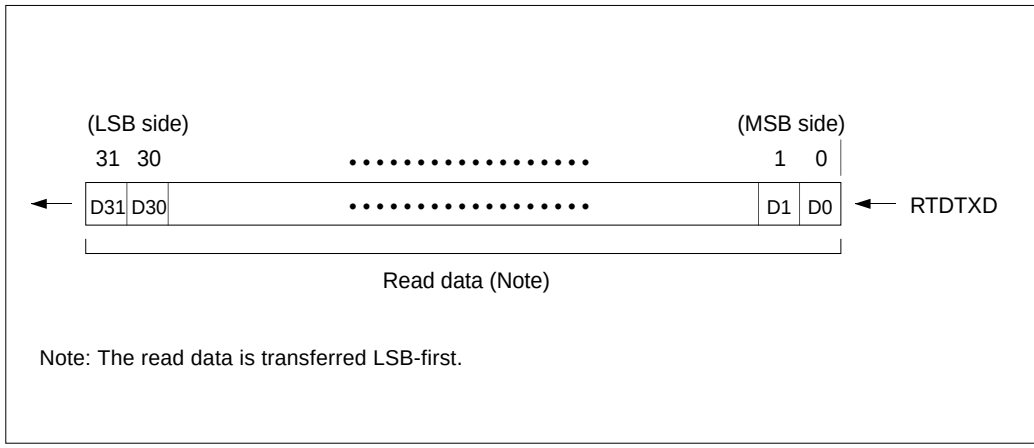
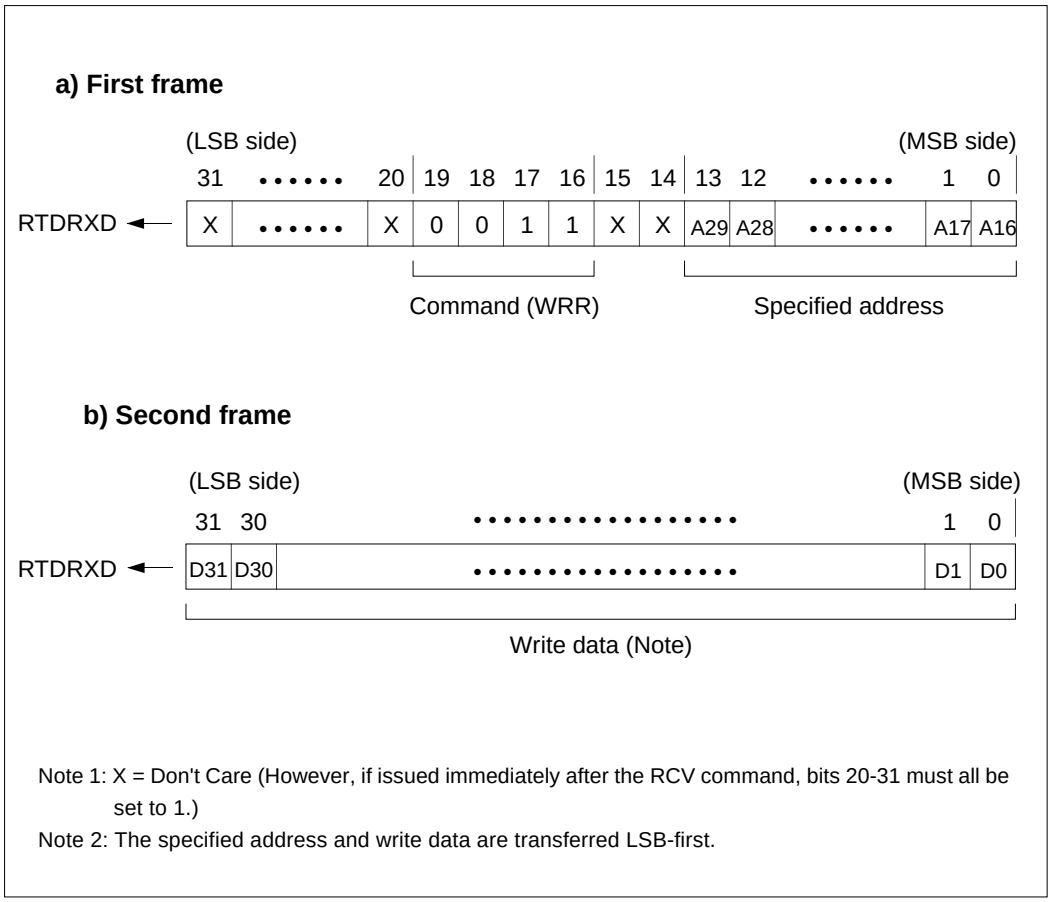


Figure 14.3.3 Read Data Transfer Format

14.3.3 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 no transfers are being performed between the CPU and internal RAM, no extra load is levied on the CPU.
The address to be read from the internal RAM can only be specified on 32-bit word boundaries. (The two low-order address bits specified by a command are ignored.) Note also that data are written to the internal RAM in units of 32 bits.
The external host should transmit the command and address in the first frame and then the write data in the second frame. The timing at which the RTD writes to the internal RAM occurs in the third frame after receiving the write data.



Note 1: X = Don't Care (However, if issued immediately after the RCV command, bits 20-31 must all be set to 1.)
Note 2: The specified address and write data are transferred LSB-first.

Figure 14.3.4 WRR Command Data Format

The RTD reads out data from the specified address before writing to the internal RAM and again reads out 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.

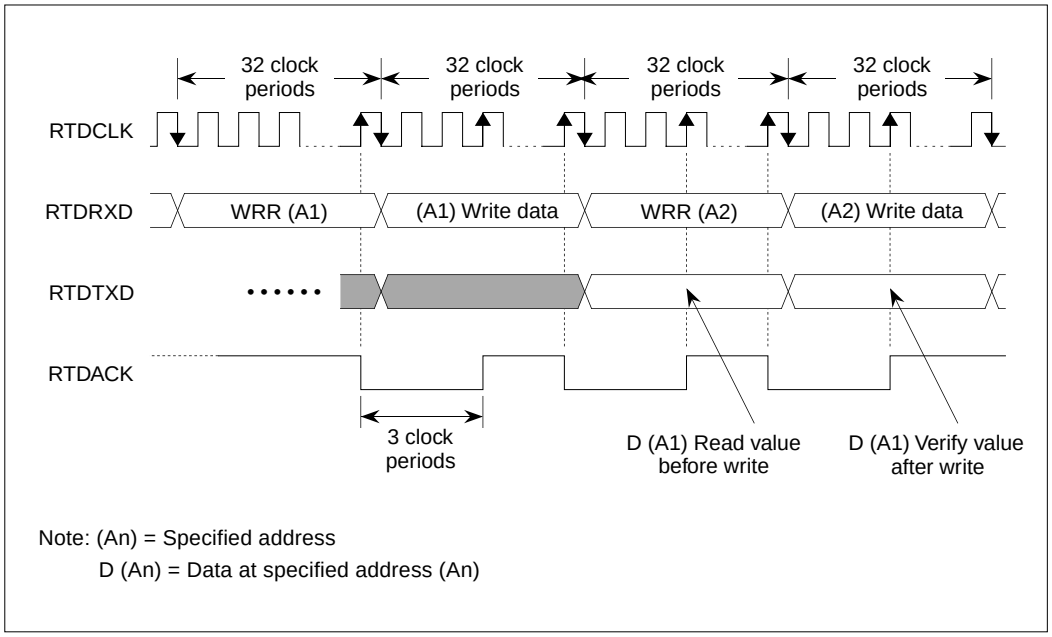
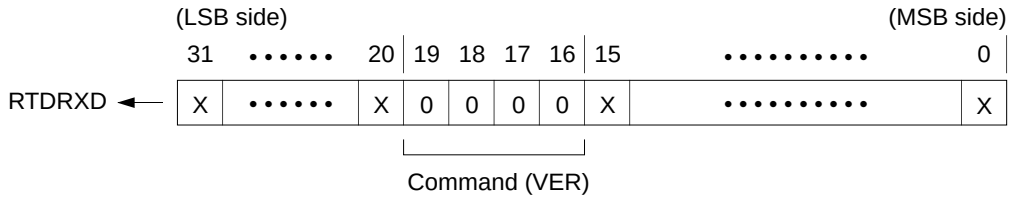


Figure 14.3.5 Operation of the WRR Command

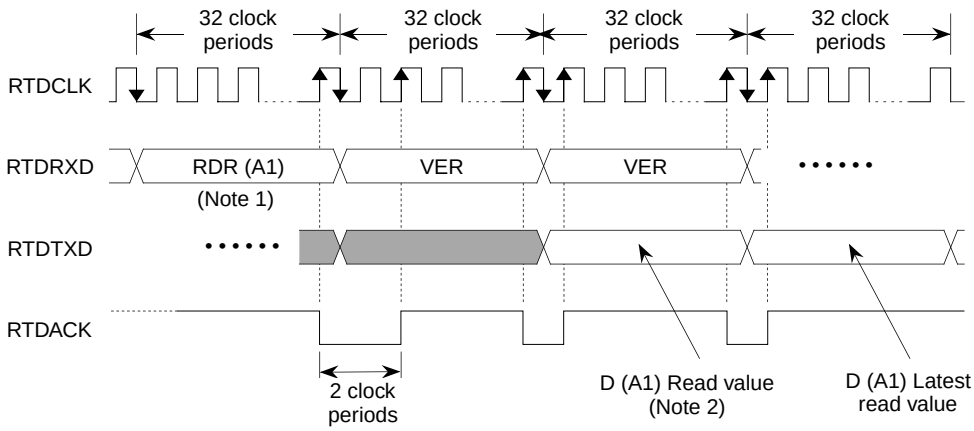
14.3.4 Operation of VER (Continuous Monitor)

When the VER (continuous monitor) command is issued, the RTD outputs data from the address that has been accessed by the instruction (either read or write) immediately before receiving the VER command.



Note: X = Don't Care (However, if issued immediately after the RCV command, bits 20-31 must all be set to 1.)

Figure 14.3.6 VER (Continuous Monitor) Command Data Format



Note 1: WRR command can also be used.
Note 2: (An) = Specified address
D (An) = Data at specified address (An)

Figure 14.3.7 Operation of the VER (Continuous Monitor) Command

14.3.5 Operation of VEI (Interrupt Request)

When the VEI (interrupt request) command is issued, the RTD outputs data from the address that has been accessed by the instruction (either read or write) immediately before receiving the VEI command.

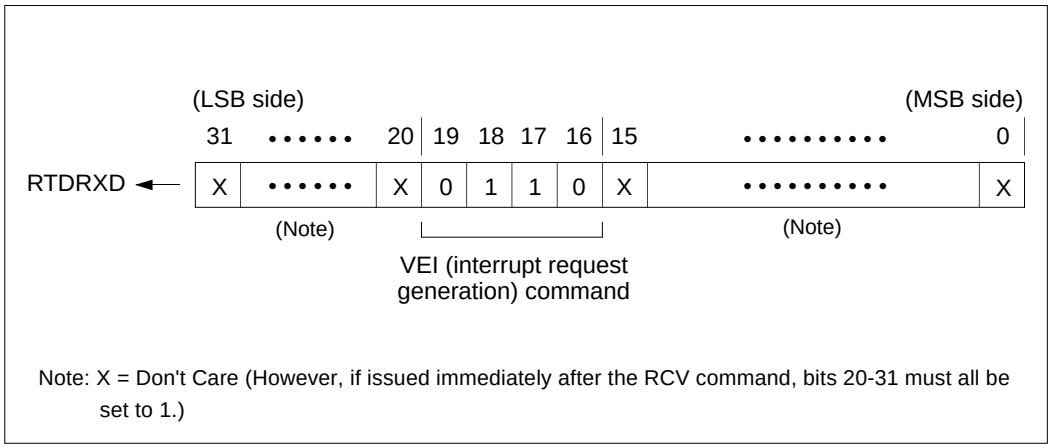


Figure 14.3.8 VEI (Interrupt Request) Command Data Format

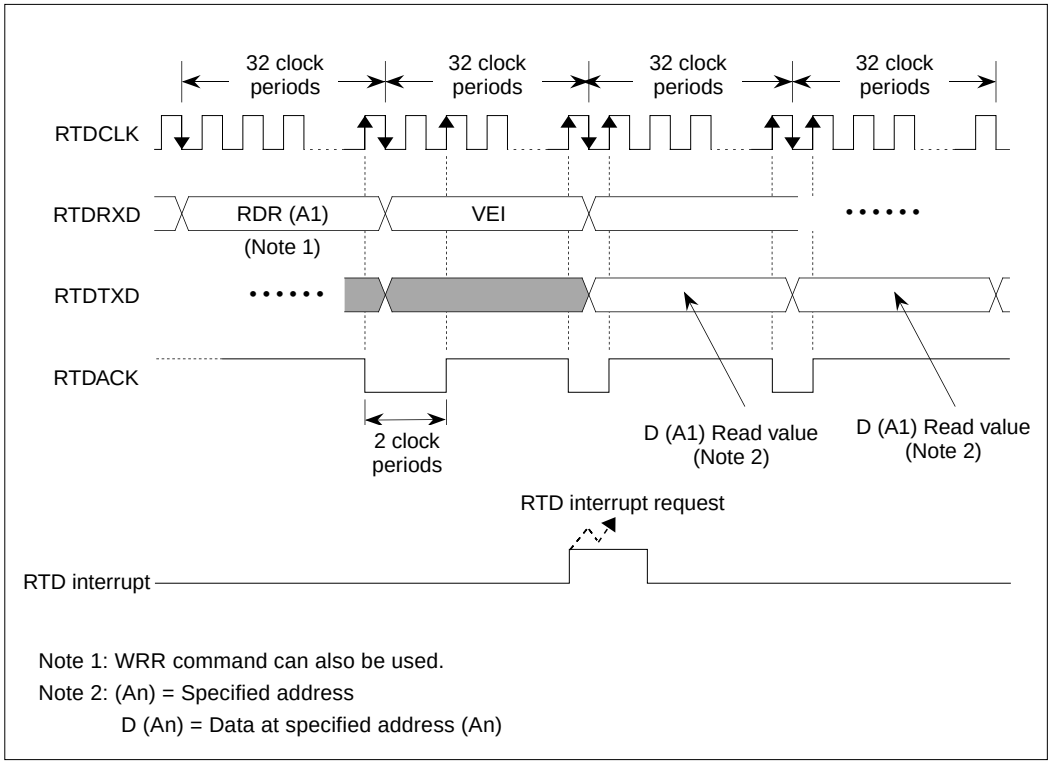
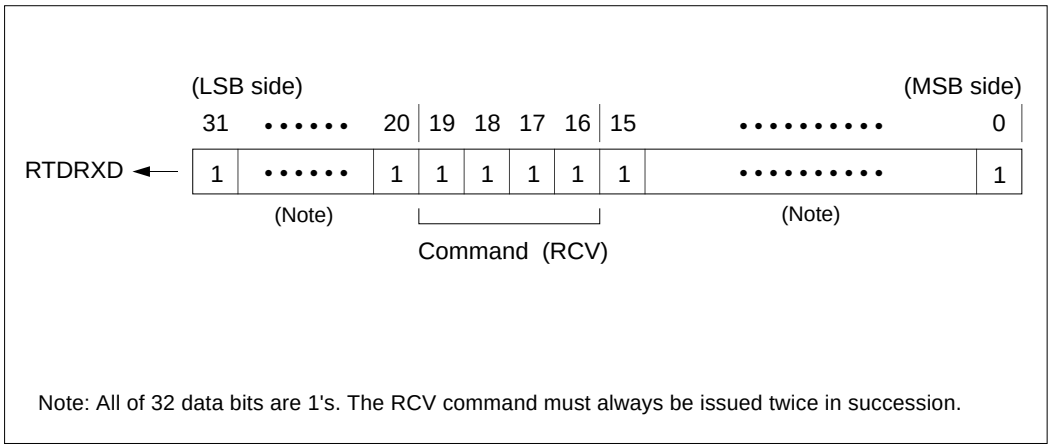


Figure 14.3.9 Operation of the VEI (Interrupt Request) Command

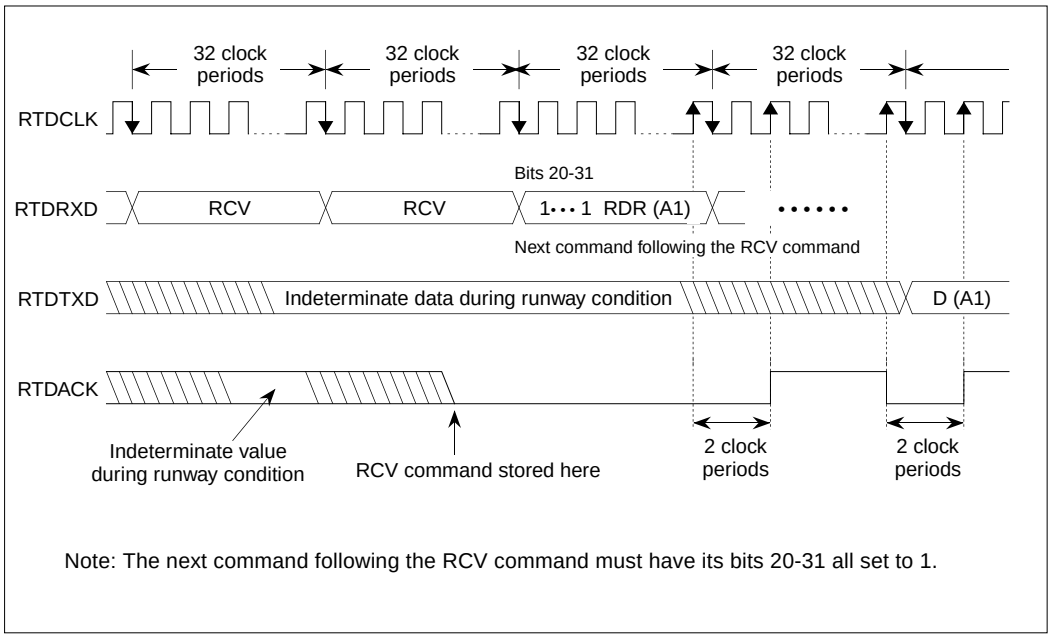
14.3.6 Operation of RCV (Recover from Runaway)

When the RTD runs out of control, the RCV (recover from runway) command can be issued to forcibly recover from the runaway condition without having to reset the system. The RCV command must always be issued twice in succession. Also, any command issued subsequently after the RCV command must have its bits 20-31 all set to 1.



Note: All of 32 data bits are 1's. The RCV command must always be issued twice in succession.

Figure 14.3.10 RCV Command Data Format



Note: The next command following the RCV command must have its bits 20-31 all set to 1.

Figure 14.3.11 Operation of the RCV Command

14.3.7 Method to Set a Specified Address when Using the RTD

When using the Real-Time Debugger (RTD), you can set low-order 16-bit addresses of the internal RAM area. Because the internal RAM area is located in a 48 KB area ranging from H'0080 4000 to H'0080 FFFF, you can set low-order 16-bit addresses of that area. However, access to any locations other than the area where the RAM resides is inhibited. Note also that two least significant address bits, A31 and A30, are always 0's because data are read and written to the internal RAM in a fixed length of 32 bits.

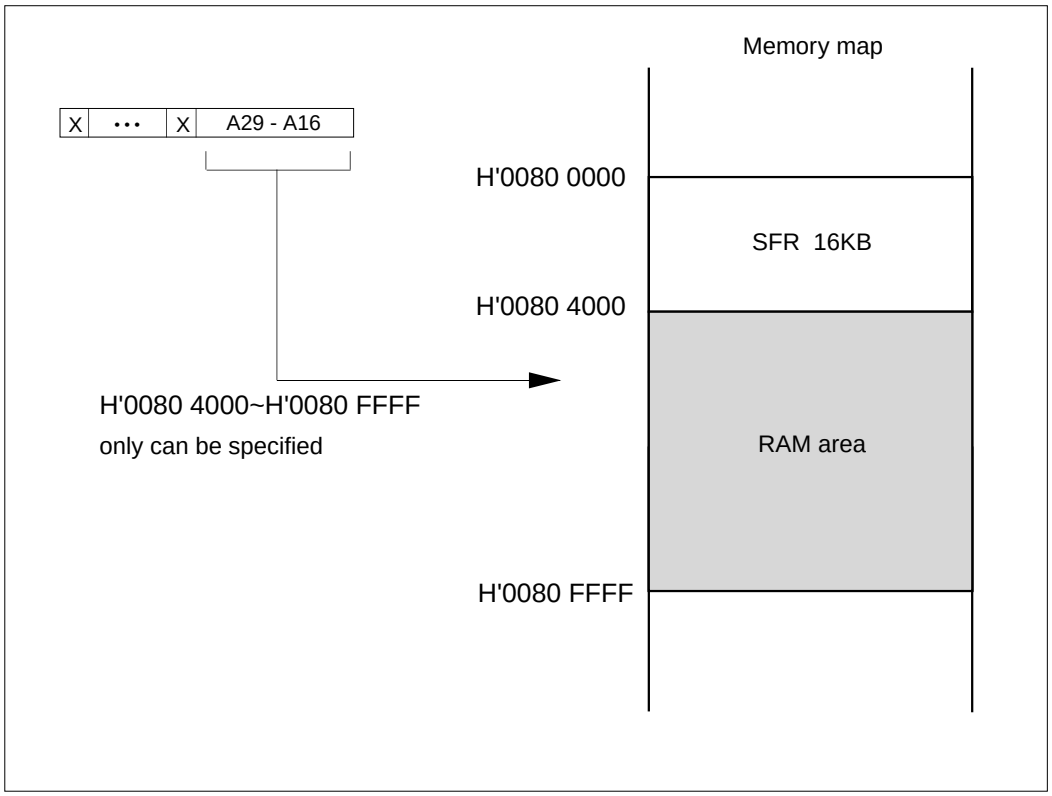


Figure 14.3.12 Method for Setting Addresses in Real-Time Debugger

14.3.8 Resetting the RTD

The RTD is reset by applying a system rest (i.e., by entering the RESET signal). The status of the RTD related output pins after a system reset are shown below.

Table 14.3.2 RTD Pin State after System Reset

Pin Name	State
RTDACK	High-level output
RTDTXD	High-level output

The first command transfer to the RTD after it was reset is initiated by transferring data to the RTDRXD pin synchronously with falling edges of RTDCLK.

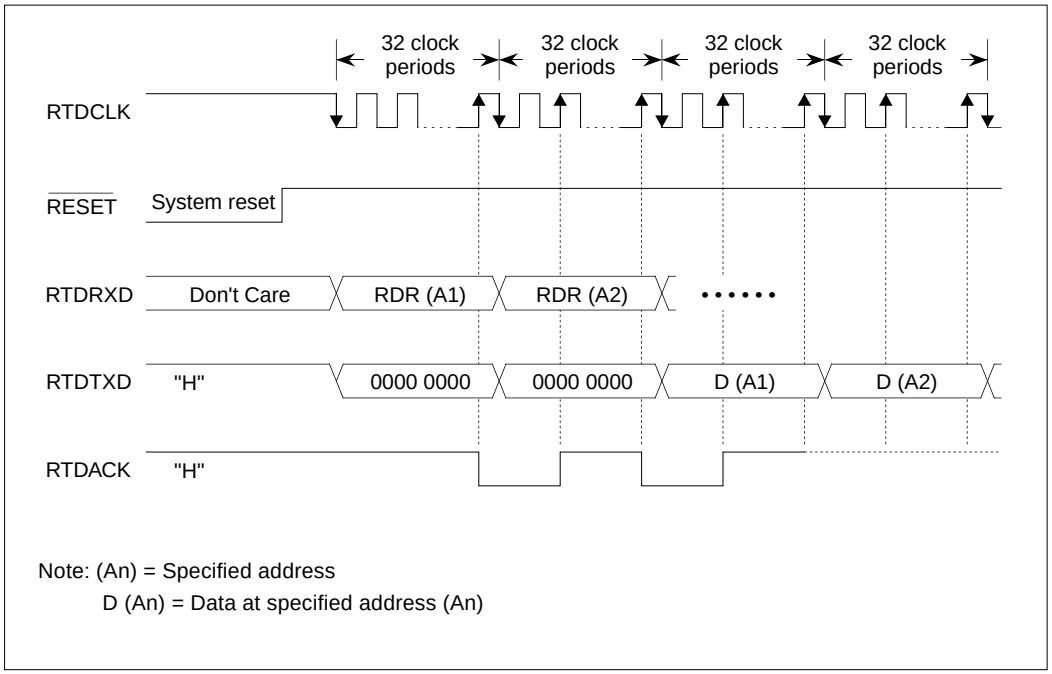


Figure 14.3.13 Command Transfer to the RTD after System Reset

14.4 Typical Connection with the Host

The host uses a serial synchronous interface to transfer data. The clock for synchronous is generated by the host. An example for connecting the RTD and host is shown below.

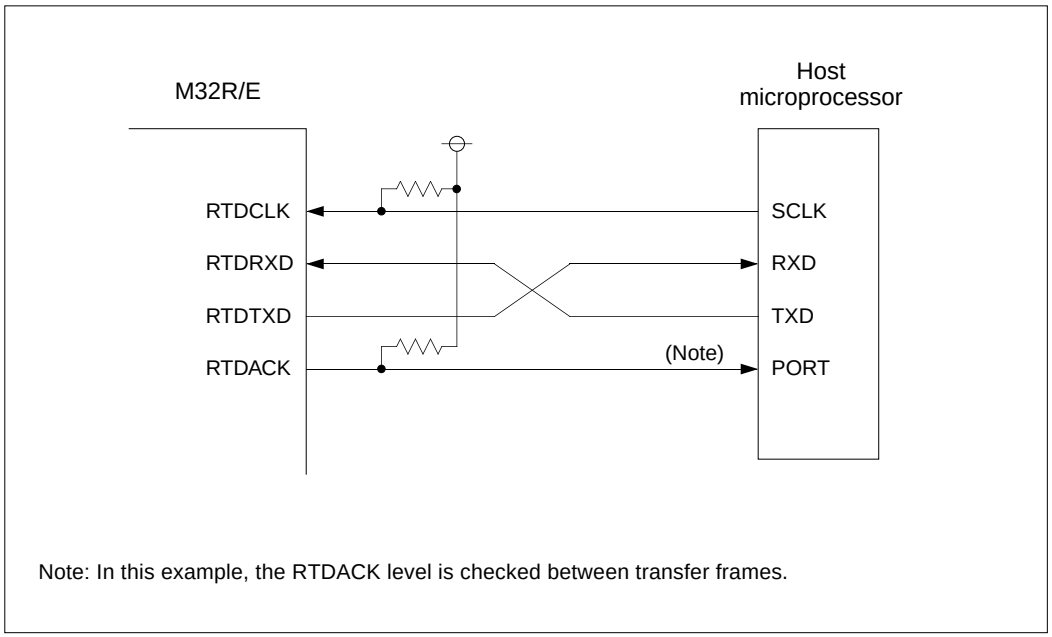


Figure 14.4.1 Connecting the RTD and Host

The RTD communication for a fixed length of 32 bits per frame generally is performed in four operations sending 8 bits at a time, because most serial interfaces transfer data in units of 8 bits. The RTDACK signal is used to verify that communication is performed normally.

After transmitting a command, the RTDACK signal is pulled low, making it possible to verify the communication status. When issuing the VER command, the RTDACK signal goes low for only one clock period. Therefore, after sending 32 bits in one frame, turn off RTDCLK output and check whether RTDACK is low. If RTDACK is low, you know that the RTD is communicating normally.

If you want to identify the type of transmitted command by the width of RTDACK, use the 32171's internal measurement timer (to count RTDCLK pulses while RTDACK is low) or create a dedicated circuit.

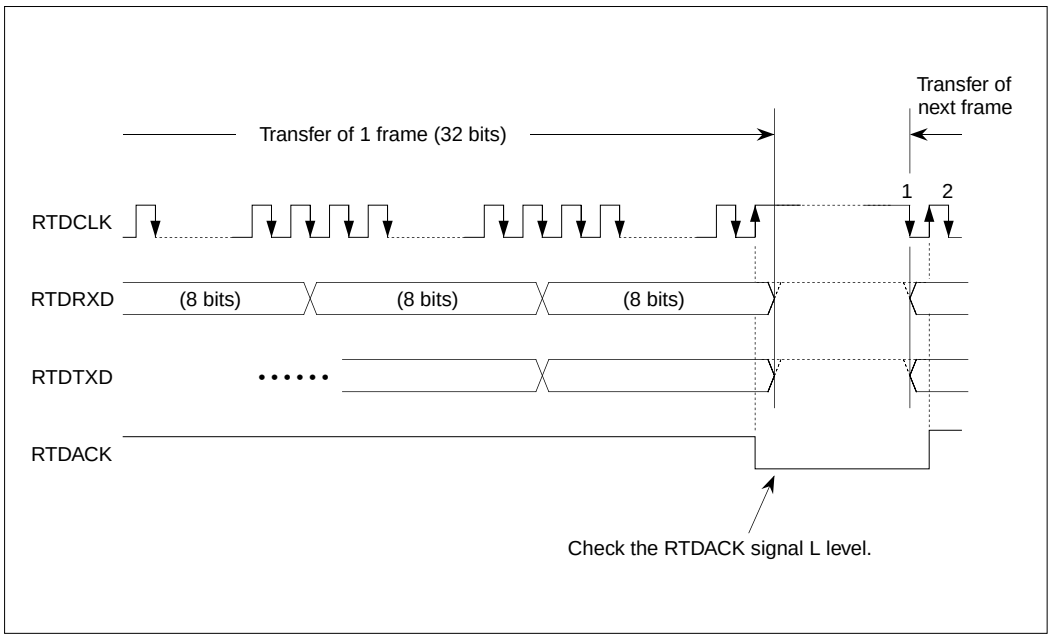


Figure 14.4.2 Typical Operation for Communication with the Host (when Issuing VER Command)

* This is a blank page.*

CHAPTER 15
PD MODULE

15.1 Outline of the PD Module

The 32172/32173 contains eight channels of event counters that can be used as a dedicated interface circuit for PD (Phase Digital) sensors. Combined with PD sensors, the PD module can perform predictive computations at high speed that are required during position detection.

Note: For the PD module to be used in combination with PD sensors, an external circuit is required separately.

Table 15.1.1 Outline of the PD Module

Item	Content
TPD	16-bit input measure timer (up-counter) One of two modes can be selected in software: • PD sensor support mode Operates corresponding to input from PD sensors. • Normal mode Operates as a free-running up-counter.
TEPiP TEPiM (i = 0,1)	16-bit input related timer (up-counter) One of two modes can be selected in software: • Event count mode Counts the number of pulses entered from an external pin • PD sensor support mode Operates corresponding to input from PD sensors.
PD calculation function	Performs various calculations corresponding to input from PD sensors and stores the result in a register (2 channels).

Table 15.1.2 PD Module Interrupt Generation Function

Signal name	PD module interrupt request source	Input to Interrupt Controller (ICU)	Sources on ICU
IRQ31	TIN0A input, TIN0B input, TIN1A input, TIN1B input, TIN0A error detection, TIN0B error detection, TIN1A error detection, TIN1B error detection	PDC input interrupt	8
IRQ2	PD0_ABD compare match, PD1_ABD compare match, PD0_PITCH compare match, PD1_PITCH compare match, PD0_ABD&PITCH compare match, PD1_ABD&PITCH compare match, PD0_S error, PD1_S error	PDC calculation interrupt	8

Table 15.1.3 PD Module DMA Transfer Request Generation Function

DMA transfer request generation function	DMA input channel
PD0_ABD compare match PD0_ABD&PITCH compare match (note)	Channels 1-9
PD1_ABD compare match PD1_ABD&PITCH compare match (note)	Channels 1-9
TIN0A input detection	Channels 0
TIN0B input detection	Channels 2
TIN1A input detection	Channels 5
TIN1B input detection	Channels 7

Note: Selected using the DMA Transfer Request Source Select Register (DMAREQSL).

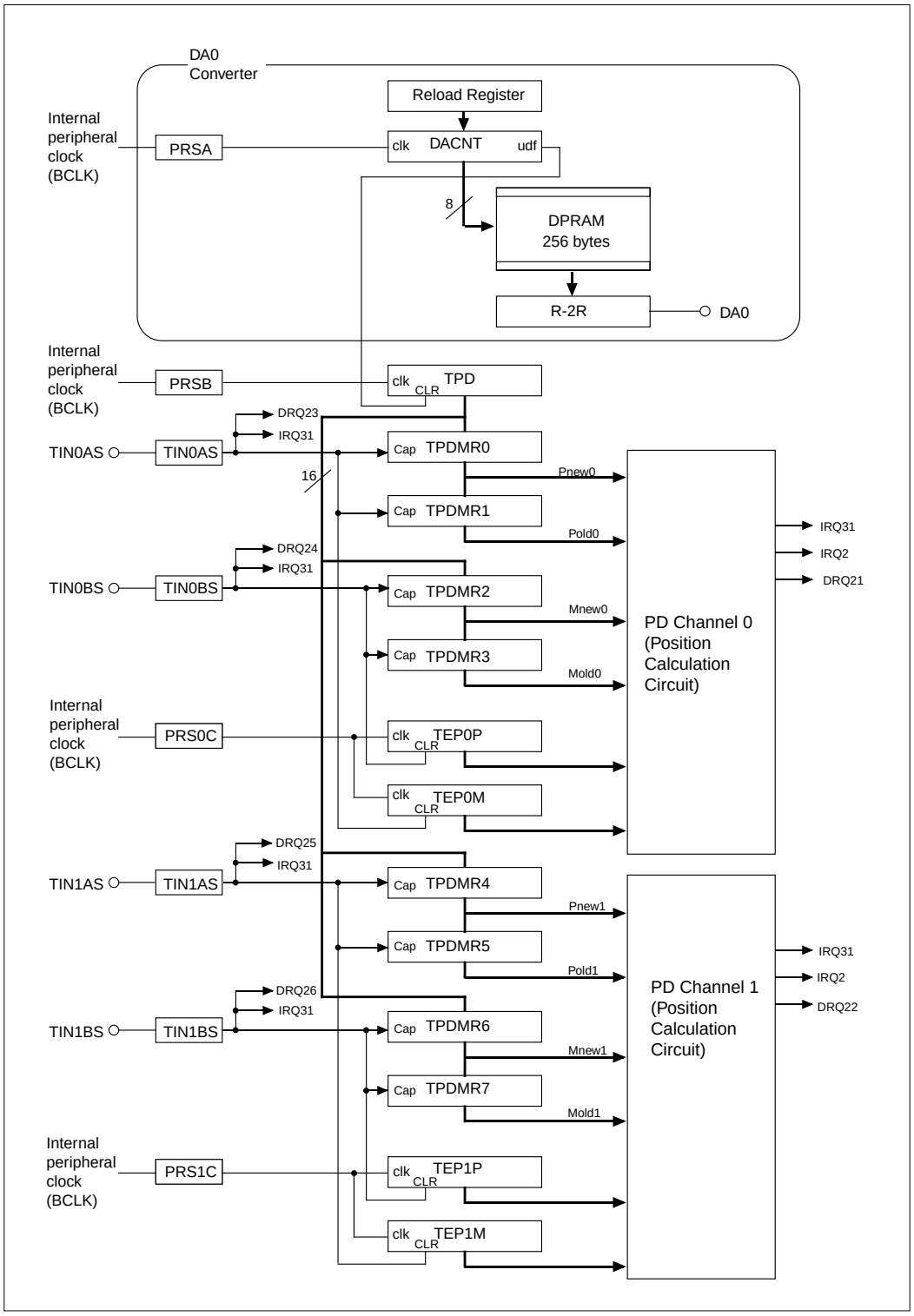


Figure 15.1.1 Block Diagram of the PD Module

15.2 PD Module Related Registers

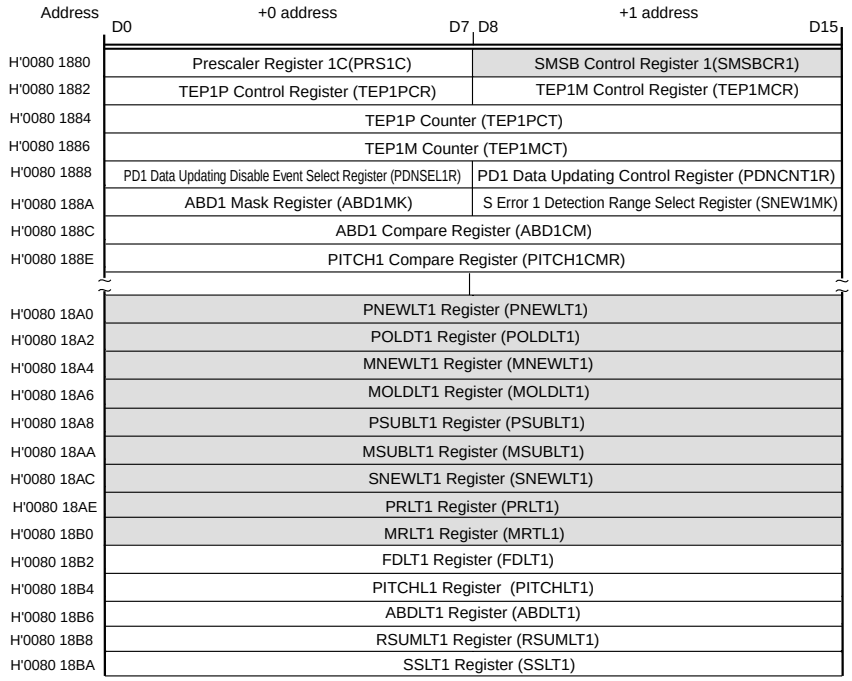
A PD module related register map is shown below.

Address	+0 address (D0 – D7)	+1 address (D8 – D15)
H'0080 1800	Prescaler Register A (PRSA)	Prescaler Register B (PRSB)
H'0080 1802	DACNT Reload Register (DACNTRL)	TIN Input Processing Control Register (TINPDCR)
H'0080 1804	TIN Interrupt Control Register (TINPDICR)	TIN Interrupt Status Register (TINPDIST)
H'0080 1806	DACNT Control Register (DACNTCR)	TPD Control Register (TPDCR)
H'0080 1808	DACNT Counter (DACNT)	
H'0080 180E	TPD Counter (TPDCT)	
H'0080 1810	TPD Measure Register 0 (TPDMR0)	
H'0080 1812	TPD Measure Register 1 (TPDMR1)	
H'0080 1814	TPD Measure Register 2 (TPDMR2)	
H'0080 1816	TPD Measure Register 3 (TPDMR3)	
H'0080 1818	TPD Measure Register 4 (TPDMR4)	
H'0080 181A	TPD Measure Register 5 (TPDMR5)	
H'0080 181C	TPD Measure Register 6 (TPDMR6)	
H'0080 181E	TPD Measure Register 7 (TPDMR7)	
H'0080 1830	PD Calculation Interrupt Control Register (PDICR)	PD Calculation Interrupt Status Register (PDIST)
H'0080 1832	Position Detection Accuracy Select Register (PDASR)	DMA Transfer Request Source Select Register (DMAREQSL)
H'0080 1840	Prescaler Register 0C (PRS0C)	SMSB Control Register 0 (SMSBCR0)
H'0080 1842	TEP0P Control Register (TEP0PCR)	TEP0M Control Register (TEP0MCR)
H'0080 1844	TEP0P Counter (TEP0PCT)	
H'0080 1846	TEP0M Counter (TEP0MCT)	
H'0080 1848	PD0 Data Updating Disable Event Select Register (PDNSEL0R)	PD0 Data Updating Control Register (PDNCNT0R)
H'0080184A	ABD0 Mask Register (ABD0MK)	S Error 0 Detection Range Select Register (SNEW0MK)
H'0080 184C	ABD0 Compare Register (ABD0CM)	
H'0080 184E	PICH0 Compare Register (PITCH0CMR)	
H'0080 1860	PNEWLT0 Register (PNEWLT0)	
H'0080 1862	POLDLT0 Register (POLDLT0)	
H'0080 1864	MNEWLT0 Register (MNEWLT0)	
H'0080 1866	MOLDLT0 Register (MOLDLT0)	
H'0080 1868	PSUBLT0 Register (PSUBLT0)	
H'0080 186A	MSUBLT0 Register (MSUBLT0)	
H'0080 186C	SNEWLT0 Register (SNEWLT0)	
H'0080 186E	PRLT0 Register (PRLT0)	
H'0080 1870	MRLT0 Register (MRLT0)	
H'0080 1872	FDLT0 Register (FDLT0)	
H'0080 1874	PITCHLT0 Register (PITCHLT0)	
H'0080 1876	ABDLT0 Register (ABDLT0)	
H'0080 1878	RSUMLT0 Register (RSUMLT0)	
H'0080 187A	SSLT0 Register (SSLT0)	

Blank areas are reserved for future use.

Note: The registers enclosed in [ ] are an intermediate register used for calculations.
Do not access these registers for read or write.

Figure 15.2.1 PD Module Register Map (1/2)



Blank areas are reserved for future use.

Note: The registers enclosed in [ ] are an intermediate register used for calculations.
Do not access these registers for read or write.

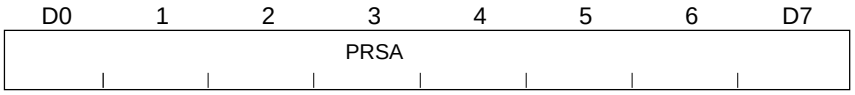
Figure 15.2.2 PD Module Register Map (2/2)

15.2.1 Prescaler Unit

The prescalers PRSA, PRSB, PRS0C, and PRS1C each consist of an 8-bit counter which generates clocks from the internal peripheral clock (20.0 MHz when the CPU clock = 40 MHz) for supply to each timer.
The prescaler value is initialized to H'00 when reset.
Also when the value of the prescaler register is rewritten, the prescaler starts operating with the new value simultaneously when the prescaler underflows.
An value in the range of H'00 to H'FF can be set in the prescaler register. The prescaler's divide ratio is given by the equation below.

$$\text{Prescaler divide ratio} = \frac{1}{\text{Prescaler set value} + 1}$$

■ Prescaler Register A (PRSA) **<Address: H'0080 1800>**



<When reset: H'00>

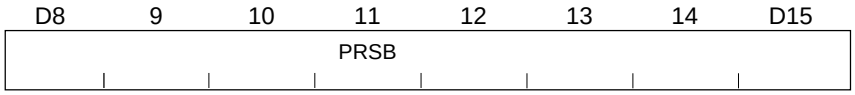
D	Bit Name	Function	R	W
0-7	PRSA	Sets the prescaler's divide ratio	○	○

The Prescaler A generates a count clock for the D-A converter's parameter table address counter (DACNT) from the internal peripheral clock (20.0 MHz when the CPU clock = 40 MHz) by dividing it by an appropriate value. When the value of Prescaler Register A is rewritten while the DACNT is operating, the prescaler starts operating with the newl value simultaneously when the prescaler underflows.
The PRSA starts generating a clock after the DACNT Control Register DACNTEN bit is set to 1.

DACNT count period = f / (PRSA + 1)

Note: When using the module along with PD sensors, the PRSA set value is subject to limitations depending on how the Position Detection Accuracy Select Register is set.

■ **Prescaler Register B (PRSB)** **<Address: H'0080 1801>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-15	PRSB	Sets the prescaler's divide-by value	○	○

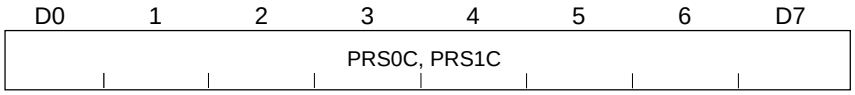
The Prescaler B generates a count clock for the 16-bit input measure counter (TPDCT) from the internal peripheral clock (20.0 MHz when the CPU clock = 40 MHz) by dividing it by an appropriate value.
When the value of Prescaler Register B is rewritten while the DACNT is operating, the prescaler starts operating with the new value simultaneously when the prescaler underflows.
The PRSB starts generating a clock after the TPD Control Register TPDEN bit is set to 1.

TPD count period = f / (PRSB + 1)

Note: When using the TPD Counter (TPDCT) along with PD sensors, make sure the set value of the PRSB is H'00.

■ **Prescaler Register 0C (PRS0C)** **<Address: H'0080 1840>**

■ **Prescaler Register 1C (PRS1C)** **<Address: H'0080 1880>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-7	PRS0C, PRS1C	Sets the prescaler's divide-by value	○	○

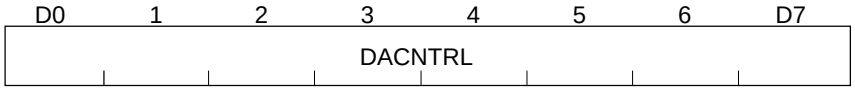
These prescalers generate count clocks for the TEPiP and TEPiM counters from the internal peripheral clock (20.0 MHz when the CPU clock = 40 MHz) by dividing it by an appropriate value. When the value of Prescaler Register 0C or 1C is rewritten while the TEPiP/TEPiM counter is operating, the prescaler starts operating with the new value simultaneously when the prescaler underflows.

Note 1: The PRS0C (PRS1C) starts generating a clock after the TEP0P (TEM1P) Control Register TEP0PEN (TEM1PEN) bit is set to 1.

Note 2: When using the TEPiP/TEPiM counters along with PD sensors, make sure the set values of the prescalers (PRS0C, PRS1C) are H'00.

15.2.2 DACNT Reload Register

■ **DACNT Reload Register (DACNTRL)** **<Address: H'0080 1802>**



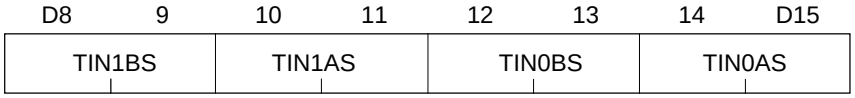
<When reset: H'00>

D	Bit Name	Function	R	W
0-7	DACNTRL	8-bit reload register value	○	○

The DACNT Reload Register is used to reload data into the DACNT Counter. Data is loaded into the DACNT Counter synchronously with the clock cycle in which the counter underflowed.
Simply because data is written to the DACNT Reload Register does not mean that the DACNT Counter is loaded with the data.

15.2.3 TIN Input Processing Control Register

■ **TIN Input Processing Control Register (TINPDCR)** **<Address: H'0080 1803>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-9	TIN1BS (Select TIN1BS input processing)	00: Input has no effect 01: Rising edge 10: Falling edge 11: Both edges	○	○
10-11	TIN1AS (Select TIN1AS input processing)	00: Input has no effect 01: Rising edge 10: Falling edge 11: Both edges	○	○
12-13	TIN0BS (Select TIN0BS input processing)	00: Input has no effect 01: Rising edge 10: Falling edge 11: Both edges	○	○
14-15	TIN0AS (Select TIN0AS input processing)	00: Input has no effect 01: Rising edge 10: Falling edge 11: Both edges	○	○

Use the TIN Input Processing Control Register to select the active edge of the TIN input signal at which to generate the measure, clear, and count source signals for each timer.

(1) Input has no effect

Input on the TIN pin has no effect, with no signals generated for each timer.

(2) Rising edge

Signals for each timer are generated upon detecting the rising edge of the TIN pin input signal.

(3) Falling edge

Signals for each timer are generated upon detecting the falling edge of the TIN pin input signal.

(4) Both edges

Signals for each timer are generated upon detecting the rising or falling edge of the TIN pin input signal.

Note: The pulse width of the TIN pin input signal must be equal to or greater than 3.5 clock periods of the internal peripheral clock.

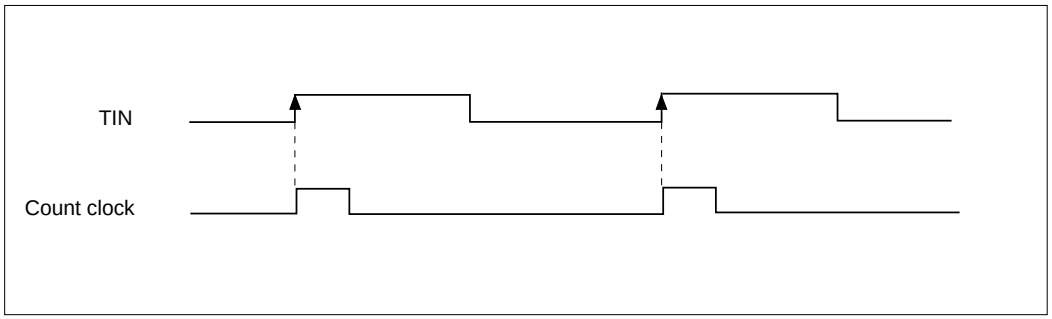


Figure 15.2.3 Rising Edge Detection

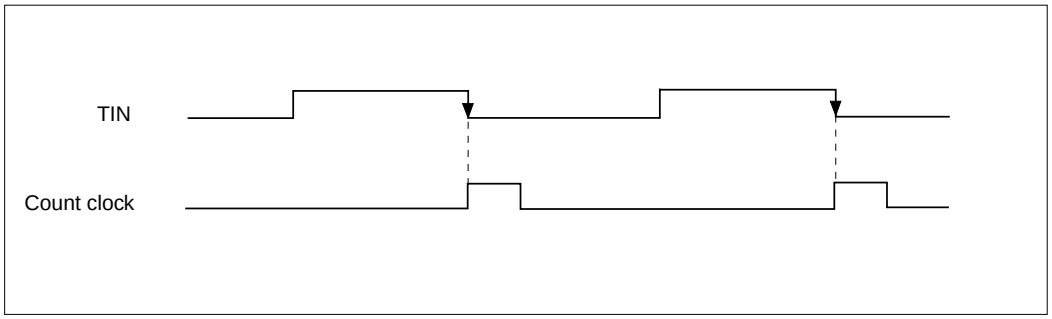


Figure 15.2.4 Falling Edge Detection

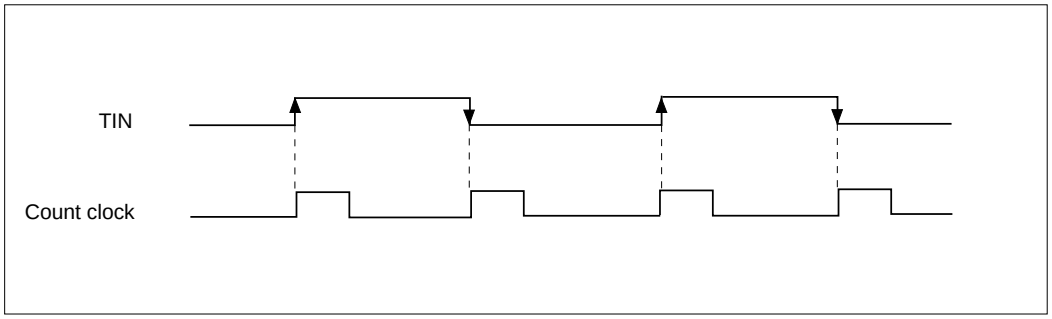


Figure 15.2.5 Both Edge Detection

15.2.4 TIN Interrupt Control Register

■ **TIN Interrupt Control Register (TINPDICR)** **<Address: H'0080 1804>**

D0	1	2	3	4	5	6	D7
TIN1BEIM	TIN1AEIM	TIN0BEIM	TIN0AEIM	TIN1BIM	TIN1AIM	TIN0BIM	TIN0AIM

<When reset: H'00>

D	Bit Name	Function	R	W
0	TIN1BEIM (Note)	0: Disables interrupt	○	○
1	TIN1AEIM (Note)	1: Enables interrupt		
2	TIN0BEIM (Note)			
3	TIN0AEIM (Note)			
4	TIN1BIM			
5	TIN1AIM			
6	TIN0BIM			
7	TIN0AIM			

This register controls disabling/enabling of interrupts for the interrupt requests generated by each TIN input processing circuit. Setting any bit to 1 in this register enables the corresponding TIN input interrupt request.

Note: The D0-D3 bits are provided for use with PD sensors. When not using the module along with PD sensors, be sure to write 0 to these bits.

15.2.5 TIN Interrupt Status Register

■ **TIN Interrupt Status Register (TINPDIST)** **<Address: H'0080 1805>**

D8	9	10	11	12	13	14	D15
TIN1BEIS	TIN1AEIS	TIN0BEIS	TIN0AEIS	TIN1BIS	TIN1AIS	TIN0BIS	TIN0AIS

<When reset: H'00>

D	Bit Name	Function	R	W
8	TIN1BEIS	0: Interrupt not requested	○	△
9	TIN1AEIS	1: Interrupt requested		
10	TIN0BEIS			
11	TIN0AEIS			
12	TIN1BIS			
13	TIN1AIS			
14	TIN0BIS			
15	TIN0AIS			

W=△ : Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

This register indicates whether there is an interrupt request for TIN input from any pin.

(1) TIN input detection error interrupt request bits (D8, D9, D10, D11)

These bits indicate interrupt requests from PD sensors.
* Detailed description being written now.

(2) TIN input detection interrupt request bits (D12, D13, D14, D15)

These bits are set to 1 when the active edge on the corresponding TIN input pin is detected, thereby indicating that an interrupt has been requested for that input.
These bits are cleared by writing 0 in software.

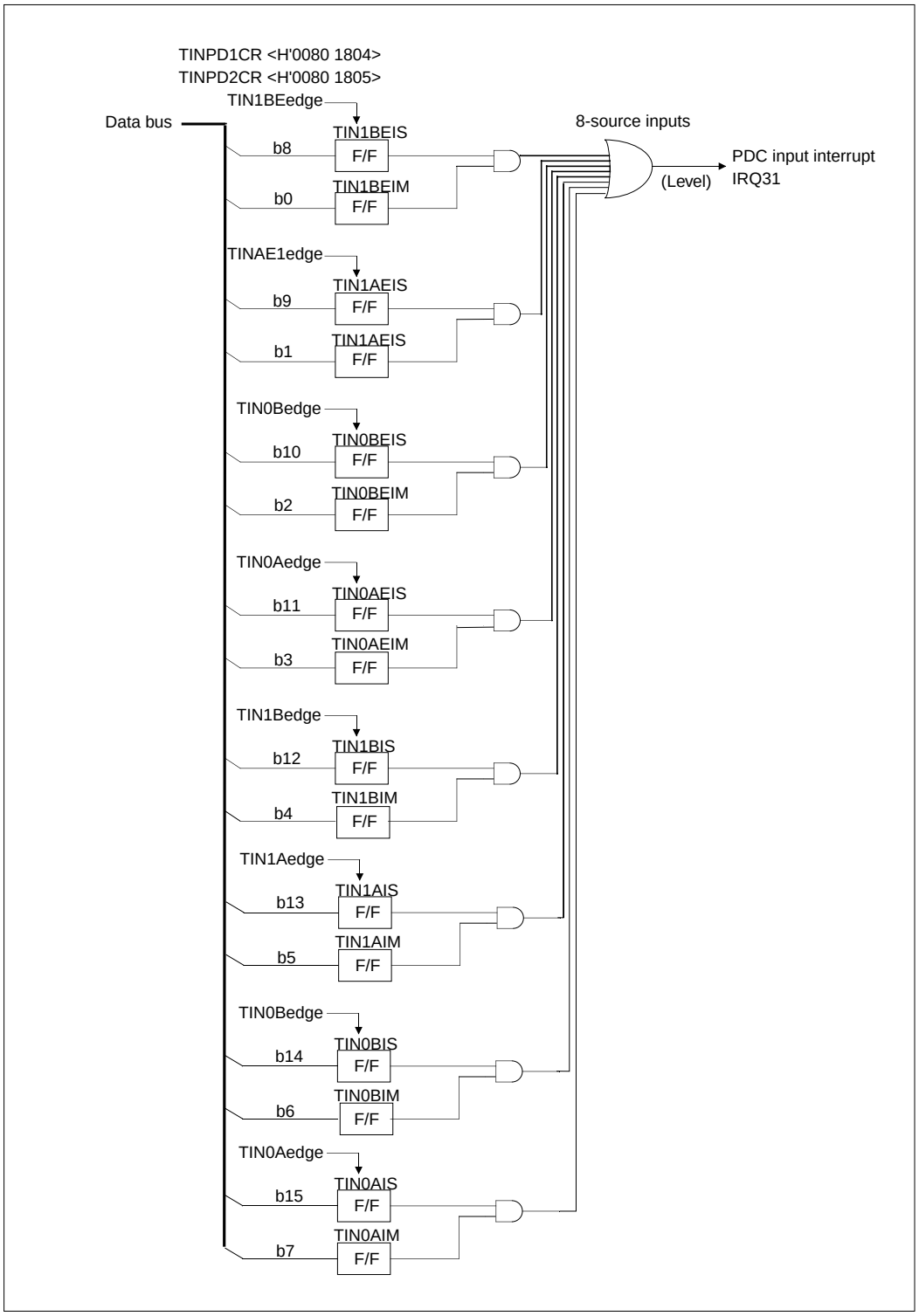
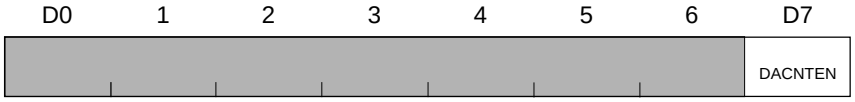


Figure 15.2.6 Block Diagram of the TIN Interrupt Circuit

15.2.6 DACNT Control Register

■ **DACNT Control Register (DACNTCR)** **<Address: H'0080 1806>**



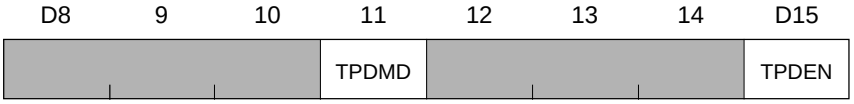
<When reset: H'00>

D	Bit Name	Function	R	W
0-6	No functions assigned		0	–
7	DACNTEN (DACNT count enable)	0: Stops count 1: Enables count	○	○

This register controls enabling/disabling of the DACNT Counter to or not to count.
When the DACNTEN bit is set to 1, the DACNT Counter starts counting down from its set value synchronously with the count period that has been set with the PRSA. Setting the DACNTEN bit to 0 disables the DACNT Counter, so that it stops counting, with the count value retained.

15.2.7 TPD Control Register

■ **TPD Control Register (TPDCR)** **<Address: H'0080 1807>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-10	No functions assigned		0	–
11	TPDMD (TPD counter operation mode)	0: Normal mode 1: PD sensor support mode	○	○
12-14	No functions assigned		0	–
15	TPDEN (TPD count enable)	0: Stops count 1: Enables count	○	○

This register selects operation modes of the TPD Counter and controls enabling/disabling of the counter.

(1) TPDMD bit (D11)

When this bit is set to 1, the TPD Counter (TPDCT) operates in PD sensor support mode.
When this bit is set to 0, the TPD Counter (TPDCT) operates as an up-counter in free-running mode.

Note 1: Rewriting this bit while the TPD Counter is operating is inhibited.

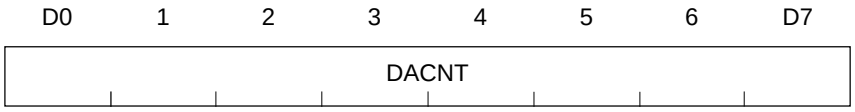
Note 2: When using the counter in PD sensor support mode, make sure the DACNT Control Register (DACNTCR)'s DACNTEN bit and this register's TPDEN bit are set to 1 simultaneously by accessing the registers in halfword.

(2) TPDEN bit (D15)

When this bit is set to 1, the TPD Counter starts counting up from its set value synchronously with the clock period that has been set with the PRSB.
Setting this bit to 0 disables the TPD Counter, so that it stops counting, with the count value retained.

15.2.8 DACNT Counter

■ **DACNT Counter (DACNT)** **<Address: H'0080 1808>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-7	DACNT	8-bit counter value	○	○

The DACNT Counter is an 8-bit down-counter which after being enabled, starts counting down from its set value synchronously with the count clock generated by the PRSA. The value of the DACNT Counter comprises the D-A converter's parameter table address value.
When the DACNT Counter underflows after reaching the minimum count (DACNT = H'00), it is reloaded with the content of the DACNT Reload Register and restarts counting down from the newly set value.

The diagram below shows operation of the DACNT Counter when the DACNT Counter and the DACNT Reload Register respectively have the values H'A0 and H'B0 set as their initial values.

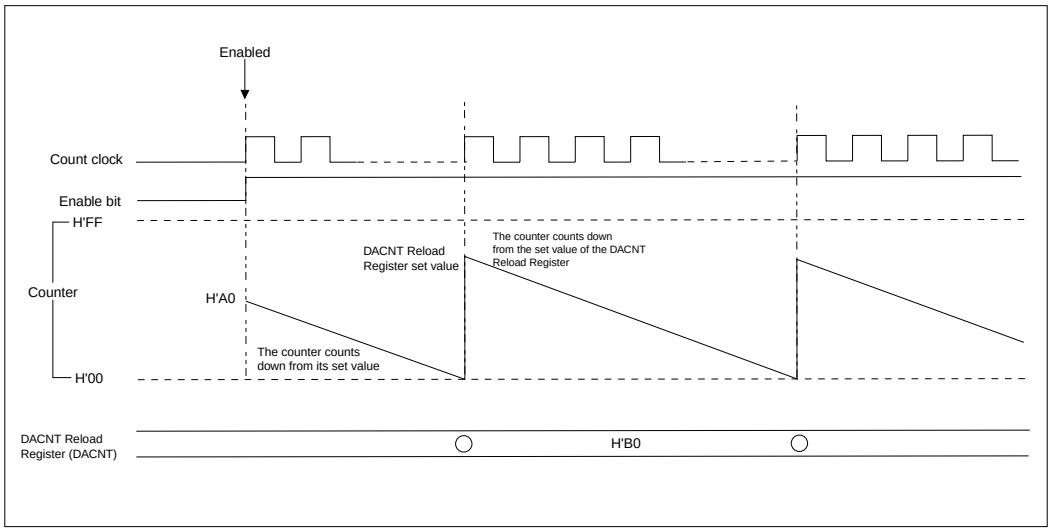
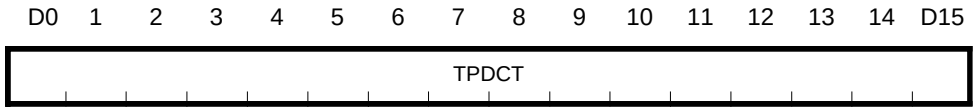


Figure 15.2.7 Example of DACNT Counter Operation

15.2.9 TPD Counter

■ **TPD Counter (TPDCT)** **<Address: H'0080 180E>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-15	TPDCT	16-bit counter value	○	○

The TPD Counter is a 16-bit up-counter which after being enabled, starts counting up from its set value synchronously with the count clock generated by the PRSB (free-running counter).
The TPD Counter has two operation modes that can be selected with the TPD Control Register TPDMD bit.
The following shows a typical operation of the TPD Counter in each mode.

(1) Normal mode

The TPD Counter operates as a free-running up-counter.
Shown below is an example of count operation of the TPD Counter when it is enabled after setting H'6000 in it as its initial value.

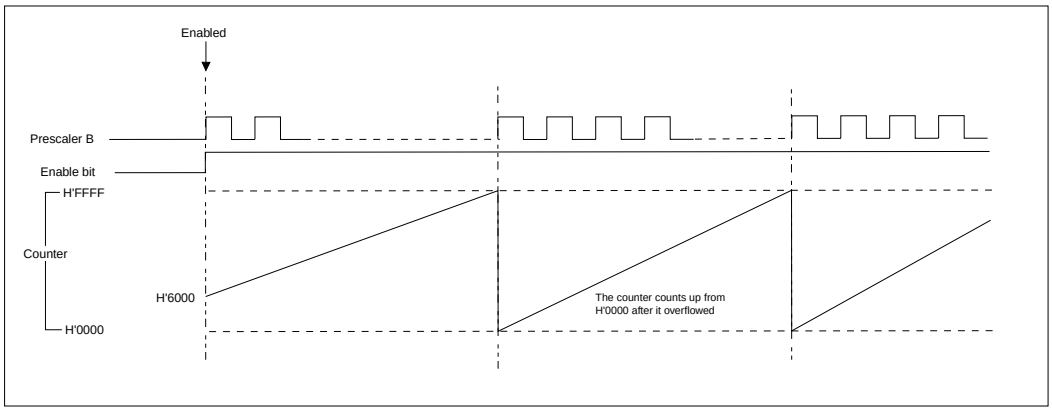


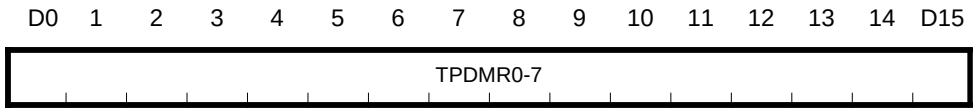
Figure 15.2.8 Typical Operation of the TPD Counter in Normal Mode

(2) PD sensor support mode

* Detailed description being written now.

15.2.10 TPD Measure Registers

■ **TPD Measure Register 0 (TPDMR0)** **<Address: H'0080 1810>**
■ **TPD Measure Register 1 (TPDMR1)** **<Address: H'0080 1812>**
■ **TPD Measure Register 2 (TPDMR2)** **<Address: H'0080 1814>**
■ **TPD Measure Register 3 (TPDMR3)** **<Address: H'0080 1816>**
■ **TPD Measure Register 4 (TPDMR4)** **<Address: H'0080 1818>**
■ **TPD Measure Register 5 (TPDMR5)** **<Address: H'0080 181A>**
■ **TPD Measure Register 6 (TPDMR6)** **<Address: H'0080 181C>**
■ **TPD Measure Register 7 (TPDMR7)** **<Address: H'0080 181E>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-15	TPDMR0-7	16-bit measured value	○	○

The TPD Measure Registers 0, 2, 4, and 6 are used to capture the count value, so that when an event input on any TIN pin is detected, the count value at that point in time is latched into the corresponding measure register.
The TPD Measure Registers 1, 3, 5, and 7 are used to capture the values of the TPD Measure Registers 0, 2, 4, and 6, so that when an event input on any TIN pin is detected, the register value at that point in time is latched into the corresponding measure register.

Note: Even when the TPD Control Register TPDEN bit = 0 (counting disabled), measure operation is performed should an event occur on any TIN input pin.

The following shows a typical operation of TPD measure inputs.

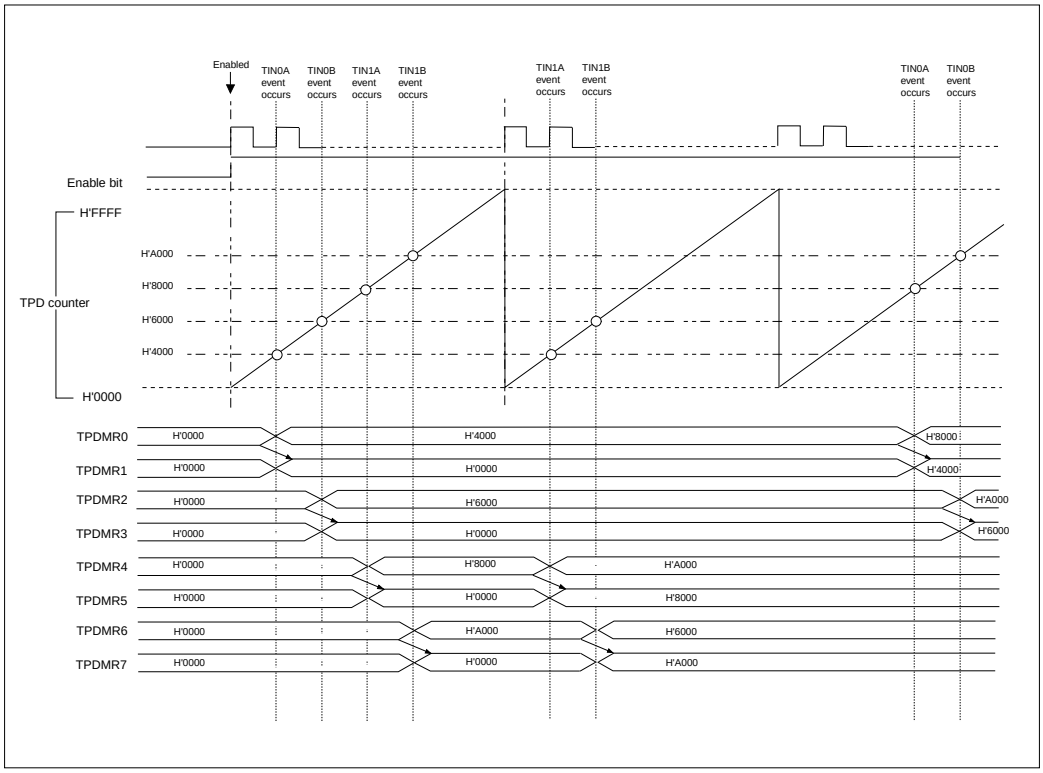


Figure 15.2.9 Example of TPD Operation for Measure Inputs

15.2.11 PD Calculation Interrupt Control Register

■ **PD Calculation Interrupt Control Register (PDICR)** **<Address: H'0080 1830>**

D0	1	2	3	4	5	6	D7
APCM1IM	SER1IM	PCM1IM	ACM1IM	APCM0IM	SER0IM	PCM0IM	ACM0IM

<When reset: H'00>

D	Bit Name	Function	R	W
0	APCM1IM	0: Disables interrupt	○	○
1	SER1IM	1: Enables interrupt		
2	PCM1IM			
3	ACM1IM			
4	APCM0IM			
5	SER0IM			
6	PCM0IM			
7	ACM0IM			

This register is used to control enabling/disabling of PD calculation interrupts.
When any bit in this register is set to 1, the corresponding interrupt of the PD Interrupt Status Register is enabled.

15.2.12 PD Calculation Interrupt Status Register

■ **PD Calculation Interrupt Status Register (PDIST)** **<Address: H'0080 1831>**

D8	9	10	11	12	13	14	D15
APCM1IS	SER1IS	PCM1IS	ACM1IS	APCM0IS	SER0IS	PCM0IS	ACM0IS

<When reset: H'00>

D	Bit Name	Function	R	W
8	APCM1IS	0: Interrupt not requested	○	△
9	SER1IS	1: Interrupt requested		
10	PCM1IS			
11	ACM1IS			
12	APCM0IS			
13	SER0IS			
14	PCM0IS			
15	ACM0IS			

W=△ : Only writing 0 is effective. Writing 1 has no effect, the bit retains the value it had before writing.

When using PD calculation interrupts, inspect this register to know which interrupt request has been generated.

(1) APCMiIS

This bit is set to 1 when the ABDi value and the ABDi Compare Register's set value matched and the PITCHi value and the PITCHi Compare Register's set value also matched.
This bit is cleared by writing 0.

(2) SERiIS

This bit is set to 1 when an S error occurred.
* Detailed description about S errors being written now.

(3) PCMiIS

This bit is set to 1 when the PITCHi value and the PITCHi Compare Register's set value matched. For details about the PITCH compare operation, see the section of this manual where the PITCHi Compare Registers are discussed.
This bit is cleared by writing 0.

(4) ACMiIS

This bit is set to 1 when the ABDi value and the ABDi Compare Register's set value matched. For details about the ABD compare operation, see the section of this manual where the ABDi Compare Registers are discussed.
This bit is cleared by writing 0.

Note: The mask function may be used for determining whether ABDi comparison matched in (1) and (4) above. For details, see the section of this manual where the ABD Mask Registers are discussed.

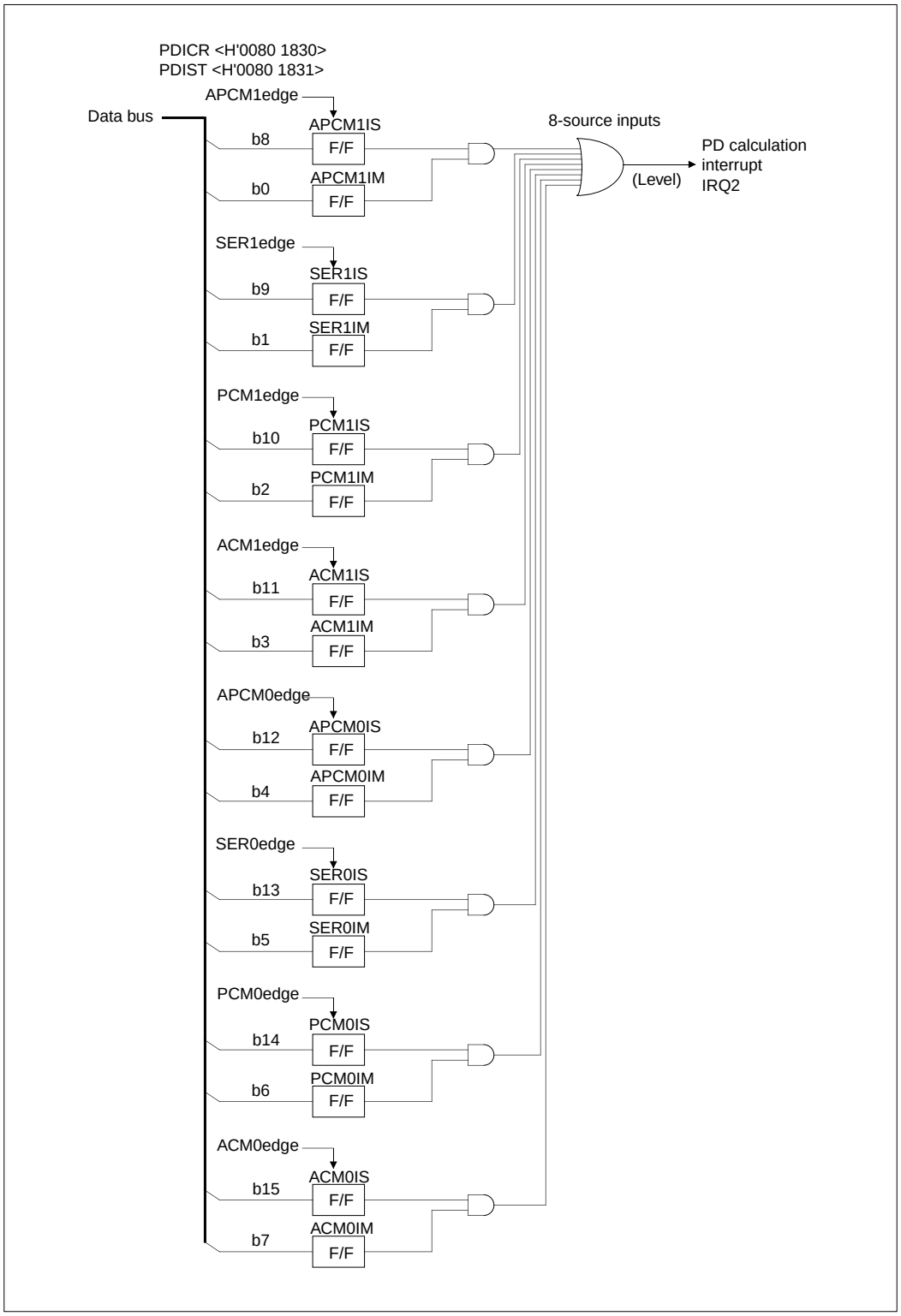
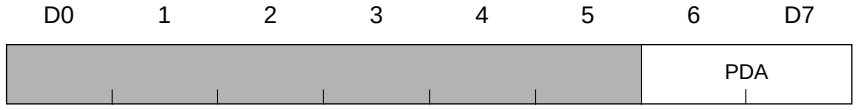


Figure 15.2.10 Block Diagram of PD Circuit Calculation Complete Interrupt

15.2.13 Position Detection Accuracy Select Register

■ **Position Detection Accuracy Select Register (PDASR)** **<Address: H'0080 1832>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6-7	PDA (Select position detection accuracy)	2'b00: 10-bit accuracy 2'b01: 11-bit accuracy 2'b10: 12-bit accuracy 2'b11: Use inhibited	○	○

This register is used when operating the module in combination with PD sensors.
The PDA bits select the accuracy of position detection. The values set in the Prescalers A and B are determined by the selected accuracy.
The table below shows the relationship between these set values and the sine waves generated by D-A conversion.

When fcpu = 40 MHz

Sine wave	D-A conversion address count frequency	D-A conversion data table usage range	Base count range (PRS = H'00)	Resolution
19.5KHz	5.0MHz (PRS = H'03)	H'00 – H'FF (256 data)	H'0000 – H'03FF	10bits
9.77KHz	2.5MHz (PRS = H'07)	H'00 – H'FF (256 data)	H'0000 – H'07FF	11bits
4.88KHz	1.25MHz (PRS = H'0F)	H'00 – H'FF (256 data)	H'0000 – H'0FFF	12bits

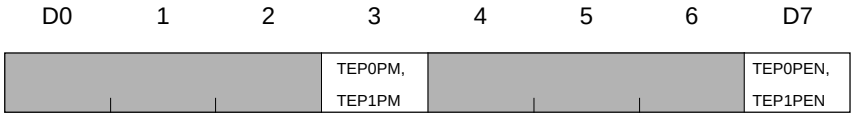
When fcpu = 32 MHz

Sine wave	D-A conversion address count frequency	D-A conversion data table usage range	Base count range (PRS = H'00)	Resolution
15.6KHz	4.0MHz (PRS = H'03)	H'00 – H'FF (256 data)	H'0000 – H'03FF	10bits
7.81KHz	2.0MHz (PRS = H'07)	H'00 – H'FF (256 data)	H'0000 – H'07FF	11bits
3.91KHz	1.0MHz (PRS = H'0F)	H'00 – H'FF (256 data)	H'0000 – H'0FFF	12bits

15.2.14 TEP Control Registers

■ **TEP0P Control Register (TEP0CR)** **<Address: H'0080 1842>**

■ **TEP1P Control Register (TEP1CR)** **<Address: H'0080 1882>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-2	No functions assigned		0	–
3	TEP0PM, TEP1PM (Operation mode)	0: Event count mode 1: PD sensor support mode	○	○
4-6	No functions assigned		0	–
7	TEP0PEN, TEP1PEN (Count enable)	0: Disables count 1: Enables count	○	○

These registers are used to select TEPiP counter operation modes and control enabling/disabling of count.

(1) TEPiPM bit

Setting this bit to 0 selects event count mode, in which the TEPiP Counter operates as an event counter (up-counter) counting events on the corresponding TIN input pin.
Setting this bit to 1 selects PD sensor support mode, in which the TEPiP Counter operates corresponding to input from PD sensors.

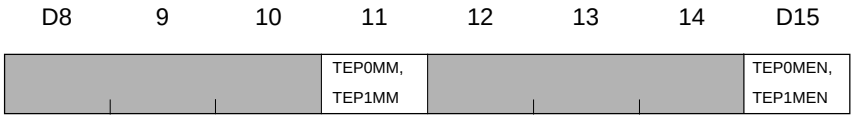
Note: Rewriting this bit while the TEPiP Counter is operating is inhibited.

(2) TEPiPEN bit

Setting this bit to 1 enables the TEPiP Counter for counting. When in event count mode, the counter counts up on each TIN event input starting from its set value; when in PD sensor support mode, the counter counts up synchronously with the clock generated by the prescaler (PRS0C or PRS1C).
Setting this bit to 0 disables the TEPiP Counter, so that it stops counting, with the count value retained.

■ **TEP0M Control Register (TEP0MCR)** **<Address: H'0080 1843>**

■ **TEP1M Control Register (TEP1MCR)** **<Address: H'0080 1883>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-10	No functions assigned		0	–
11	TEP0MM, TEP1MM (Operation mode)	0: Event count mode 1: PD sensor support mode	○	○
12-14	No functions assigned		0	–
15	TEP0MEN, TEP1MEN (Count enable)	0: Disables count 1: Enables count	○	○

These registers are used to select TEPiM counter operation modes and control enabling/disabling of count.

(1) TEPiMM bit

Setting this bit to 0 selects event count mode, in which the TEPiM Counter operates as an event counter (up-counter) counting events on the corresponding TIN input pin.
Setting this bit to 1 selects PD sensor support mode, in which the TEPiM Counter operates corresponding to input from PD sensors.

Note: Rewriting this bit while the TEPiM Counter is operating is inhibited.

(2) TEPiMEN bit

Setting this bit to 1 enables the TEPiM Counter for counting. When in event count mode, the counter counts up on each TIN event input starting from its set value; when in PD sensor support mode, the counter counts up synchronously with the clock generated by the prescaler (PRS0C or PRS1C).
Setting this bit to 0 disables the TEPiM Counter, so that it stops counting, with the count value retained.

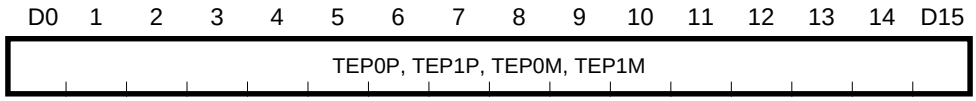
15.2.15 TEP Counters

■ **TEP0P Counter (TEP0PCT)** **<Address: H'0080 1844>**

■ **TEP1P Counter (TEP1PCT)** **<Address: H'0080 1884>**

■ **TEP0M Counter (TEP0MCT)** **<Address: H'0080 1846>**

■ **TEP1M Counter (TEP1MCT)** **<Address: H'0080 1886>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-15	TEP0P TEP1P TEP0M TEP1M	16-bit counter value	○	○

The TEPiP/TEPiM Counters are a 16-bit up-counter.
The TEPiP/TEPiM Counters have two operations that can be selected with the TEPiP/TEPiM Control Register TEPiPEN/TEPiMEN bit. A typical operation of the TEPiP/TEPiM Counters in each mode is shown.

(1) Event count mode

The counter counts up each time an event input on the corresponding TIN pin is detected.

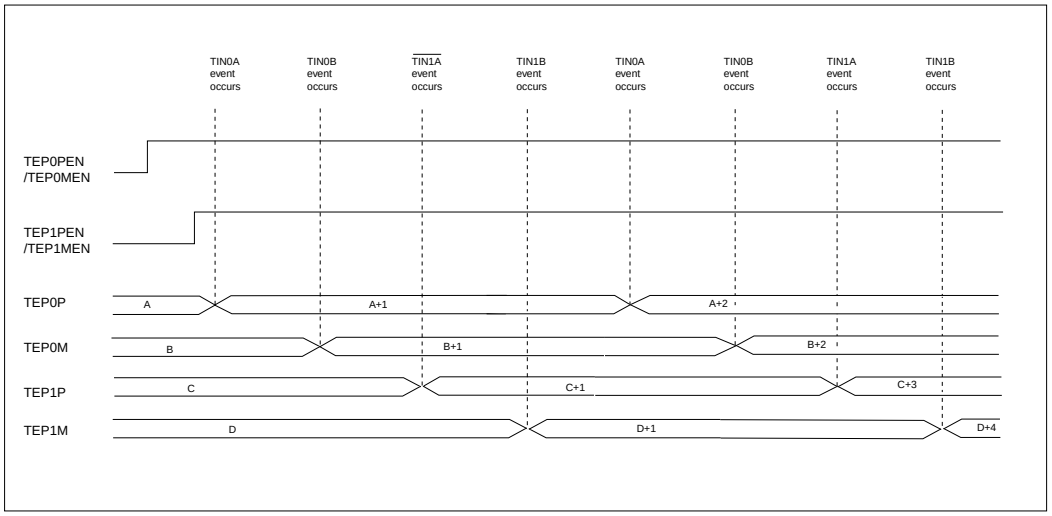


Figure 15.2.11 Event Count Mode

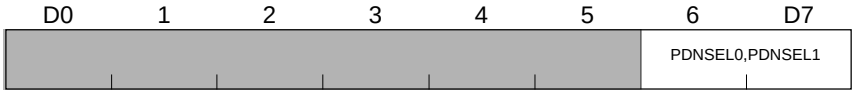
(2) PD sensor support mode

*Detailed description being written now.

15.2.16 PD Data Updating Disable Event Select Registers

■ **PD0 Data Updating Disable Event Select Register (PDNSEL0R) <Address: H'0080 1848>**

■ **PD1 Data Updating Disable Event Select Register (PDNSEL1R) <Address: H'0080 1888>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6-7	PDNSEL0, PDNSEL1 (Select data updating disable event)	00: No selection 01: TOM0_6 timer event 10: TIN16 event input 11: Settings inhibited	○	○

The PD calculation block always performs calculation based on the latest measured value, but the register in which the calculation result is stored can have its data updated under software control. This register is used to select an event that disables updating of said register data. Consequently, the data stored in the register at the time the selected event occurred can be retained until updating is reenabled. Updating of the register data is enabled by setting the PDi Data Updating Control Register directly in software.

(1) No selection (PDNSELi = H'00)

When data updating all needs to be enabled/disabled in software, set the PDNSELi bits to '00.'

(2) TOM0_6 timer event (PDNSELi = H'01)

Data updating is disabled by occurrence of TOM0_6 timer event.

(3) TIN16 event input (PDNSELi = H'10)

Data updating is disabled by occurrence of TIN16 event input.

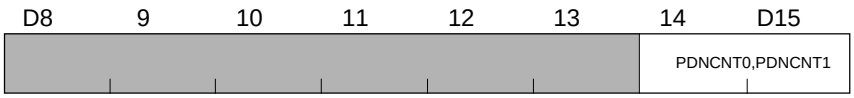
(4) Settings inhibited (PDNSELi = H'11)

Setting the PDNSELi bits to '11' is inhibited.

15.2.17 PD Data Updating Control Registers

■ **PD0 Data Updating Control Register (PDNCNT0R)** **<Address: H'0080 1849>**

■ **PD1 Data Updating Control Register (PDNCNT1R)** **<Address: H'0080 1889>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-13	No functions assigned		0	–
14-15	PDNCNT0, PDNCNT1 (Data updating enable)	0: Enables data updating 1: Disables data updating	○	○

This register controls updating of the data stored in the PD calculation result register.

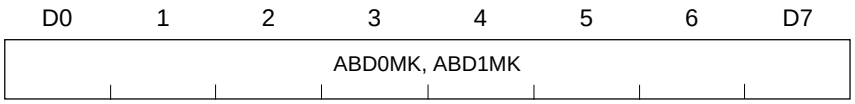
[Set conditions]

- The bits in this register can be set by writing 1 in software. This is always possible regardless of how the PDi Data Updating Disable Event Select Register is set.
- When TOM0_6 timer event has been selected with the PDi Data Updating Disable Event Select Register, the bits in this register are set to 1 by occurrence of TOM0_6 timer event. If this timer event occurs at the same time the PDNCNTi bits are cleared to 0 in software, the former has priority (the bit is set).
- When external input has been selected with the PDi Data Updating Disable Event Select Register, the bits in this register are set to 1 by occurrence of external input event. If this external input event occurs at the same time the PDNCNTi bits are cleared to 0 in software, the former has priority (the bit is set).

15.2.18 ABD Mask Registers

■ **ABD0 Mask Register (ABD0MK)** **<Address: H'0080 184A>**

■ **ABD1 Mask Register (ABD1MK)** **<Address: H'0080 188A>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-7	ABD0MK, ABD1MK (ABD compare match mask)	0: Mask 1: Compare	○	○

When comparing the ABDiLT Register and ABDi Compare Register values to determine whether they match, this register may be used to mask the low-order bits of the comparison result. When any bit in this register is set to 0, its corresponding bit of the comparison result is handled as "Don't care."
The diagram below shows the relationship of bits between the ABDi Mask Register and the ABDiLT Register and ABDi Compare Register.

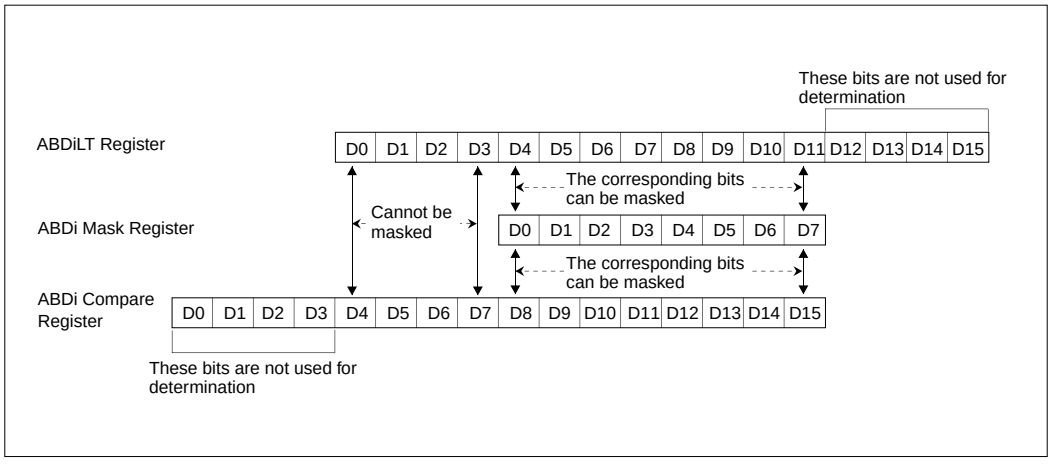
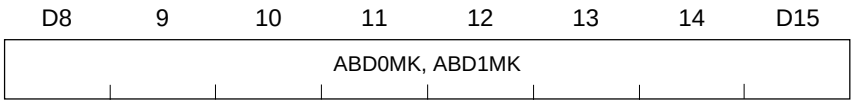


Figure 15.2.12 Data Formats when Comparing Register Values to Determine Matching

15.2.19 S Error Detection Range Select Registers

■ **S Error 0 Detection Range Select Register (SNEW0MK)** **<Address: H'0080 184B>**

■ **S Error 1 Detection Range Select Register (SNEW1MK)** **<Address: H'0080 188B>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-15	SNEW0MK SNEW1MK (Select S error detection range)	Settings Detection range (Note) b'00000000 : SS≧256, SS≦-257 b'10000000 : SS≧128, SS≦-129 b'11000000 : SS≧64, SS≦-65 b'11100000 : SS≧32, SS≦-33 b'11110000 : SS≧16, SS≦-17 b'11111000 : SS≧8, SS≦-9 b'11111100 : SS≧4, SS≦-5 b'11111110 : SS≧2, SS≦-3 b'11111111 : SS≧1, SS≦-2	○	○

Note: The detection ranges shown here assume the SSLT Register D11 bit is the least significant bit.

These registers are used to set the S error detection range.

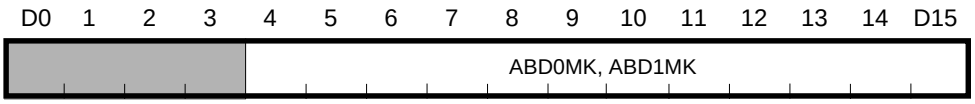
- About the definition of S error

  * Detailed description being written now.

15.2.20 ABD Compare Registers

■ **ABD0 Compare Register (ABD0CM)** **<Address: H'0080 184C>**

■ **ABD1 Compare Register (ABD1CM)** **<Address: H'0080 188C>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-3	No functions assigned		0	–
4-5	ABD0CM, ABD1CM	12-bit compare value	○	○

The value set in this register is compared with the ABDiLT Register value and when they match, an ABD compare match interrupt request is set. At this time, the low-order bits of the comparison result can be masked out as "Don't care" bits by using the ABDi Mask Register.

Note 1: When setting the ABDiCM Register, always be sure to set the bits in this register to 0 that correspond to those which are set to 0 in the ABDi Mask Register.

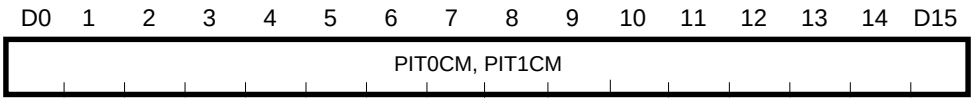
Note 2: ABD compare match processing is always performed based on the latest measured value, regardless of how the PDi Data Updating Control Register is set.

Note 3: Pay attention to the relationship of bits between the ABDi Mask Register and ABDiLT Register. (For details, see the description of the ABD Mask Register.)

15.2.21 PITCH Compare Registers

■ **PITCH0 Compare Register (PITCH0CMR)** **<Address: H'0080 184E>**

■ **PITCH1 Compare Register (PITCH1CMR)** **<Address: H'0080 188E>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-15	PIT0CM, PIT1CM	16-bit compare value	○	○

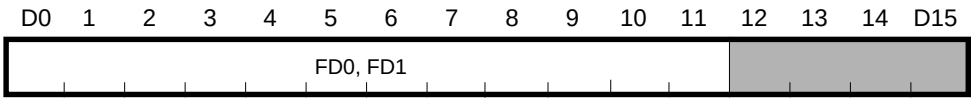
The value set in this register is compared with the PITCHi Counter Register value and when they match, a PITCH compare match interrupt request is set.

Note: PITCH compare match processing is always performed based on the latest measured value, regardless of how the PDi Data Updating Control Register is set.

15.2.22 FDLT Registers

■ **FDLT0 Register (FDLT0)** **<Address: H'0080 1872>**

■ **FDLT1 Register (FDLT1)** **<Address: H'0080 18B2>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-11	FD0, FD1	12-bit FD value	○	–
12-15	No functions assigned		0	–

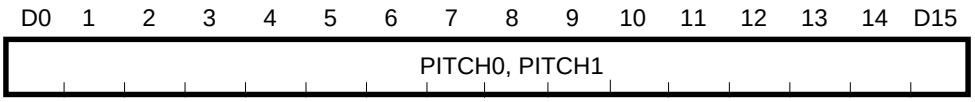
This register is used to store the value equivalent to a change of position.

Note: To read data from this register, access it as signed halfword data.

15.2.23 PITCHLT Registers

■ **PITCHLT0 Register (PITCHLT0)** **<Address: H'0080 1874>**

■ **PITCHLT1 Register (PITCHLT1)** **<Address: H'0080 18B4>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-15	PITCH0, PITCH1	16-bit PITCH counter value	○	○

Note 1: This register must always be accessed in halfwords.

Note 2: When performing predictive calculations, access the PITCHLT and ABDLT Registers wordwise to read out data by using an LD instruction.

The PITCH counter value is stored in this register.

- PITCH Counter

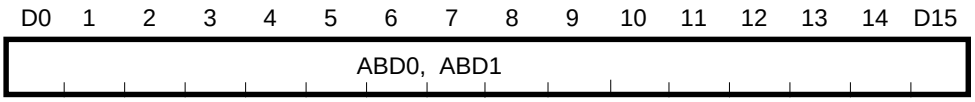
  The PITCHi Counter is an up/down-counter whose count direction is determined by comparing the two high-order bits of the latest and the previous ABDiLTD Register values. The following shows how the up/down direction actually is determined.

New {AB0, AB1}	Old {AB0, AB1}	Count value
LL	HH	+1
HH	LL	–1

15.2.24 ABDLT Registers

■ **ABDLT0 Register (ABDLT0)** **<Address: H'0080 1876>**

■ **ABDLT1 Register (ABDLT1)** **<Address: H'0080 18B6>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-15	ABD0, ABD1	12-bit ABD value	○	–

Note 1: This register must always be accessed in halfwords.

Note 2: When performing predictive calculations, access the PITCHLT and ABDLT Registers wordwise to read out data by using an LD instruction.

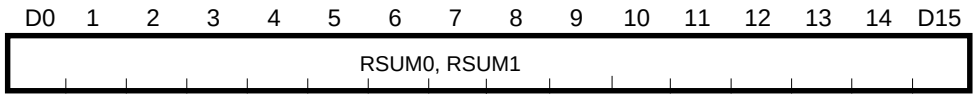
The measured position information is stored in the D0-D11 bits of this register. The D12-D15 bits are always 0 when read out.

However, it is the D11-D15 bits when 11-bit accuracy is selected or the D10-D15 bits when 10-bit accuracy is selected that are always 0 when read out.

15.2.25 RSUMLT Registers

■ **RSUMLT0 Register (RSUMLT0)** **<Address: H'0080 1878>**

■ **RSUMLT1 Register (RSUMLT1)** **<Address: H'0080 18B8>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-15	RSUM0, RSUM1	16-bit correction factor	○	–

Note: This register must always be accessed as signed halfword data

This register is used to store the correction factor necessary for predictive calculations. The predictive position I is calculated using the value in this register by the equation below.

$$I = \{ABDLT, ABDLT\} + (FDLT \times RSUMLT) / T$$

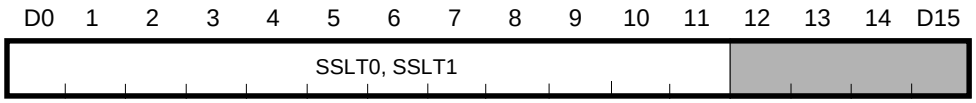
Note 1: T = H'1000.

Note 2: Access the PITCHLT and ABDLT Registers wordwise to read out data by using an LD instruction.

15.2.26 SSLT Registers

■ **SSLT0 Register (SSLT0)** **<Address: H'0080 187A>**

■ **SSLT1 Register (SSLT1)** **<Address: H'0080 18BA>**



<When reset: H'0000>

D	Bit Name	Function	R	W
0-11	SSLT0, SSLT1	12-bit SSLT value	○	–
12-15	No functions assigned		0	–

Note: This register must always be accessed in halfwords.

This register is used to store the latest value of the SSLT data used for S error judgment. The four low-order bits are 0 when read out.

15.3 Initialization for PD Sensor Support

When using the PD module in combination with PD sensors, follow the initialization procedure described below to initialize it.

(1) Setting the Input/output Port Operation Mode Register

The TINA0, TENB0, TINA1, and TENB1 pins are dual-function pins shared with input/output ports each. Therefore, select the function of these pins using the said register.

(2) Setting the D-A Converter, Parameter Table, and D-A Control Register

To generate sine waves, set the data of the D-A Converter Parameter Table (0-255) in the DA0 Data Registers 0-255. Then set DA0 for continuous mode and enable it for output using the D-A Control Register for the D-A Converter.

Note 1: Use all of the D-A Converter Parameter Table (0-255) to define the sine wave.
Note 2: For details on how to set the D-A Control Register, see Chapter 16, "D-A Converters."

(3) Setting the Position Detection Accuracy Select Register

Select the accuracy of position detection from 10-bit through 12-bit resolutions.

(4) Setting the Prescalers A and B

Use the Prescaler A to set the address count period for the DA0 Data Register. The value to be set in the Prescaler A depends on the selected accuracy of position detection, so set the appropriate value as shown below.

- When 10-bit accuracy is selected: H'03
- When 11-bit accuracy is selected: H'07
- When 12-bit accuracy is selected: H'0F

For each of these settings, the sine wave period Fsin is given by the equation below.

$$Fsin = Fcpu/ \{2 \times (1+PRSA) \times 256\}$$ (Fcpu: CPU clock frequency)

Use the Prescaler B to set the count period of the measure counter TPDCT. Always be sure to set the value H'00 in this register. In this case, the TPDCT count period is Fcpu/2.

(5) Setting the DACNT Reload Register

Because all of the DA0 Data Registers 0-255 must be used to generate sine waves, set the value H'FF in this register. Each time the DACNT counter underflows, it is reloaded with the content of this register and starts counting down over again.

(6) Setting the DACNT Counter Register

Set the value H'FF in this register as the DACNT Counter's initial value. Consequently, an analog voltage value corresponding to the value set in DA0 Data Register 255 is output from the DA0 pin.

(7) Setting the PD related registers

- Setting the Prescalers 0C and 1C
  Always write the value H'00 in these registers.

- Setting the PD0 and PD1 Data Updating Disable Event Select Registers
  Use these registers to set data-updating disable events.

- Setting the PD0 and PD1 Data Updating Control Registers
  Use these registers to enable or disable updating of PD data.

- Setting the ABD0 and ABD1 Mask Registers
  When using the ABD compare match function, select the bits to be masked.

- Setting the S Error 0 and S Error 1 Detection Range Select Registers
  Use these registers to set the S error detection range.

- Setting the ABD0 and ABD1 Compare Registers
  When using ABD compare match, set the compare value in these registers.

- Setting the PITCH0 and PITCH1 Compare Registers
  When using PITCH compare match, set the compare value in these registers.

(8) Setting the Interrupt Controller

When using PD interrupts, set the priority level of each interrupt.

(9) Setting the TIN and PD Interrupt Control Registers

Use the TIN Interrupt Control Register to enable or disable TIN related interrupts.
Use the PD Interrupt Control Register to enable or disable PD calculation related interrupts.

(10) Setting the DACNT and TPD Control Registers

Set the DACNT Control Register DACNTEN bit to enable the counter. The DACNT Counter starts counting down, generating sine waves.
Set the value H'11 in the TPD Control Register to enable counting in PD sensor support mode.

Note: Set these two registers simultaneously by using a STH instruction.

(11) Setting the TEP0P, TEP0M, TEP1P, and TEP1M Control Registers

Set the value H'11 in each register to enable counting in PD sensor support mode.

Note: Use all of the data tables 0-255 to define the sine wave.

(12) Setting the TIN Input Processing Control Register

Select rising-edge detection for each TIN input.

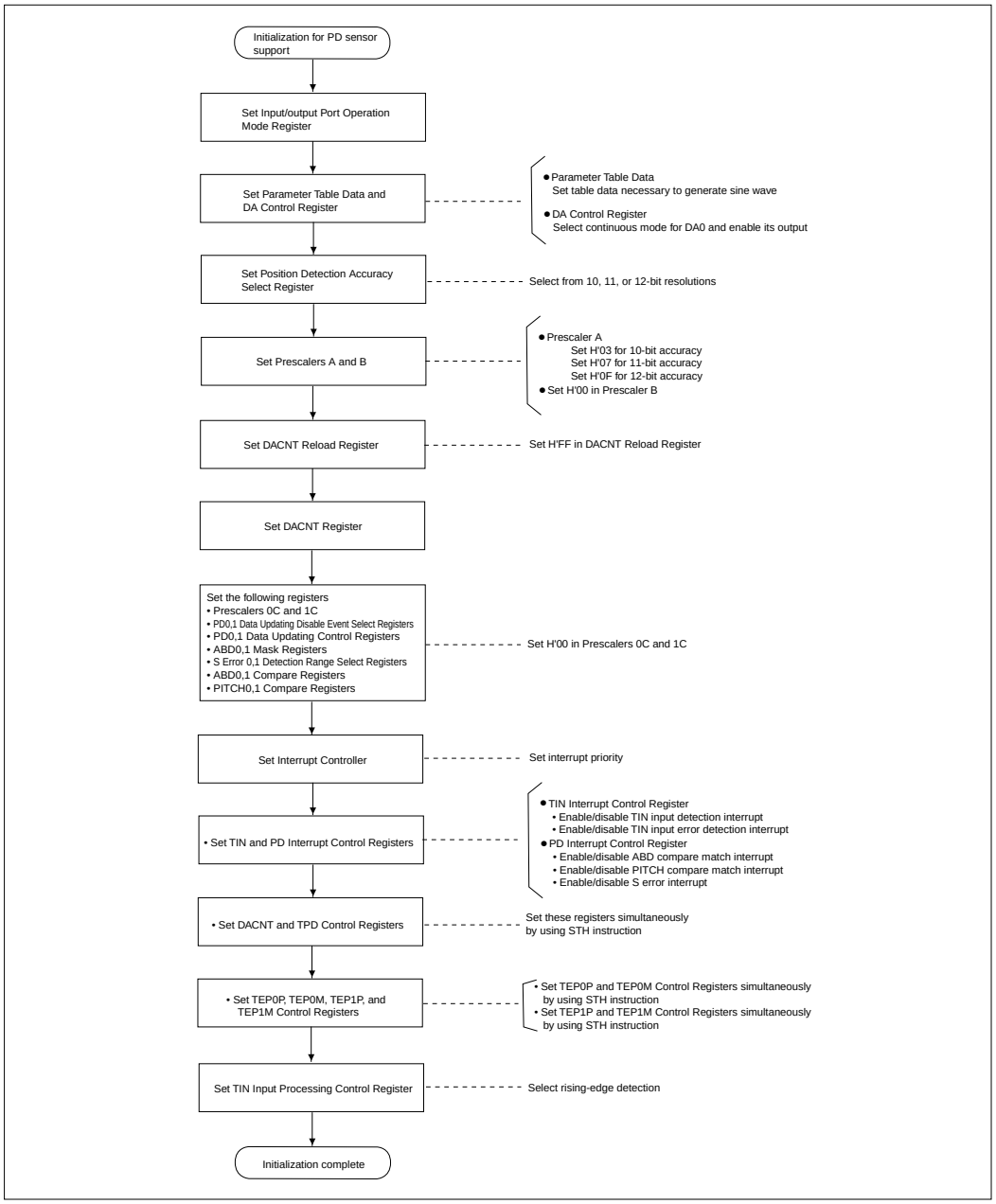


Figure 15.3.1 PD Sensor Support Initialization Flow

15.4 Precautions on Using the PD Module

- PD calculation processing is always performed based on the latest measured value regardless of how the PDi Data Updating Control Register is set. Therefore, even after disabling data updating, it is possible to read out the calculation result based on the latest event input by altering register settings back again. In no case will event inputs pass undetected even when data updating is disabled.

- When performing predictive calculations, be sure to disable updating of PD data before reading out the FDLT, PITCHLT, ABDLT, and RSUMLT Registers.

CHAPTER 16

D-A CONVERTERS

16.1 Outline of the D-A Converters

The 32172/32173 contains two 8-bit D-A converters (D-A0 and D-A1 Converters). D-A conversion is performed in either single mode or continuous mode (D-A0 Converter only).

Single mode: The analog values corresponding to the D-A Converter Register (DA0CNV, DA1CNV) values are output from the DA0 and DA1 pins.

Continuous mode: The values set in the DA0 Data Register n (n = 0-255) are sequentially converted to produce analog values. (Continuous mode is available for only the D-A0 Converter, and not for the D-A1 Converter.)

Table 16.1.1 outlines the D-A Converters. Figures 16.1.1 and 16.1.2 show block diagrams of the D-A Converters. Figure 16.1.3 shows an equivalent circuit of the D-A Converters.

Table 16.1.1 Outline of the D-A Converters

Item	Content
Analog output	2 channels
D-A conversion method	R-2R method
Resolution	8 bits
Output voltage	0–VREF
Conversion mode Continuous mode (Note)	Single mode
Parameter table	256-byte parameter table for output use
Conversion speed	TBD

Note: Available for only the D-A0 Converter, and not available for the D-A1 Converter.

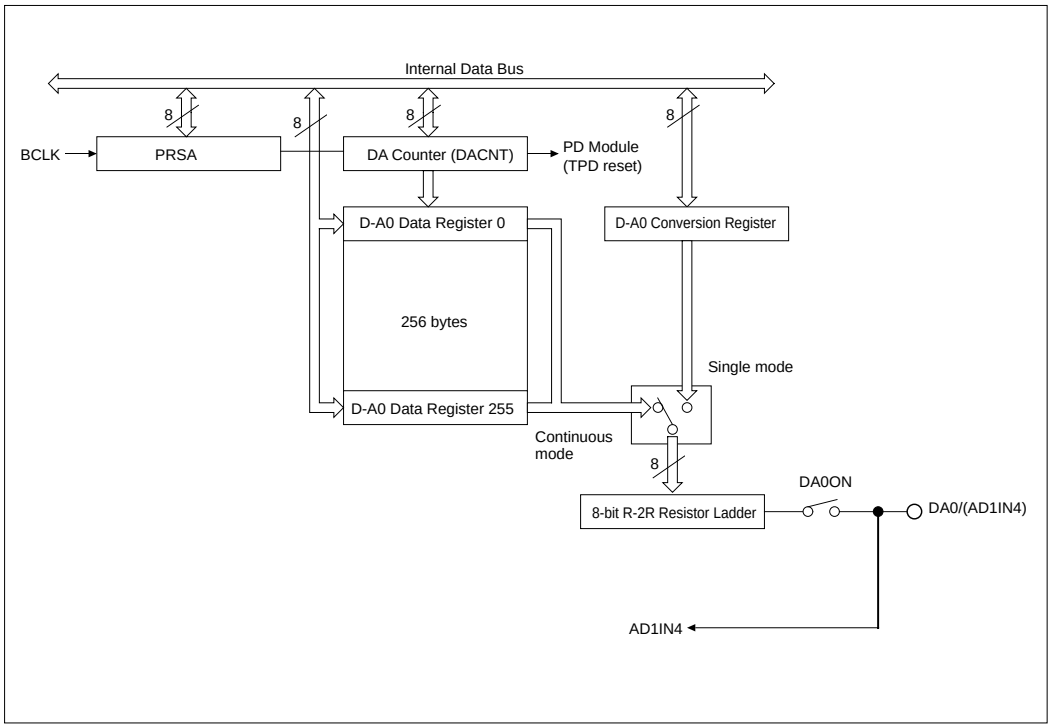


Figure 16.1.1 Block Diagram of the D-A0 Converter

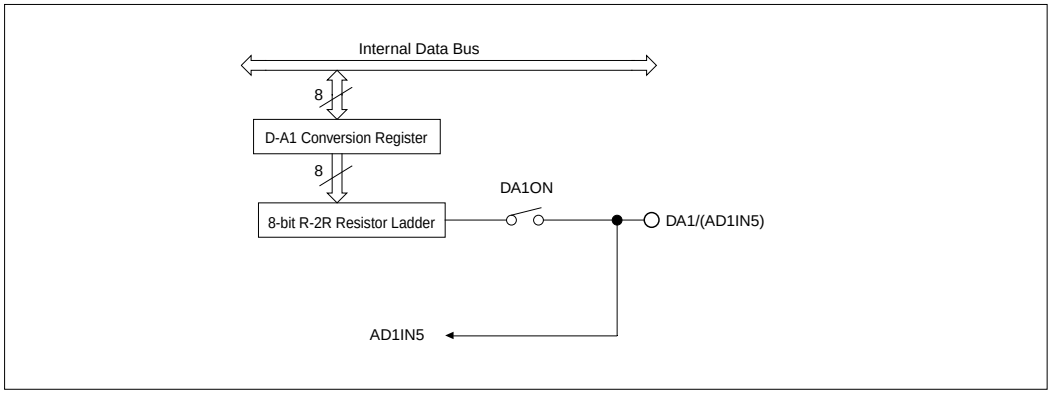
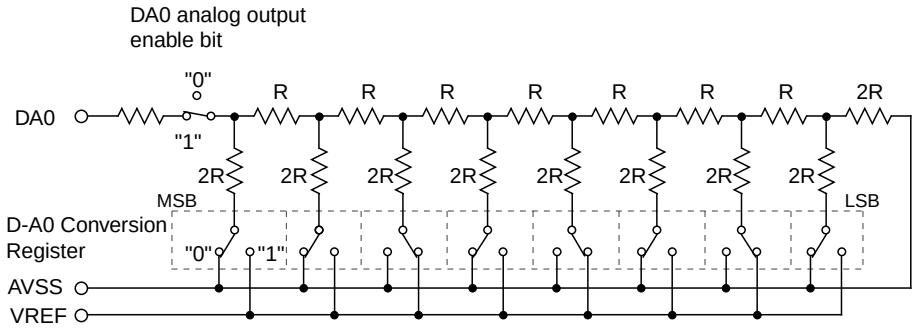


Figure 16.1.2 Block Diagram of the D-A1 Converter



Note 1: This applies to the case where the value of the D-A0 Conversion Register is H'2A.

Note 2: This circuit is the same as for the D-A1 Converter.

Note 3: When not using the D-A converters, set the D-A Control Register's analog output enable bit (DA0ON, DA1ON) to 0 and write the data H'00 to the D-A Conversion Register (DA0CNV, DA1CNV) to prevent current from flowing into the R-2R resistors. This helps to minimize the unwanted current consumption.

Figure 16.1.3 Equivalent Circuit Diagram of the D-A0 Converter

16.2 D-A Converter Related Registers

A D-A converter related register map is shown below.

Address	+0 address D0	D7 D8	+1 address D15
H'0080 1800	Prescaler Register A (PRSA)		
H'0080 1802	DACNT Reload Register (DACNTRL)		
H'0080 1804			
H'0080 1806	DACNT Control Register (DACNTCR)		
H'0080 1808	DACNT Counter (DACNT)		
H'0080 1C78	D-A0 Conversion Register (DA0CNV)		
H'0080 1C7A	D-A1 Conversion Register (DA1CNV)		
H'0080 1C7C	D-A Control Register (DACR)		
H'0080 1D00	D-A0 Data Register 0 (DA0DT0)		D-A0 Data Register 1 (DA0DT1)
H'0080 1D02	D-A0 Data Register 2 (DA0DT2)		D-A0 Data Register 3 (DA0DT3)
H'0080 1D04	D-A0 Data Register 4 (DA0DT4)		D-A0 Data Register 5 (DA0DT5)
H'0080 1D06	D-A0 Data Register 6 (DA0DT6)		D-A0 Data Register 7 (DA0DT7)
H'0080 1D08	D-A0 Data Register 8 (DA0DT8)		D-A0 Data Register 9 (DA0DT9)
H'0080 1D0A	D-A0 Data Register 10 (DA0DT10)		D-A0 Data Register 11 (DA0DT11)
H'0080 1D0C	D-A0 Data Register 12 (DA0DT12)		D-A0 Data Register 13 (DA0DT13)
H'0080 1D0E	D-A0 Data Register 14 (DA0DT14)		D-A0 Data Register 15 (DA0DT15)
H'0080 1D10	D-A0 Data Register 16 (DA0DT16)		D-A0 Data Register 17 (DA0DT17)
H'0080 1D12	D-A0 Data Register 18 (DA0DT18)		D-A0 Data Register 19 (DA0DT19)
H'0080 1D14	D-A0 Data Register 20 (DA0DT20)		D-A0 Data Register 21 (DA0DT21)
H'0080 1D16	D-A0 Data Register 22 (DA0DT22)		D-A0 Data Register 23 (DA0DT23)
H'0080 1D18	D-A0 Data Register 24 (DA0DT24)		D-A0 Data Register 25 (DA0DT25)
H'0080 1D1A	D-A0 Data Register 26 (DA0DT26)		D-A0 Data Register 27 (DA0DT27)
H'0080 1D1C	D-A0 Data Register 28 (DA0DT28)		D-A0 Data Register 29 (DA0DT29)
H'0080 1D1E	D-A0 Data Register 30 (DA0DT30)		D-A0 Data Register 31 (DA0DT31)
H'0080 1D20	D-A0 Data Register 32 (DA0DT32)		D-A0 Data Register 33 (DA0DT33)
H'0080 1D22	D-A0 Data Register 34 (DA0DT34)		D-A0 Data Register 35 (DA0DT35)
H'0080 1D24	D-A0 Data Register 36 (DA0DT36)		D-A0 Data Register 37 (DA0DT37)
H'0080 1D26	D-A0 Data Register 38 (DA0DT38)		D-A0 Data Register 39 (DA0DT39)
H'0080 1D28	D-A0 Data Register 40 (DA0DT40)		D-A0 Data Register 41 (DA0DT41)
H'0080 1D2A	D-A0 Data Register 42 (DA0DT42)		D-A0 Data Register 43 (DA0DT43)
H'0080 1D2C	D-A0 Data Register 44 (DA0DT44)		D-A0 Data Register 45 (DA0DT45)
H'0080 1D2E	D-A0 Data Register 46 (DA0DT46)		D-A0 Data Register 47 (DA0DT47)
H'0080 1D30	D-A0 Data Register 48 (DA0DT48)		D-A0 Data Register 49 (DA0DT49)
H'0080 1D32	D-A0 Data Register 50 (DA0DT50)		D-A0 Data Register 51 (DA0DT51)
H'0080 1D34	D-A0 Data Register 52 (DA0DT52)		D-A0 Data Register 53 (DA0DT53)
H'0080 1D36	D-A0 Data Register 54 (DA0DT54)		D-A0 Data Register 55 (DA0DT55)
H'0080 1D38	D-A0 Data Register 56 (DA0DT56)		D-A0 Data Register 57 (DA0DT57)
H'0080 1D3A	D-A0 Data Register 58 (DA0DT58)		D-A0 Data Register 59 (DA0DT59)
H'0080 1D3C	D-A0 Data Register 60 (DA0DT60)		D-A0 Data Register 61 (DA0DT61)
H'0080 1D3E	D-A0 Data Register 62 (DA0DT62)		D-A0 Data Register 63 (DA0DT63)
H'0080 1D40	D-A0 Data Register 64 (DA0DT64)		D-A0 Data Register 65 (DA0DT65)
H'0080 1D42	D-A0 Data Register 66 (DA0DT66)		D-A0 Data Register 67 (DA0DT67)
H'0080 1D44	D-A0 Data Register 68 (DA0DT68)		D-A0 Data Register 69 (DA0DT69)
H'0080 1D46	D-A0 Data Register 70 (DA0DT70)		D-A0 Data Register 71 (DA0DT71)
H'0080 1D48	D-A0 Data Register 72 (DA0DT72)		D-A0 Data Register 73 (DA0DT73)
H'0080 1D4A	D-A0 Data Register 74 (DA0DT74)		D-A0 Data Register 75 (DA0DT75)
H'0080 1D4C	D-A0 Data Register 76 (DA0DT76)		D-A0 Data Register 77 (DA0DT77)
H'0080 1D4E	D-A0 Data Register 78 (DA0DT78)		D-A0 Data Register 79 (DA0DT79)

Blank areas are reserved for future use.

Figure 16.2.1 SFR Area Register Map (1/3)

Address	+0 address D0	D7	+1 address D8	D15
H'0080 1D50	D-A0 Data Register 80 (DA0DT80)		D-A0 Data Register 81 (DA0DT81)	
H'0080 1D52	D-A0 Data Register 82 (DA0DT82)		D-A0 Data Register 83 (DA0DT83)	
H'0080 1D54	D-A0 Data Register 84 (DA0DT84)		D-A0 Data Register 85 (DA0DT85)	
H'0080 1D56	D-A0 Data Register 86 (DA0DT86)		D-A0 Data Register 87 (DA0DT87)	
H'0080 1D58	D-A0 Data Register 88 (DA0DT88)		D-A0 Data Register 89 (DA0DT89)	
H'0080 1D5A	D-A0 Data Register 90 (DA0DT90)		D-A0 Data Register 91 (DA0DT91)	
H'0080 1D5C	D-A0 Data Register 92 (DA0DT92)		D-A0 Data Register 93 (DA0DT93)	
H'0080 1D5E	D-A0 Data Register 94 (DA0DT94)		D-A0 Data Register 95 (DA0DT95)	
H'0080 1D60	D-A0 Data Register 96 (DA0DT96)		D-A0 Data Register 97 (DA0DT97)	
H'0080 1D62	D-A0 Data Register 98 (DA0DT98)		D-A0 Data Register 99 (DA0DT99)	
H'0080 1D64	D-A0 Data Register 100 (DA0DT100)		D-A0 Data Register 101 (DA0DT101)	
H'0080 1D66	D-A0 Data Register 102 (DA0DT102)		D-A0 Data Register 103 (DA0DT103)	
H'0080 1D68	D-A0 Data Register 104 (DA0DT104)		D-A0 Data Register 105 (DA0DT105)	
H'0080 1D6A	D-A0 Data Register 106 (DA0DT106)		D-A0 Data Register 107 (DA0DT107)	
H'0080 1D6C	D-A0 Data Register 108 (DA0DT108)		D-A0 Data Register 109 (DA0DT109)	
H'0080 1D6E	D-A0 Data Register 110 (DA0DT110)		D-A0 Data Register 111 (DA0DT111)	
H'0080 1D70	D-A0 Data Register 112 (DA0DT112)		D-A0 Data Register 113 (DA0DT113)	
H'0080 1D72	D-A0 Data Register 114 (DA0DT114)		D-A0 Data Register 115 (DA0DT115)	
H'0080 1D74	D-A0 Data Register 116 (DA0DT116)		D-A0 Data Register 117 (DA0DT117)	
H'0080 1D76	D-A0 Data Register 118 (DA0DT118)		D-A0 Data Register 119 (DA0DT119)	
H'0080 1D78	D-A0 Data Register 120 (DA0DT120)		D-A0 Data Register 121 (DA0DT121)	
H'0080 1D7A	D-A0 Data Register 122 (DA0DT122)		D-A0 Data Register 123 (DA0DT123)	
H'0080 1D7C	D-A0 Data Register 124 (DA0DT124)		D-A0 Data Register 125 (DA0DT125)	
H'0080 1D7E	D-A0 Data Register 126 (DA0DT126)		D-A0 Data Register 127 (DA0DT127)	
H'0080 1D80	D-A0 Data Register 128 (DA0DT128)		D-A0 Data Register 129 (DA0DT129)	
H'0080 1D82	D-A0 Data Register 130 (DA0DT130)		D-A0 Data Register 131 (DA0DT131)	
H'0080 1D84	D-A0 Data Register 132 (DA0DT132)		D-A0 Data Register 133 (DA0DT133)	
H'0080 1D86	D-A0 Data Register 134 (DA0DT134)		D-A0 Data Register 135 (DA0DT135)	
H'0080 1D88	D-A0 Data Register 136 (DA0DT136)		D-A0 Data Register 137 (DA0DT137)	
H'0080 1D8A	D-A0 Data Register 138 (DA0DT138)		D-A0 Data Register 139 (DA0DT139)	
H'0080 1D8C	D-A0 Data Register 140 (DA0DT140)		D-A0 Data Register 141 (DA0DT141)	
H'0080 1D8E	D-A0 Data Register 142 (DA0DT142)		D-A0 Data Register 143 (DA0DT143)	
H'0080 1D90	D-A0 Data Register 144 (DA0DT144)		D-A0 Data Register 145 (DA0DT145)	
H'0080 1D92	D-A0 Data Register 146 (DA0DT146)		D-A0 Data Register 147 (DA0DT147)	
H'0080 1D94	D-A0 Data Register 148 (DA0DT148)		D-A0 Data Register 149 (DA0DT149)	
H'0080 1D96	D-A0 Data Register 150 (DA0DT150)		D-A0 Data Register 151 (DA0DT151)	
H'0080 1D98	D-A0 Data Register 152 (DA0DT152)		D-A0 Data Register 153 (DA0DT153)	
H'0080 1D9A	D-A0 Data Register 154 (DA0DT154)		D-A0 Data Register 155 (DA0DT155)	
H'0080 1D9C	D-A0 Data Register 156 (DA0DT156)		D-A0 Data Register 157 (DA0DT157)	
H'0080 1D9E	D-A0 Data Register 158 (DA0DT158)		D-A0 Data Register 159 (DA0DT159)	
H'0080 1DA0	D-A0 Data Register 160 (DA0DT160)		D-A0 Data Register 161 (DA0DT161)	
H'0080 1DA2	D-A0 Data Register 162 (DA0DT162)		D-A0 Data Register 163 (DA0DT163)	
H'0080 1DA4	D-A0 Data Register 164 (DA0DT164)		D-A0 Data Register 165 (DA0DT165)	
H'0080 1DA6	D-A0 Data Register 166 (DA0DT166)		D-A0 Data Register 167 (DA0DT167)	
H'0080 1DA8	D-A0 Data Register 168 (DA0DT168)		D-A0 Data Register 169 (DA0DT169)	

Figure 16.2.2 SFR Area Register Map (2/3)

Address	+0 address D0 D7	+1 address D8 D15
H'0080 1DAA	D-A0 Data Register 170 (DA0DT170)	D-A0 Data Register 171 (DA0DT171)
H'0080 1DAC	D-A0 Data Register 172 (DA0DT172)	D-A0 Data Register 173 (DA0DT173)
H'0080 1DAE	D-A0 Data Register 174 (DA0DT174)	D-A0 Data Register 175 (DA0DT175)
H'0080 1DB0	D-A0 Data Register 176 (DA0DT176)	D-A0 Data Register 177 (DA0DT177)
H'0080 1DB2	D-A0 Data Register 178 (DA0DT178)	D-A0 Data Register 179 (DA0DT179)
H'0080 1DB4	D-A0 Data Register 180 (DA0DT180)	D-A0 Data Register 181 (DA0DT181)
H'0080 1DB6	D-A0 Data Register 182 (DA0DT182)	D-A0 Data Register 183 (DA0DT183)
H'0080 1DB8	D-A0 Data Register 184 (DA0DT184)	D-A0 Data Register 185 (DA0DT185)
H'0080 1DBA	D-A0 Data Register 186 (DA0DT186)	D-A0 Data Register 187 (DA0DT187)
H'0080 1DBC	D-A0 Data Register 188 (DA0DT188)	D-A0 Data Register 189 (DA0DT189)
H'0080 1DBE	D-A0 Data Register 190 (DA0DT190)	D-A0 Data Register 191 (DA0DT191)
H'0080 1DC0	D-A0 Data Register 192 (DA0DT192)	D-A0 Data Register 193 (DA0DT193)
H'0080 1DC2	D-A0 Data Register 194 (DA0DT194)	D-A0 Data Register 195 (DA0DT195)
H'0080 1DC4	D-A0 Data Register 196 (DA0DT196)	D-A0 Data Register 197 (DA0DT197)
H'0080 1DC6	D-A0 Data Register 198 (DA0DT198)	D-A0 Data Register 199 (DA0DT199)
H'0080 1DC8	D-A0 Data Register 200 (DA0DT200)	D-A0 Data Register 201 (DA0DT201)
H'0080 1DCA	D-A0 Data Register 202 (DA0DT202)	D-A0 Data Register 203 (DA0DT203)
H'0080 1DCC	D-A0 Data Register 204 (DA0DT204)	D-A0 Data Register 205 (DA0DT205)
H'0080 1DCE	D-A0 Data Register 206 (DA0DT206)	D-A0 Data Register 207 (DA0DT207)
H'0080 1DD0	D-A0 Data Register 208 (DA0DT208)	D-A0 Data Register 209 (DA0DT209)
H'0080 1DD2	D-A0 Data Register 210 (DA0DT210)	D-A0 Data Register 211 (DA0DT211)
H'0080 1DD4	D-A0 Data Register 212 (DA0DT212)	D-A0 Data Register 213 (DA0DT213)
H'0080 1DD6	D-A0 Data Register 214 (DA0DT214)	D-A0 Data Register 215 (DA0DT215)
H'0080 1DD8	D-A0 Data Register 216 (DA0DT216)	D-A0 Data Register 217 (DA0DT217)
H'0080 1DDA	D-A0 Data Register 218 (DA0DT218)	D-A0 Data Register 219 (DA0DT219)
H'0080 1DDC	D-A0 Data Register 220 (DA0DT220)	D-A0 Data Register 221 (DA0DT221)
H'0080 1DDE	D-A0 Data Register 222 (DA0DT222)	D-A0 Data Register 223 (DA0DT223)
H'0080 1DE0	D-A0 Data Register 224 (DA0DT224)	D-A0 Data Register 225 (DA0DT225)
H'0080 1DE2	D-A0 Data Register 226 (DA0DT226)	D-A0 Data Register 227 (DA0DT227)
H'0080 1DE4	D-A0 Data Register 228 (DA0DT228)	D-A0 Data Register 229 (DA0DT229)
H'0080 1DE6	D-A0 Data Register 230 (DA0DT230)	D-A0 Data Register 231 (DA0DT231)
H'0080 1DE8	D-A0 Data Register 232 (DA0DT232)	D-A0 Data Register 233 (DA0DT233)
H'0080 1DEA	D-A0 Data Register 234 (DA0DT234)	D-A0 Data Register 235 (DA0DT235)
H'0080 1DEC	D-A0 Data Register 236 (DA0DT236)	D-A0 Data Register 237 (DA0DT237)
H'0080 1DEE	D-A0 Data Register 238 (DA0DT238)	D-A0 Data Register 239 (DA0DT239)
H'0080 1DF0	D-A0 Data Register 240 (DA0DT240)	D-A0 Data Register 241 (DA0DT241)
H'0080 1DF2	D-A0 Data Register 242 (DA0DT242)	D-A0 Data Register 243 (DA0DT243)
H'0080 1DF4	D-A0 Data Register 244 (DA0DT244)	D-A0 Data Register 245 (DA0DT245)
H'0080 1DF6	D-A0 Data Register 246 (DA0DT246)	D-A0 Data Register 247 (DA0DT247)
H'0080 1DF8	D-A0 Data Register 248 (DA0DT248)	D-A0 Data Register 249 (DA0DT249)
H'0080 1DFA	D-A0 Data Register 250 (DA0DT250)	D-A0 Data Register 251 (DA0DT251)
H'0080 1DFC	D-A0 Data Register 252 (DA0DT252)	D-A0 Data Register 253 (DA0DT253)
H'0080 1DFE	D-A0 Data Register 254 (DA0DT254)	D-A0 Data Register 255 (DA0DT255)

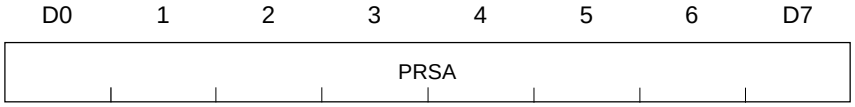
Figure 16.2.3 SFR Area Register Map (3/3)

16.2.1 Prescaler Unit

The Prescaler PRSA consists of an 8-bit counter which generates a clock from the internal peripheral clock (20.0 MHz when the CPU clock = 40 MHz) for supply to the DACNT Counter.
The prescaler value is initialized to H'00 when reset.
Also when the value of the prescaler register is rewritten, the prescaler starts operating with the new value simultaneously when the prescaler underflows.
Any value in the range of H'00 to H'FF can be set in the prescaler register. The prescaler's divide ratio is given by the equation below.

$$\text{Prescaler divide ratio} = \frac{f}{\text{Prescaler set value} + 1}$$

■ Prescaler Register A (PRSA) **<Address: H'0080 1800>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-7	PRSA	Sets the prescaler's divide-by value	○	○

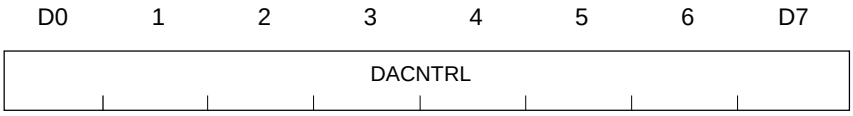
The Prescaler A generates a count clock for the D-A converter's parameter table address counter (DACNT) from the internal peripheral clock (20.0 MHz when the CPU clock = 40 MHz) by dividing it by an appropriate value. When the value of Prescaler Register A is rewritten while the DACNT is operating, the prescaler starts operating with the newly set value simultaneously when it underflows after reaching the minimum count.
The PRSA starts generating a clock after the DACNT Control Register DACNTEN bit is set to 1.

DACNT count period = f / (PRSA + 1)

Note: When using the D-A converter along with PD sensors, the PRSA set value is subject to limitations depending on how the Position Detection Accuracy Select Register is set.

16.2.2 DACNT Reload Register

■ **DACNT Reload Register (DACNTRL)** **<Address: H'0080 1802>**



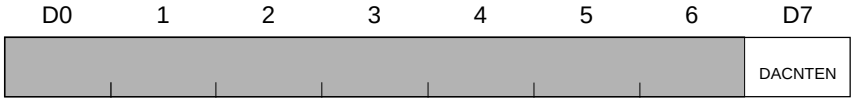
<When reset: H'00>

D	Bit Name	Function	R	W
0-7	DACNTRL	8-bit reload register value	○	○

The DACNT Reload Register is used to reload data into the DACNT Counter. Data is loaded into the DACNT Counter synchronously with the clock cycle in which the counter underflowed.
Simply because data is written to the DACNT Reload Register does not mean that the DACNT Counter is loaded with the data.

16.2.3 DACNT Control Register

■ **DACNT Control Register (DACNTCR)** **<Address: H'0080 1806>**



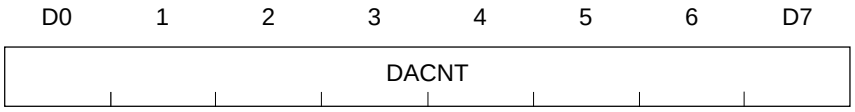
<When reset: H'00>

D	Bit Name	Function	R	W
0-6	No functions assigned		0	–
7	DACNTEN	0: Stops count 1: Enables count	○	○

This register controls enabling/disabling of the DACNT Counter to or not to count.
When the DACNTEN bit is set to 1, the DACNT Counter starts counting down from its set value synchronously with the count period that has been set with the PRSA. Setting the DACNTEN bit to 0 disables the DACNT Counter, so that it stops counting, with the count value retained.

16.2.4 DACNT Counter

■ **DACNT Counter (DACNT)** **<Address: H'0080 1808>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-7	DACNT	8-bit counter value	○	○

The DACNT Counter is an 8-bit down-counter which after being enabled, starts counting down from its set value synchronously with the count clock generated by the PRSA. The value of the DACNT Counter comprises the D-A converter's parameter table address value. When the DACNT Counter underflows after reaching the minimum count (DACNT = H'00), it is reloaded with the content of the DACNT Reload Register and restarts counting down from the newly set value.

The diagram below shows operation of the DACNT Counter when the DACNT Counter and the DACNT Reload Register respectively have the values H'A0 and H'B0 set as their initial values.

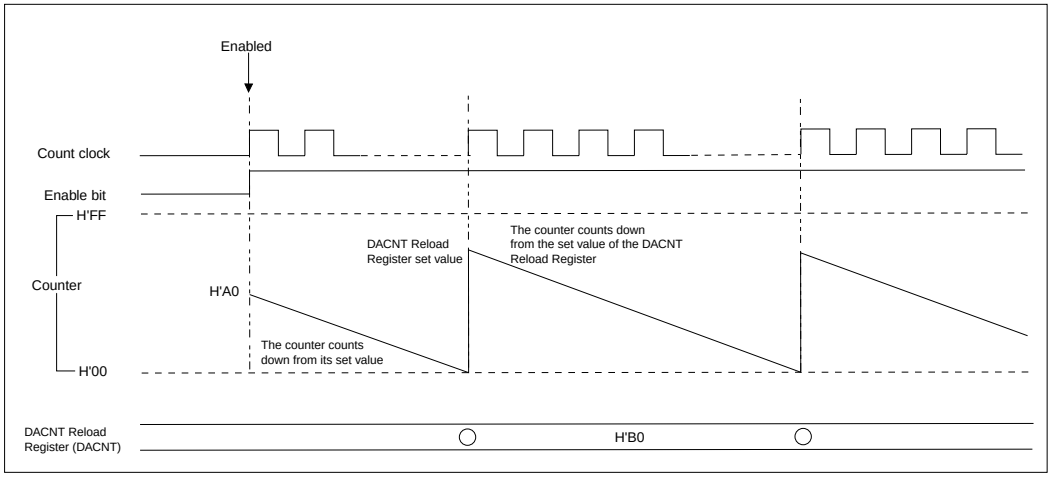
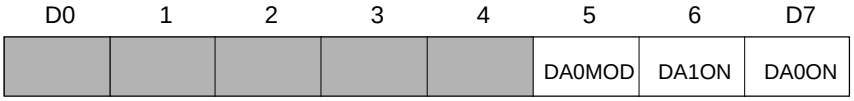


Figure 16.2.4 Example of DACNT Counter Operation

16.2.5 D-A Control Register

■ **D-A Control Register (DACR)** **<Address: H'0080 1C7C>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-4	No functions assigned		0	–
5	DA0MO (Select DA0 mode)	0: Single mode 1: Continuous mode	○	○
6	DA1ON (Enable DA1 analog output)	0: Disables output 1: Enables output	○	○
7	DA0ON (Enable DA0 analog output)			

This register is used to select conversion mode of the D-A0 Converter and control whether or not to send the D-A conversion result to the output pin (DA0, DA1).

(1) DA0MO (DA0 mode select) bit (D5)

This bit selects conversion mode of the D-A0 Converter (single or continuous mode). When DA0MO = 0, single mode is selected; when DA0MO = 1, continuous mode is selected.

Note: The D-A1 Converter does not have continuous mode. (This mode is available for only the D-A0 Converter.)

(2) DA1ON (DA1 analog output enable) bit (D6)

This bit controls whether or not to output the conversion result of the D-A1 Converter. Setting this bit to 1 enables analog output, so that the conversion result is output to the external pin DA1. Setting this bit to 0 disables analog output, so that the conversion result is not output to the external pin DA1.

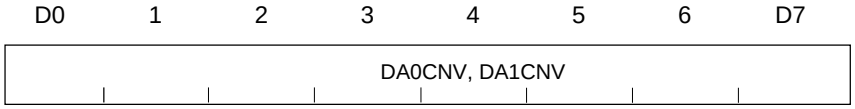
(3) DA0ON (DA0 analog output enable) bit (D7)

This bit controls whether or not to output the conversion result of the D-A0 Converter. Setting this bit to 1 enables analog output, so that the conversion result is output to the external pin DA0. Setting this bit to 0 disables analog output, so that the conversion result is not output to the external pin DA0.

16.2.6 D-A Conversion Registers

■ **D-A0 Conversion Register (DA0CNV)** **<Address: H'0080 1C78>**

■ **D-A1 Conversion Register (DA1CNV)** **<Address: H'0080 1C7A>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-7	DA0CNV, DA1CNV	8-bit D-A conversion data (single mode)	○	○

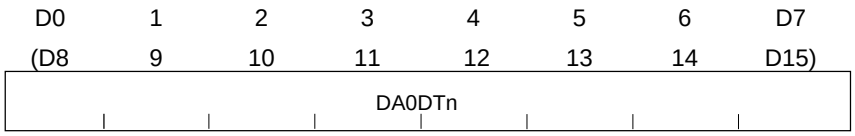
Writing a value (0-255) to the D-A Conversion Register (DA0CNV, DA1CNV) causes D-A conversion to start. The analog voltage V that is output by the D-A converter depends on the value n (n = 0-255) that has been set in the D-A Conversion Register (DA0CNV, DA1CNV).

$$V = VREF \times \frac{n}{256}$$

Note VREF : reference voltage

16.2.7 D-A0 Data Registers

■ **D-A0 Data Register n (DA0DTn)** **<Address: H'0080 1D00 to H'0080 1DFF>**



<When reset: indeterminate>

D	Bit Name	Function	R	W
0-7	DA0DTn	8-bit D-A conversion data (continuous mode)	○	○

The D-A0 Data Registers n (n = 0-255) are used for D-A conversion in continuous mode. The values set in the D-A0 Data Registers are sequentially D-A converted and output as analog quantities.

16.3 Functional Description of the D-A Converters

D-A conversion is performed in either single or continuous mode (the latter for only the D-A0 Converter).

16.3.1 Single Mode

In this mode, an analog value corresponding to the value set in the D-A Conversion Register (DAiCNV) is output from the external pin DAi.
Setting a value in the D-A Conversion Register (DAiCNV) causes D-A conversion to start. If the D-A Control Register DAiON bit has been set to 1 (analog output enabled), the converted analog value is output from the external pin DAi. The converter continues generating analog output from the external pin DAi until the DAiON bit is set to 0 (analog output disabled). The output voltage V is determined by the equation given below.

$$V = VREF \times \frac{\text{Set value of the D-A conversion register}}{256}$$

Note VREF : reference voltage

16.3.2 Continuous Mode

The values set in the D-A0 Data Registers n (n = 0-255) are sequentially converted into analog quantities which are then output from the external pin DA0.
Setting a value in the D-A Data Register n causes D-A conversion to start. If the D-A Control Register DA0ON bit (D7) has been set to 1 (analog output enabled), the converted analog value is output from the external pin DA0. The converter continues generating analog output from the external pin DA0 until the DA0ON bit is set to 0 (analog output disabled).

*** This is a blank page ***

CHAPTER 17

EXTERNAL BUS INTERFACE

17.1 External Bus Interface Related Signals

The 32172/32173 has the signals listed below that are associated with the external bus interface. These signals can be used in external extended mode or processor mode.

(1) Address

The external bus interface outputs 19-bit address (A12-A30) for addressing a 1-Mbyte space. The least significant A31 is not output. In an external write cycle, the external bus interface outputs the $\overline{BHW}$ and $\overline{BLW}$ signals to indicate the valid byte position on the 16-bit data bus at which to write data. In a read cycle, data is read always in 16 bits and the data at only the valid byte position is transferred.

Note: During external extended mode, the A12 and A13 pins are shared with the $\overline{CS2}$ and $\overline{CS3}$ pins, respectively. Therefore, functions of these pins must be selected with the Port P4 and P22 Peripheral Output Select Register. (In processor mode, these pins always function as A12 and A13.)

(2) Chip selects ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, and $\overline{CS3}$)

These signals are output in external extended mode or processor mode. The $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, and $\overline{CS3}$ signals each indicate one of the external extended areas spaced every 2 Mbytes. The $\overline{CS0}$ signal points to a 2-Mbyte area when in processor mode or a 1-Mbyte area when in external extended mode. (For details, see Chapter 3, "Address Space.")

Note: During external extended mode, the A12 and A13 pins are shared with the $\overline{CS2}$ and $\overline{CS3}$ pins, respectively. Therefore, functions of these pins must be selected with the Port P4 and P22 Peripheral Output Select Register. (In processor mode, these pins always function as A12 and A13, so that the $\overline{CS2}$ and $\overline{CS3}$ signals are not output.)

(3) Read strobe ($\overline{RD}$)

This signal is output during an external read cycle, indicating the timing at which to read data. This signal is driven high when writing to the bus or accessing the internal function.

(4) Byte high write/byte high enable ($\overline{BHW}$/$\overline{BHE}$)

The pin function is changed using the Bus Mode Control Register (BUSMODC).

When BUSMOD = 0, Byte High Write ($\overline{BHW}$) is selected, indicating that valid data is transferred on the upper byte side (DB0-DB7) of the data bus during external write access. This output is held high during external read and when accessing an internal function.

When BUSMOD = 1, Byte High Enable ($\overline{BHE}$) is selected, indicating that valid data is transferred on the upper byte side (DB0-DB7) of the data bus during external access. This output is held high when accessing an internal function.

(5) Byte low write/byte low enable ($\overline{BLW}$/$\overline{BLE}$)

The pin function is changed using the Bus Mode Control Register (BUSMODC).

When BUSMOD = 0, Byte Low Write ($\overline{BLW}$) is selected, indicating that valid data is transferred on the lower byte side (DB8-DB15) of the data bus during external write access. This output is held high during an external read cycle.

When BUSMOD = 1, Byte Low Enable ($\overline{BLE}$) is selected, indicating that valid data is transferred on the upper byte side (DB8-DB15) of the data bus during external access. This output is held high when accessing an internal function.

(6) Data bus (DB0-DB15)

This is a 16-bit data bus used to access an external device.

(7) System clock/write (BCLK/$\overline{WR}$)

The pin function is changed using the Bus Mode Control Register (BUSMODC).

When BUSMOD = 0, System Clock (BCLK) is selected, generating the system clock needed for the synchronized design of external devices. When the CPU clock is 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 setting the P7 Operation Mode Register P70MOD bit to 0.

BUSMOD = 1, Write ($\overline{WR}$) is selected, indicating that valid data is transferred on the data bus during external write access. This output is held high during an external read cycle and when accessing an internal function.

(8) Wait ($\overline{WAIT}$)

When the 32172/32173 started an external bus cycle, it automatically inserts wait cycles while the $\overline{WAIT}$ signal is asserted. For details, see Chapter 18, "Wait Controller." When not using the WAIT function, this pin can be used as P71 by setting the P7 Operation Mode Register P71MOD bit to 0.
For external access, one or m ore wait states are always inserted. Therefore, access to an external device incurs at least one wait state (total 2 BCLK cycles).

(9) Hold control ($\overline{HREQ}$, $\overline{HACK}$)

A hold state refers to a condition where bus access is halted and all of the bus interface related pins are placed in the high-impedance state. While in a hold state, any external bus master can use the system bus to transfer data.
A hold state is entered into by pulling the $\overline{HREQ}$ pin input. After the hold request is accepted, the

$\overline{\text{HACK}}$ pin outputs a low-level signal during hold and while going to a hold state. To return from the hold state to a normal operating state, release the $\overline{\text{HREQ}}$ input signal back high. When not using the HREQ and HACK functions, these pins can be used as P72 and P73 by setting the P7 Operation Mode Register P72MOD and P73MOD bits to 0.
The table below shows the state of each pin during hold.

Table 17.1.1 Pin State during Hold Period

Pin name	Pin state or operation
A12-A30, DB0-DB15, $\overline{\text{CS0}}$, $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, $\overline{\text{CS3}}$,$\overline{\text{RD}}$, $\overline{\text{BHW}}$, $\overline{\text{BLW}}$, $\overline{\text{BHE}}$, $\overline{\text{BLE}}$, $\overline{\text{WR}}$	High-impedance state
$\overline{\text{HACK}}$	Outputs a low
Other pins (e.g., ports and timer outputs)	Normal operation

(10) Port operation mode register

When the CPU is set to operate in external extended or processor mode, ports P0-P4 and P22 have their functions changed to signal pins for external access. When reset, these pins handle port signals.

Port P7 has its function changed to signal pins for external access by setting the Port P7 Operation Mode Register and Port P7 Peripheral Output Select Register.

■ P0 Operation Mode Register (P0MOD) <Address: H'0080 0740>

D0	1	2	3	4	5	6	D7
P00MOD	P01MOD	P02MOD	P03MOD	P04MOD	P05MOD	P06MOD	P07MOD

<When reset: H'00>

D	Bit Name	Function	R	W
0	P00MOD (Port P00 operation mode)	0: DB0 1: P00	○	○
1	P01MOD (Port P01 operation mode)	0: DB1 1: P01	○	○
2	P02MOD (Port P02 operation mode)	0: DB2 1: P02	○	○
3	P03MOD (Port P03 operation mode)	0: DB3 1: P03	○	○
4	P04MOD (Port P04 operation mode)	0: DB4 1: P04	○	○
5	P05MOD (Port P05 operation mode)	0: DB5 1: P05	○	○
6	P06MOD (Port P06 operation mode)	0: DB6 1: P06	○	○
7	P07MOD (Port P07 operation mode)	0: DB7 1: P07	○	○

Note: Settings made to the P0 Operation Mode Register are effective only when the CPU is operating in external extended mode.

■ **P1 Operation Mode Register (P1MOD)** **<Address: H'0080 0741>**

D8	9	10	11	12	13	14	D15
P10MOD	P11MOD	P12MOD	P13MOD	P14MOD	P15MOD	P16MOD	P17MOD

<When reset: H'00>

D	Bit Name	Function	R	W
8	P10MOD (Port P10 operation mode)	0: DB8 1: P10	○	○
9	P11MOD (Port P11 operation mode)	0: DB9 1: P11	○	○
10	P12MOD (Port P12 operation mode)	0: DB10 1: P12	○	○
11	P13MOD (Port P13 operation mode)	0: DB11 1: P13	○	○
12	P14MOD (Port P14 operation mode)	0: DB12 1: P14	○	○
13	P15MOD (Port P15 operation mode)	0: DB13 1: P15	○	○
14	P16MOD (Port P16 operation mode)	0: DB14 1: P16	○	○
15	P17MOD (Port P17 operation mode)	0: DB15 1: P17	○	○

Note: Settings made to the P1 Operation Mode Register are effective only when the CPU is operating in external extended mode.

■ **P2 Operation Mode Register (P2MOD)** **<Address: H'0080 0742>**

D0	1	2	3	4	5	6	D7
P20MOD	P21MOD	P22MOD	P23MOD	P24MOD	P25MOD	P26MOD	P27MOD

<When reset: H'00>

D	Bit Name	Function	R	W
0	P20MOD (Port P20 operation mode)	0: A23 1: P20	○	○
1	P21MOD (Port P21 operation mode)	0: A24 1: P21	○	○
2	P22MOD (Port P22 operation mode)	0: A25 1: P22	○	○
3	P23MOD (Port P23 operation mode)	0: A26 1: P23	○	○
4	P24MOD (Port P24 operation mode)	0: A27 1: P24	○	○
5	P25MOD (Port P25 operation mode)	0: A288 1: P25	○	○
6	P26MOD (Port P26 operation mode)	0: A29 1: P26	○	○
7	P27MOD (Port P27 operation mode)	0: A30 1: P27	○	○

Note: Settings made to the P2 Operation Mode Register are effective only when the CPU is operating in external extended mode.

■ **P3 Operation Mode Register (P3MOD)** **<Address: H'0080 0743>**

D8	9	10	11	12	13	14	D15
P30MOD	P31MOD	P32MOD	P33MOD	P34MOD	P35MOD	P36MOD	P37MOD

<When reset: H'00>

D	Bit Name	Function	R	W
8	P30MOD (Port P30 operation mode)	0: A15 1: P30	○	○
9	P31MOD (Port P31 operation mode)	0: A16 1: P31	○	○
10	P32MOD (Port P32 operation mode)	0: A17 1: P32	○	○
11	P33MOD (Port P33 operation mode)	0: A18 1: P33	○	○
12	P34MOD (Port P34 operation mode)	0: A19 1: P34	○	○
13	P35MOD (Port P35 operation mode)	0: A20 1: P35	○	○
14	P36MOD (Port P36 operation mode)	0: A21 1: P36	○	○
15	P37MOD (Port P37 operation mode)	0: A22 1: P37	○	○

Note: Settings made to the P3 Operation Mode Register are effective only when the CPU is operating in external extended mode.

■ **P4 Operation Mode Register (P4MOD)** **<Address: H'0080 0744>**

D0	1	2	3	4	5	6	D7
	P41MOD	P42MOD	P43MOD	P44MOD	P45MOD		P47MOD

<When reset: H'00>

D	Bit Name	Function	R	W
0	No functions assigned		0	–
1	P41MOD (Port P41 operation mode)	0: $\overline{BLW}/\overline{BLE}$ 1: P41	○	○
2	P42MOD (Port P42 operation mode)	0: $\overline{BHW}/\overline{BHE}$ 1: P42	○	○
3	P43MOD (Port P43 operation mode)	0: $\overline{RD}$ 1: P43	○	○
4	P44MOD (Port P44 operation mode)	0: $\overline{CS0}$ 1: P44	○	○
5	P45MOD (Port P45 operation mode)	0: $\overline{CS1}$ 1: P45	○	○
6	No functions assigned		0	–
7	P47MOD (Port P47 operation mode)	0: A14 1: P47	○	○

Note: Settings made to the P4 Operation Mode Register are effective only when the CPU is operating in external extended mode.

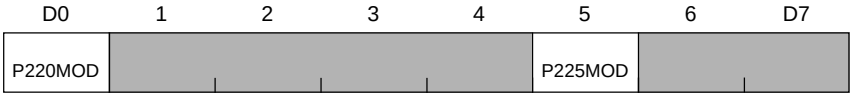
■ **P7 Operation Mode Register (P7MOD)** **<Address: H'0080 0747>**

D8	9	10	11	12	13	14	D15
P70MOD	P71MOD	P72MOD	P73MOD	P74MOD	P75MOD	P76MOD	P77MOD

<When reset: H'00>

D	Bit Name	Function	R	W
8	P70MOD	0: P70	○	○
	(Port P70 operation mode)	1: BCLK/$\overline{WR}$		
9	P71MOD	0: P71	○	○
	(Port P71 operation mode)	1: $\overline{WAIT}$		
10	P72MOD	0: P72	○	○
	(Port P72 operation mode)	1: $\overline{HREQ}$		
11	P73MOD	0: P73	○	○
	(Port P73 operation mode)	1: $\overline{HACK}$/TXD3		
12	P74MOD	0: P74	○	○
	(Port P74 operation mode)	1: RTDTXD		
13	P75MOD	0: P75	○	○
	(Port P75 operation mode)	1: RTDRXD		
14	P76MOD	0: P76	○	○
	(Port P76 operation mode)	1: RTDACK		
15	P77MOD	0: P77	○	○
	(Port P77 operation mode)	1: RTDCLK		

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



<When reset: H'00>

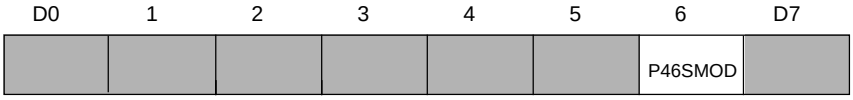
D	Bit Name	Function	R	W
0	P220MOD (Port P220 operation mode)	0: P220 1: CTX0	○	○
1-4	No functions assigned		0	–
5	P225MOD (Port P225 operation mode)	0: P225 1: Use inhibited	○	○
6-7	No functions assigned		0	–

Note 1: P221 is a CAN0 input-only pin.

Note 2: P225 has its pin function changed depending on how the MOD0 and MOD1 pins are set. Also, because this pin has a debug event function, its use requires caution.

Note 3: P222-P224, P226, and P227 are not included.

■ **P4 Peripheral Output Select Register (P4SMOD)** **<Address: H'0080 0764>**



<When reset: H'00>

D	Bit Name	Function	R	W
0-5	No functions assigned		0	–
6	P46SMOD (Select port P46 peripheral output)	0: A13 1: $\overline{CS3}$	○	○
7	No functions assigned		0	–

■ **P22 Peripheral Output Select Register (P22SMOD)** **<Address: H'0080 0776>**

<When reset: H'00>

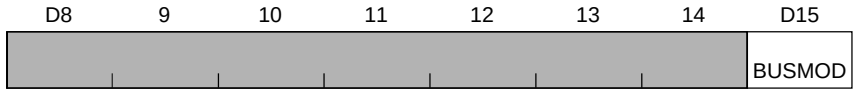
D	Bit Name	Function	R	W
0-4	No functions assigned		0	–
5	P225SMOD (Select port P225 peripheral output)	0: A12 1: $\overline{CS2}$	○	○
6-7	No functions assigned		0	–

Note: Settings made to the P22 Peripheral Output Select Register are effective only when the CPU is operating in external extended mode.

(11) Bus Mode Control Register (BUSMODC)

The microcomputer contains a function to select between two external bus modes.

■ **Bus Mode Control Register (BUSMODC)** **<Address: H'0080 077F>**



<When reset: H'00>

D	Bit Name	Function	R	W
8-14	No functions assigned		0	–
15	BUSMOD (Bus mode control)	0: WR signal separate mode 1: Byte enable separate mode	○	○

This register is used to facilitate memory connections during processor and external extended modes.

When the bus mode control (BUSMOD) bit = 0, the WR signal is output separately for each byte area. When in this mode, the $\overline{RD}$, $\overline{BHW}$, $\overline{BLW}$, $\overline{BCLK}$, and $\overline{WAIT}$ signals can be used. When connecting memory chips in boot mode, however, the Bus Mode Control Register has no effect, so that the external bus interface operates assuming that the bus mode control (BUSMOD) bit = 0.

When the bus mode control (BUSMOD) bit = 1, the byte enable signal is output separately for each byte area. When in this mode, the $\overline{RD}$, $\overline{BHE}$, $\overline{BLE}$, $\overline{WR}$, and $\overline{WAIT}$ signals can be used. When a WAIT control circuit is configured, timing must be controlled external to the chip because BCLK is not output.

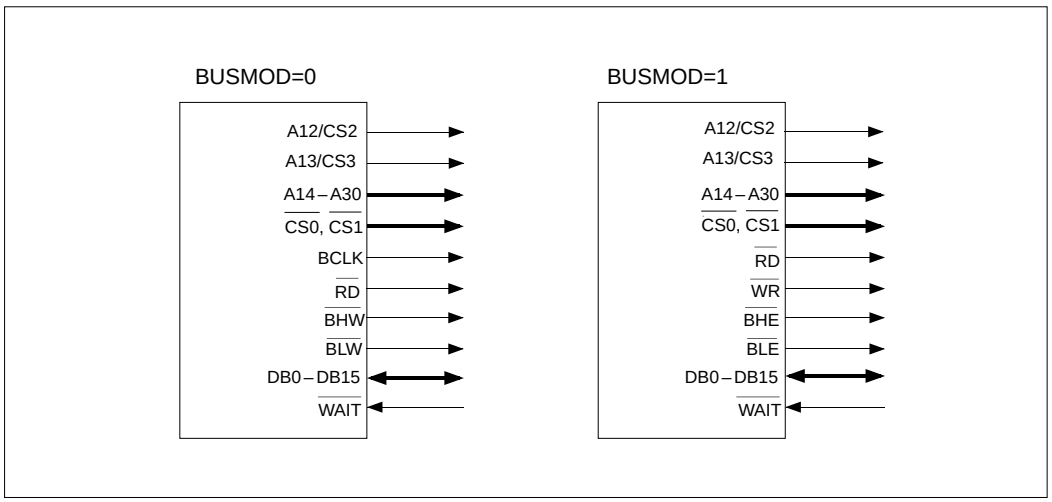


Figure 17.1.1 Pin Functions when Bus Modes are Changed

17.2 Read/Write Operations

(1) When Bus Mode Control Register = 0 (WR signal separate mode)

External read/write operations are performed using the address and data buses and the $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{RD}$, $\overline{BHW}$, $\overline{BLW}$, $\overline{WAIT}$, and BCLK signals. In an external read cycle, the $\overline{RD}$ signal goes low while the $\overline{BHW}$ and $\overline{BLW}$ signals both go high, so that the data at only the necessary byte position is read.
In an external write cycle, the $\overline{BHW}$ or $\overline{BLW}$ signal for the byte position to write is asserted low, allowing data to be written at that position.

When an external bus cycle starts, wait states are inserted as long as $\overline{WAIT}$ remains low. Therefore, the $\overline{WAIT}$ signal must always be held high unless necessary. Note that an external bus cycle, even during the shortest access, has at least one wait state inserted (shortest bus cycle consists of 2 BCLK cycles).

Note: $\overline{CS2}$ and $\overline{CS3}$ can be output in only external extended mode.

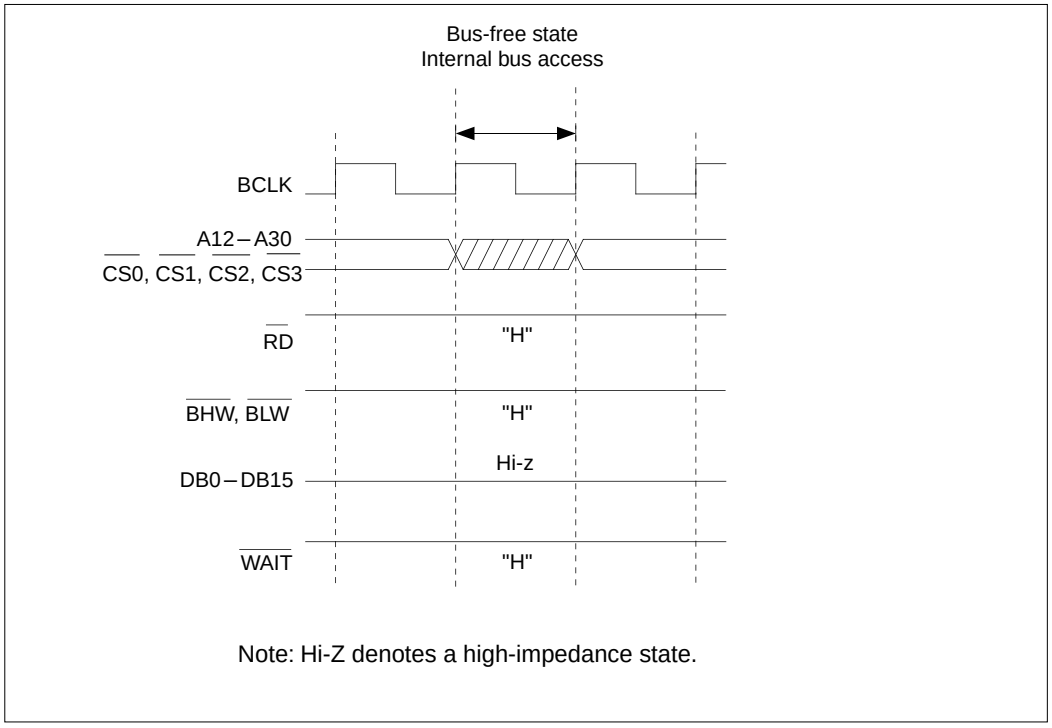


Figure 17.2.1 Read/Write Operations during Bus-free State/Internal Bus Access

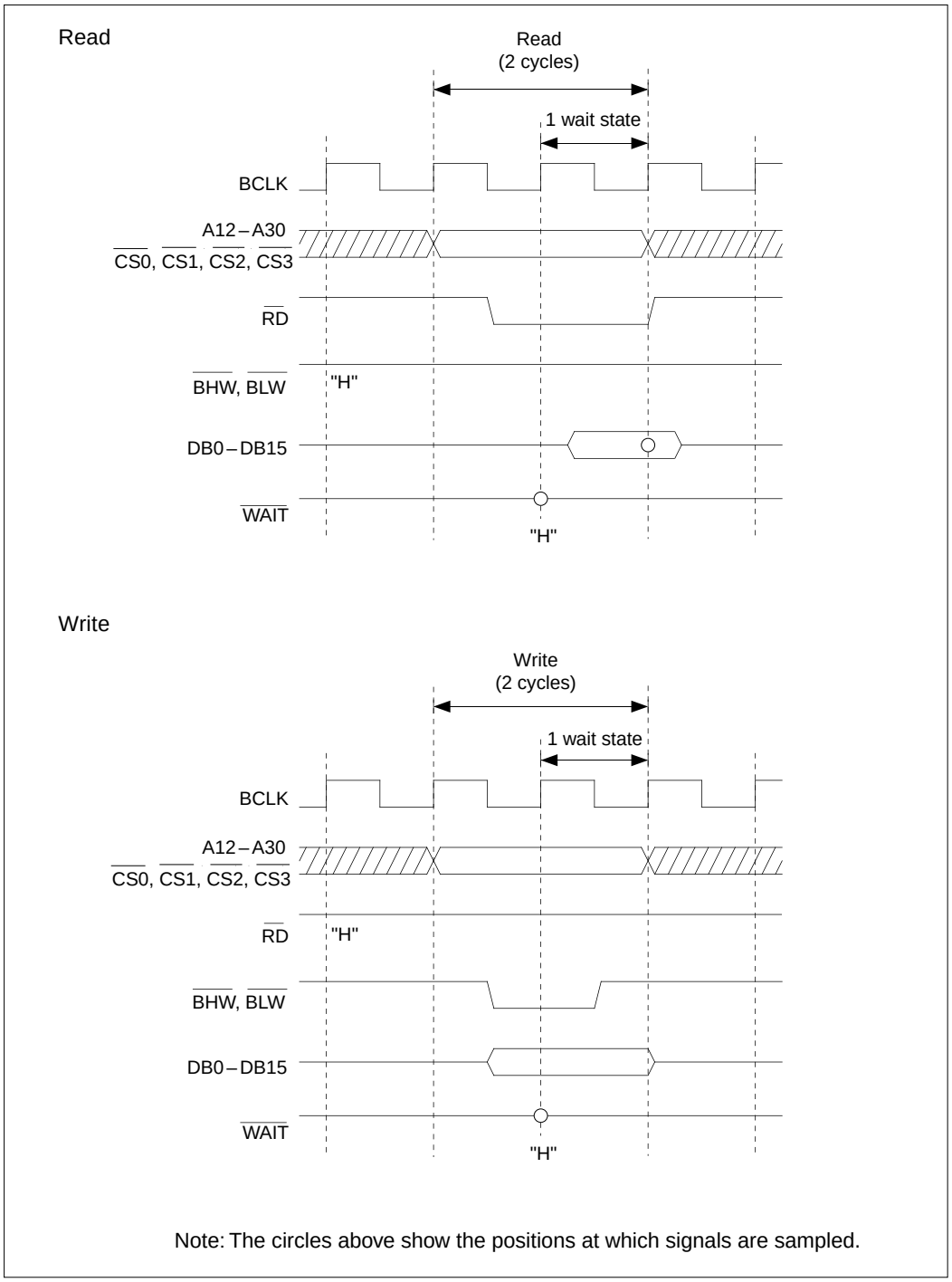


Figure 17.2.2 Read/Write Timing (during Shortest External Access)

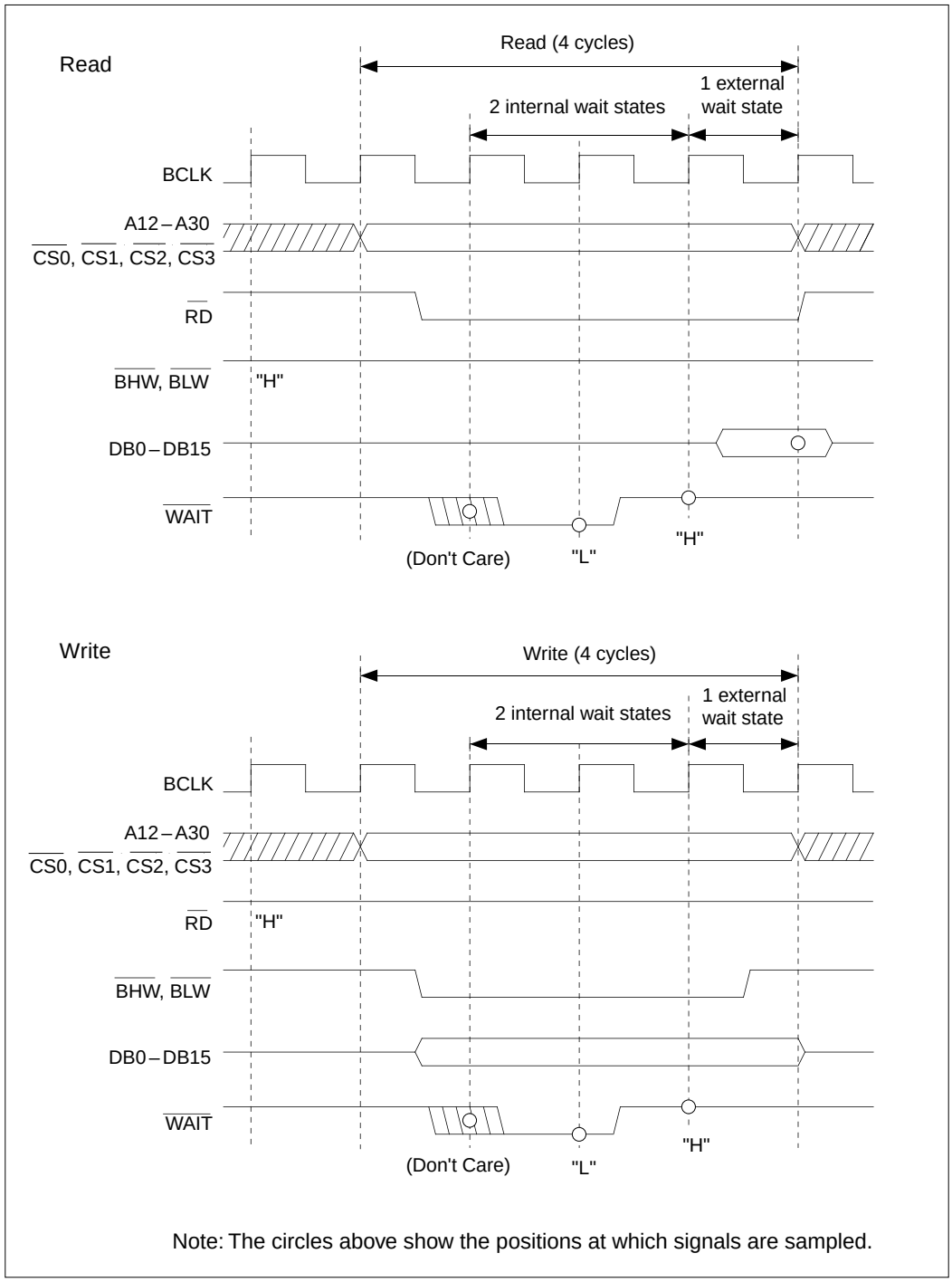


Figure 17.2.3 Read/Write Timing (Accessed with 2 Internal + 1 External Wait States)

(2) When Bus Mode Control Register = 1 (WR signal separate mode)

External read/write operations are performed using the address and data buses and the $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{RD}$, $\overline{BHE}$, $\overline{BLE}$, $\overline{WAIT}$, and $\overline{WR}$ signals. In an external read cycle, the $\overline{RD}$ signal goes low and the $\overline{BHE}$ or $\overline{BLE}$ signal for the byte position to read goes low, so that the data at only the necessary byte position is read.
In an external write cycle, the $\overline{WR}$ signal goes low and the $\overline{BHE}$ or $\overline{BLE}$ signal for the byte position to write is asserted low, allowing data to be written at the necessary byte position.

When an external bus cycle starts, wait states are inserted as long as $\overline{WAIT}$ remains low. Therefore, the $\overline{WAIT}$ signal must always be held high unless necessary. Note that an external bus cycle, even during the shortest access, has at least one wait state inserted (shortest bus cycle consists of 2 BCLK cycles). When not using the WAIT function, set the P7 Operation Mode Register P71MOD bit to 0. The pin can be used as P71.

Note: $\overline{CS2}$ and $\overline{CS3}$ can be output in only external extended mode.

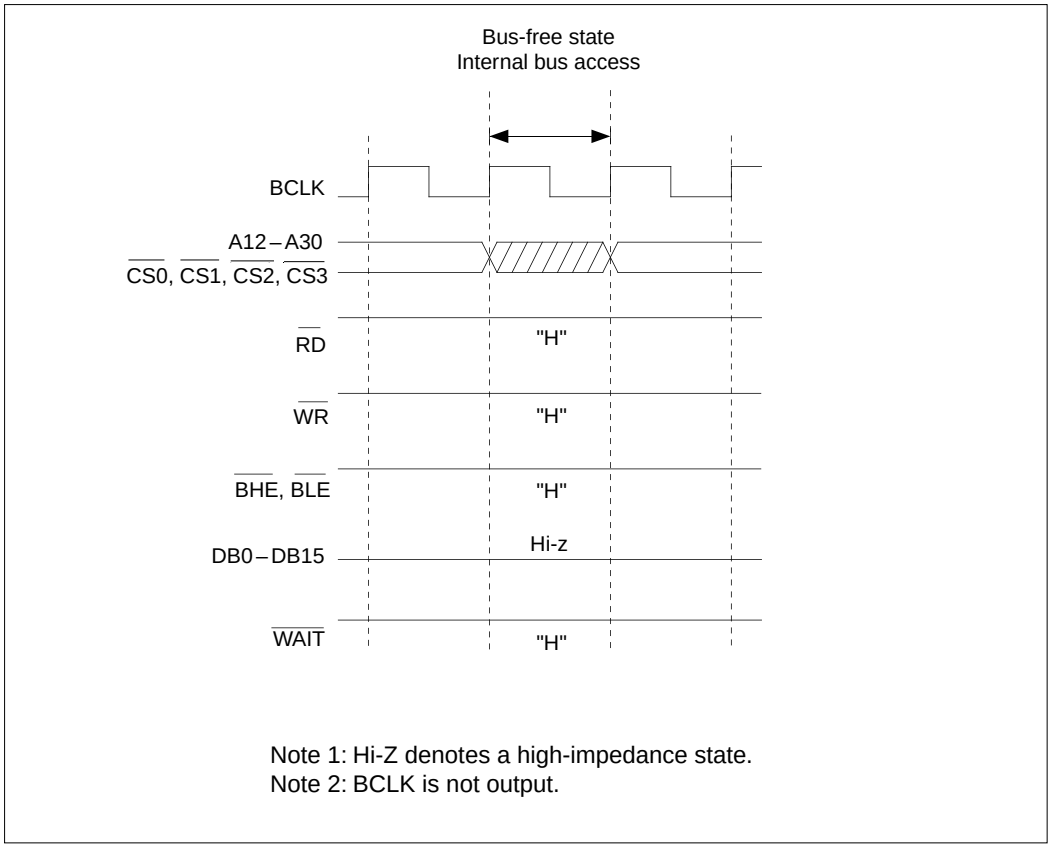
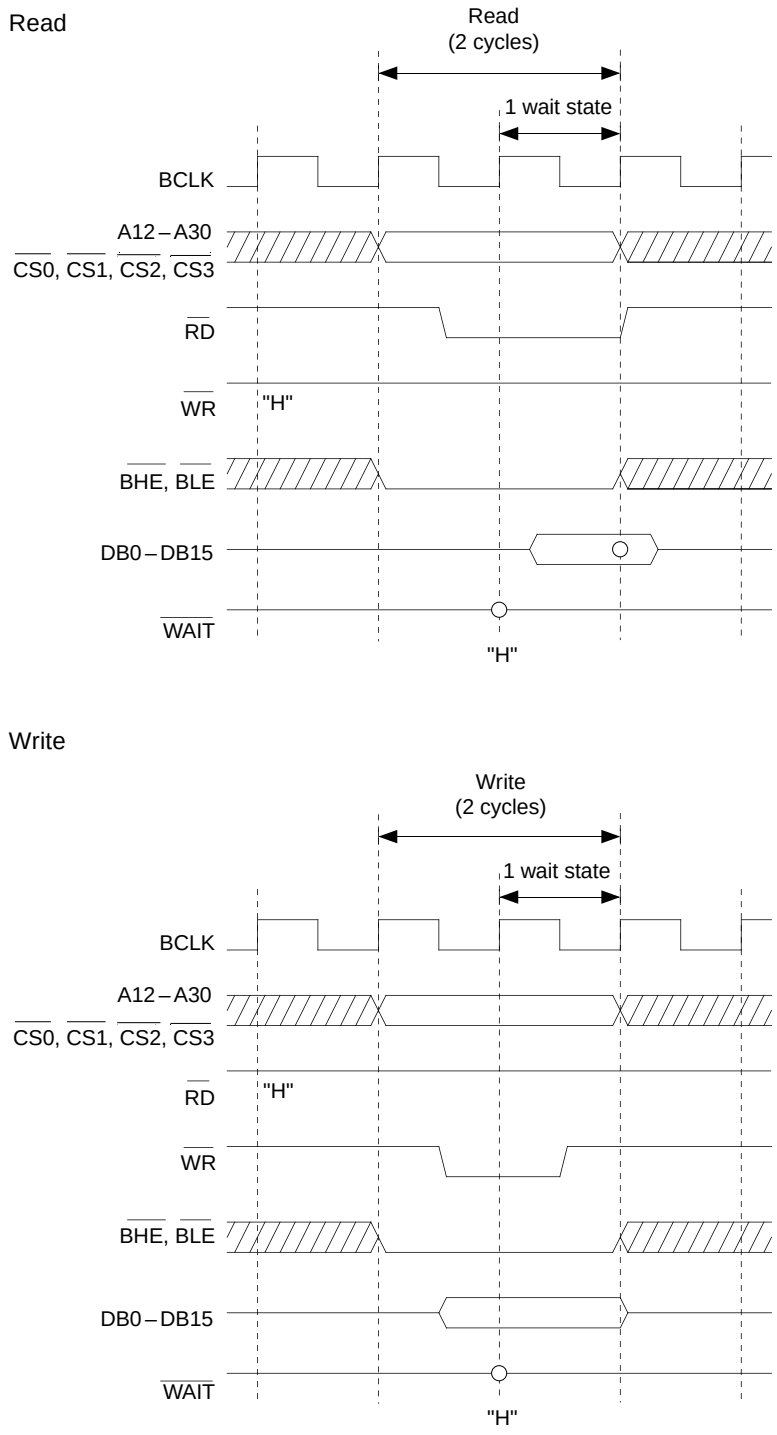
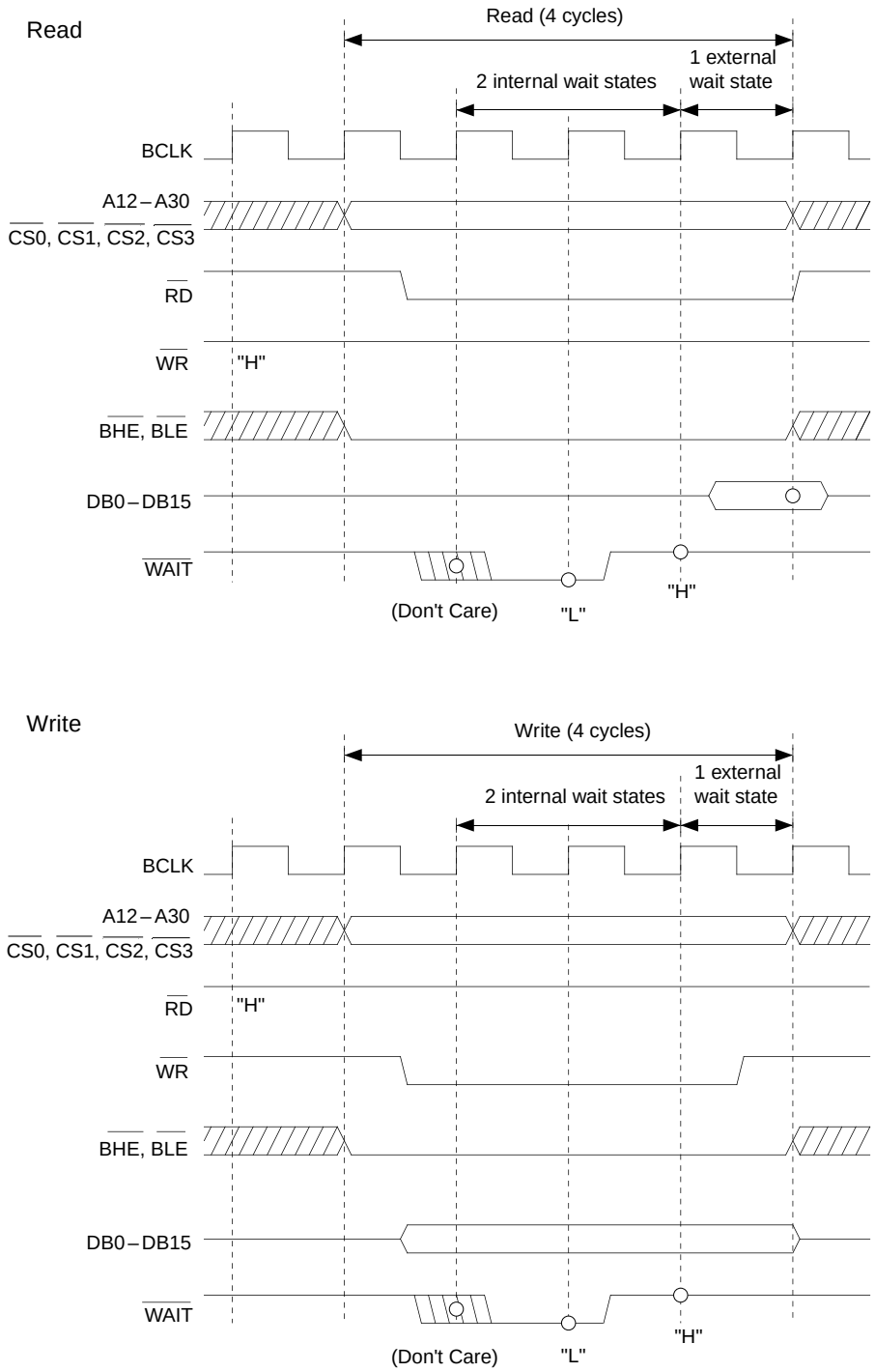


Figure 17.2.4 Read/Write Operations during Bus-free State/Internal Bus Access



Note 1: The circles above show the positions at which signals are sampled.
Note 2: BCLK is not output.

Figure 17.2.5 Read/Write Timing (during Shortest External Access)



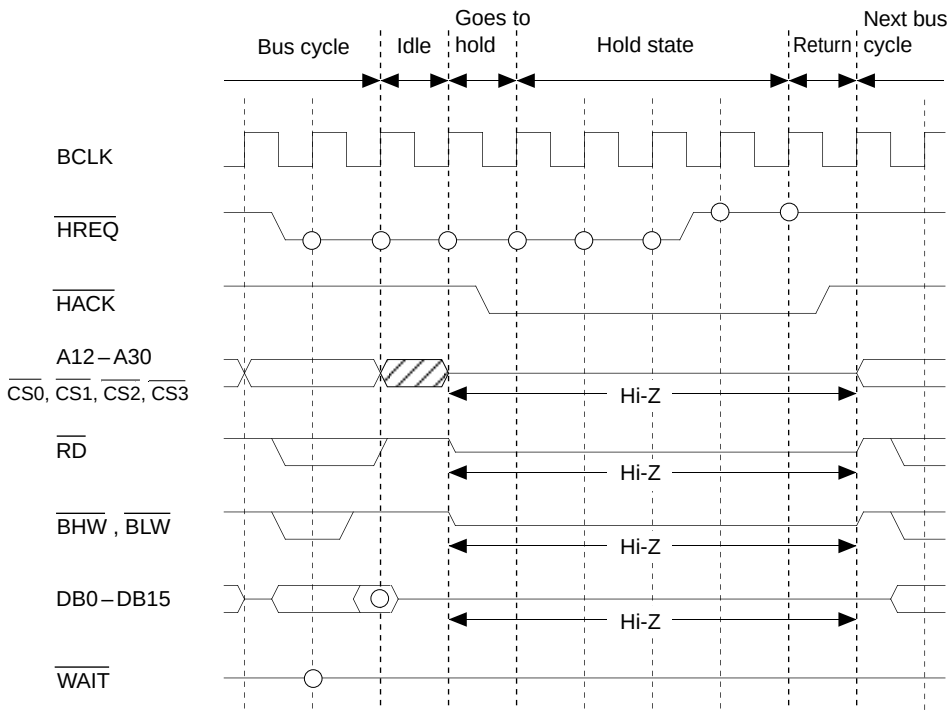
Note 1: The circles above show the positions at which signals are sampled.
Note 2: BCLK is not output.

Figure 17.2.6 Read/Write Timing (Accessed with 2 Internal + 1 External Wait States)

17.3 Bus Arbitration

(1) When Bus Mode Control Register = 0 (WR signal separate mode)

When the input signal at the $\overline{\text{HREQ}}$ pin is asserted low and the hold request is accepted, the CPU enters a hold state in which it outputs a low from the $\overline{\text{HACK}}$ pin. During hold, the bus related signals go to a high-impedance state, allowing data transfers to be performed on the system bus. To return from the hold state to a normal operating state, release the $\overline{\text{HREQ}}$ input signal back high.

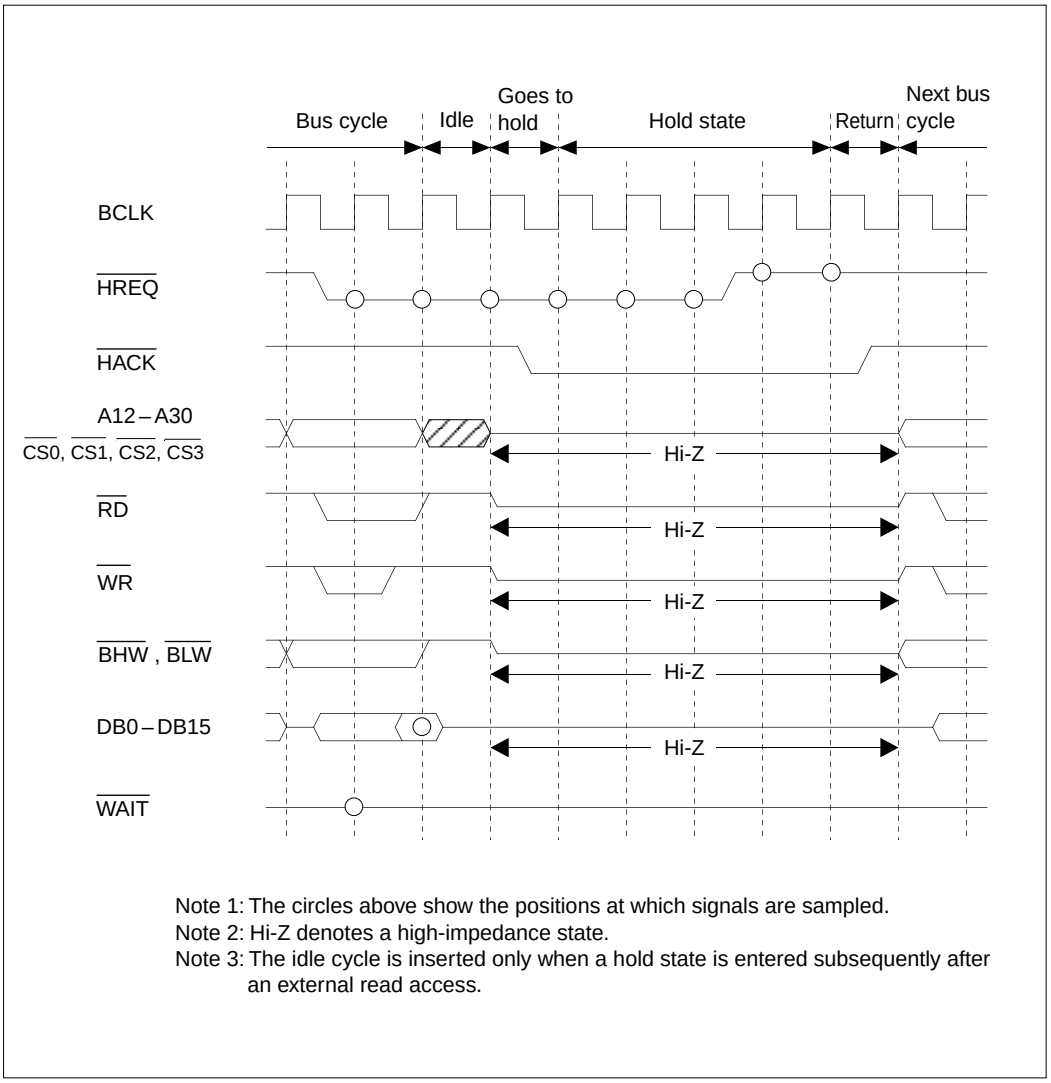


Note 1: The circles above show the positions at which signals are sampled.
Note 2: Hi-Z denotes a high-impedance state.
Note 3: The idle cycle is inserted only when a hold state is entered subsequently after an external read access.

Figure 17.3.1 Bus Arbitration Timing

(2) When Bus Mode Control Register = 1 (Byte enable separate mode)

When the input signal at the $\overline{HREQ}$ pin is asserted low and the hold request is accepted, the CPU enters a hold state in which it outputs a low from the $\overline{HACK}$ pin. During hold, the bus related signals go to a high-impedance state, allowing data transfers to be performed on the system bus. To return from the hold state to a normal operating state, release the $\overline{HREQ}$ input signal back high.



Note 1: The circles above show the positions at which signals are sampled.
Note 2: Hi-Z denotes a high-impedance state.
Note 3: The idle cycle is inserted only when a hold state is entered subsequently after an external read access.

Figure 17.3.2 Bus Arbitration Timing

17.4 Example for Connecting External Extension Memory

(1) When Bus Mode Control Register = 0 (with two memory blocks connected)

Figure 17.4.1 shows a typical connection diagram for the microcomputer when using external extension memory (external extended mode).

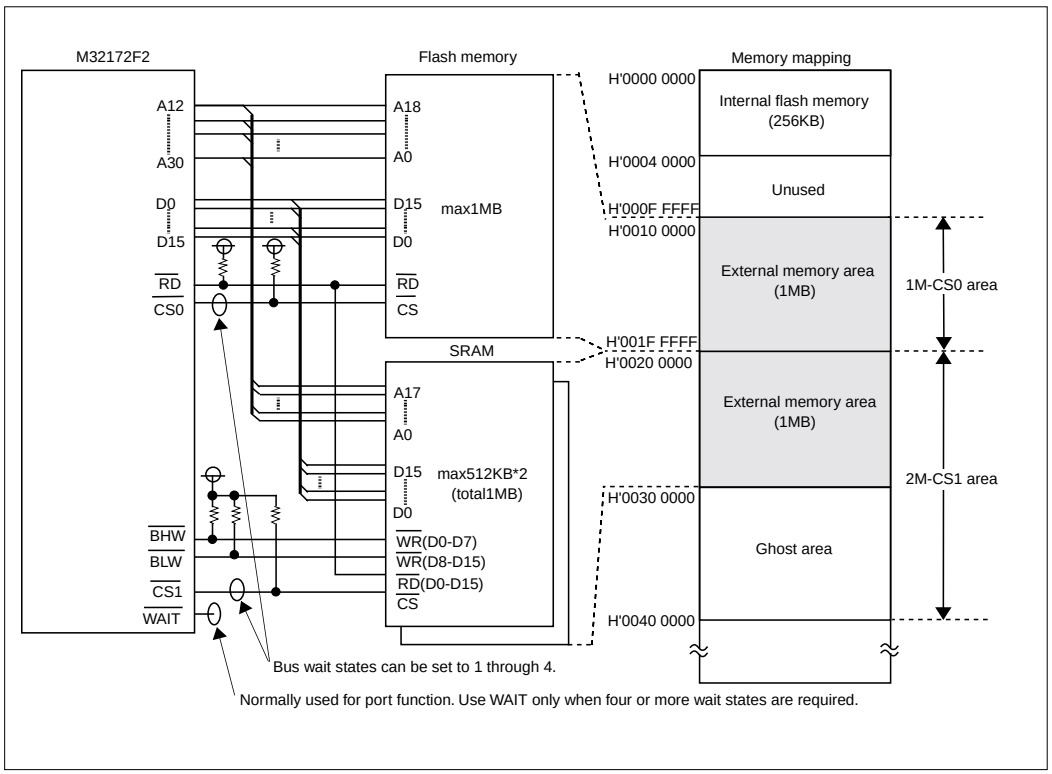
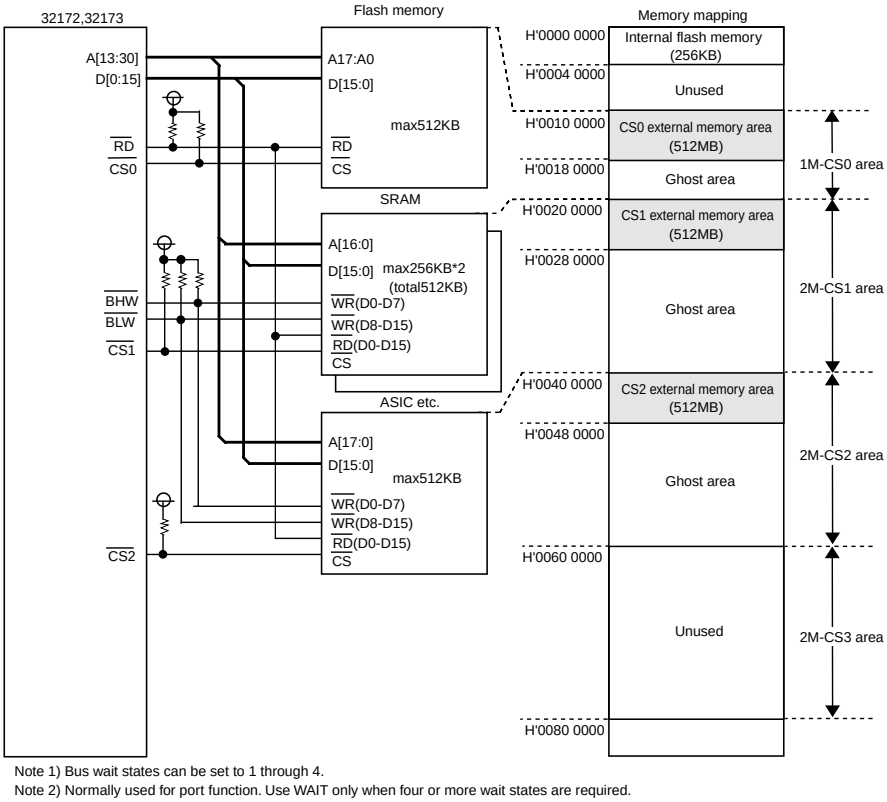


Figure 17.4.1 Typical Connection of External Extension Memory Using Two Memory Blocks (when BUSMOD = 0)

Note: The M32R/E address and data are comprised of bit 0 = MSB and bit 15 = LSB. When connecting external extension memory to the chip, the MSB and the LSB sides must be reversed.

(2) When Bus Mode Control Register = 0 (with three memory blocks connected)

Figure 17.4.2 shows a typical connection diagram for the microcomputer when using external extension memory (external extended mode).



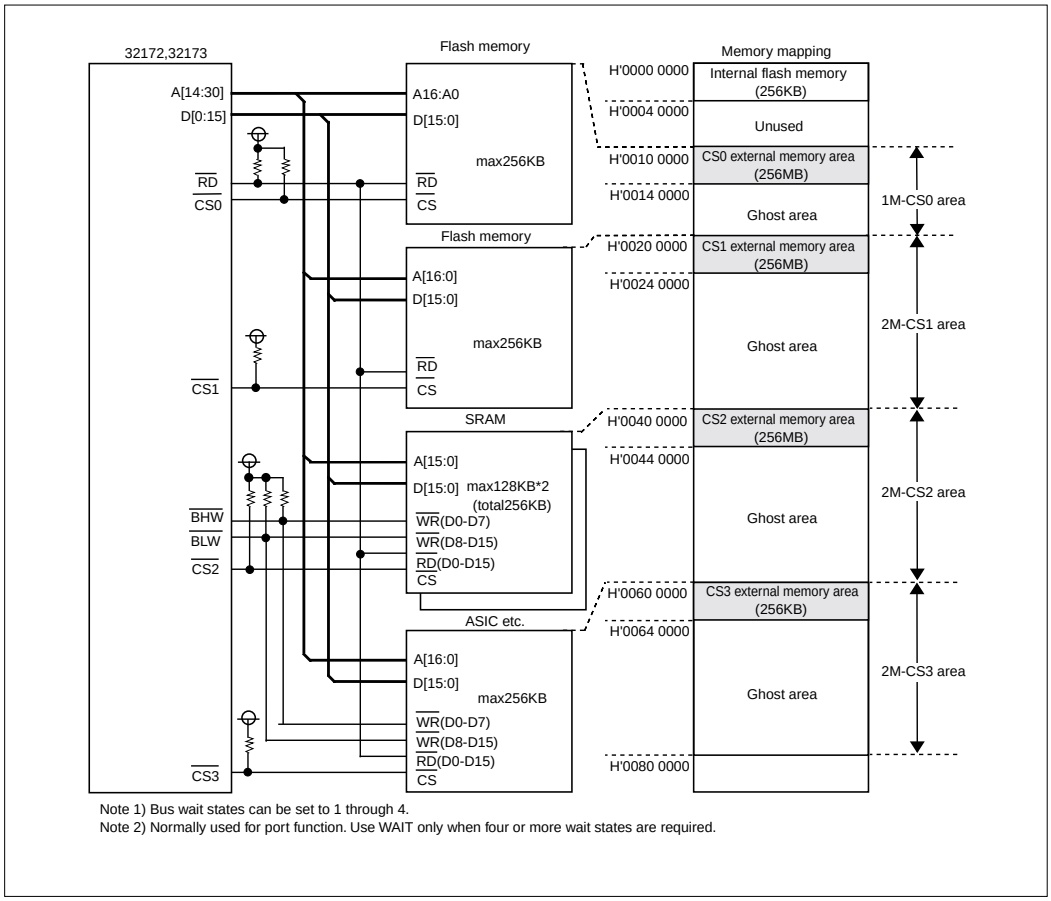
Note 1) Bus wait states can be set to 1 through 4.
Note 2) Normally used for port function. Use WAIT only when four or more wait states are required.

Figure 17.4.2 Typical Connection of External Extension Memory Using Three Memory Blocks (when BUSMOD = 0)

Note: The M32R/E address and data are comprised of bit 0 = MSB and bit 15 = LSB. When connecting external extension memory to the chip, the MSB and the LSB sides must be reversed.

(3) When Bus Mode Control Register = 0 (with four memory blocks connected)

Figure 17.4.3 shows a typical connection diagram for the microcomputer when using external extension memory (external extended mode).



Note 1) Bus wait states can be set to 1 through 4.
Note 2) Normally used for port function. Use WAIT only when four or more wait states are required.

Figure 17.4.3 Typical Connection of External Extension Memory Using four Memory Blocks (when BUSMOD = 0)

Note: The M32R/E address and data are comprised of bit 0 = MSB and bit 15 = LSB. When connecting external extension memory to the chip, the MSB and the LSB sides must be reversed.

(4) When Bus Mode Control Register = 1 (with two memory blocks connected)

Figure 17.4.4 shows a typical connection diagram for the microcomputer when using external extension memory (external extended mode).

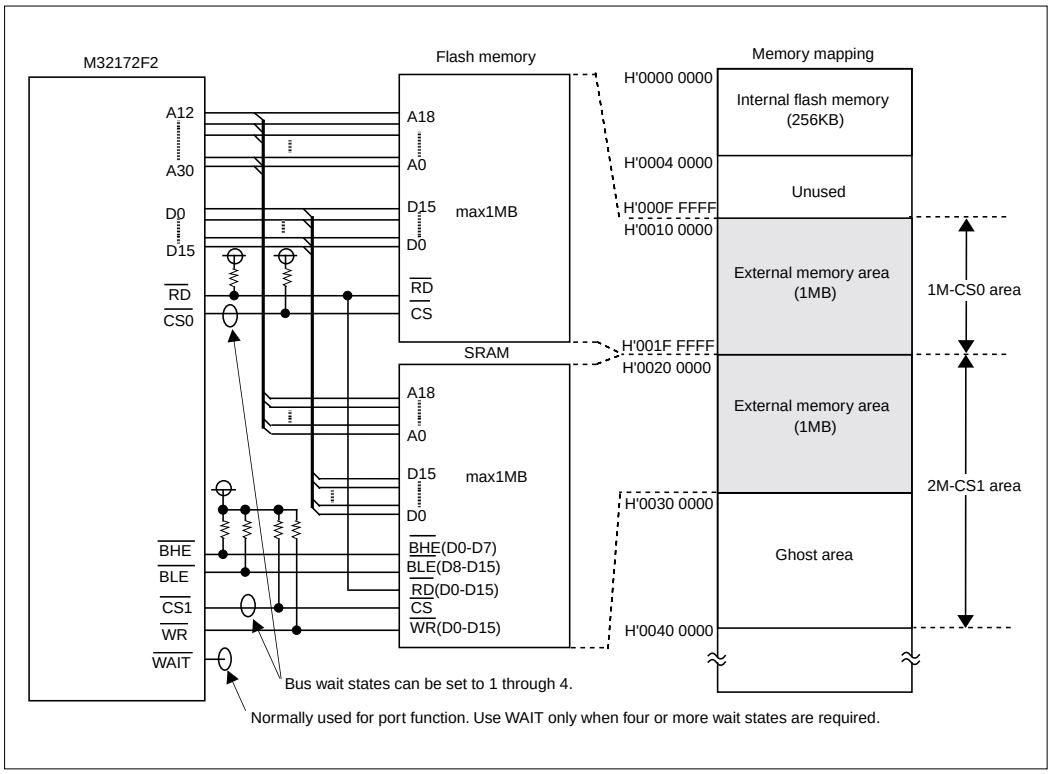


Figure 17.4.4 Typical Connection of External Extension Memory Using Two Memory Blocks (when BUSMOD = 1)

Note: The M32R/E address and data are comprised of bit 0 = MSB and bit 15 = LSB. When connecting external extension memory to the chip, the MSB and the LSB sides must be reversed.

(5) When Bus Mode Control Register = 1 using 8/16-bit data bus memory blocks in combination

Figure 17.4.3 shows a typical connection diagram for the microcomputer when using an 8-bit data bus memory which is located in the CS0 area and a 16-bit data bus memory which is located in the CS1 area. (External extension memory can only be used in external extended and processor modes).

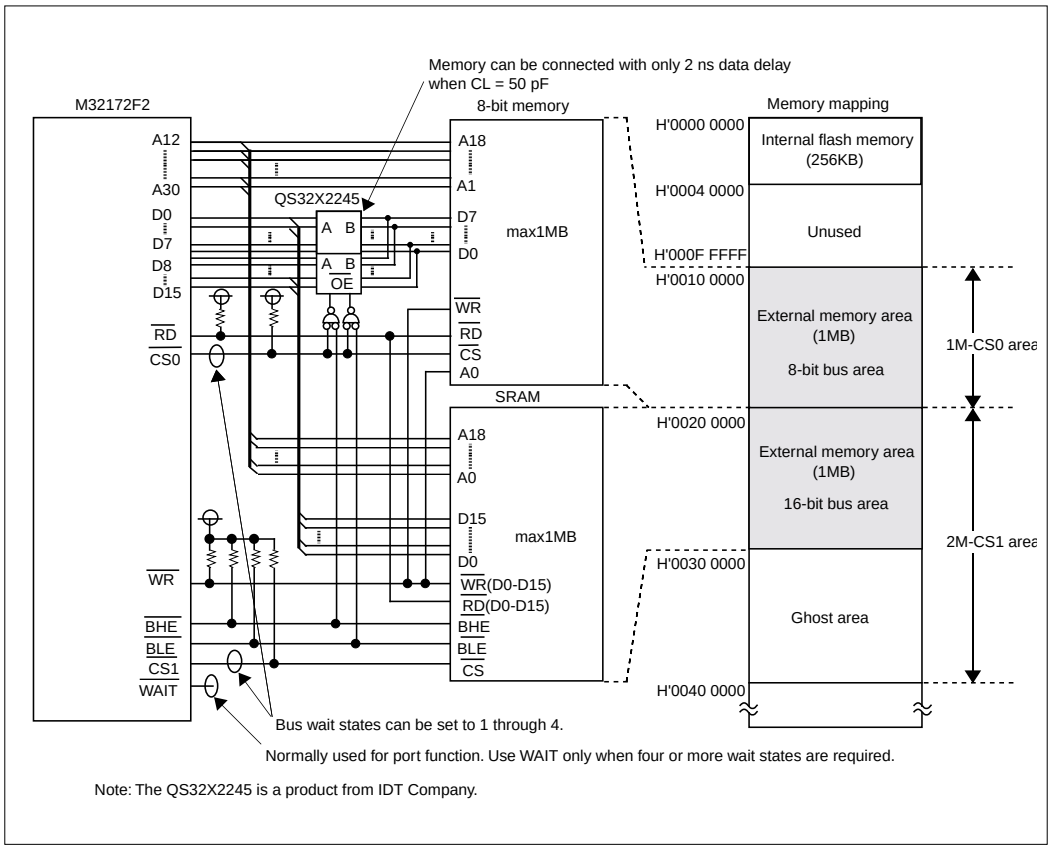


Figure 17.4.5 Typical Connection of External Extension Memory for the M32172 (when BUSMOD = 1 using 8/16-bit Memory Blocks in Combination)

Note: The M32R/E address and data are comprised of bit 0 = MSB and bit 15 = LSB. When connecting external extension memory to the chip, the MSB and the LSB sides must be reversed.

CHAPTER 18
WAIT CONTROLLER

18.1 Outline of the Wait Controller

The Wait Controller controls the number of wait cycle inserted in the bus cycle when accessing an external extended area. The Wait Controller is outlined below.

Table 18.1.1 Outline of the Wait Controller

Item	Specification
Target space	Following spaces are controlled depending on operation mode: Single-chip mode : No target space (Settings made to the Wait Controller have no effect) External extended mode : CS0 area (1M Mbytes), CS1 area (2 Mbytes), CS2 area (2 Mbytes), or CS3 area (2 Mbytes) Processor mode : CS0 area (2 Mbytes), CS1 area (2 Mbytes)
Number of wait cycles that can be inserted	One to four wait cycles set in software + any wait cycles inserted by $\overline{\text{WAIT}}$ pin input (Bus cycle during external access has at least one wait cycle inserted which is the shortest access possible.)

In external extended, four chip select signals ($\overline{\text{CS0}}$, $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, and $\overline{\text{CS3}}$) are output for external extended areas. The four external extended areas corresponding to these four chip select signals respectively are referred to as the CS0, CS1, CS2, and CS3 areas.

In processor mode, two chip select signals ($\overline{\text{CS0}}$ and $\overline{\text{CS1}}$) are output for external extended areas. The two external extended areas corresponding to these two chip select signals respectively are referred to as the CS0 and CS1 areas.

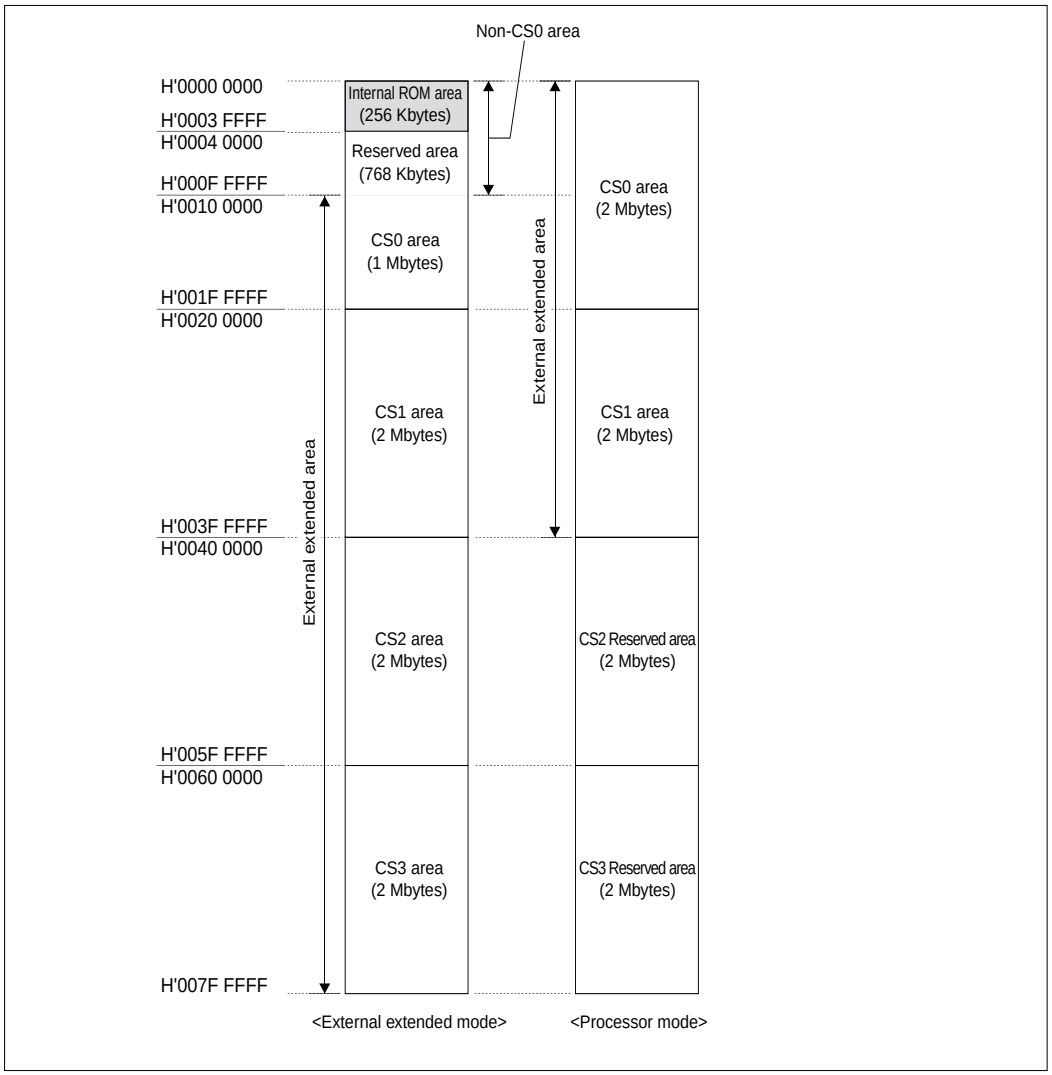


Figure 18.1.1 Address Map of the CS0 through CS3 Areas

When accessing an external extended area, the Wait Controller controls the number of wait cycles to be inserted in the bus cycle by checking the number of wait cycles set in software and those set by an input signal from the $\overline{\text{WAIT}}$ pin.

The number of wait cycles that can be controlled by software is 1 to 4. (External access incurs at least one wait cycle which is the shortest bus cycle possible.)

If the $\overline{\text{WAIT}}$ pin input remains low in the last cycle of the internal wait state set by software, the wait cycle is extended. Then when the $\overline{\text{WAIT}}$ pin input is released back high, the wait cycle is terminated and the next new bus cycle occurs.

Table 18.1.2 Number of Wait Cycles That Can Be Set by the Wait Controller during External Extended Mode

Valid chip select signal	External extended area	Address	Number of wait cycles inserted
CS0 , CS1	CS0 area	H'0010 0000–H'001F FFFF	One to four wait cycles inserted by setting in software + any wait cycles inserted by $\overline{\text{WAIT}}$ pin input (Software settings have priority, however.)
	CS1 area	H'0020 0000–H'002F FFFF (Note 1)	
CS0 , CS1, CS2	CS0 area	H'0010 0000–H'0017 FFFF (Note 2)	
	CS1 area	H'0020 0000–H'0027 FFFF (Note 3)	
	CS2 area	H'0040 0000–H'0047 FFFF (Note 4)	
CS0 , CS1, CS3	CS0 area	H'0010 0000–H'0013 FFFF, H'0018 0000–H'001B FFFF (Note 5)	
	CS1 area	H'0020 0000–H'0023 FFFF, H'0028 0000–H'002B FFFF (Note 6)	
	CS3 area	H'0060 0000–H'0063 FFFF H'0068 0000–H'006B FFFF (Note 7)	
CS0 , CS1, CS2, CS3	CS0 area	H'0010 0000–H'0013 FFFF (Note 8)	
	CS1 area	H'0020 0000–H'0023 FFFF (Note 9)	
	CS2 area	H'0040 0000–H'0043 FFFF (Note 10)	
	CS3 area	H'0060 0000–H'0063 FFFF (Note 11)	

Note 1: A ghost of the CS1 area appears in the area ranging in address from H'0030 0000 to H'003F FFFF.
Note 2: A ghost of the CS0 area appears in the area ranging in address from H'0018 0000 to H'001F FFFF.
Note 3: A ghost of the CS1 area appears in the area ranging in address from H'0028 0000 to H'003F FFFF.
Note 4: A ghost of the CS2 area appears in the area ranging in address from H'0048 0000 to H'005F FFFF.
Note 5: A ghost of the CS0 area appears in areas ranging in address from H'0014 0000 to H'0017 FFFF and from H'001C 0000 to H'001F FFFF.
Note 6: A ghost of the CS1 area appears in areas ranging in address from H'0024 0000 to H'0027 FFFF and from H'002C 0000 to H'003F FFFF.
Note 7: A ghost of the CS3 area appears in areas ranging in address from H'0064 0000 to H'0067 FFFF and from H'006C 0000 to H'007F FFFF.
Note 8: A ghost of the CS0 area appears in the area ranging in address from H'0014 0000 to H'001F FFFF.
Note 9: A ghost of the CS1 area appears in the area ranging in address from H'0024 0000 to H'003F FFFF.
Note 10: A ghost of the CS2 area appears in the area ranging in address from H'0044 0000 to H'005F FFFF.
Note 11: A ghost of the CS3 area appears in the area ranging in address from H'0064 0000 to H'007F FFFF.

Table 18.1.3 Number of Wait Cycles That Can Be Set by the Wait Controller during Processor Mode

External extended area	Address	Number of wait cycles inserted
CS0 area	H'0000 0000–H'000F FFFF (Note 1)	One to four wait cycles inserted by setting in software + any wait cycles inserted by $\overline{\text{WAIT}}$ pin input (Software settings have priority, however.)
CS1 area	H'0020 0000–H'002F FFFF (Note 2)	

Note 1: A (1 Mbyte) ghost of the CS0 area appears in the area ranging in address from H'0010 0000 to H'001F FFFF.
Note 2: A (1 Mbyte) ghost of the CS1 area appears in the area ranging in address from H'0030 0000 to H'003F FFFF.
Note 3: There is no CS2 and CS3 area in the processor mode.

18.2 Wait Controller Related Registers

A wait controller related register map is shown below.

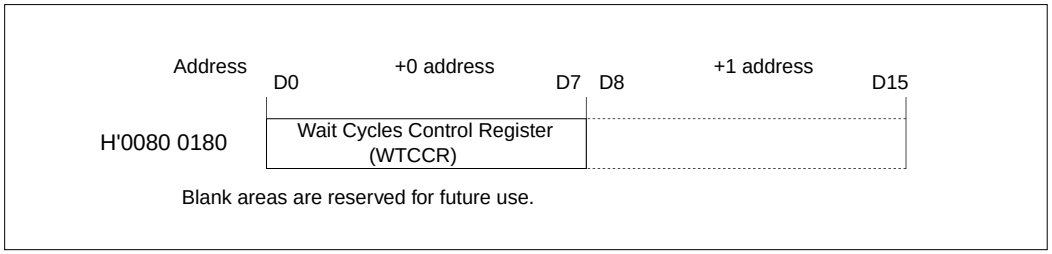
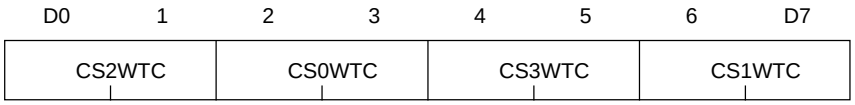


Figure 18.2.1 Wait Controller Related Register Map

18.2.1 Wait States Control Register

■ **Wait States Control Register (WTCCR)** **<Address: H'0080 0180>**



<When reset: H'00>

D	Bit Name	Function	R	W
0,1	CS2WTC (Control the number of CS2 wait states)	00: 4 wait cycles (when reset) 01: 3 wait cycles 10: 2 wait cycles 11: 1 wait cycle	○	○
2,3	CS0WTC (Control the number of CS0 wait states)	00: 4 wait cycles (when reset) 01: 3 wait cycles 10: 2 wait cycles 11: 1 wait cycle	○	○
4,5	CS3WTC (Control the number of CS3 wait states)	00: 4 wait cycles (when reset) 01: 3 wait cycles 10: 2 wait cycles 11: 1 wait cycle	○	○
6,7	CS1WTC (Control the number of CS1 wait states)	00: 4 wait cycles (when reset) 01: 3 wait cycles 10: 2 wait cycles 11: 1 wait cycle	○	○

18.3 Typical Operation of the Wait Controller

The following shows a typical operation of the wait controller. The wait controller can control bus access in the range of 2 to 5 cycles. If more access cycles than that are needed, use the WAIT function in combination with the wait controller.

(1) When Bus Mode Control Register = 0

External read/write operations are performed using the address bus, data bus, and signals $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{RD}$, $\overline{BHW}$, $\overline{BLW}$, $\overline{WAIT}$, and BCLK.

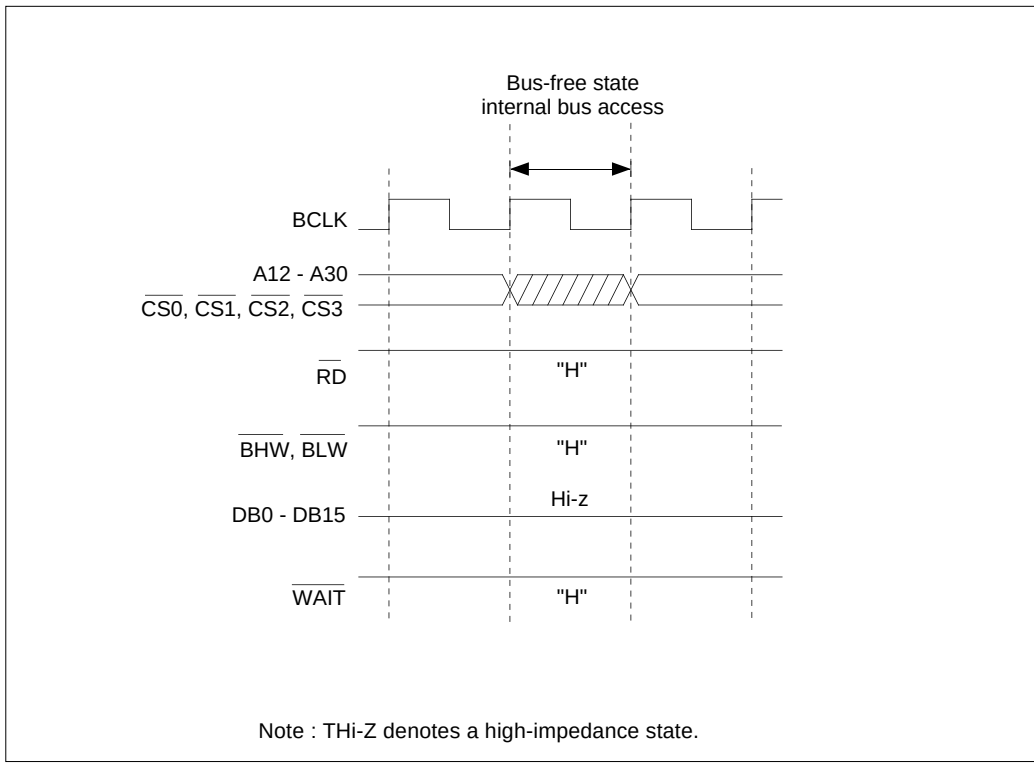
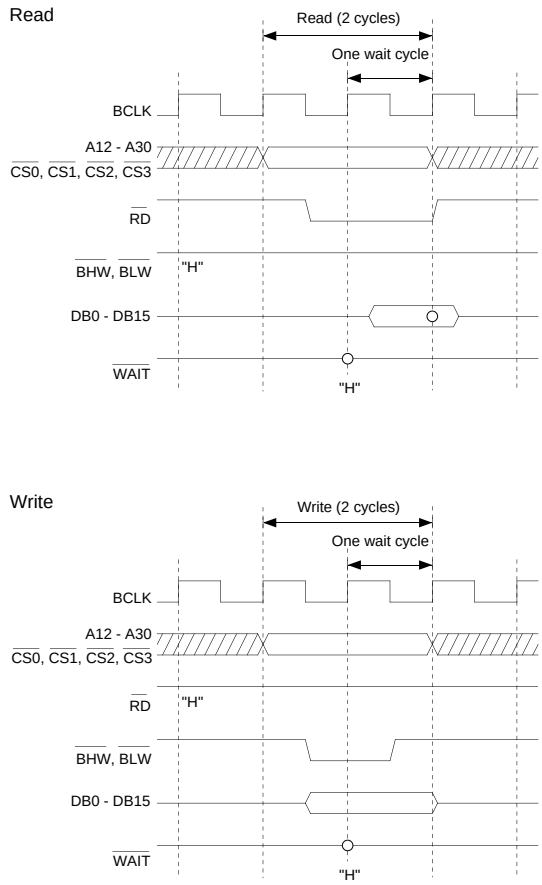
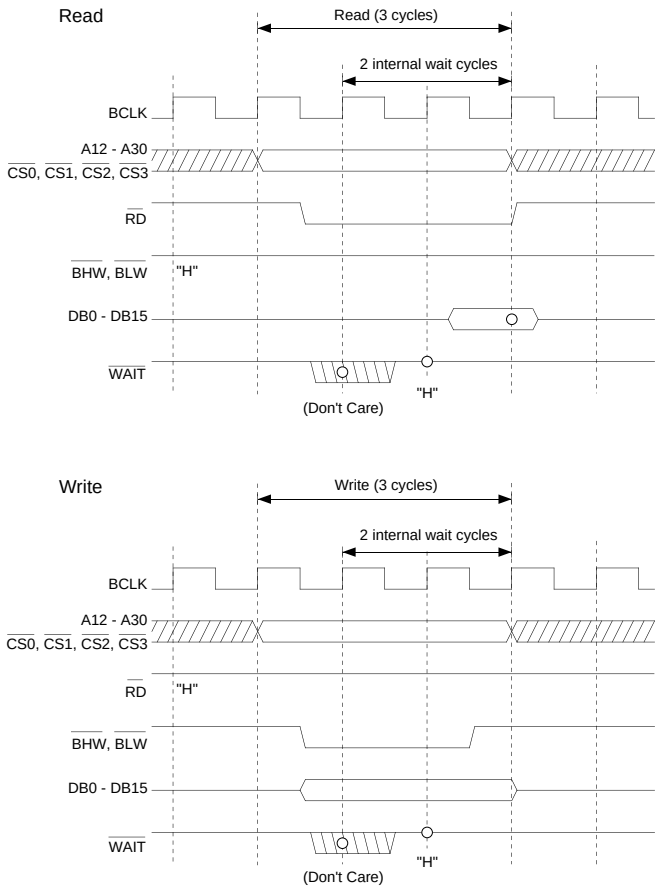


Figure 18.3.1 Internal Bus Access during Bus Free State



Note : Circles ○ above indicate points at which signals are sampled.

Figure 18.3.2 Read/Write Timing (for Access with 1 Internal Wait Cycle)



Note : Circles ○ above indicate points at which signals are sampled.

Figure 18.3.3 Read/Write Timing (for Access with 2 Internal Wait Cycles)

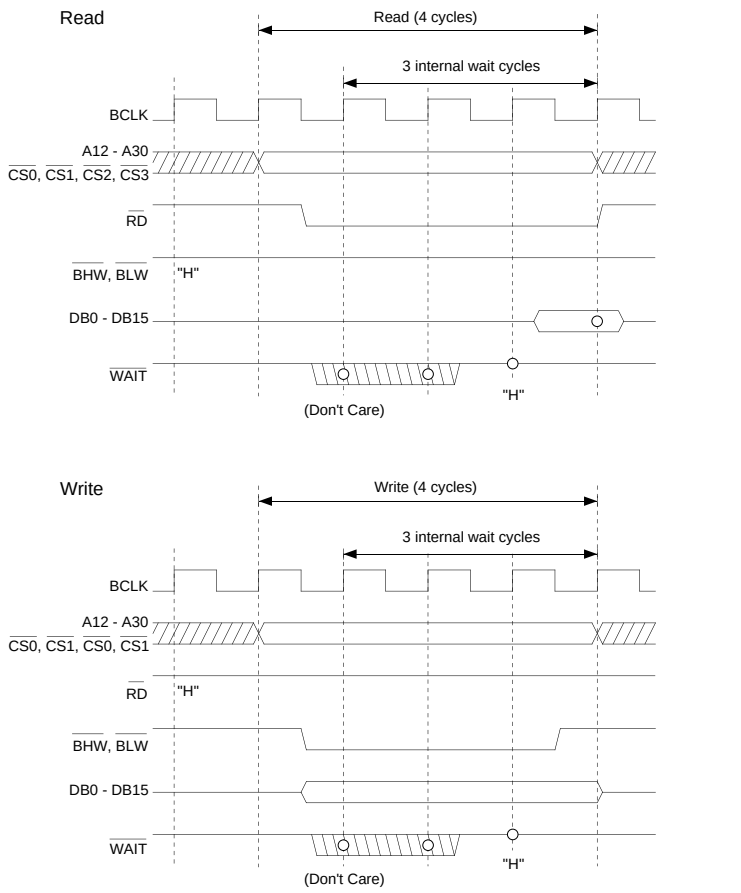


Figure 18.3.4 Read/Write Timing (for Access with 3 Internal Wait Cycles)

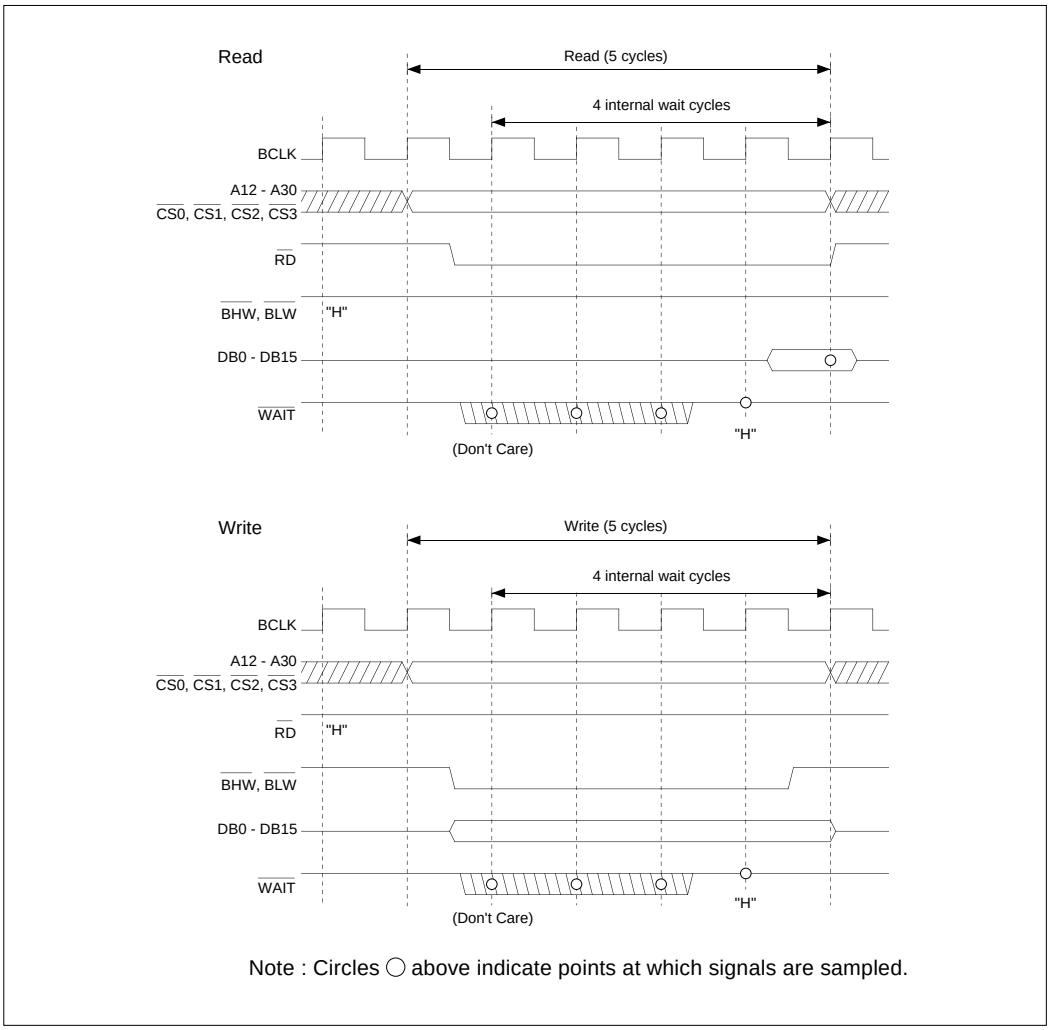
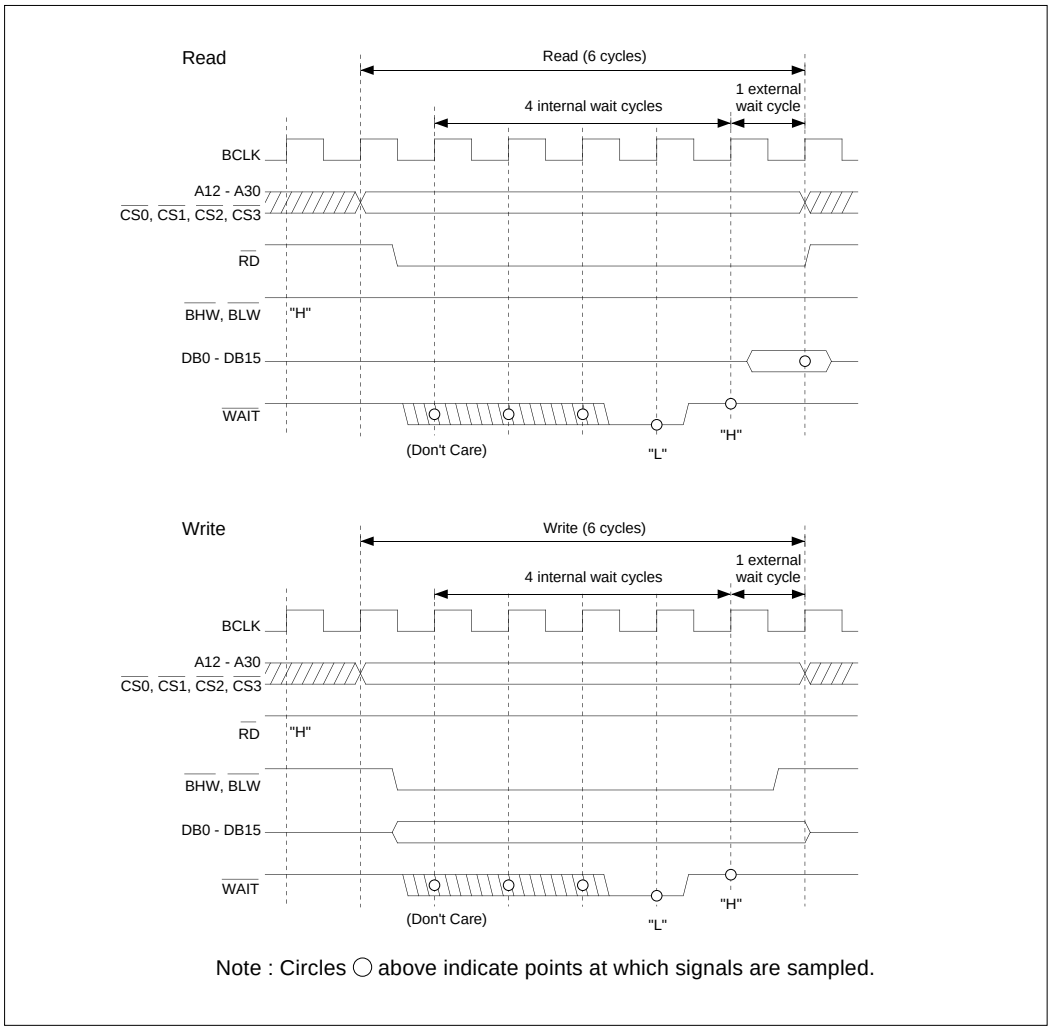


Figure 18.3.5 Read/Write Timing (for Access with 4 Internal Wait Cycles)



Note : Circles ○ above indicate points at which signals are sampled.

Figure 18.3.6 Read/Write Timing (for Access with 4 Internal and 1 External Wait Cycles)

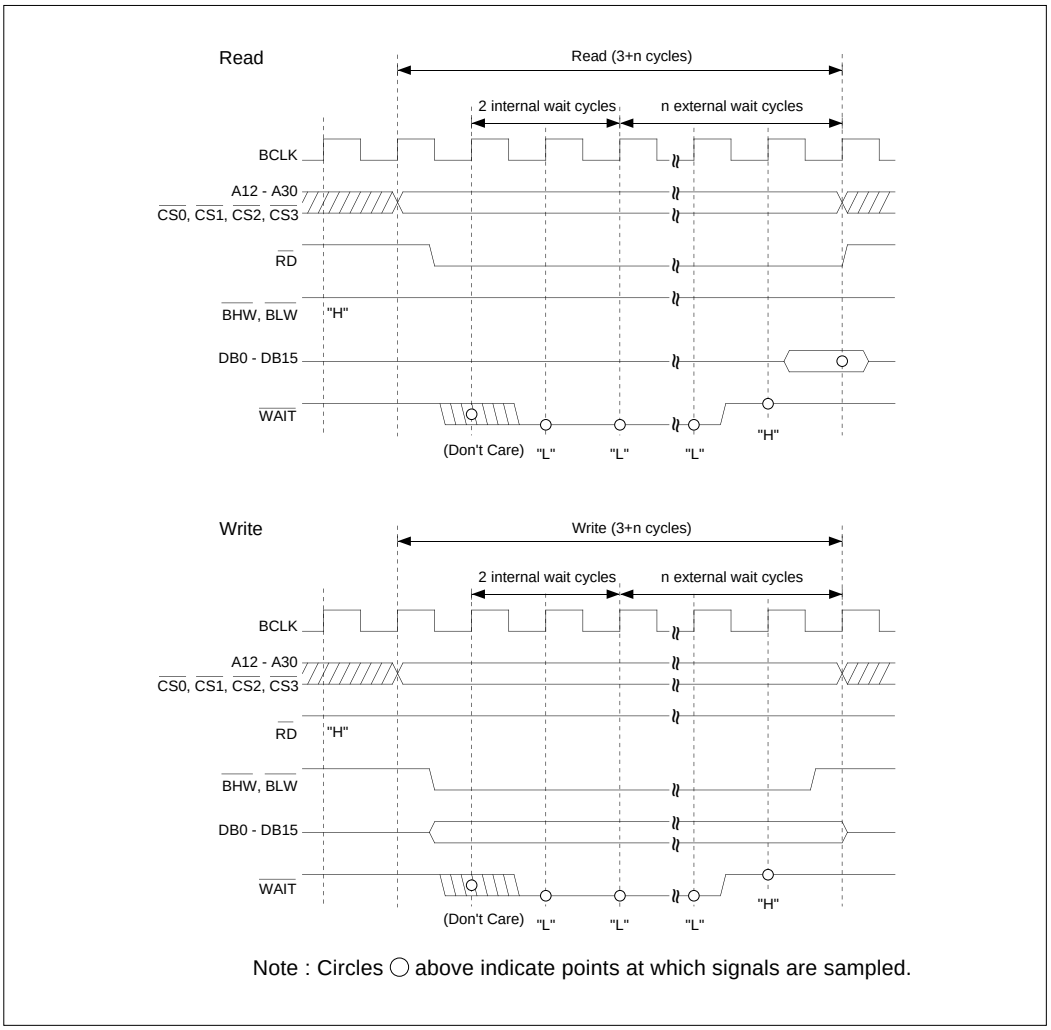


Figure 18.3.7 Read/Write Timing (for Access with 2 Internal and n External Wait Cycles)

(2) When Bus Mode Control Register = 1

External read/write operations are performed using the address bus, data bus, and signals $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{RD}$, $\overline{BHE}$, $\overline{BLE}$, $\overline{WAIT}$, and $\overline{WR}$.

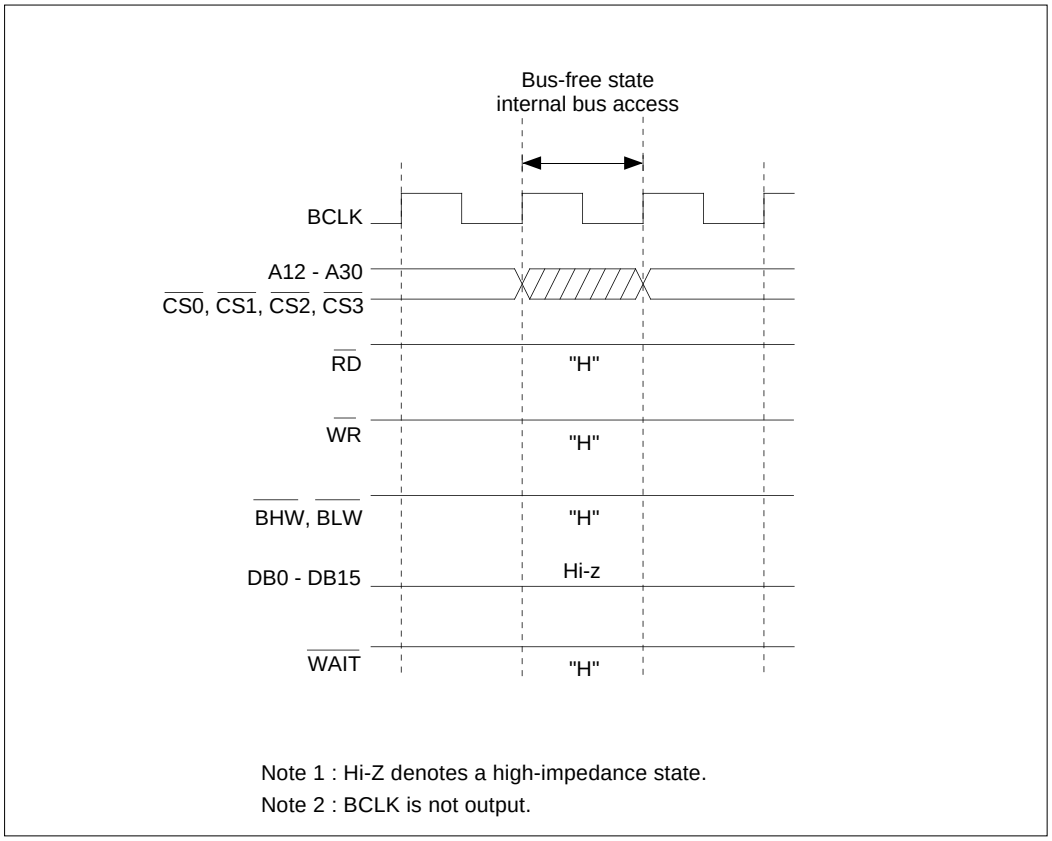
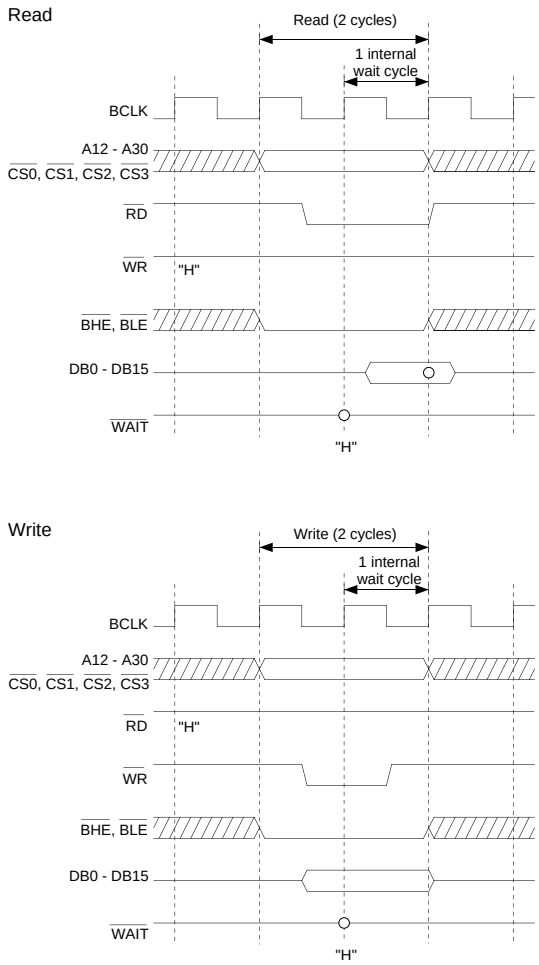
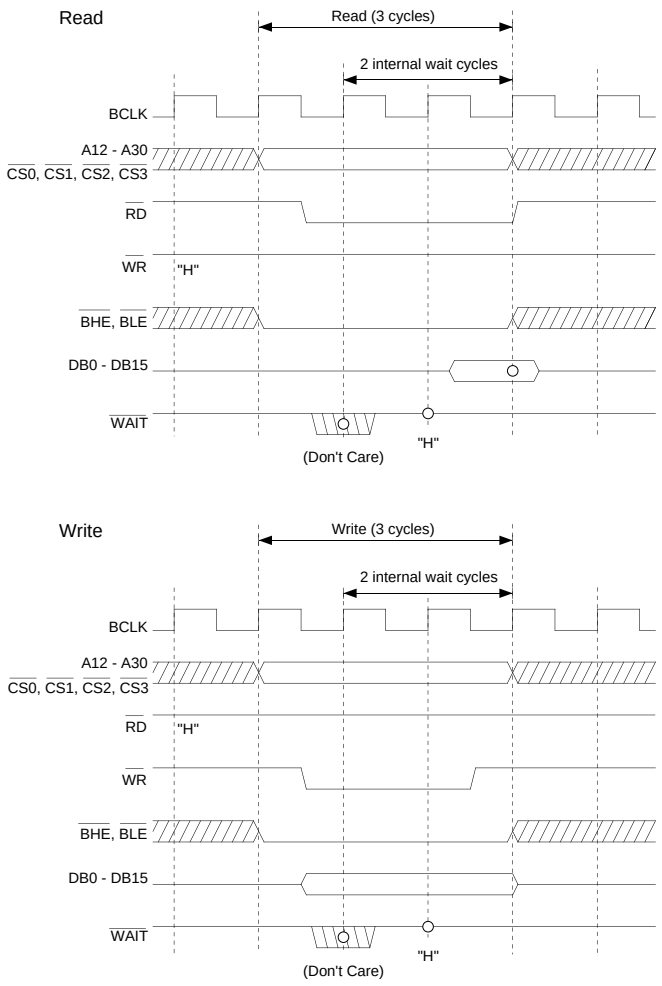


Figure 18.3.8 Internal Bus Access during Bus Free State



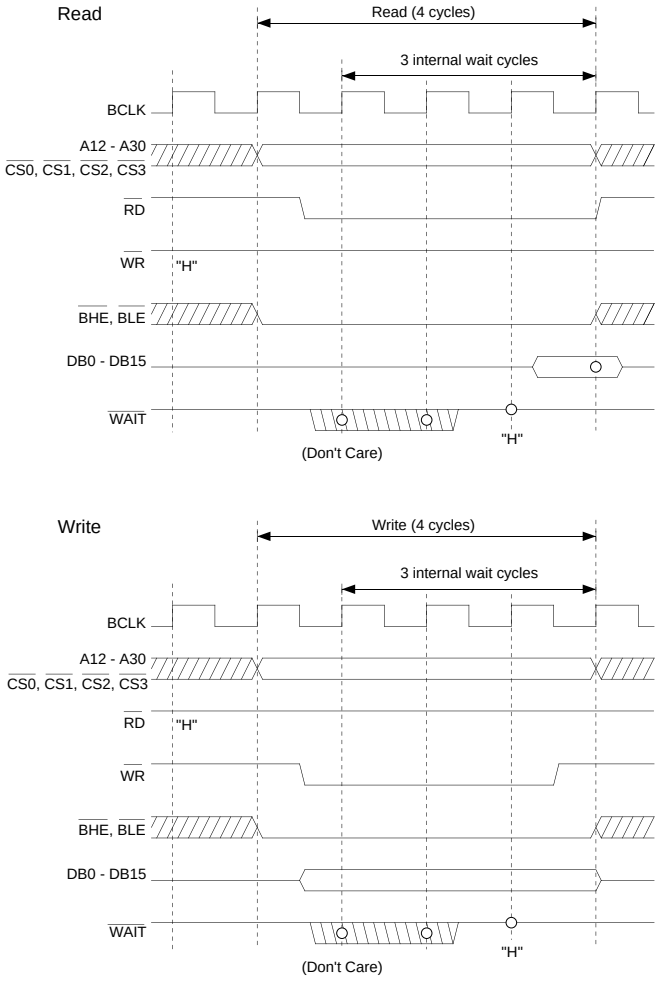
Note 1 : Circles ○ above indicate points at which signals are sampled.
Note 2 : BCLK is not output.

Figure 18.3.9 Read/Write Timing (for Access with 1 Internal Wait Cycle)



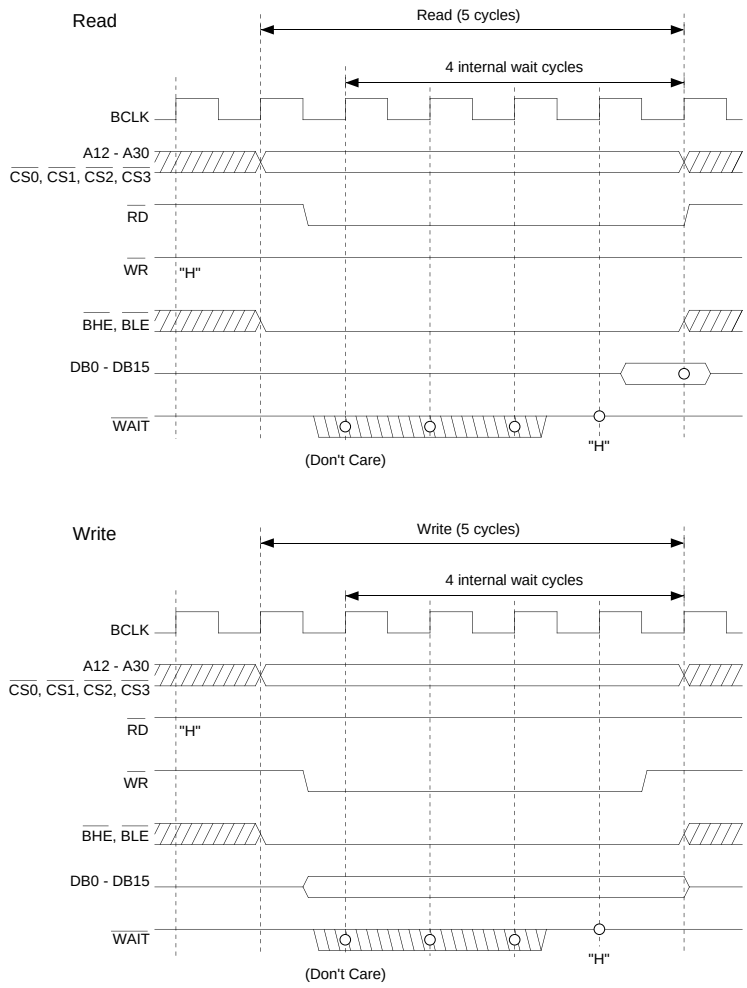
Note 1 : Circles ○ above indicate points at which signals are sampled.
Note 2 : BCLK is not output.

Figure 18.3.10 Read/Write Timing (for Access with 2 Internal Wait Cycles)



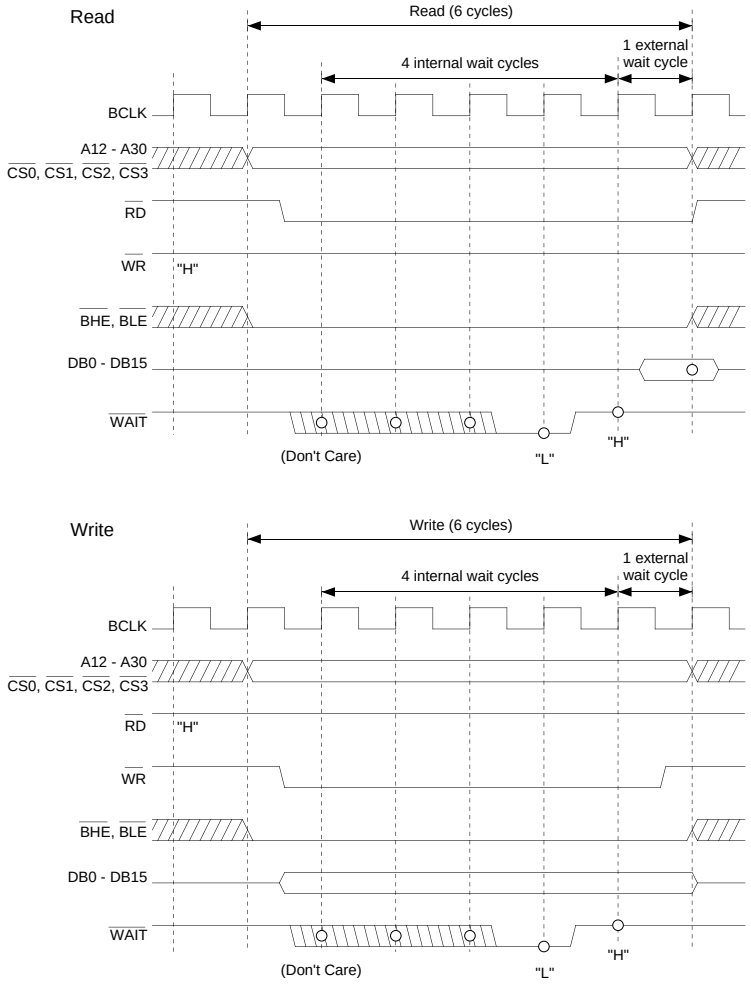
Note 1 : Circles ○ above indicate points at which signals are sampled.
Note 2 : BCLK is not output.

Figure 18.3.11 Read/Write Timing (for Access with 3 Internal Wait Cycles)



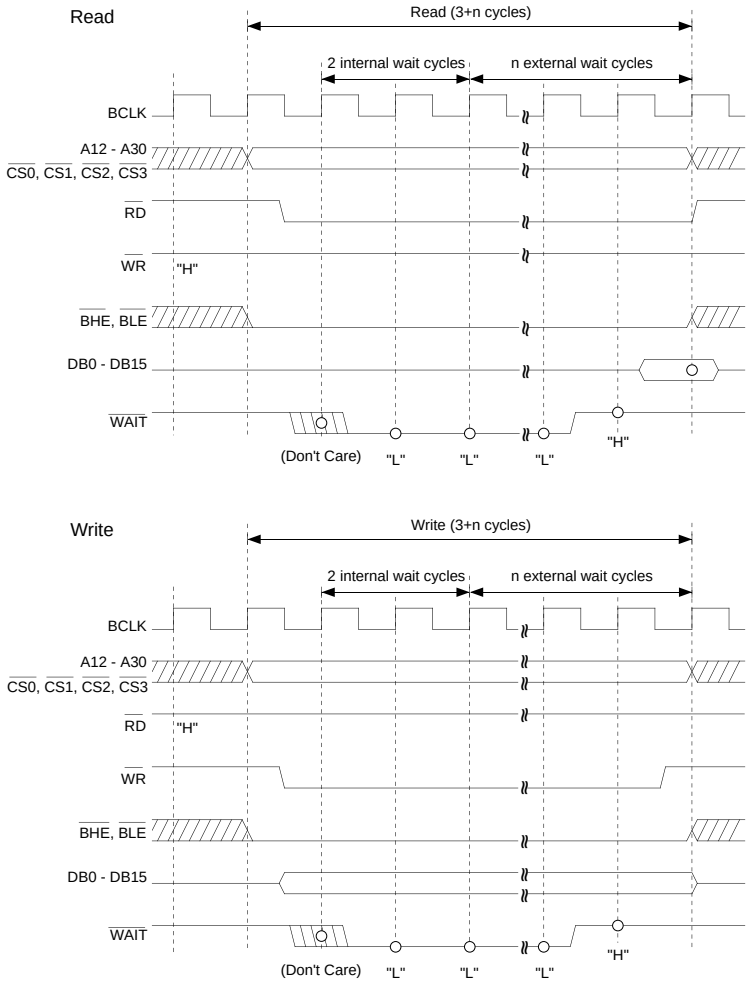
Note 1 : Circles ○ above indicate points at which signals are sampled.
Note 2 : BCLK is not output.

Figure 18.3.12 Read/Write Timing (for Access with 4 Internal Wait Cycles)



Note 1 : Circles ○ above indicate points at which signals are sampled.
Note 2 : BCLK is not output.

Figure 18.3.13 Read/Write Timing (for Access with 4 Internal and 1 External Wait Cycles)



Note 1 : Circles ○ above indicate points at which signals are sampled.
Note 2 : BCLK is not output.

Figure 18.3.14 Read/Write Timing (for Access with 2 Internal and n External Wait Cycles)

* This is a blank page.*

CHAPTER 19

RAM BACKUP MODE

19.1 Outline

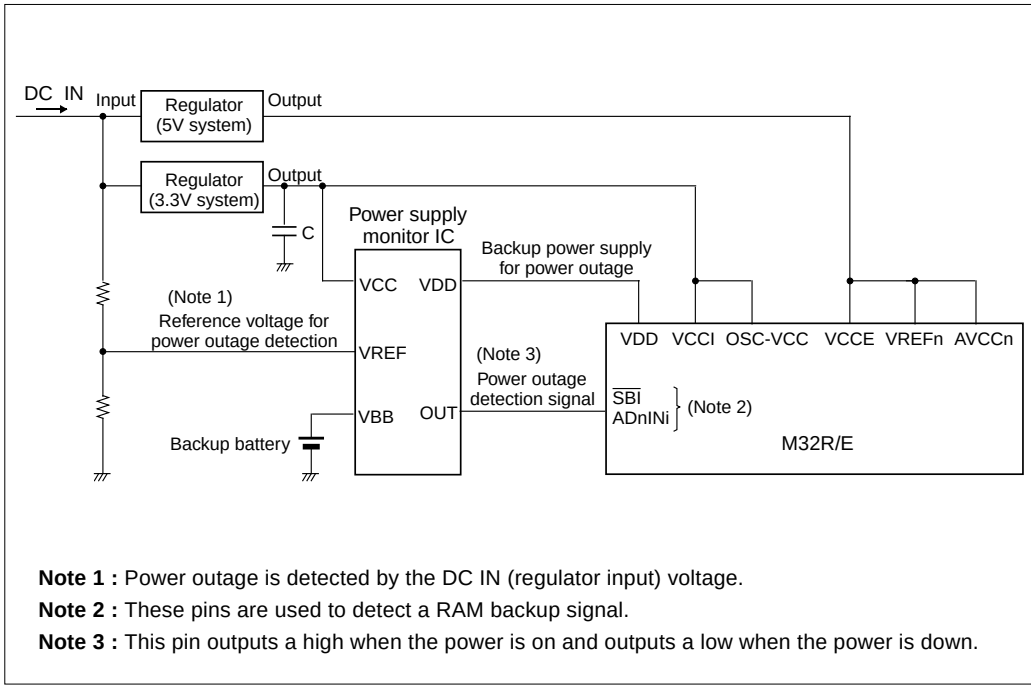
In RAM backup mode, the contents of the internal RAM are retained while the power is turned off. RAM backup mode is used for the following two purposes:

- Back up the internal RAM data when the power is down
- Turn off the power to the CPU whenever necessary to save on the system's power consumption

The 32R/E CPU is placed in RAM backup mode by applying a voltage of 2.0-3.3 V to the VDD 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. Also, because all pins except VDD are held low during RAM backup mode, power consumption in the system can effectively reduced.

19.2 Example of RAM Backup when Power is Down

A typical circuit for RAM backup at power outage is shown in Figure 19.2.1. The following explains how the RAM can be backed up by using this circuit as an example.

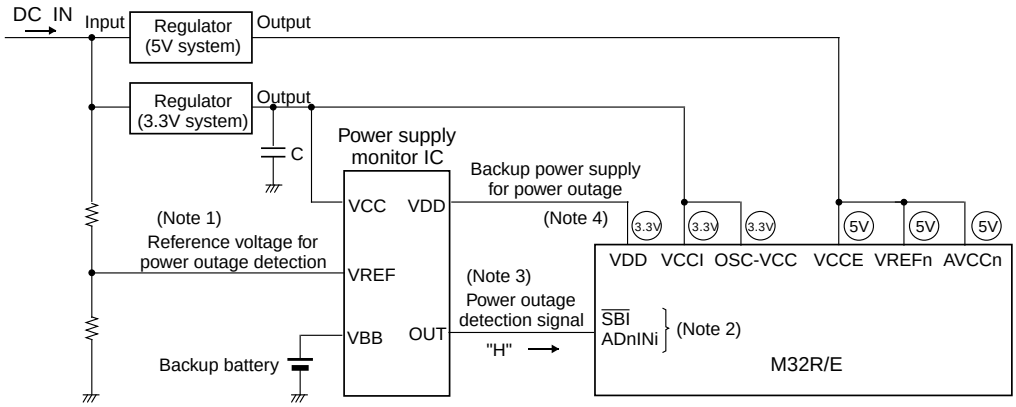


Note 1 : Power outage is detected by the DC IN (regulator input) voltage.
Note 2 : These pins are used to detect a RAM backup signal.
Note 3 : This pin outputs a high when the power is on and outputs a low when the power is down.

Figure 19.2.1 Typical Circuit for RAM Backup at Power Outage

19.2.1 Normal Operating State

Figure 19.2.2 shows the normal operating state of the M32R/E. During normal operation, input on the $\overline{\text{SBI}}$ pin or ADnINi (i = 0-15) pin used for RAM backup signal detection remains high.



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 high when the power is on and outputs a low when the power is down.
Note 4 : Backup power supply = 2.0 to 3.3 V

Figure 19.2.2 Normal Operating State

19.2.2 RAM Backup State

Shown in Figure 19.2.3 is the power outage RAM backup state of the M32R/E. When the power supply goes down, the power supply monitor IC starts feeding current from the backup battery to the M32R/E. Also, the power supply monitor IC's power outage detection pin outputs a low, causing the $\overline{SBI}$ pin or ADnINi pin input to go low, which generates a RAM backup signal (① in Figure 17.2.3). Whether the power is down or not must be determined 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 settings.

(1) Create check data to verify after returning from RAM backup to normal mode whether the RAM data has been retained normally (② in Figure 19.2.3).

When the power supply to VCC goes down after settings in (1), the voltage applied to the VDD pin becomes 2.0-3.3 V and voltages applied to all other pins drop to 0 V, and the M32R/E thereby enters RAM backup mode (③ in Figure 19.2.3).

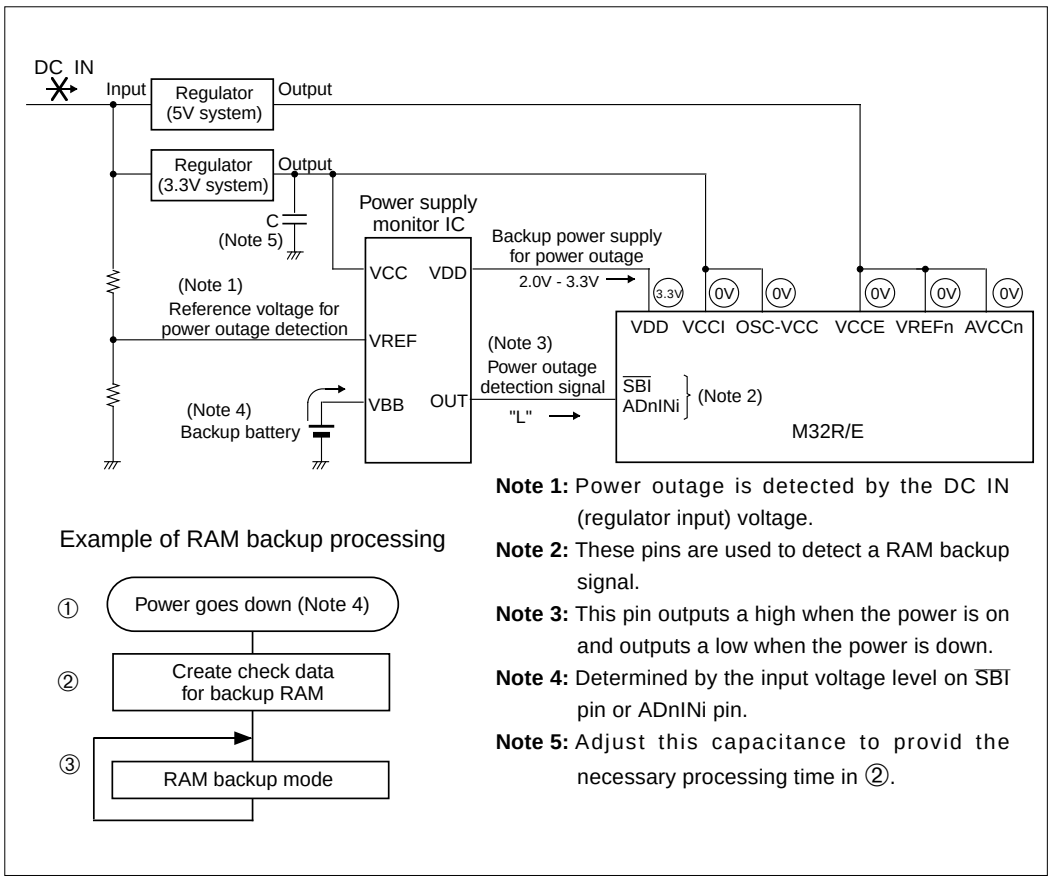
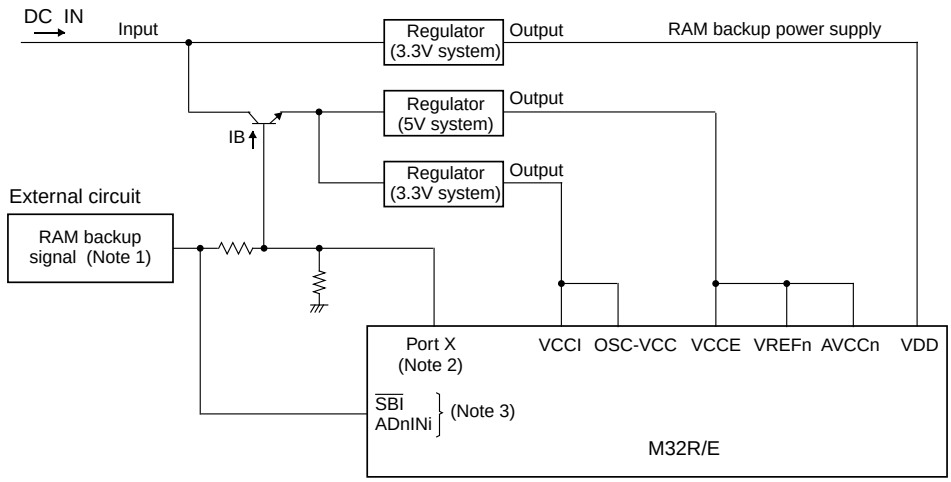


Figure 19.2.3 RAM Backup State at Power Outage

19.3 Example of RAM Backup for Saving Power Consumption

Figure 19.3.1 shows a typical circuit for RAM backup to save on power consumption. The following explains how the RAM is backed up for the purpose of low-power operation by using this circuit as an example.



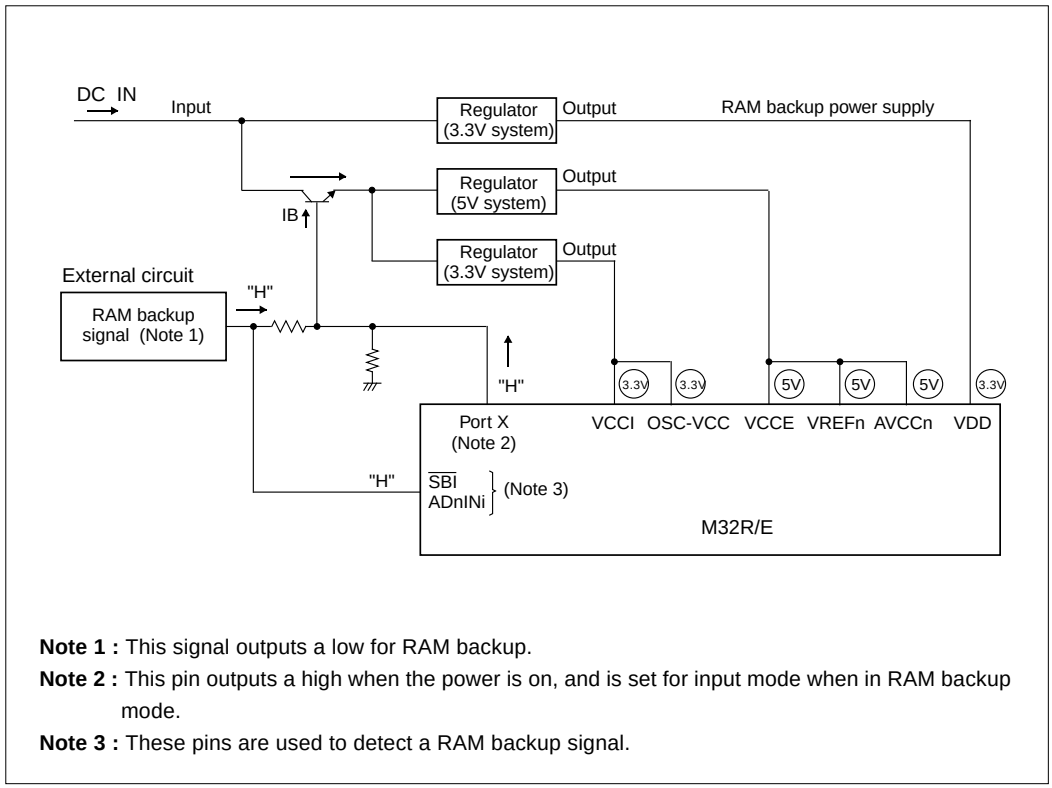
Note 1 : This signal outputs a low for RAM backup.
Note 2 : This pin outputs a high when the power is on, and is set for input mode when in RAM backup mode.
Note 3 : These pins are used to detect a RAM backup signal.

Figure 19.3.1 Typical Circuit for RAM Backup to Save on Power Consumption

19.3.1 Normal Operating State

Figure 19.3.2 shows the normal operating state of the M32R/E. During normal operation, the RAM backup signal output by the external signal is high. Also, input on the $\overline{SBI}$ pin or ADnINi (i = 0-15) pin used for RAM backup signal detection remains high.

Port X, which is the transistor's base connecting pin, should output a high. This causes the transistor's base voltage, IB, to go high, so that current is fed from the power supply to the VCC pin via the transistor.



Note 1 : This signal outputs a low for RAM backup.
Note 2 : This pin outputs a high 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 19.3.2 Normal Operating State

19.3.2 RAM Backup State

Figure 19.3.3 shows the RAM backup state of the M32R/E. Figure 19.3.4 shows a RAM backup sequence. When the external circuit outputs a low, input on the $\overline{SBI}$ pin or ADnINi pin goes low. A low on these input pins generates a RAM backup signal (A and ① in Figure 19.3.3). To enable RAM backup mode, make the following settings.

(1) Create check data to verify after returning from RAM backup to normal mode whether the RAM data has been retained normally (② in Figure 19.3.3).

(2) To materialize low-power operation, set all programmable input/output pins except port X for input mode (or for output mode, with pins outputting a low) (③ in Figure 19.3.3).

(3) Set port X for input mode (B and √ in Figure 19.3.3). This causes the transistor's base voltage, IB, to go low, so that no current flows from the power supply to the VCC pin via the transistor (C in Figure 19.3.3). Consequently, the power to the VCC pin is shut off (D in Figure 19.3.3).

Due to settings in (1) to (3), the voltage applied to the VDD pin becomes 3.3 V ± 10% and voltages applied to all other pins drop to 0 V, thus placing the M32R/E in RAM backup mode (√ in Figure 19.2.3).

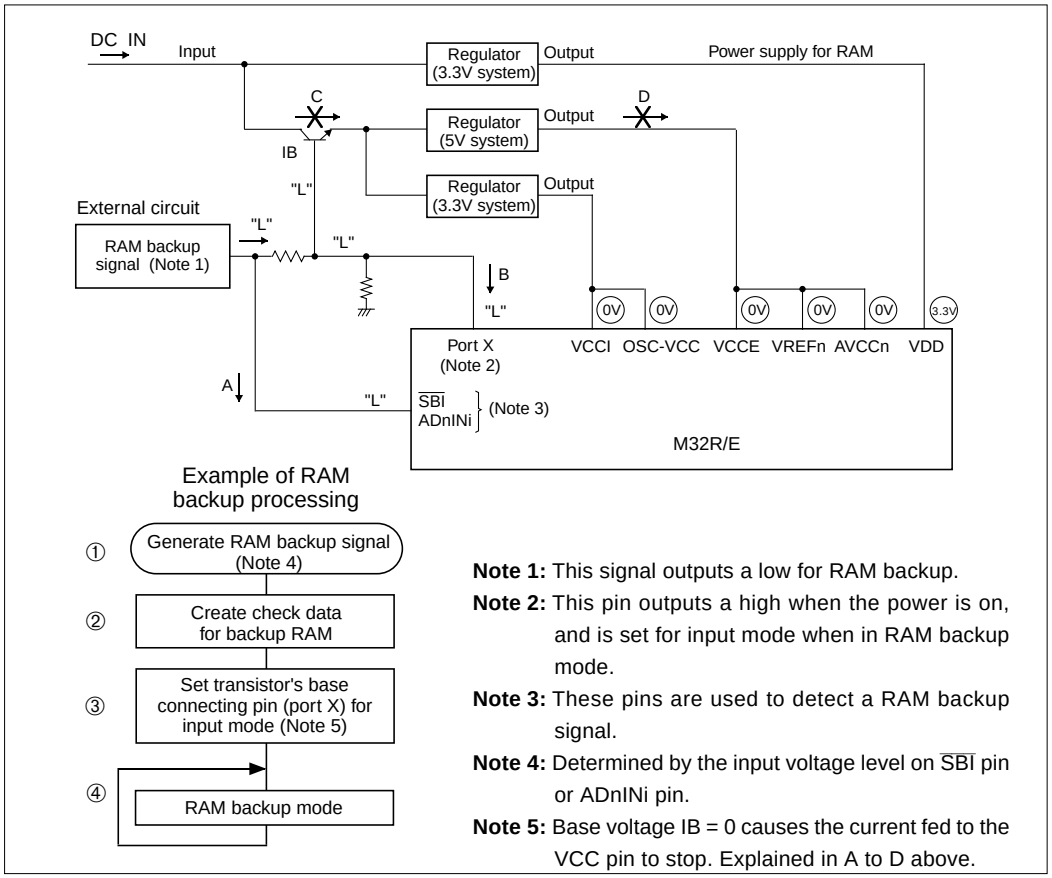


Figure 19.3.3 RAM Backup State for Low-Power Operation

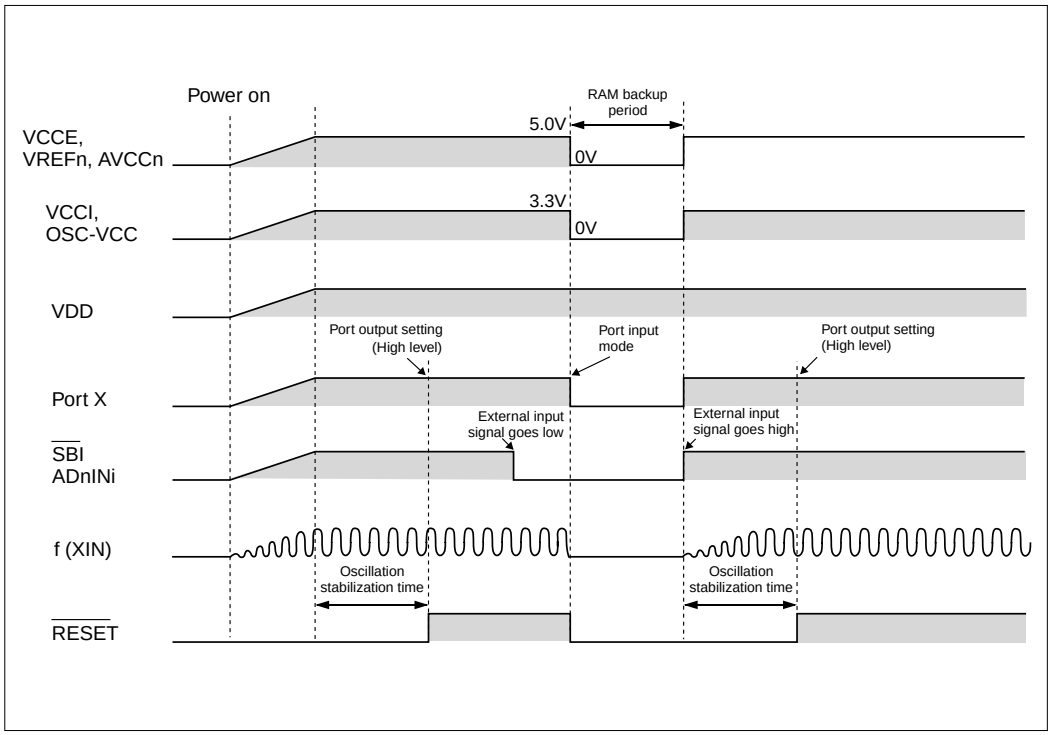


Figure 19.3.4 Example of RAM Backup Sequence for Low-Power Operation

19.3.3 Precautions to Be Observed at Power-on

When changing port X from input mode to output mode after power-on, pay attention to the following.

If port X is set for output mode while no data is set in the Port X Data Register, the port's initial output level is indeterminate. Therefore, be sure to set the output high level in the Port X Data Register before you set port X for output mode. Unless this method is followed, port output may go low at the same time port output is set after the clock oscillation has stabilized, causing the device to enter RAM backup mode.

19.4 Exiting RAM Backup Mode (Wakeup)

Processing to exit RAM backup mode and return to normal operation is referred to as "wakeup processing." Figure 19.4.1 shows an example of wakeup processing. Wakeup processing is initiated by reset input. The following shows how to execute wakeup processing.

(1) Reset the device (① in Figure 19.4.1). For details about reset, refer to Chapter 7, "Reset."

(2) Set port X for output mode and output a high from the port (② in Figure 19.4.1).(Note)

(3) Check the RAM contents against the check data created before entering RAM backup mode (③ in Figure 19.4.1).

(4) If the RAM contents and check data did not match when checked in (3), initialize the RAM (④ in Figure 19.4.1). If the RAM contents and check data matched, use the retained data in the program.

(5) After initializing each internal circuit (⑤ in Figure 19.4.1), return the main routine (⑥ in Figure 19.4.1).

Note: For wakeup from power outage RAM backup mode, settings for port X are unnecessary.

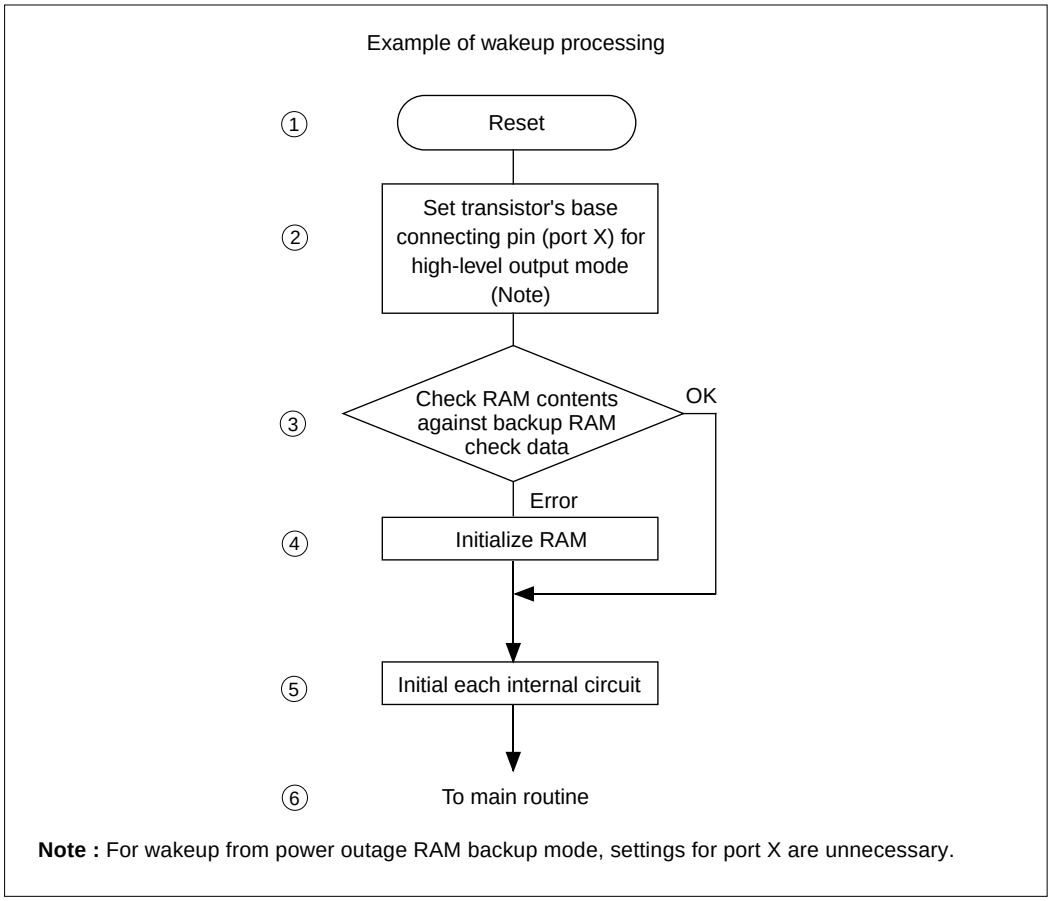


Figure 19.4.1 Wakeup Processing

* This is a blank page.*

CHAPTER 20
OSCILLATION CIRCUIT

20.1 Oscillator Circuit

The M32R/E contains an oscillator circuit that supplies operating clocks for the CPU core, internal peripheral I/O, and internal memory. The frequency fed to the clock input pin (XIN) is multiplied by 4 by the 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 internal peripheral clock, which is the operating clock for the internal peripheral I/O.

20.1.1 Example of an Oscillator Circuit

A clock generating circuit can be configured by connecting a ceramic (or crystal) resonator between the XIN and XOUT pins external to the chip. Figure 20.1.1 below shows an example of a system clock generating circuit using a resonator connected external to the chip and an RC network connected to the PLL circuit control pin (VCNT). For constants Rf, CIN, COUT, and Rd, consult your resonator manufacturer to determine the appropriate values.
When you use an externally sourced clock signal without using the internal oscillator circuit, connect the external clock signal to the XIN pin and leave the XOUT pin open.

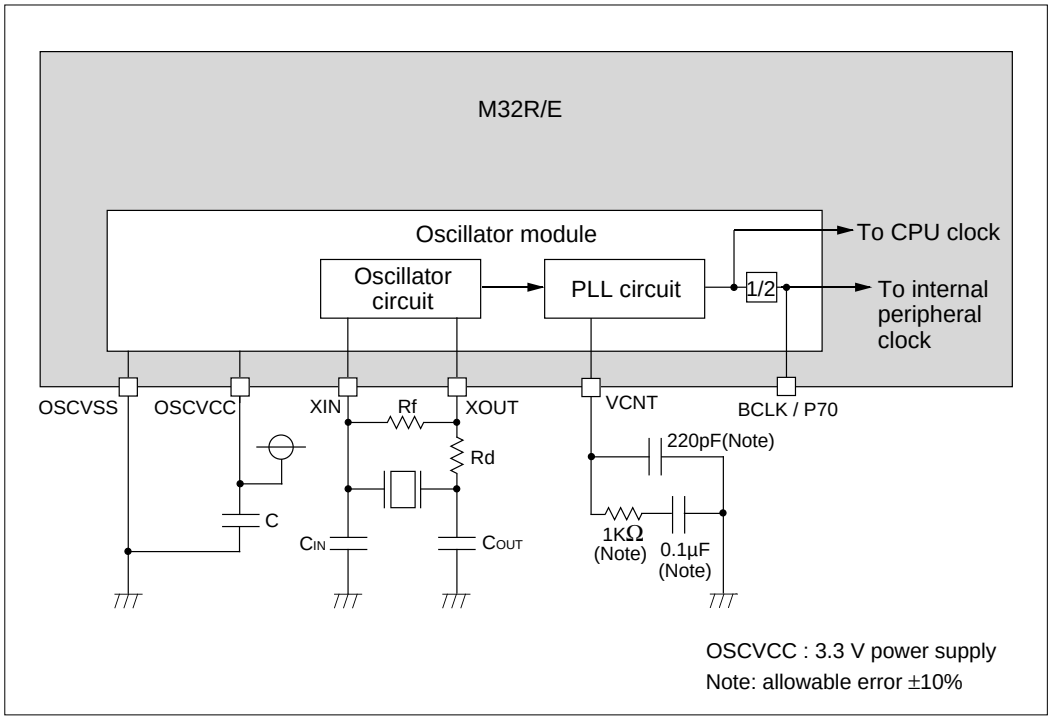


Figure 20.1.1 Example of a System Clock Generating Circuit

20.1.2 System Clock Output Function

A clock whose frequency is twice the input frequency can be output from the BCLK pin. The BCLK pin is shared with port P70. When you use this pin to output the system clock, set the P7 Operation Mode Register (P7MOD)'s D8 bit to 1. Configuration of the P7 Operation Mode Register is shown below.

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

D8	9	10	11	12	13	14	D15
P70MOD	P71MOD	P72MOD	P73MOD	P74MOD	P75MOD	P76MOD	P77MOD

<When reset : H'00>

D	Bit Name	Function	R	W
8	P70MOD	0 : P70	○	○
	(Port P70 operation mode)	1 : BCLK		
9	P71MOD	0 : P71	○	○
	(Port P71 operation mode)	1 : $\overline{\text{WAIT}}$		
10	P72MOD	0 : P72	○	○
	(Port P72 operation mode)	1 : $\overline{\text{HREQ}}$		
11	P73MOD	0 : P73	○	○
	(Port P73 operation mode)	1 : $\overline{\text{HACK}}$		
12	P74MOD	0 : P74	○	○
	(Port P74 operation mode)	1 : RTDTXD		
13	P75MOD	0 : P75	○	○
	(Port P75 operation mode)	1 : RTDRXD		
14	P76MOD	0 : P76	○	○
	(Port P76 operation mode)	1 : RTDACK		
15	P77MOD	0 : P77	○	○
	(Port P77 operation mode)	1 : RTDCLK		

20.1.3 Oscillation Stabilization Time at Power-on

The oscillator circuit comprised of a ceramic (or crystal) resonator has a finite time after power-on at which its oscillation is instable. Therefore, create a certain amount of oscillation stabilization time that suits the oscillator circuit used. Figure 20.1.2 shows an oscillation stabilization time at power-on.

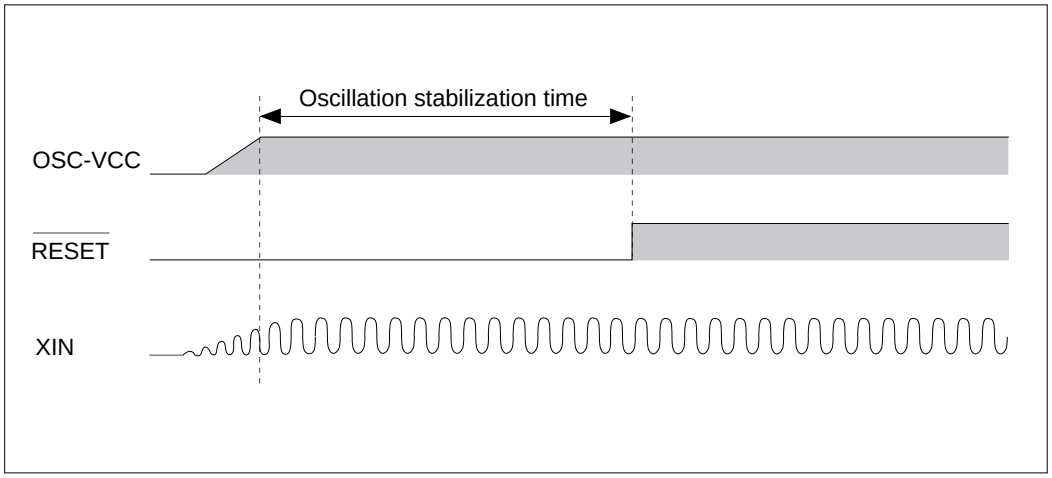


Figure 20.1.2 Oscillation Stabilization Time at Power-on

20.2 Clock Generator Circuit

The clock generator supplies independent clocks to the CPU and internal peripheral circuits.

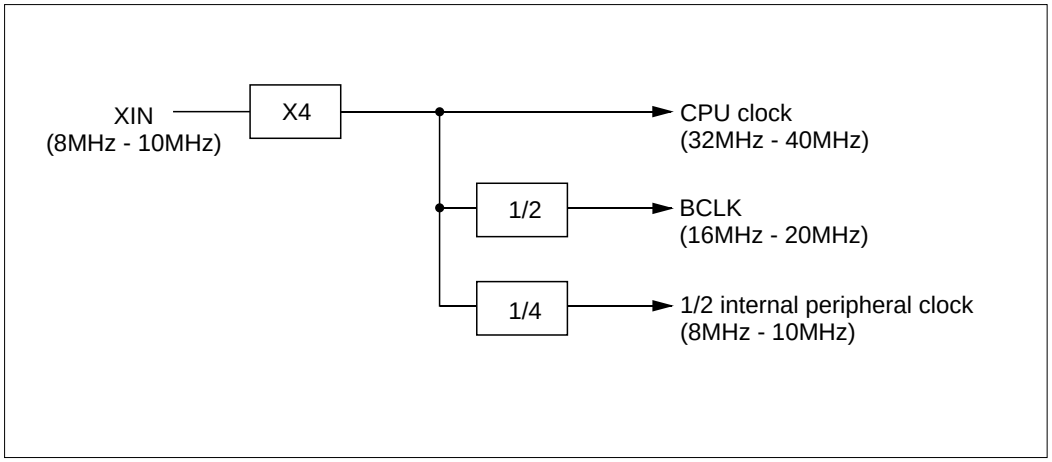


Figure 20.2.1 Configuration of the Clock Generator Circuit

* This is a blank page.*

CHAPTER 21
JTAG

21.1 Outline of the JTAG

The 32172/32173 contains a JTAG (Joint Test Action Group) interface based on 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, refer to the IEEE Std. 1149.1a-1993 documentation.

The functions of JTAG interface related pins mounted on the 32172/32173 are shown below.

Table 21.1.1 JTAG Pin Functions

Type	Symbol	Pin Name	I/O	Function
TAP (Note)	JTCK	Test clock	Input	Clock input to the test circuit.
	JTDI	Test data input	Input	Synchronous serial data input pin used to enter test instruction code and test data. This input is sampled on rising edges of JTCK.
	JTDO	Test data output	output	Synchronous serial data output pin used to output test instruction code and test data. This signal changes state on falling edges of JTCK, and is output only in Shift-IR or Shift-DR state.
	JTMS	Test mode select	Input	Test mode select input to control the test circuit's state transitions. This input is sampled on rising edges of JTCK.
	JTRST	Test reset	Input	Active-low test reset input to initialize the test circuit asynchronously. To ensure that the test circuit is reset without fail, JTMS signal input must be held high while this signal changes state from low to high.

Note: TAP = Test Access Port, a JTAG interface stipulated in IEEE 1149.1.

21.2 Configuration of the JTAG Circuit

The 32172/32173's JTAG circuit consists of the following blocks:

- Instruction register to hold instruction codes which are fetched through the boundary-scan path
- A set of data registers which are accessed through the boundary-scan path
- Test access port (abbreviated TAP) controller to control the JTAG unit's state transitions
- Control logic to select input, output, etc.

A configuration of the JTAG circuit is shown below.

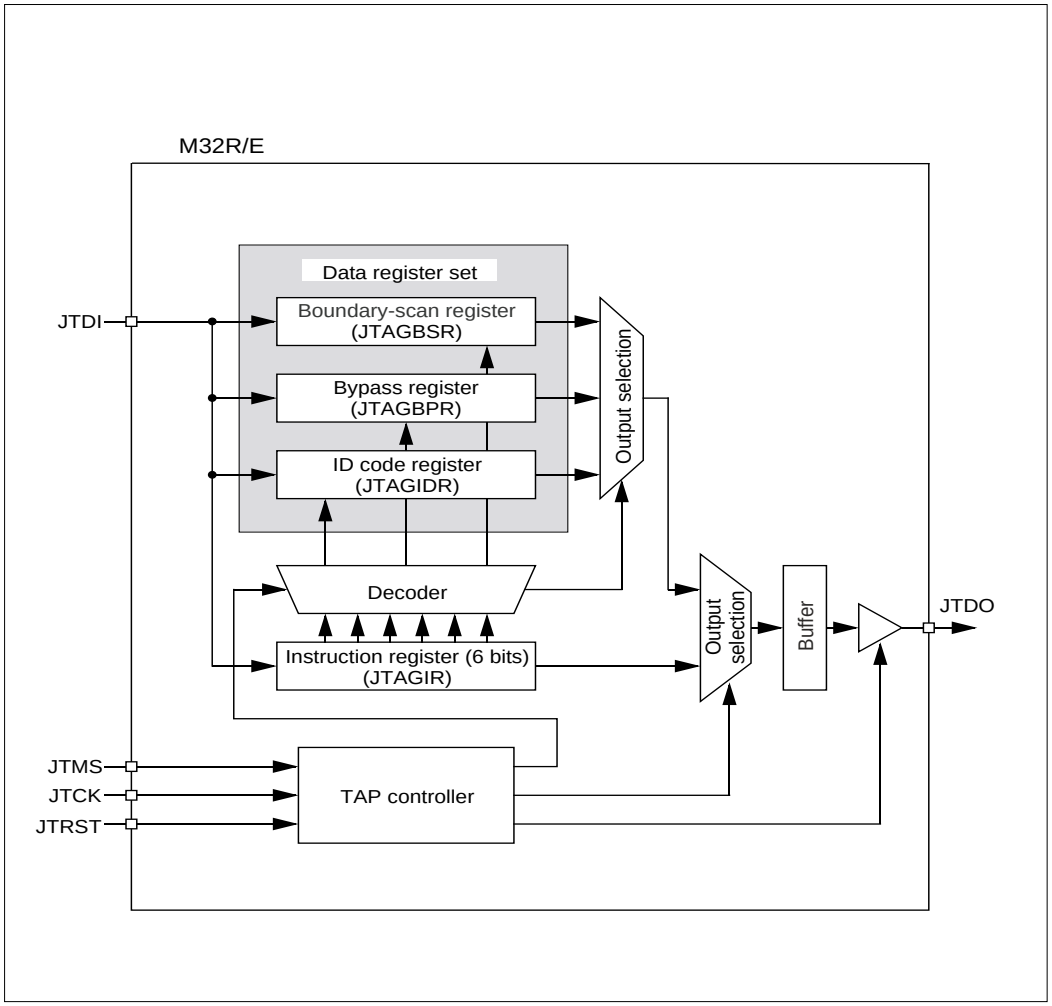


Figure 21.2.1 Configuration of the JTAG Circuit

21.3 JTAG Registers

21.3.1 Instruction Register (JTAGIR)

The Instruction Register (JTAGIR) is a 6-bit register to hold instruction code. This register is set in IR path sequence. The instructions set in this register determine the data register to be selected in the subsequent DR path sequence.

When test is reset (to initialize the test circuit), the initial value of this register is b'000010 (IDCODE instruction). After a test reset, the IDCODE Register is selected as the data register until an instruction code is set by an external device. In "Capture-IR" state, this register always has b'110001 (fixed value) loaded into it. Therefore, when in "Shift-IR" state, no matter what value was set in this register, b'110001 is always output from the JTDO pin (sequentially beginning with LSB). However, this value normally is not handled as instruction code.

Shown below is outside the scope of guaranteed operations. Note that if this operation is performed, the device may inadvertently handle b'110001 as instruction code, which makes it unable to operate normally.

[Capture-IR] → [Exit1-IR] → [Update-IR]

The 32172/32173's JTAG interface supports the following instructions:

- Three instructions stipulated as essential in IEEE 1149.1 (EXTEST, SAMPLE/PRELOAD, BYPASS)
- Device ID register access instruction (IDCODE)

Table 21.3.1 JTAG Instruction List

Instruction Code	Abbreviation	Operation
b'000000	EXTEST	Tests circuit/board-level connections outside the chip.
b'000001	SAMPLE/PRELOAD	Samples operating circuit status and outputs the sampled status from JTDO pin, while at the same time entering the data used for boundary-scan test from the JTDI pin and presets it in Boundary Scan Register.
b'000010	IDCODE	Selects ID Code Register and outputs device and manufacturer identification data from JTDO pin.
b'111111	BYPASS	Selects Bypass Register and inspects or sets data.

Note 1: Do not set any other instruction code.

Note 2: For details about "IR path sequence," "DR path sequence," "Test reset," "Capture-IR" state, "Shift-IR" state, "Exit1-IR" state, and "Update-IR" state, refer to Section 21.4.

21.3.2 Data Registers

(1) Boundary Scan Register (JTAGBSR)

The Boundary Scan Register is a 471-bit register used to perform boundary-scan test. Bits in this register are assigned to each pin on the 32172/32173.

Connected between the JTDI and JTDO pins, this register is selected when issuing EXTEST or SAMPLE/PRELOAD instruction. In "Capture-DR" state, this register captures the status of input pins or internal logic output values. In "Shift-DR" state, while outputting the sampled value, it is used to set pin functions (input/output pin and tristate output pin direction) and output values by entering data for boundary-scan test.

(2) Bypass Register (JTAGBPR)

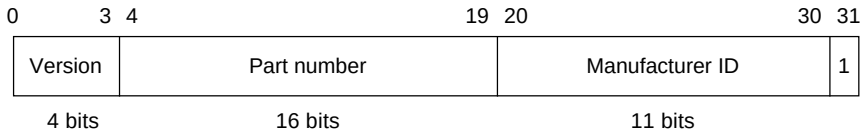
The Bypass Register is a 1-bit register used to bypass boundary-scan passes when the 32172/32173 is not the target of boundary-scan test. Connected between the JTDI and JTDO pins, this register is selected when issuing BYPASS instruction. This register when in "Capture-DR" state has b'0 (fixed value) loaded into it.

(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 0000
- 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. When in "Capture-DR" state, this register has the said IDCODE data loaded into it, which is output from the JTDO pin in "Shift_DR" state.
This register is a read-only register, so that the data written from the JTDI pin during DR pass sequence is ignored. Therefore, make sure JTDI input = low during "Shift-DR" state.



Note: For details about "Capture-DR" and "Shift-DR" states, refer to Section 21.4.

21.4 Basic Operation of the JTAG

21.4.1 Outline of JTAG Operation

The instruction and data registers basically are accessed in the following three operations, which are performed based on state transitions of the TAP controller. 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 register operation, the input data is loaded into the shift register stage.

- **Shift operation**

  The register is accessed from outside through the boundary-scan path. The sampled value is output to an external device at the same time data is set from outside. As register operation, bits are shifted right between each shift register stage.

- **Update operation**

  The data set from outside during shift is driven. As register operation, the value set in the shift register stage is transferred to the parallel output stage.

The JTAG interface undergoes transitions of internal state depending on JTMS input as it performs the following two operations. In either case, the operation basically is performed in order of Capture $\rightarrow$ Shift $\rightarrow$ Update.

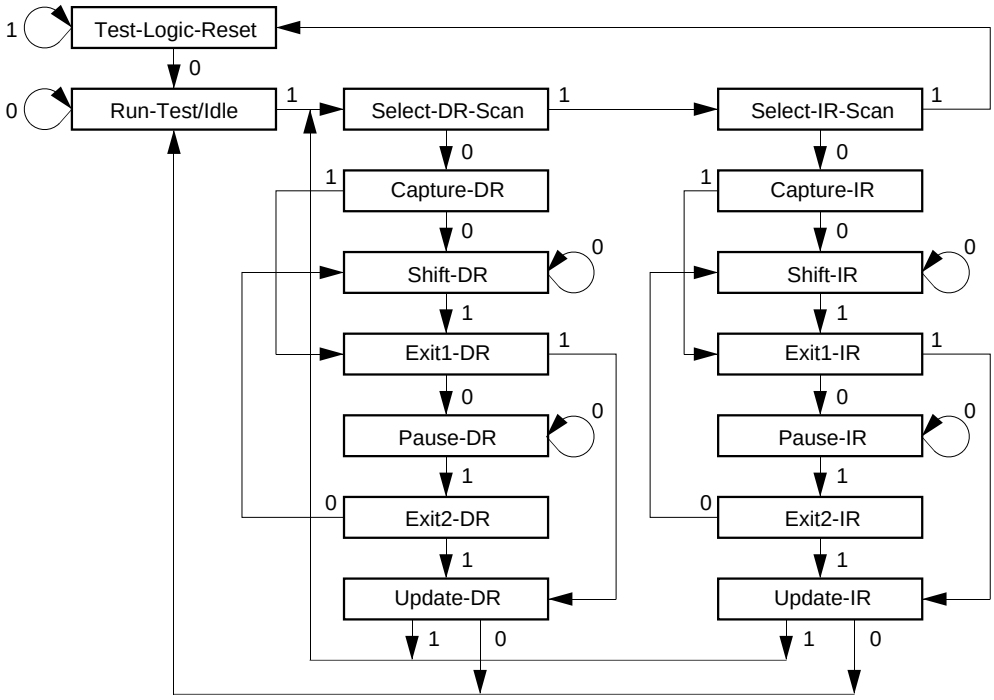
- **IR path sequence**

  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**

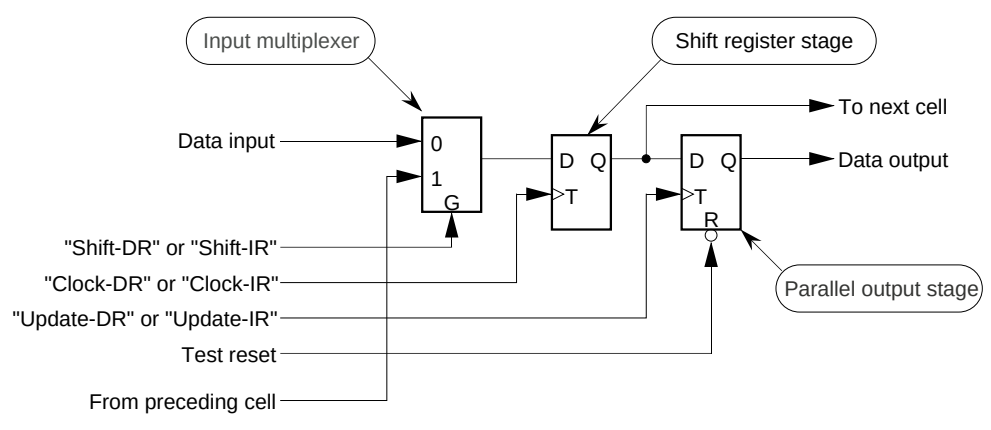
  The selected data register is operated on to inspect or set data.

The state transitions of the TAP controller and the basic configuration of the 32171's JTAG related registers are shown below.



Note : Values (0 and 1) in this diagram denote the state of JTMS input signal.

Figure 21.4.1 TAP Controller State Transition



Note : Shown here is the basic configuration, and the configuration of DR and IR does not all have to be like this.

Figure 21.4.2 Basic Configuration of JTAG Related Registers

21.4.2 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) Enter JTMS = high for a period of two JTCK cycles from "Run-Test/Idle" state to go to "Select-IR-Scan" state.

(2) Set JTMS = low to go to "Capture-IR" state. At this time, b'110001 (fixed value) is set in the instruction register's shift register stage.

(3) Subsequently, enter JTMS = low to go to "Shift-IR" state. In "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, the instruction code serially entered from the JTDI pin is set in the shift register stage bit by bit. Because instruction code is set in the instruction register which is comprised of 6 bits, the "Shift-IR" state continues for a period of 6 JTCK cycles. To stop the shift operation in the middle, go to "Pause-IR" state via temporarily "Exit1-IR" state (by setting JTMS input from high to low). Also, to return from "Pause-IR" state, go to "Shift-IR" state via temporarily "Exit1-IR" state (by setting JTMS input from high to low).

(4) By setting JTMS = high, go from "Shift-IR" state to "Exit1-IR" state. This completes the shift operation.

(5) Subsequently, enter JTMS = high to go to "Update-IR" state. In "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, thus, JTAG instruction decoding begins.

(6) Subsequently, enter JTMS = high to go to "Select-DR-Scan" state or JTMS = low to go to "Run-Test/Idle" state.

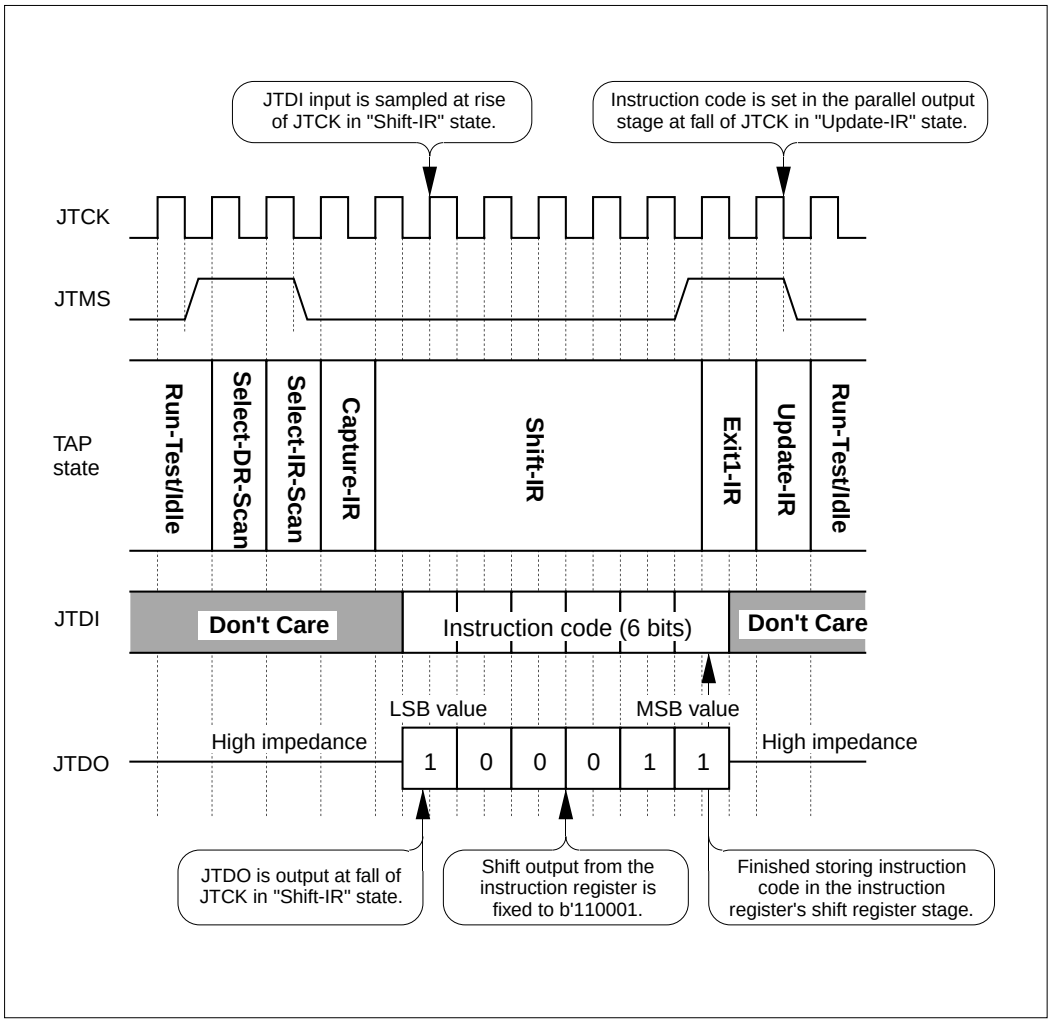


Figure 21.4.3 IR Path Sequence

21.4.3 DR Path Sequence

The data register that was selected during the IR path sequence prior to the DR path sequence is operated on to inspect or set data in it. The DR path sequence is performed following the procedure described below.

(1) Enter JTMS = high for a period of one JTCK cycle from "Run-Test/Idle" state to go to "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.

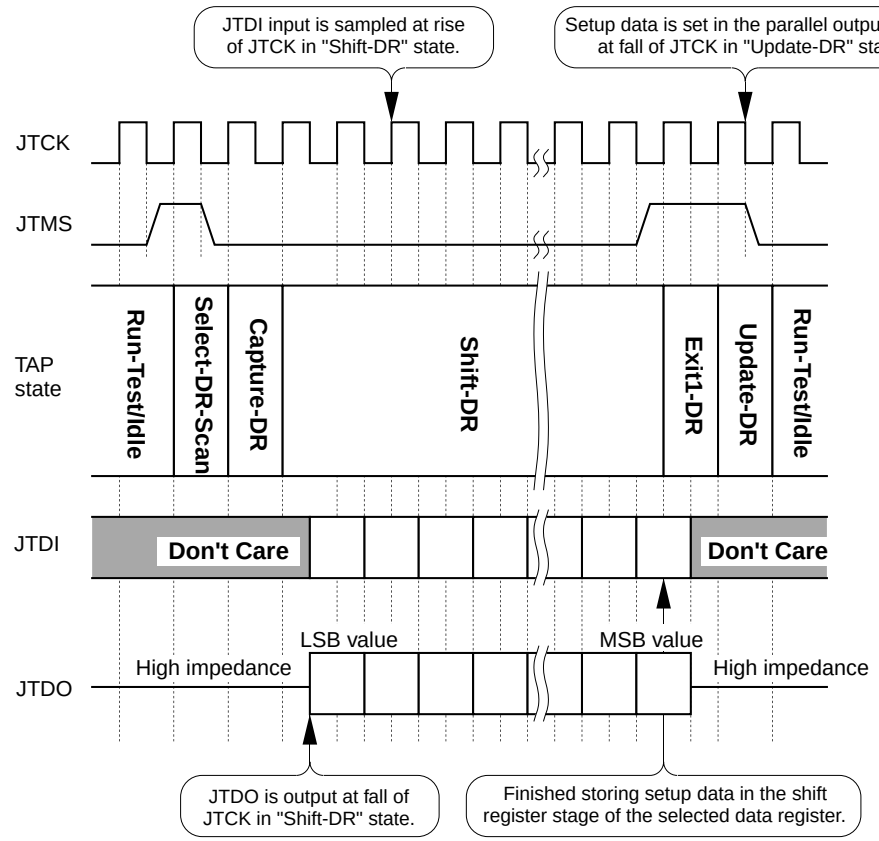
(2) Set JTMS = low to go to "Capture-DR" state. At this time, the result of boundary-scan test or the fixed data defined for each register is set in the data register's shift register stage.

(3) Subsequently, enter JTMS = low to go to "Shift-DR" state. In "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 in. At the same time, the setup data serially entered from the JTDI pin is set in the data register's shift register stage bit by bit. By continuing the "Shift-DR" state as long as the number of bits of the selected data register (by entering JTMS = low), all bits of data can be set in and read out from the shift register stage. To stop the shift operation in the middle, go to "Pause-DR" state via temporarily "Exit1-DR" state (by setting JTMS input from high to low). Also, to return from "Pause-DR" state, go to "Shift-DR" state via temporarily "Exit1-DR" state (by setting JTMS input from high to low).

(4) Set JTMS = high to go from "Shift-DR" state to "Exit2-DR" state. This completes the shift operation.

(5) Subsequently, enter JTMS = high to go to "Update-DR" state. In "Update-DR" state, the data that was set in the data register's shift register stage is transferred to the parallel output stage and, thus, the setup data becomes ready for use.

(6) Subsequently, enter JTMS = high to go to "Select-DR-Scan" state or JTMS = low to go to "Run-Test/Idle" state.



Note: The shift operation of the data register for the shift register stage is right-shifted, therefore, the output from JTDO is from the LSB side. Input to JTDI starts from the value to be set in LSB side.

Figure 21.4.4 DR Path Sequence

21.4.4 Examining 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, enter test reset (to initialize the test circuit). Test reset can be entered by one of the following two methods:

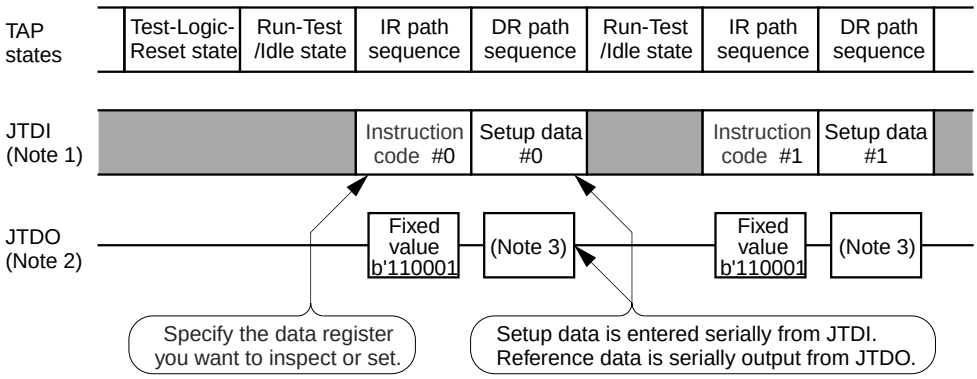
- Pull JTRST pin input low
- Drive JTMS pin input high and enter JTCK for 5 cycles or more

(2) Set JTMS = low to go to "Run-Test/Idle" state. To continue the idle state, hold JTMS input low.

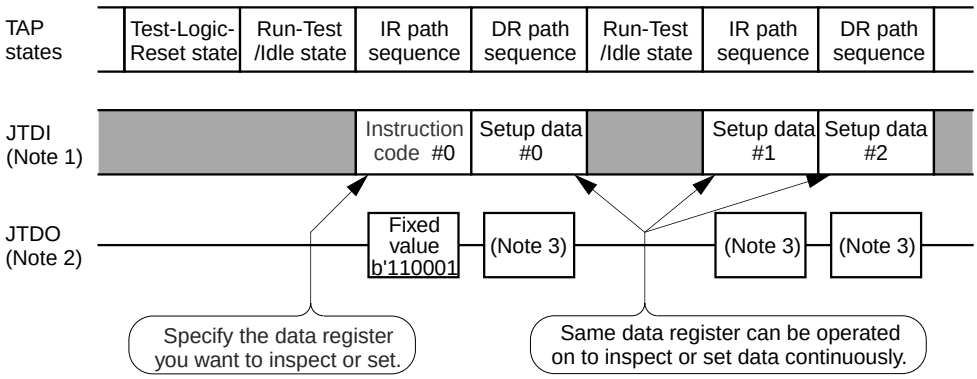
(3) Set JTMS = high to exit "Run-Test/Idle" state and perform IR path sequence. In IR path sequence, specify the data register you want to inspect or set.

(4) Subsequently, perform DR path sequence. For the data register specified in IR path sequence, enter setup data from the JTDI pin and read out reference data from the JTDO pin.

(5) If after DR path sequence is completed you want to proceed and perform IR path sequence or DR path sequence, enter JTMS = high to return to "Select-DR-Scan" state. If after a series of IR and DR path sequence processing is completed you want to wait for the next processing, enter JTMS = low to go to "Run-Test/Idle" state and retain the state.



(1) Basic access



(2) Continuous access to the same data register

Note 1 : The setup value for each register must be entered from the JTDI pin beginning with the LSB.

Note 2 : The value of each register is output from the JTDO pin beginning with the LSB. The JTDO pin outputs valid data in only "Shift-IR" state of IR path sequence and "Shift-DR" state of DR path sequence. In all other states, the JTDO pin is tristated (high impedance).

Note 3 : Data can only be read out from the data register which is selected by the instruction that was set in the immediately preceding IR path sequence. Output in the selected data register's shift register stage is the value that was sampled during "Capture-DR" state.

Figure 21.4.5 Continuous JTAG Access

21.5 Boundary Scan Description Language

The Boundary Scan Description Language (abbreviated BSDL) is stipulated in supplements to "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 helps to precisely describe the functions of standard-compliant 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, it 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 contains 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. This determines the logic type of input, output, input/output, buffer, or link of each pin that defines the logical direction of signal flow.

- **Physical pin map**

  The physical pin map correlates the chip's logical ports to the physical pins on each package. Use of separate names for each map makes it possible to define multiple 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. It is also possible to write instructions exclusive to the chip.

- **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 closest to the test data output (JTDO) pin, and the cell with the largest number is located closest to the test data input (JTDI) pin. Cells also contain related other information which includes cell type, logical port corresponding to cell, logical function of cell, safety value, control cell number, disable value, and result value.

The BSDL for the 32172/32173 shown in the pages to follow have been prepared for use in test engineering for the purpose of PCB design and those stipulated in IEEE 1149.1 standards.

```
--------------------------------------------------------------------------
--  Boundary Scan Description Language (BSDL) for
--  M32173F2VFP: M32R/E M32173 Group, Flash 256KB, 144P6Q
--------------------------------------------------------------------------
--  Modification History
--   Date       Author       Version
--  Created   '00/10/05   MITSUBISHI    Ver. 0.0
--  Modified  '--/--/--
--------------------------------------------------------------------------

entity M32173F2VFP is

        generic (PHYSICAL_PIN_MAP : string := "P6Q144");

        port (
                P221                    :in             bit;
                P225                    :inout          bit;
                OSCVSS_3                :linkage        bit;
                XIN                     :in             bit;
                XOUT                    :buffer         bit;
                OSCVCC_6                :linkage        bit;
                VCNT_7                  :linkage        bit;
                P30                     :inout          bit;
                P31                     :inout          bit;
                P32                     :inout          bit;
                P33                     :inout          bit;
                P34                     :inout          bit;
                P35                     :inout          bit;
                P36                     :inout          bit;
                P37                     :inout          bit;
                P20                     :inout          bit;
                P21                     :inout          bit;
                P22                     :inout          bit;
                P23                     :inout          bit;
                VCCE_20                 :linkage        bit;
                VSS_21                  :linkage        bit;
                P24                     :inout          bit;
                P25                     :inout          bit;
                P26                     :inout          bit;
                P27                     :inout          bit;
                P00                     :inout          bit;
                P01                     :inout          bit;
                P02                     :inout          bit;
                P03                     :inout          bit;
                P04                     :inout          bit;
                P05                     :inout          bit;
                P06                     :inout          bit;
                P07                     :inout          bit;
                P10                     :inout          bit;
                P11                     :inout          bit;
                P12                     :inout          bit;
                P13                     :inout          bit;
                P14                     :inout          bit;
                P15                     :inout          bit;
                P16                     :inout          bit;
                P17                     :inout          bit;
                VREF_42                 :linkage        bit;
                AVCC_43                 :linkage        bit;
                AD0IN0                  :linkage        bit;
                AD0IN1                  :linkage        bit;
```

Figure 21.5.1 BSDL Description for the 32173(1/11)

```
AD0IN2          :linkage    bit;
AD0IN3          :linkage    bit;
AD0IN4          :linkage    bit;
AD0IN5          :linkage    bit;
AD0IN6          :linkage    bit;
AD0IN7          :linkage    bit;
AD1IN0          :linkage    bit;
AD1IN1          :linkage    bit;
AD1IN2          :linkage    bit;
AD1IN3          :linkage    bit;
DA0             :linkage    bit;
DA1             :linkage    bit;
P172            :in         bit;
P173            :in         bit;
AVSS_60         :linkage    bit;
VCCI_61         :linkage    bit;
VSS_62          :linkage    bit;
P174            :inout      bit;
P175            :inout      bit;
VCCE_65         :linkage    bit;
P82             :inout      bit;
P83             :inout      bit;
P84             :inout      bit;
P85             :inout      bit;
P86             :inout      bit;
P87             :inout      bit;
VSS_72          :linkage    bit;
FVCC_73         :linkage    bit;
P61             :inout      bit;
P62             :inout      bit;
P63             :inout      bit;
P64             :in         bit;
P70             :inout      bit;
P71             :inout      bit;
P72             :inout      bit;
P73             :inout      bit;
P74             :inout      bit;
P75             :inout      bit;
P76             :inout      bit;
P77             :inout      bit;
P93             :in         bit;
P94             :inout      bit;
P95             :inout      bit;
P96             :inout      bit;
P97             :in         bit;
RESET           :in         bit;
MOD0            :in         bit;
MOD1            :in         bit;
FP              :in         bit;
VCCE_95         :linkage    bit;
VSS_96          :linkage    bit;
P110            :inout      bit;
P111            :inout      bit;
P112            :inout      bit;
P113            :inout      bit;
P114            :inout      bit;
P115            :inout      bit;
P116            :inout      bit;
P117            :inout      bit;
P100            :inout      bit;
```

Figure 21.5.2 BSDL Description for the 32173(2/11)

```
P101                    :inout          bit;
P102                    :inout          bit;
VDD_108                 :linkage        bit;
TMS                     :in             bit;
TCK                     :in             bit;
TRST                    :in             bit;
TDO                     :out            bit;
TDI                     :in             bit;
P103                    :inout          bit;
P104                    :inout          bit;
P105                    :inout          bit;
P106                    :inout          bit;
P107                    :inout          bit;
P124                    :in             bit;
P125                    :in             bit;
P126                    :in             bit;
P127                    :in             bit;
VCCI_123                :linkage        bit;
P130                    :in             bit;
P131                    :in             bit;
P132                    :in             bit;
P133                    :in             bit;
P134                    :in             bit;
P135                    :in             bit;
P136                    :in             bit;
P137                    :in             bit;
VCCE_132                :linkage        bit;
P150                    :inout          bit;
P153                    :inout          bit;
P41                     :inout          bit;
P42                     :inout          bit;
VCCI_137                :linkage        bit;
VSS_138                 :linkage        bit;
P43                     :inout          bit;
P44                     :inout          bit;
P45                     :inout          bit;
P46                     :inout          bit;
P47                     :inout          bit;
P220                    :inout          bit
);

use STD_1149_1_1994.all;

attribute COMPONENT_CONFORMANCE of M32173F2VFP : entity is "STD_1149_1_1993";

attribute PIN_MAP of M32173F2VFP : entity is PHYSICAL_PIN_MAP;

constant P6Q144 : PIN_MAP_STRING :=
"P221                   :1," &
"P225                   :2," &
"OSCVSS_3               :3," &
"XIN                    :4," &
"XOUT                   :5," &
"OSCVCC_6               :6," &
"VCNT_7                 :7," &
"P30                    :8," &
"P31                    :9," &
"P32                    :10," &
"P33                    :11," &
"P34                    :12," &
```

Figure 21.5.3 BSDL Description for the 32173(3/11)

```
"P35                :13," &
"P36                :14," &
"P37                :15," &
"P20                :16," &
"P21                :17," &
"P22                :18," &
"P23                :19," &
"VCCE_20            :20," &
"VSS_21             :21," &
"P24                :22," &
"P25                :23," &
"P26                :24," &
"P27                :25," &
"P00                :26," &
"P01                :27," &
"P02                :28," &
"P03                :29," &
"P04                :30," &
"P05                :31," &
"P06                :32," &
"P07                :33," &
"P10                :34," &
"P11                :35," &
"P12                :36," &
"P13                :37," &
"P14                :38," &
"P15                :39," &
"P16                :40," &
"P17                :41," &
"VREF_42            :42," &
"AVCC_43            :43," &
"AD0IN0             :44," &
"AD0IN1             :45," &
"AD0IN2             :46," &
"AD0IN3             :47," &
"AD0IN4             :48," &
"AD0IN5             :49," &
"AD0IN6             :50," &
"AD0IN7             :51," &
"AD1IN0             :52," &
"AD1IN1             :53," &
"AD1IN2             :54," &
"AD1IN3             :55," &
"DA0                :56," &
"DA1                :57," &
"P172               :58," &
"P173               :59," &
"AVSS_60            :60," &
"VCCI_61            :61," &
"VSS_62             :62," &
"P174               :63," &
"P175               :64," &
"VCCE_65            :65," &
"P82                :66," &
"P83                :67," &
"P84                :68," &
"P85                :69," &
"P86                :70," &
"P87                :71," &
"VSS_72             :72," &
```

Figure 21.5.4 BSDL Description for the 32173(4/11)

```
"FVCC_73        :73," &
"P61            :74," &
"P62            :75," &
"P63            :76," &
"P64            :77," &
"P70            :78," &
"P71            :79," &
"P72            :80," &
"P73            :81," &
"P74            :82," &
"P75            :83," &
"P76            :84," &
"P77            :85," &
"P93            :86," &
"P94            :87," &
"P95            :88," &
"P96            :89," &
"P97            :90," &
"RESET          :91," &
"MOD0           :92," &
"MOD1           :93," &
"FP             :94," &
"VCCE_95        :95," &
"VSS_96         :96," &
"P110           :97," &
"P111           :98," &
"P112           :99," &
"P113           :100," &
"P114           :101," &
"P115           :102," &
"P116           :103," &
"P117           :104," &
"P100           :105," &
"P101           :106," &
"P102           :107," &
"VDD_108        :108," &
"TMS            :109," &
"TCK            :110," &
"TRST           :111," &
"TDO            :112," &
"TDI            :113," &
"P103           :114," &
"P104           :115," &
"P105           :116," &
"P106           :117," &
"P107           :118," &
"P124           :119," &
"P125           :120," &
"P126           :121," &
"P127           :122," &
"VCCI_123       :123," &
"P130           :124," &
"P131           :125," &
"P132           :126," &
"P133           :127," &
"P134           :128," &
"P135           :129," &
"P136           :130," &
"P137           :131," &
"VCCE_132       :132," &
```

Figure 21.5.5 BSDL Description for the 32173(5/11)

```
"P150                   :133," &
"P153                   :134," &
"P41                    :135," &
"P42                    :136," &
"VCCI_137               :137," &
"VSS_138                :138," &
"P43                    :139," &
"P44                    :140," &
"P45                    :141," &
"P46                    :142," &
"P47                    :143," &
"P220                   :144" ;

attribute TAP_SCAN_IN of TDI : signal is true;
attribute TAP_SCAN_MODE of TMS : signal is true;
attribute TAP_SCAN_OUT of TDO : signal is true;

attribute TAP_SCAN_CLOCK of TCK : signal is (5.0e6, BOTH);
attribute TAP_SCAN_RESET of TRST : signal is true;

attribute INSTRUCTION_LENGTH of M32173F2VFP : entity is 6;

attribute INSTRUCTION_OPCODE of M32173F2VFP : entity is
        "BYPASS             (111111)," &
        "SAMPLE             (000001)," &
        "EXTEST             (000000)," &
        "IDCODE             (000010)," &
        "USERCODE           (000011)," &
        "MDM_SYSTEM         (001000)," &
        "MDM_CONTROL        (001001)," &
        "MDM_SETUP          (001010)," &
        "MTM_CONTROL        (001111)," &
        "MON_CODE           (010000)," &
        "MON_DATA           (010001)," &
        "MON_PARAM          (010010)," &
        "MON_ACCESS         (010011)," &
        "DMA_RADDR          (011000)," &
        "DMA_RDATA          (011001)," &
        "DMA_RTYPE          (011010)," &
        "DMA_ACCESS         (011011)," &
        "RTDENB             (100000)" ;

attribute INSTRUCTION_CAPTURE of M32173F2VFP : entity is "110001";

attribute INSTRUCTION_PRIVATE of M32173F2VFP : entity is
        "MDM_SYSTEM," &
        "MDM_CONTROL," &
        "MDM_SETUP," &
        "MTM_CONTROL," &
        "MON_CODE," &
        "MON_DATA," &
        "MON_PARAM," &
        "MON_ACCESS," &
        "DMA_RADDR," &
        "DMA_RDATA," &
        "DMA_RTYPE," &
        "DMA_ACCESS," &
        "RTDENB" ;
```

Figure 21.5.6 BSDL Description for the 32173(6/11)

```
                attribute IDCODE_REGISTER of M32173F2VFP : entity is
                        "0000" &                        -- version
                        "0011001000100001" & -- part number
                        "00000011100" &                 -- manufacturer's identity
                        "1";                            -- required by 1149.1

---             attribute USERCODE_REGISTER of M32173F2VFP : entity is
---                     "0000 0000 0000 0000" &         -- reserved
---                     "0000" &                        -- reserved
---                     "0001" &                        -- ROM
---                     "0000" &                        -- ISA
---                     "0001";                         -- SDI version

                attribute REGISTER_ACCESS of M32173F2VFP : entity is
                        "Bypass                         (BYPASS)," &
                        "Boundary                       (SAMPLE, EXTEST)," &
                        "Device_ID                      (IDCODE)," &
                        "USERCODE_REG[32]               (USERCODE)," &
                        "MDM_SYSTEM_REG[17]             (MDM_SYSTEM)," &
                        "MDM_CONTROL_REG[20]            (MDM_CONTROL)," &
                        "MDM_SETUP_REG[2]               (MDM_SETUP)," &
                        "MTM_CONTROL_REG[4]             (MTM_CONTROL)," &
                        "MON_CODE_REG[32]               (MON_CODE)," &
                        "MON_DATA_REG[32]               (MON_DATA)," &
                        "MON_PARAM_REG[32]              (MON_PARAM)," &
                        "MON_ACCESS_REG[4]              (MON_ACCESS)," &
                        "DMA_RADDR_REG[32]              (DMA_RADDR)," &
                        "DMA_RDATA_REG[32]              (DMA_RDATA)," &
                        "DMA_RTYPE_REG[3]               (DMA_RTYPE)," &
                        "DMA_ACCESS_REG[3]              (DMA_ACCESS)," &
                        "RTDENB_REG[1]                  (RTDENB)";

                attribute BOUNDARY_LENGTH of M32173F2VFP : entity is 265;

                attribute BOUNDARY_REGISTER of M32173F2VFP : entity is
                 --
                 -- num  cell       port                function            safe [ccell disval rslt]
                 --
                        "264 (BC_4,     P103,           observe_only,   X)," &
                        "263 (BC_1,     P103,           output3,        X, 262, 0, Z)," &
                        "262 (BC_1,     *,              control,        0)," &
                        "261 (BC_4,     P104,           observe_only,   X)," &
                        "260 (BC_1,     P104,           output3,        X, 259, 0, Z)," &
                        "259 (BC_1,     *,              control,        0)," &
                        "258 (BC_4,     P105,           observe_only,   X)," &
                        "257 (BC_1,     P105,           output3,        X, 256, 0, Z)," &
                        "256 (BC_1,     *,              control,        0)," &
                        "255 (BC_4,     P106,           observe_only,   X)," &
                        "254 (BC_1,     P106,           output3,        X, 253, 0, Z)," &
                        "253 (BC_1,     *,              control,        0)," &
                        "252 (BC_4,     P107,           observe_only,   X)," &
                        "251 (BC_1,     P107,           output3,        X, 250, 0, Z)," &
                        "250 (BC_1,     *,              control,        0)," &
                        "249 (BC_4,     P124,           observe_only,   X)," &
                        "248 (BC_4,     P125,           observe_only,   X)," &
                        "247 (BC_4,     P126,           observe_only,   X)," &
                        "246 (BC_4,     P127,           observe_only,   X)," &
                        "245 (BC_4,     P130,           observe_only,   X)," &
                        "244 (BC_4,     P131,           observe_only,   X)," &
                        "243 (BC_4,     P132,           observe_only,   X)," &
```

Figure 21.5.7 BSDL Description for the 32173(7/11)

```
"242 (BC_4,    P133,    observe_only,    X)," &
"241 (BC_4,    P134,    observe_only,    X)," &
"240 (BC_4,    P135,    observe_only,    X)," &
"239 (BC_4,    P136,    observe_only,    X)," &
"238 (BC_4,    P137,    observe_only,    X)," &
"237 (BC_4,    P150,    observe_only,    X)," &
"236 (BC_1,    P150,    output3,         X, 235, 0, Z)," &
"235 (BC_1,    *,       control,         0)," &
"234 (BC_4,    P153,    observe_only,    X)," &
"233 (BC_1,    P153,    output3,         X, 232, 0, Z)," &
"232 (BC_1,    *,       control,         0)," &
"231 (BC_4,    P41,     observe_only,    X)," &
"230 (BC_1,    P41,     output3,         X, 229, 0, Z)," &
"229 (BC_1,    *,       control,         0)," &
"228 (BC_4,    P42,     observe_only,    X)," &
"227 (BC_1,    P42,     output3,         X, 226, 0, Z)," &
"226 (BC_1,    *,       control,         0)," &
"225 (BC_4,    P43,     observe_only,    X)," &
"224 (BC_1,    P43,     output3,         X, 223, 0, Z)," &
"223 (BC_1,    *,       control,         0)," &
"222 (BC_4,    P44,     observe_only,    X)," &
"221 (BC_1,    P44,     output3,         X, 220, 0, Z)," &
"220 (BC_1,    *,       control,         0)," &
"219 (BC_4,    P45,     observe_only,    X)," &
"218 (BC_1,    P45,     output3,         X, 217, 0, Z)," &
"217 (BC_1,    *,       control,         0)," &
"216 (BC_4,    P46,     observe_only,    X)," &
"215 (BC_1,    P46,     output3,         X, 214, 0, Z)," &
"214 (BC_1,    *,       control,         0)," &
"213 (BC_4,    P47,     observe_only,    X)," &
"212 (BC_1,    P47,     output3,         X, 211, 0, Z)," &
"211 (BC_1,    *,       control,         0)," &
"210 (BC_4,    P220,    observe_only,    X)," &
"209 (BC_1,    P220,    output3,         X, 208, 0, Z)," &
"208 (BC_1,    *,       control,         0)," &
"207 (BC_4,    P221,    observe_only,    X)," &
"206 (BC_4,    P225,    observe_only,    X)," &
"205 (BC_1,    P225,    output3,         X, 204, 0, Z)," &
"204 (BC_1,    *,       control,         0)," &
"203 (BC_4,    P30,     observe_only,    X)," &
"202 (BC_1,    P30,     output3,         X, 201, 0, Z)," &
"201 (BC_1,    *,       control,         0)," &
"200 (BC_4,    P31,     observe_only,    X)," &
"199 (BC_1,    P31,     output3,         X, 198, 0, Z)," &
"198 (BC_1,    *,       control,         0)," &
"197 (BC_4,    P32,     observe_only,    X)," &
"196 (BC_1,    P32,     output3,         X, 195, 0, Z)," &
"195 (BC_1,    *,       control,         0)," &
"194 (BC_4,    P33,     observe_only,    X)," &
"193 (BC_1,    P33,     output3,         X, 192, 0, Z)," &
"192 (BC_1,    *,       control,         0)," &
"191 (BC_4,    P34,     observe_only,    X)," &
"190 (BC_1,    P34,     output3,         X, 189, 0, Z)," &
"189 (BC_1,    *,       control,         0)," &
"188 (BC_4,    P35,     observe_only,    X)," &
"187 (BC_1,    P35,     output3,         X, 186, 0, Z)," &
"186 (BC_1,    *,       control,         0)," &
"185 (BC_4,    P36,     observe_only,    X)," &
"184 (BC_1,    P36,     output3,         X, 183, 0, Z)," &
"183 (BC_1,    *,       control,         0)," &
```

Figure 21.5.8 BSDL Description for the 32173(8/11)

```
"182 (BC_4,     P37,        observe_only,   X)," &
"181 (BC_1,     P37,        output3,        X, 180, 0, Z)," &
"180 (BC_1,     *,          control,        0)," &
"179 (BC_4,     P20,        observe_only,   X)," &
"178 (BC_1,     P20,        output3,        X, 177, 0, Z)," &
"177 (BC_1,     *,          control,        0)," &
"176 (BC_4,     P21,        observe_only,   X)," &
"175 (BC_1,     P21,        output3,        X, 174, 0, Z)," &
"174 (BC_1,     *,          control,        0)," &
"173 (BC_4,     P22,        observe_only,   X)," &
"172 (BC_1,     P22,        output3,        X, 171, 0, Z)," &
"171 (BC_1,     *,          control,        0)," &
"170 (BC_4,     P23,        observe_only,   X)," &
"169 (BC_1,     P23,        output3,        X, 168, 0, Z)," &
"168 (BC_1,     *,          control,        0)," &
"167 (BC_4,     P24,        observe_only,   X)," &
"166 (BC_1,     P24,        output3,        X, 165, 0, Z)," &
"165 (BC_1,     *,          control,        0)," &
"164 (BC_4,     P25,        observe_only,   X)," &
"163 (BC_1,     P25,        output3,        X, 162, 0, Z)," &
"162 (BC_1,     *,          control,        0)," &
"161 (BC_4,     P26,        observe_only,   X)," &
"160 (BC_1,     P26,        output3,        X, 159, 0, Z)," &
"159 (BC_1,     *,          control,        0)," &
"158 (BC_4,     P27,        observe_only,   X)," &
"157 (BC_1,     P27,        output3,        X, 156, 0, Z)," &
"156 (BC_1,     *,          control,        0)," &
"155 (BC_4,     P00,        observe_only,   X)," &
"154 (BC_1,     P00,        output3,        X, 153, 0, Z)," &
"153 (BC_1,     *,          control,        0)," &
"152 (BC_4,     P01,        observe_only,   X)," &
"151 (BC_1,     P01,        output3,        X, 150, 0, Z)," &
"150 (BC_1,     *,          control,        0)," &
"149 (BC_4,     P02,        observe_only,   X)," &
"148 (BC_1,     P02,        output3,        X, 147, 0, Z)," &
"147 (BC_1,     *,          control,        0)," &
"146 (BC_4,     P03,        observe_only,   X)," &
"145 (BC_1,     P03,        output3,        X, 144, 0, Z)," &
"144 (BC_1,     *,          control,        0)," &
"143 (BC_4,     P04,        observe_only,   X)," &
"142 (BC_1,     P04,        output3,        X, 141, 0, Z)," &
"141 (BC_1,     *,          control,        0)," &
"140 (BC_4,     P05,        observe_only,   X)," &
"139 (BC_1,     P05,        output3,        X, 138, 0, Z)," &
"138 (BC_1,     *,          control,        0)," &
"137 (BC_4,     P06,        observe_only,   X)," &
"136 (BC_1,     P06,        output3,        X, 135, 0, Z)," &
"135 (BC_1,     *,          control,        0)," &
"134 (BC_4,     P07,        observe_only,   X)," &
"133 (BC_1,     P07,        output3,        X, 132, 0, Z)," &
"132 (BC_1,     *,          control,        0)," &
"131 (BC_4,     P10,        observe_only,   X)," &
"130 (BC_1,     P10,        output3,        X, 129, 0, Z)," &
"129 (BC_1,     *,          control,        0)," &
"128 (BC_4,     P11,        observe_only,   X)," &
"127 (BC_1,     P11,        output3,        X, 126, 0, Z)," &
"126 (BC_1,     *,          control,        0)," &
"125 (BC_4,     P12,        observe_only,   X)," &
"124 (BC_1,     P12,        output3,        X, 123, 0, Z)," &
"123 (BC_1,     *,          control,        0)," &
```

Figure 21.5.9 BSDL Description for the 32173(9/11)

```
"122 (BC_4,    P13,     observe_only,  X)," &
"121 (BC_1,    P13,     output3,       X, 120, 0, Z)," &
"120 (BC_1,    *,       control,       0)," &
"119 (BC_4,    P14,     observe_only,  X)," &
"118 (BC_1,    P14,     output3,       X, 117, 0, Z)," &
"117 (BC_1,    *,       control,       0)," &
"116 (BC_4,    P15,     observe_only,  X)," &
"115 (BC_1,    P15,     output3,       X, 114, 0, Z)," &
"114 (BC_1,    *,       control,       0)," &
"113 (BC_4,    P16,     observe_only,  X)," &
"112 (BC_1,    P16,     output3,       X, 111, 0, Z)," &
"111 (BC_1,    *,       control,       0)," &
"110 (BC_4,    P17,     observe_only,  X)," &
"109 (BC_1,    P17,     output3,       X, 108, 0, Z)," &
"108 (BC_1,    *,       control,       0)," &
"107 (BC_4,    P172,    observe_only,  X)," &
"106 (BC_4,    P173,    observe_only,  X)," &
"105 (BC_4,    P174,    observe_only,  X)," &
"104 (BC_1,    P174,    output3,       X, 103, 0, Z)," &
"103 (BC_1,    *,       control,       0)," &
"102 (BC_4,    P175,    observe_only,  X)," &
"101 (BC_1,    P175,    output3,       X, 100, 0, Z)," &
"100 (BC_1,    *,       control,       0)," &
"99  (BC_4,    P82,     observe_only,  X)," &
"98  (BC_1,    P82,     output3,       X, 97, 0, Z)," &
"97  (BC_1,    *,       control,       0)," &
"96  (BC_4,    P83,     observe_only,  X)," &
"95  (BC_1,    P83,     output3,       X, 94, 0, Z)," &
"94  (BC_1,    *,       control,       0)," &
"93  (BC_4,    P84,     observe_only,  X)," &
"92  (BC_1,    P84,     output3,       X, 91, 0, Z)," &
"91  (BC_1,    *,       control,       0)," &
"90  (BC_4,    P85,     observe_only,  X)," &
"89  (BC_1,    P85,     output3,       X, 88, 0, Z)," &
"88  (BC_1,    *,       control,       0)," &
"87  (BC_4,    P86,     observe_only,  X)," &
"86  (BC_1,    P86,     output3,       X, 85, 0, Z)," &
"85  (BC_1,    *,       control,       0)," &
"84  (BC_4,    P87,     observe_only,  X)," &
"83  (BC_1,    P87,     output3,       X, 82, 0, Z)," &
"82  (BC_1,    *,       control,       0)," &
"81  (BC_4,    P61,     observe_only,  X)," &
"80  (BC_1,    P61,     output3,       X, 79, 0, Z)," &
"79  (BC_1,    *,       control,       0)," &
"78  (BC_4,    P62,     observe_only,  X)," &
"77  (BC_1,    P62,     output3,       X, 76, 0, Z)," &
"76  (BC_1,    *,       control,       0)," &
"75  (BC_4,    P63,     observe_only,  X)," &
"74  (BC_1,    P63,     output3,       X, 73, 0, Z)," &
"73  (BC_1,    *,       control,       0)," &
"72  (BC_4,    P64,     observe_only,  X)," &
"71  (BC_4,    P70,     observe_only,  X)," &
"70  (BC_1,    P70,     output3,       X, 69, 0, Z)," &
"69  (BC_1,    *,       control,       0)," &
"68  (BC_4,    P71,     observe_only,  X)," &
"67  (BC_1,    P71,     output3,       X, 66, 0, Z)," &
"66  (BC_1,    *,       control,       0)," &
"65  (BC_4,    P72,     observe_only,  X)," &
"64  (BC_1,    P72,     output3,       X, 63, 0, Z)," &
"63  (BC_1,    *,       control,       0)," &
```

Figure 21.5.10 BSDL Description for the 32173(10/11)

```
                "62 (BC_4,      P73,            observe_only,   X)," &
                "61 (BC_1,      P73,            output3,        X, 60, 0, Z)," &
                "60 (BC_1,      *,              control,        0)," &
                "59 (BC_4,      P74,            observe_only,   X)," &
                "58 (BC_1,      P74,            output3,        X, 57, 0, Z)," &
                "57 (BC_1,      *,              control,        0)," &
                "56 (BC_4,      P75,            observe_only,   X)," &
                "55 (BC_1,      P75,            output3,        X, 54, 0, Z)," &
                "54 (BC_1,      *,              control,        0)," &
                "53 (BC_4,      P76,            observe_only,   X)," &
                "52 (BC_1,      P76,            output3,        X, 51, 0, Z)," &
                "51 (BC_1,      *,              control,        0)," &
                "50 (BC_4,      P77,            observe_only,   X)," &
                "49 (BC_1,      P77,            output3,        X, 48, 0, Z)," &
                "48 (BC_1,      *,              control,        0)," &
                "47 (BC_4,      P93,            observe_only,   X)," &
                "46 (BC_4,      P94,            observe_only,   X)," &
                "45 (BC_1,      P94,            output3,        X, 44, 0, Z)," &
                "44 (BC_1,      *,              control,        0)," &
                "43 (BC_4,      P95,            observe_only,   X)," &
                "42 (BC_1,      P95,            output3,        X, 41, 0, Z)," &
                "41 (BC_1,      *,              control,        0)," &
                "40 (BC_4,      P96,            observe_only,   X)," &
                "39 (BC_1,      P96,            output3,        X, 38, 0, Z)," &
                "38 (BC_1,      *,              control,        0)," &
                "37 (BC_4,      P97,            observe_only,   X)," &
                "36 (BC_4,      RESET,          observe_only,   X)," &
                "35 (BC_4,      MOD0,           observe_only,   X)," &
                "34 (BC_4,      MOD1,           observe_only,   X)," &
                "33 (BC_4,      FP,             observe_only,   X)," &
                "32 (BC_4,      P110,           observe_only,   X)," &
                "31 (BC_1,      P110,           output3,        X, 30, 0, Z)," &
                "30 (BC_1,      *,              control,        0)," &
                "29 (BC_4,      P111,           observe_only,   X)," &
                "28 (BC_1,      P111,           output3,        X, 27, 0, Z)," &
                "27 (BC_1,      *,              control,        0)," &
                "26 (BC_4,      P112,           observe_only,   X)," &
                "25 (BC_1,      P112,           output3,        X, 24, 0, Z)," &
                "24 (BC_1,      *,              control,        0)," &
                "23 (BC_4,      P113,           observe_only,   X)," &
                "22 (BC_1,      P113,           output3,        X, 21, 0, Z)," &
                "21 (BC_1,      *,              control,        0)," &
                "20 (BC_4,      P114,           observe_only,   X)," &
                "19 (BC_1,      P114,           output3,        X, 18, 0, Z)," &
                "18 (BC_1,      *,              control,        0)," &
                "17 (BC_4,      P115,           observe_only,   X)," &
                "16 (BC_1,      P115,           output3,        X, 15, 0, Z)," &
                "15 (BC_1,      *,              control,        0)," &
                "14 (BC_4,      P116,           observe_only,   X)," &
                "13 (BC_1,      P116,           output3,        X, 12, 0, Z)," &
                "12 (BC_1,      *,              control,        0)," &
                "11 (BC_4,      P117,           observe_only,   X)," &
                "10 (BC_1,      P117,           output3,        X, 9, 0, Z)," &
                "9  (BC_1,      *,              control,        0)," &
                "8  (BC_4,      P100,           observe_only,   X)," &
                "7  (BC_1,      P100,           output3,        X, 6, 0, Z)," &
                "6  (BC_1,      *,              control,        0)," &
                "5  (BC_4,      P101,           observe_only,   X)," &
                "4  (BC_1,      P101,           output3,        X, 3, 0, Z)," &
                "3  (BC_1,      *,              control,        0)," &
                "2  (BC_4,      P102,           observe_only,   X)," &
                "1  (BC_1,      P102,           output3,        X, 0, 0, Z)," &
                "0  (BC_1,      *,              control,        0)";

end M32173F2VFP;
```

Figure 21.5.11 BSDL Description for the 32173(11/11)

```
--------------------------------------------------------------------------
--  Boundary Scan Description Language (BSDL) for
--  M32172F2VFP: M32R/E M32172 Group, Flash 256KB, 144P6Q
--------------------------------------------------------------------------
--  Modification History
--   Date       Author       Version
--  Created   '00/10/05   MITSUBISHI    Ver. 0.0
--  Modified  '--/--/--
--------------------------------------------------------------------------

entity M32172F2VFP is

        generic (PHYSICAL_PIN_MAP : string := "P6Q144");

        port (
                P221                    :in             bit;
                P225                    :inout          bit;
                OSCVSS_3                :linkage        bit;
                XIN                     :in             bit;
                XOUT                    :buffer         bit;
                OSCVCC_6                :linkage        bit;
                VCNT_7                  :linkage        bit;
                P30                     :inout          bit;
                P31                     :inout          bit;
                P32                     :inout          bit;
                P33                     :inout          bit;
                P34                     :inout          bit;
                P35                     :inout          bit;
                P36                     :inout          bit;
                P37                     :inout          bit;
                P20                     :inout          bit;
                P21                     :inout          bit;
                P22                     :inout          bit;
                P23                     :inout          bit;
                VCCE_20                 :linkage        bit;
                VSS_21                  :linkage        bit;
                P24                     :inout          bit;
                P25                     :inout          bit;
                P26                     :inout          bit;
                P27                     :inout          bit;
                P00                     :inout          bit;
                P01                     :inout          bit;
                P02                     :inout          bit;
                P03                     :inout          bit;
                P04                     :inout          bit;
                P05                     :inout          bit;
                P06                     :inout          bit;
                P07                     :inout          bit;
                P10                     :inout          bit;
                P11                     :inout          bit;
                P12                     :inout          bit;
                P13                     :inout          bit;
                P14                     :inout          bit;
                P15                     :inout          bit;
                P16                     :inout          bit;
                P17                     :inout          bit;
                VREF_42                 :linkage        bit;
                AVCC_43                 :linkage        bit;
                AD0IN0                  :linkage        bit;
                AD0IN1                  :linkage        bit;
```

Figure 21.5.12 BSDL Description for the 32172(1/11)

```
AD0IN2          :linkage    bit;
AD0IN3          :linkage    bit;
AD0IN4          :linkage    bit;
AD0IN5          :linkage    bit;
AD0IN6          :linkage    bit;
AD0IN7          :linkage    bit;
AD1IN0          :linkage    bit;
AD1IN1          :linkage    bit;
AD1IN2          :linkage    bit;
AD1IN3          :linkage    bit;
DA0             :linkage    bit;
DA1             :linkage    bit;
P172            :in         bit;
P173            :in         bit;
AVSS_60         :linkage    bit;
VCCI_61         :linkage    bit;
VSS_62          :linkage    bit;
P174            :inout      bit;
P175            :inout      bit;
VCCE_65         :linkage    bit;
P82             :inout      bit;
P83             :inout      bit;
P84             :inout      bit;
P85             :inout      bit;
P86             :inout      bit;
P87             :inout      bit;
VSS_72          :linkage    bit;
FVCC_73         :linkage    bit;
P61             :inout      bit;
P62             :inout      bit;
P63             :inout      bit;
P64             :in         bit;
P70             :inout      bit;
P71             :inout      bit;
P72             :inout      bit;
P73             :inout      bit;
P74             :inout      bit;
P75             :inout      bit;
P76             :inout      bit;
P77             :inout      bit;
P93             :in         bit;
P94             :inout      bit;
P95             :inout      bit;
P96             :inout      bit;
P97             :in         bit;
RESET           :in         bit;
MOD0            :in         bit;
MOD1            :in         bit;
FP              :in         bit;
VCCE_95         :linkage    bit;
VSS_96          :linkage    bit;
P110            :inout      bit;
P111            :inout      bit;
P112            :inout      bit;
P113            :inout      bit;
P114            :inout      bit;
P115            :inout      bit;
P116            :inout      bit;
P117            :inout      bit;
P100            :inout      bit;
```

Figure 21.5.13 BSDL Description for the 32172(2/11)

```
P101                    :inout          bit;
P102                    :inout          bit;
VDD_108                 :linkage        bit;
TMS                     :in             bit;
TCK                     :in             bit;
TRST                    :in             bit;
TDO                     :out            bit;
TDI                     :in             bit;
P103                    :inout          bit;
P104                    :inout          bit;
P105                    :inout          bit;
P106                    :inout          bit;
P107                    :inout          bit;
P124                    :in             bit;
P125                    :in             bit;
P126                    :in             bit;
P127                    :in             bit;
VCCI_123                :linkage        bit;
P130                    :in             bit;
P131                    :in             bit;
P132                    :in             bit;
P133                    :in             bit;
P134                    :in             bit;
P135                    :in             bit;
P136                    :in             bit;
P137                    :in             bit;
VCCE_132                :linkage        bit;
P150                    :inout          bit;
P153                    :inout          bit;
P41                     :inout          bit;
P42                     :inout          bit;
VCCI_137                :linkage        bit;
VSS_138                 :linkage        bit;
P43                     :inout          bit;
P44                     :inout          bit;
P45                     :inout          bit;
P46                     :inout          bit;
P47                     :inout          bit;
P220                    :inout          bit
);

use STD_1149_1_1994.all;

attribute COMPONENT_CONFORMANCE of M32172F2VFP : entity is "STD_1149_1_1993";

attribute PIN_MAP of M32172F2VFP : entity is PHYSICAL_PIN_MAP;

constant P6Q144 : PIN_MAP_STRING :=
"P221                   :1," &
"P225                   :2," &
"OSCVSS_3               :3," &
"XIN                    :4," &
"XOUT                   :5," &
"OSCVCC_6               :6," &
"VCNT_7                 :7," &
"P30                    :8," &
"P31                    :9," &
"P32                    :10," &
"P33                    :11," &
"P34                    :12," &
```

Figure 21.5.14 BSDL Description for the 32172(3/11)

```
"P35                :13," &
"P36                :14," &
"P37                :15," &
"P20                :16," &
"P21                :17," &
"P22                :18," &
"P23                :19," &
"VCCE_20            :20," &
"VSS_21             :21," &
"P24                :22," &
"P25                :23," &
"P26                :24," &
"P27                :25," &
"P00                :26," &
"P01                :27," &
"P02                :28," &
"P03                :29," &
"P04                :30," &
"P05                :31," &
"P06                :32," &
"P07                :33," &
"P10                :34," &
"P11                :35," &
"P12                :36," &
"P13                :37," &
"P14                :38," &
"P15                :39," &
"P16                :40," &
"P17                :41," &
"VREF_42            :42," &
"AVCC_43            :43," &
"AD0IN0             :44," &
"AD0IN1             :45," &
"AD0IN2             :46," &
"AD0IN3             :47," &
"AD0IN4             :48," &
"AD0IN5             :49," &
"AD0IN6             :50," &
"AD0IN7             :51," &
"AD1IN0             :52," &
"AD1IN1             :53," &
"AD1IN2             :54," &
"AD1IN3             :55," &
"DA0                :56," &
"DA1                :57," &
"P172               :58," &
"P173               :59," &
"AVSS_60            :60," &
"VCCI_61            :61," &
"VSS_62             :62," &
"P174               :63," &
"P175               :64," &
"VCCE_65            :65," &
"P82                :66," &
"P83                :67," &
"P84                :68," &
"P85                :69," &
"P86                :70," &
"P87                :71," &
"VSS_72             :72," &
```

Figure 21.5.15 BSDL Description for the 32172(4/11)

```
"FVCC_73        :73," &
"P61            :74," &
"P62            :75," &
"P63            :76," &
"P64            :77," &
"P70            :78," &
"P71            :79," &
"P72            :80," &
"P73            :81," &
"P74            :82," &
"P75            :83," &
"P76            :84," &
"P77            :85," &
"P93            :86," &
"P94            :87," &
"P95            :88," &
"P96            :89," &
"P97            :90," &
"RESET          :91," &
"MOD0           :92," &
"MOD1           :93," &
"FP             :94," &
"VCCE_95        :95," &
"VSS_96         :96," &
"P110           :97," &
"P111           :98," &
"P112           :99," &
"P113           :100," &
"P114           :101," &
"P115           :102," &
"P116           :103," &
"P117           :104," &
"P100           :105," &
"P101           :106," &
"P102           :107," &
"VDD_108        :108," &
"TMS            :109," &
"TCK            :110," &
"TRST           :111," &
"TDO            :112," &
"TDI            :113," &
"P103           :114," &
"P104           :115," &
"P105           :116," &
"P106           :117," &
"P107           :118," &
"P124           :119," &
"P125           :120," &
"P126           :121," &
"P127           :122," &
"VCCI_123       :123," &
"P130           :124," &
"P131           :125," &
"P132           :126," &
"P133           :127," &
"P134           :128," &
"P135           :129," &
"P136           :130," &
"P137           :131," &
"VCCE_132       :132," &
```

Figure 21.5.16 BSDL Description for the 32172(5/11)

```
"P150                   :133," &
"P153                   :134," &
"P41                    :135," &
"P42                    :136," &
"VCCI_137               :137," &
"VSS_138                :138," &
"P43                    :139," &
"P44                    :140," &
"P45                    :141," &
"P46                    :142," &
"P47                    :143," &
"P220                   :144" ;

attribute TAP_SCAN_IN of TDI : signal is true;
attribute TAP_SCAN_MODE of TMS : signal is true;
attribute TAP_SCAN_OUT of TDO : signal is true;

attribute TAP_SCAN_CLOCK of TCK : signal is (5.0e6, BOTH);
attribute TAP_SCAN_RESET of TRST : signal is true;

attribute INSTRUCTION_LENGTH of M32172F2VFP : entity is 6;

attribute INSTRUCTION_OPCODE of M32172F2VFP : entity is
        "BYPASS             (111111)," &
        "SAMPLE             (000001)," &
        "EXTEST             (000000)," &
        "IDCODE             (000010)," &
        "USERCODE           (000011)," &
        "MDM_SYSTEM         (001000)," &
        "MDM_CONTROL        (001001)," &
        "MDM_SETUP          (001010)," &
        "MTM_CONTROL        (001111)," &
        "MON_CODE           (010000)," &
        "MON_DATA           (010001)," &
        "MON_PARAM          (010010)," &
        "MON_ACCESS         (010011)," &
        "DMA_RADDR          (011000)," &
        "DMA_RDATA          (011001)," &
        "DMA_RTYPE          (011010)," &
        "DMA_ACCESS         (011011)," &
        "RTDENB             (100000)" ;

attribute INSTRUCTION_CAPTURE of M32172F2VFP : entity is "110001";

attribute INSTRUCTION_PRIVATE of M32172F2VFP : entity is
        "MDM_SYSTEM," &
        "MDM_CONTROL," &
        "MDM_SETUP," &
        "MTM_CONTROL," &
        "MON_CODE," &
        "MON_DATA," &
        "MON_PARAM," &
        "MON_ACCESS," &
        "DMA_RADDR," &
        "DMA_RDATA," &
        "DMA_RTYPE," &
        "DMA_ACCESS," &
        "RTDENB" ;
```

Figure 21.5.17 BSDL Description for the 32172(6/11)

```
                attribute IDCODE_REGISTER of M32172F2VFP : entity is
                        "0000" &                        -- version
                        "0011001000100001" & -- part number
                        "00000011100" &                 -- manufacturer's identity
                        "1";                            -- required by 1149.1

---             attribute USERCODE_REGISTER of M32172F2VFP : entity is
---                     "0000 0000 0000 0000" &         -- reserved
---                     "0000" &                        -- reserved
---                     "0001" &                        -- ROM
---                     "0000" &                        -- ISA
---                     "0001";                         -- SDI version

                attribute REGISTER_ACCESS of M32172F2VFP : entity is
                        "Bypass                         (BYPASS)," &
                        "Boundary                       (SAMPLE, EXTEST)," &
                        "Device_ID                      (IDCODE)," &
                        "USERCODE_REG[32]               (USERCODE)," &
                        "MDM_SYSTEM_REG[17]             (MDM_SYSTEM)," &
                        "MDM_CONTROL_REG[20]            (MDM_CONTROL)," &
                        "MDM_SETUP_REG[2]               (MDM_SETUP)," &
                        "MTM_CONTROL_REG[4]             (MTM_CONTROL)," &
                        "MON_CODE_REG[32]               (MON_CODE)," &
                        "MON_DATA_REG[32]               (MON_DATA)," &
                        "MON_PARAM_REG[32]              (MON_PARAM)," &
                        "MON_ACCESS_REG[4]              (MON_ACCESS)," &
                        "DMA_RADDR_REG[32]              (DMA_RADDR)," &
                        "DMA_RDATA_REG[32]              (DMA_RDATA)," &
                        "DMA_RTYPE_REG[3]               (DMA_RTYPE)," &
                        "DMA_ACCESS_REG[3]              (DMA_ACCESS)," &
                        "RTDENB_REG[1]                  (RTDENB)";

                attribute BOUNDARY_LENGTH of M32172F2VFP : entity is 265;

                attribute BOUNDARY_REGISTER of M32172F2VFP : entity is
                 --
                 -- num  cell      port            function        safe [ccell disval rslt]
                 --
                        "264 (BC_4,     P103,           observe_only,   X)," &
                        "263 (BC_1,     P103,           output3,        X, 262, 0, Z)," &
                        "262 (BC_1,     *,              control,        0)," &
                        "261 (BC_4,     P104,           observe_only,   X)," &
                        "260 (BC_1,     P104,           output3,        X, 259, 0, Z)," &
                        "259 (BC_1,     *,              control,        0)," &
                        "258 (BC_4,     P105,           observe_only,   X)," &
                        "257 (BC_1,     P105,           output3,        X, 256, 0, Z)," &
                        "256 (BC_1,     *,              control,        0)," &
                        "255 (BC_4,     P106,           observe_only,   X)," &
                        "254 (BC_1,     P106,           output3,        X, 253, 0, Z)," &
                        "253 (BC_1,     *,              control,        0)," &
                        "252 (BC_4,     P107,           observe_only,   X)," &
                        "251 (BC_1,     P107,           output3,        X, 250, 0, Z)," &
                        "250 (BC_1,     *,              control,        0)," &
                        "249 (BC_4,     P124,           observe_only,   X)," &
                        "248 (BC_4,     P125,           observe_only,   X)," &
                        "247 (BC_4,     P126,           observe_only,   X)," &
                        "246 (BC_4,     P127,           observe_only,   X)," &
                        "245 (BC_4,     P130,           observe_only,   X)," &
                        "244 (BC_4,     P131,           observe_only,   X)," &
                        "243 (BC_4,     P132,           observe_only,   X)," &
```

Figure 21.5.18 BSDL Description for the 32172(7/11)

```
"242 (BC_4,     P133,     observe_only,   X)," &
"241 (BC_4,     P134,     observe_only,   X)," &
"240 (BC_4,     P135,     observe_only,   X)," &
"239 (BC_4,     P136,     observe_only,   X)," &
"238 (BC_4,     P137,     observe_only,   X)," &
"237 (BC_4,     P150,     observe_only,   X)," &
"236 (BC_1,     P150,     output3,        X, 235, 0, Z)," &
"235 (BC_1,     *,        control,        0)," &
"234 (BC_4,     P153,     observe_only,   X)," &
"233 (BC_1,     P153,     output3,        X, 232, 0, Z)," &
"232 (BC_1,     *,        control,        0)," &
"231 (BC_4,     P41,      observe_only,   X)," &
"230 (BC_1,     P41,      output3,        X, 229, 0, Z)," &
"229 (BC_1,     *,        control,        0)," &
"228 (BC_4,     P42,      observe_only,   X)," &
"227 (BC_1,     P42,      output3,        X, 226, 0, Z)," &
"226 (BC_1,     *,        control,        0)," &
"225 (BC_4,     P43,      observe_only,   X)," &
"224 (BC_1,     P43,      output3,        X, 223, 0, Z)," &
"223 (BC_1,     *,        control,        0)," &
"222 (BC_4,     P44,      observe_only,   X)," &
"221 (BC_1,     P44,      output3,        X, 220, 0, Z)," &
"220 (BC_1,     *,        control,        0)," &
"219 (BC_4,     P45,      observe_only,   X)," &
"218 (BC_1,     P45,      output3,        X, 217, 0, Z)," &
"217 (BC_1,     *,        control,        0)," &
"216 (BC_4,     P46,      observe_only,   X)," &
"215 (BC_1,     P46,      output3,        X, 214, 0, Z)," &
"214 (BC_1,     *,        control,        0)," &
"213 (BC_4,     P47,      observe_only,   X)," &
"212 (BC_1,     P47,      output3,        X, 211, 0, Z)," &
"211 (BC_1,     *,        control,        0)," &
"210 (BC_4,     P220,     observe_only,   X)," &
"209 (BC_1,     P220,     output3,        X, 208, 0, Z)," &
"208 (BC_1,     *,        control,        0)," &
"207 (BC_4,     P221,     observe_only,   X)," &
"206 (BC_4,     P225,     observe_only,   X)," &
"205 (BC_1,     P225,     output3,        X, 204, 0, Z)," &
"204 (BC_1,     *,        control,        0)," &
"203 (BC_4,     P30,      observe_only,   X)," &
"202 (BC_1,     P30,      output3,        X, 201, 0, Z)," &
"201 (BC_1,     *,        control,        0)," &
"200 (BC_4,     P31,      observe_only,   X)," &
"199 (BC_1,     P31,      output3,        X, 198, 0, Z)," &
"198 (BC_1,     *,        control,        0)," &
"197 (BC_4,     P32,      observe_only,   X)," &
"196 (BC_1,     P32,      output3,        X, 195, 0, Z)," &
"195 (BC_1,     *,        control,        0)," &
"194 (BC_4,     P33,      observe_only,   X)," &
"193 (BC_1,     P33,      output3,        X, 192, 0, Z)," &
"192 (BC_1,     *,        control,        0)," &
"191 (BC_4,     P34,      observe_only,   X)," &
"190 (BC_1,     P34,      output3,        X, 189, 0, Z)," &
"189 (BC_1,     *,        control,        0)," &
"188 (BC_4,     P35,      observe_only,   X)," &
"187 (BC_1,     P35,      output3,        X, 186, 0, Z)," &
"186 (BC_1,     *,        control,        0)," &
"185 (BC_4,     P36,      observe_only,   X)," &
"184 (BC_1,     P36,      output3,        X, 183, 0, Z)," &
"183 (BC_1,     *,        control,        0)," &
```

Figure 21.5.19 BSDL Description for the 32172(8/11)

```
"182 (BC_4,     P37,        observe_only,   X)," &
"181 (BC_1,     P37,        output3,        X, 180, 0, Z)," &
"180 (BC_1,     *,          control,        0)," &
"179 (BC_4,     P20,        observe_only,   X)," &
"178 (BC_1,     P20,        output3,        X, 177, 0, Z)," &
"177 (BC_1,     *,          control,        0)," &
"176 (BC_4,     P21,        observe_only,   X)," &
"175 (BC_1,     P21,        output3,        X, 174, 0, Z)," &
"174 (BC_1,     *,          control,        0)," &
"173 (BC_4,     P22,        observe_only,   X)," &
"172 (BC_1,     P22,        output3,        X, 171, 0, Z)," &
"171 (BC_1,     *,          control,        0)," &
"170 (BC_4,     P23,        observe_only,   X)," &
"169 (BC_1,     P23,        output3,        X, 168, 0, Z)," &
"168 (BC_1,     *,          control,        0)," &
"167 (BC_4,     P24,        observe_only,   X)," &
"166 (BC_1,     P24,        output3,        X, 165, 0, Z)," &
"165 (BC_1,     *,          control,        0)," &
"164 (BC_4,     P25,        observe_only,   X)," &
"163 (BC_1,     P25,        output3,        X, 162, 0, Z)," &
"162 (BC_1,     *,          control,        0)," &
"161 (BC_4,     P26,        observe_only,   X)," &
"160 (BC_1,     P26,        output3,        X, 159, 0, Z)," &
"159 (BC_1,     *,          control,        0)," &
"158 (BC_4,     P27,        observe_only,   X)," &
"157 (BC_1,     P27,        output3,        X, 156, 0, Z)," &
"156 (BC_1,     *,          control,        0)," &
"155 (BC_4,     P00,        observe_only,   X)," &
"154 (BC_1,     P00,        output3,        X, 153, 0, Z)," &
"153 (BC_1,     *,          control,        0)," &
"152 (BC_4,     P01,        observe_only,   X)," &
"151 (BC_1,     P01,        output3,        X, 150, 0, Z)," &
"150 (BC_1,     *,          control,        0)," &
"149 (BC_4,     P02,        observe_only,   X)," &
"148 (BC_1,     P02,        output3,        X, 147, 0, Z)," &
"147 (BC_1,     *,          control,        0)," &
"146 (BC_4,     P03,        observe_only,   X)," &
"145 (BC_1,     P03,        output3,        X, 144, 0, Z)," &
"144 (BC_1,     *,          control,        0)," &
"143 (BC_4,     P04,        observe_only,   X)," &
"142 (BC_1,     P04,        output3,        X, 141, 0, Z)," &
"141 (BC_1,     *,          control,        0)," &
"140 (BC_4,     P05,        observe_only,   X)," &
"139 (BC_1,     P05,        output3,        X, 138, 0, Z)," &
"138 (BC_1,     *,          control,        0)," &
"137 (BC_4,     P06,        observe_only,   X)," &
"136 (BC_1,     P06,        output3,        X, 135, 0, Z)," &
"135 (BC_1,     *,          control,        0)," &
"134 (BC_4,     P07,        observe_only,   X)," &
"133 (BC_1,     P07,        output3,        X, 132, 0, Z)," &
"132 (BC_1,     *,          control,        0)," &
"131 (BC_4,     P10,        observe_only,   X)," &
"130 (BC_1,     P10,        output3,        X, 129, 0, Z)," &
"129 (BC_1,     *,          control,        0)," &
"128 (BC_4,     P11,        observe_only,   X)," &
"127 (BC_1,     P11,        output3,        X, 126, 0, Z)," &
"126 (BC_1,     *,          control,        0)," &
"125 (BC_4,     P12,        observe_only,   X)," &
"124 (BC_1,     P12,        output3,        X, 123, 0, Z)," &
"123 (BC_1,     *,          control,        0)," &
```

Figure 21.5.20 BSDL Description for the 32172(9/11)

```
"122 (BC_4,     P13,        observe_only,   X)," &
"121 (BC_1,     P13,        output3,        X, 120, 0, Z)," &
"120 (BC_1,     *,          control,        0)," &
"119 (BC_4,     P14,        observe_only,   X)," &
"118 (BC_1,     P14,        output3,        X, 117, 0, Z)," &
"117 (BC_1,     *,          control,        0)," &
"116 (BC_4,     P15,        observe_only,   X)," &
"115 (BC_1,     P15,        output3,        X, 114, 0, Z)," &
"114 (BC_1,     *,          control,        0)," &
"113 (BC_4,     P16,        observe_only,   X)," &
"112 (BC_1,     P16,        output3,        X, 111, 0, Z)," &
"111 (BC_1,     *,          control,        0)," &
"110 (BC_4,     P17,        observe_only,   X)," &
"109 (BC_1,     P17,        output3,        X, 108, 0, Z)," &
"108 (BC_1,     *,          control,        0)," &
"107 (BC_4,     P172,       observe_only,   X)," &
"106 (BC_4,     P173,       observe_only,   X)," &
"105 (BC_4,     P174,       observe_only,   X)," &
"104 (BC_1,     P174,       output3,        X, 103, 0, Z)," &
"103 (BC_1,     *,          control,        0)," &
"102 (BC_4,     P175,       observe_only,   X)," &
"101 (BC_1,     P175,       output3,        X, 100, 0, Z)," &
"100 (BC_1,     *,          control,        0)," &
"99  (BC_4,     P82,        observe_only,   X)," &
"98  (BC_1,     P82,        output3,        X, 97, 0, Z)," &
"97  (BC_1,     *,          control,        0)," &
"96  (BC_4,     P83,        observe_only,   X)," &
"95  (BC_1,     P83,        output3,        X, 94, 0, Z)," &
"94  (BC_1,     *,          control,        0)," &
"93  (BC_4,     P84,        observe_only,   X)," &
"92  (BC_1,     P84,        output3,        X, 91, 0, Z)," &
"91  (BC_1,     *,          control,        0)," &
"90  (BC_4,     P85,        observe_only,   X)," &
"89  (BC_1,     P85,        output3,        X, 88, 0, Z)," &
"88  (BC_1,     *,          control,        0)," &
"87  (BC_4,     P86,        observe_only,   X)," &
"86  (BC_1,     P86,        output3,        X, 85, 0, Z)," &
"85  (BC_1,     *,          control,        0)," &
"84  (BC_4,     P87,        observe_only,   X)," &
"83  (BC_1,     P87,        output3,        X, 82, 0, Z)," &
"82  (BC_1,     *,          control,        0)," &
"81  (BC_4,     P61,        observe_only,   X)," &
"80  (BC_1,     P61,        output3,        X, 79, 0, Z)," &
"79  (BC_1,     *,          control,        0)," &
"78  (BC_4,     P62,        observe_only,   X)," &
"77  (BC_1,     P62,        output3,        X, 76, 0, Z)," &
"76  (BC_1,     *,          control,        0)," &
"75  (BC_4,     P63,        observe_only,   X)," &
"74  (BC_1,     P63,        output3,        X, 73, 0, Z)," &
"73  (BC_1,     *,          control,        0)," &
"72  (BC_4,     P64,        observe_only,   X)," &
"71  (BC_4,     P70,        observe_only,   X)," &
"70  (BC_1,     P70,        output3,        X, 69, 0, Z)," &
"69  (BC_1,     *,          control,        0)," &
"68  (BC_4,     P71,        observe_only,   X)," &
"67  (BC_1,     P71,        output3,        X, 66, 0, Z)," &
"66  (BC_1,     *,          control,        0)," &
"65  (BC_4,     P72,        observe_only,   X)," &
"64  (BC_1,     P72,        output3,        X, 63, 0, Z)," &
"63  (BC_1,     *,          control,        0)," &
```

Figure 21.5.21 BSDL Description for the 32172(10/11)

```
                "62 (BC_4,      P73,            observe_only,   X)," &
                "61 (BC_1,      P73,            output3,        X, 60, 0, Z)," &
                "60 (BC_1,      *,              control,        0)," &
                "59 (BC_4,      P74,            observe_only,   X)," &
                "58 (BC_1,      P74,            output3,        X, 57, 0, Z)," &
                "57 (BC_1,      *,              control,        0)," &
                "56 (BC_4,      P75,            observe_only,   X)," &
                "55 (BC_1,      P75,            output3,        X, 54, 0, Z)," &
                "54 (BC_1,      *,              control,        0)," &
                "53 (BC_4,      P76,            observe_only,   X)," &
                "52 (BC_1,      P76,            output3,        X, 51, 0, Z)," &
                "51 (BC_1,      *,              control,        0)," &
                "50 (BC_4,      P77,            observe_only,   X)," &
                "49 (BC_1,      P77,            output3,        X, 48, 0, Z)," &
                "48 (BC_1,      *,              control,        0)," &
                "47 (BC_4,      P93,            observe_only,   X)," &
                "46 (BC_4,      P94,            observe_only,   X)," &
                "45 (BC_1,      P94,            output3,        X, 44, 0, Z)," &
                "44 (BC_1,      *,              control,        0)," &
                "43 (BC_4,      P95,            observe_only,   X)," &
                "42 (BC_1,      P95,            output3,        X, 41, 0, Z)," &
                "41 (BC_1,      *,              control,        0)," &
                "40 (BC_4,      P96,            observe_only,   X)," &
                "39 (BC_1,      P96,            output3,        X, 38, 0, Z)," &
                "38 (BC_1,      *,              control,        0)," &
                "37 (BC_4,      P97,            observe_only,   X)," &
                "36 (BC_4,      RESET,          observe_only,   X)," &
                "35 (BC_4,      MOD0,           observe_only,   X)," &
                "34 (BC_4,      MOD1,           observe_only,   X)," &
                "33 (BC_4,      FP,             observe_only,   X)," &
                "32 (BC_4,      P110,           observe_only,   X)," &
                "31 (BC_1,      P110,           output3,        X, 30, 0, Z)," &
                "30 (BC_1,      *,              control,        0)," &
                "29 (BC_4,      P111,           observe_only,   X)," &
                "28 (BC_1,      P111,           output3,        X, 27, 0, Z)," &
                "27 (BC_1,      *,              control,        0)," &
                "26 (BC_4,      P112,           observe_only,   X)," &
                "25 (BC_1,      P112,           output3,        X, 24, 0, Z)," &
                "24 (BC_1,      *,              control,        0)," &
                "23 (BC_4,      P113,           observe_only,   X)," &
                "22 (BC_1,      P113,           output3,        X, 21, 0, Z)," &
                "21 (BC_1,      *,              control,        0)," &
                "20 (BC_4,      P114,           observe_only,   X)," &
                "19 (BC_1,      P114,           output3,        X, 18, 0, Z)," &
                "18 (BC_1,      *,              control,        0)," &
                "17 (BC_4,      P115,           observe_only,   X)," &
                "16 (BC_1,      P115,           output3,        X, 15, 0, Z)," &
                "15 (BC_1,      *,              control,        0)," &
                "14 (BC_4,      P116,           observe_only,   X)," &
                "13 (BC_1,      P116,           output3,        X, 12, 0, Z)," &
                "12 (BC_1,      *,              control,        0)," &
                "11 (BC_4,      P117,           observe_only,   X)," &
                "10 (BC_1,      P117,           output3,        X, 9, 0, Z)," &
                "9  (BC_1,      *,              control,        0)," &
                "8  (BC_4,      P100,           observe_only,   X)," &
                "7  (BC_1,      P100,           output3,        X, 6, 0, Z)," &
                "6  (BC_1,      *,              control,        0)," &
                "5  (BC_4,      P101,           observe_only,   X)," &
                "4  (BC_1,      P101,           output3,        X, 3, 0, Z)," &
                "3  (BC_1,      *,              control,        0)," &
                "2  (BC_4,      P102,           observe_only,   X)," &
                "1  (BC_1,      P102,           output3,        X, 0, 0, Z)," &
                "0  (BC_1,      *,              control,        0)";

end M32172F2VFP;
```

Figure 21.5.22 BSDL Description for the 32172(11/11)

21.6 Precautions on Board Design when Connecting the JTAG

The JTAG pins must have their wiring lengths matched during board design. This is necessary to accomplish fast, highly reliable communication with the JTAG tool.

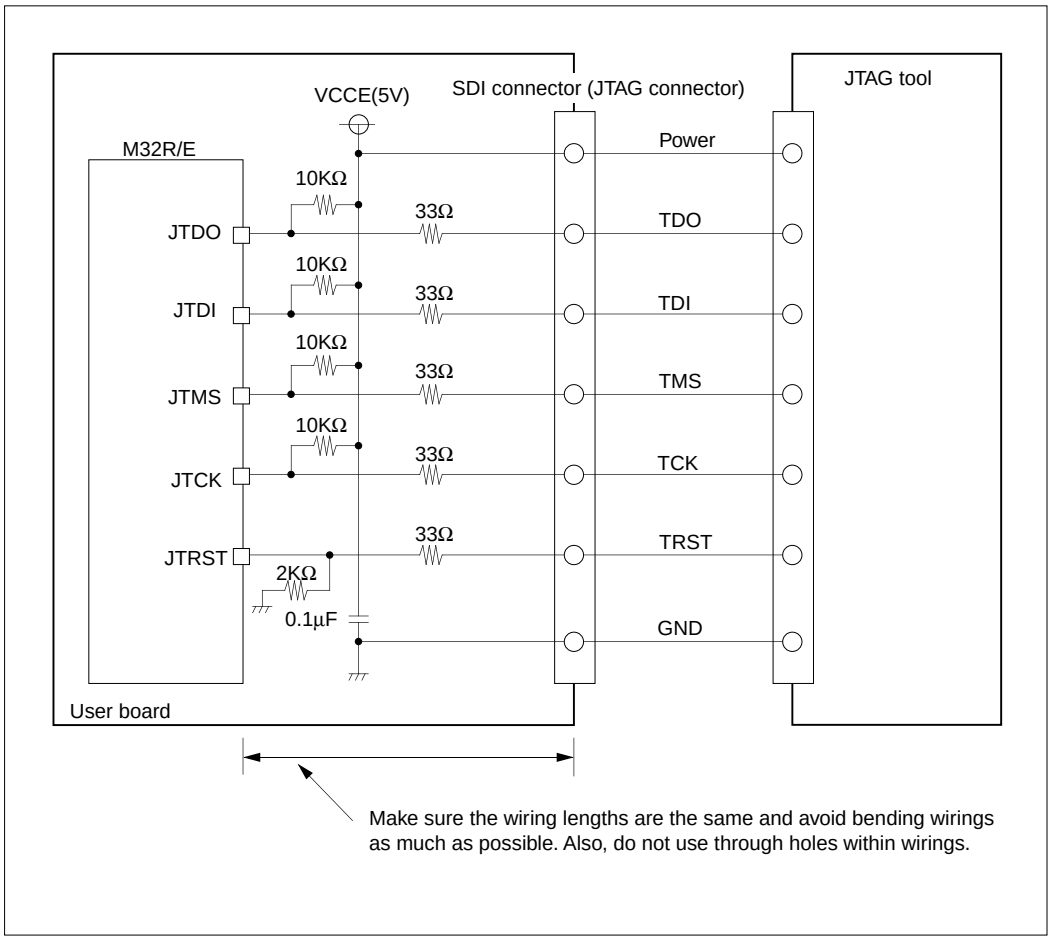


Figure 21.6.1 Precautions on Connecting the User Board and JTAG

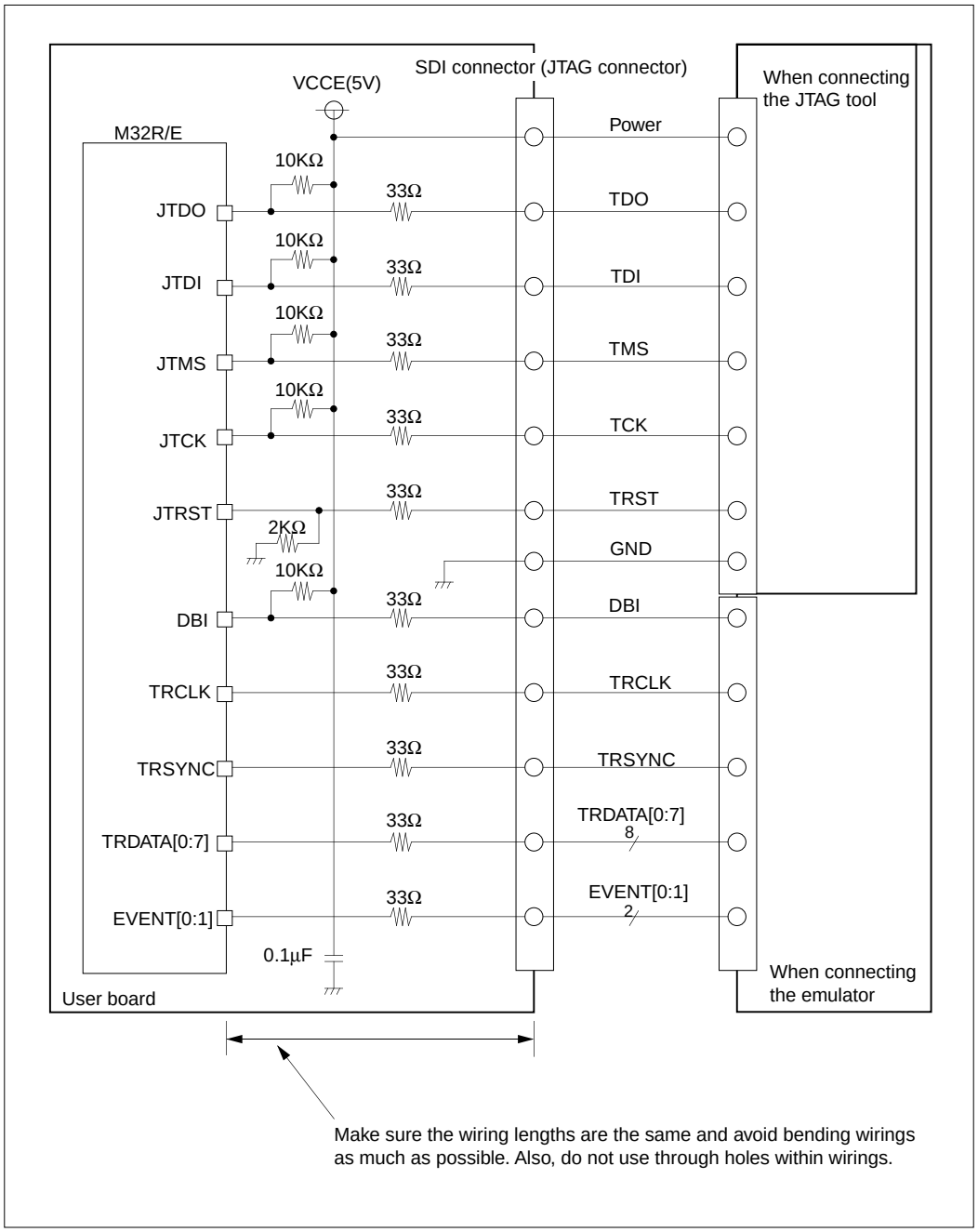


Figure 21.6.2 Precautions on Connecting the User Board and JTAG (for the 255FBGA)

21.7 Processing Pins when Not Using the JTAG

When not using the JTAG, make sure the pins on the microcomputer are processed properly, as shown below.

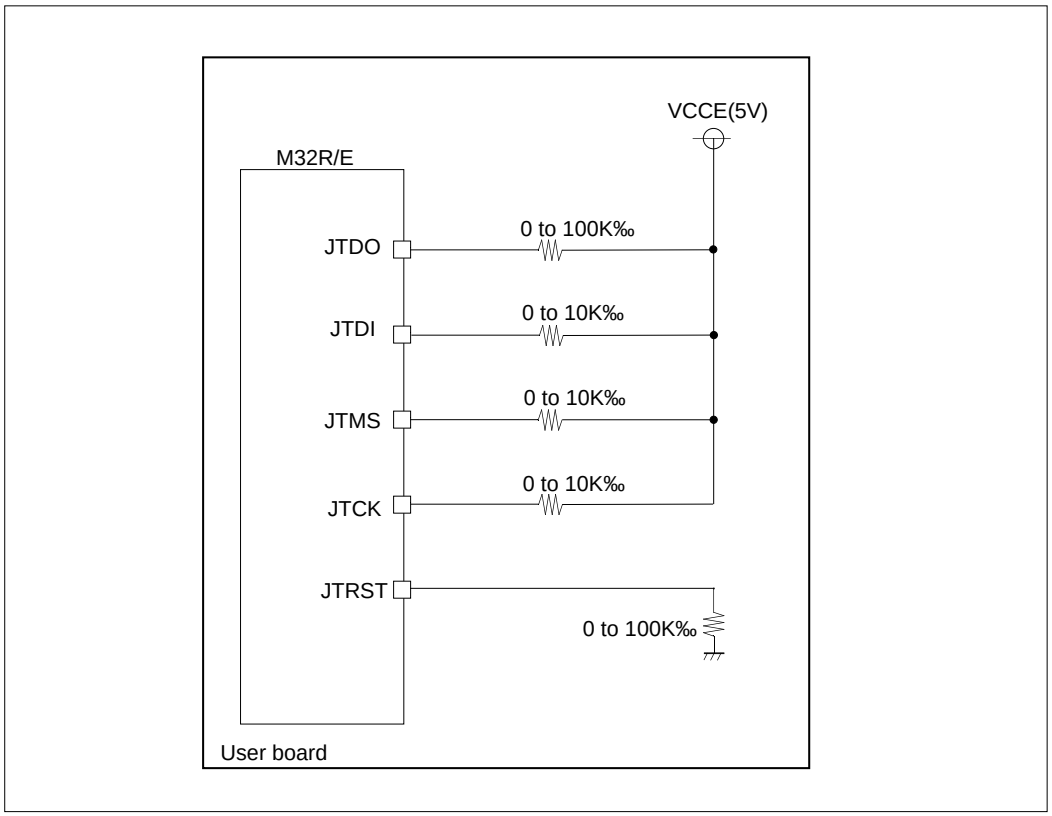


Figure 21.7.1 Processing Pins when Not Using the JTAG (for the 144LGFP)

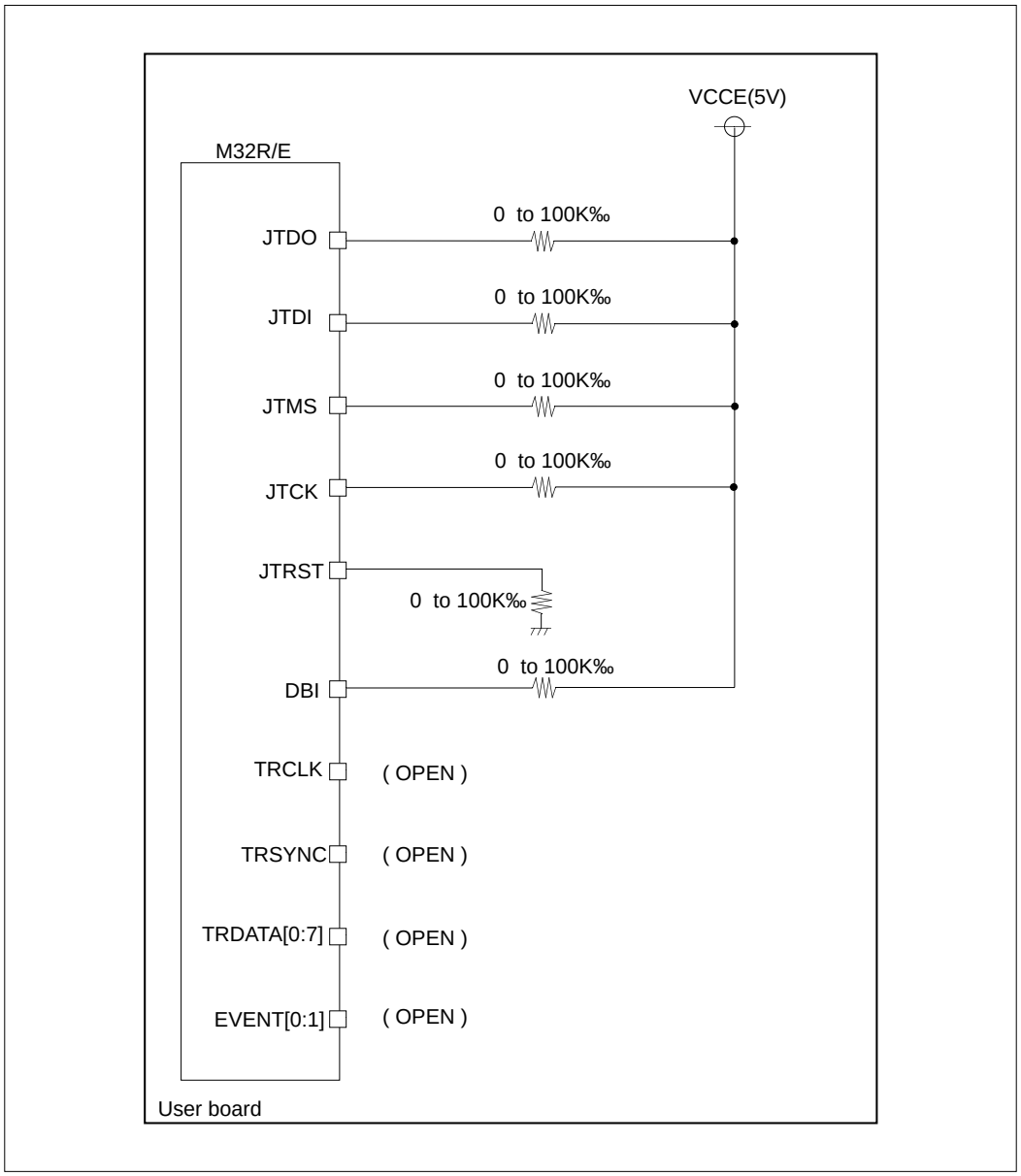


Figure 21.7.2 Processing Pins when Not Using the JTAG (for the 175FBGA)

CHAPTER 22

POWER-UP/POWER-SHUTDOWN SEQUENCE

22.1 Configuration of the Power Supply Circuit

To materialize high-speed operation at low power, the M32R/E is designed in such a way that its external interface circuits operate at 5 V power supply and all other circuits operate at 3.3 V. This requires that control timing of both 5 V and 3.3 V power supplies be considered when designing your circuit.

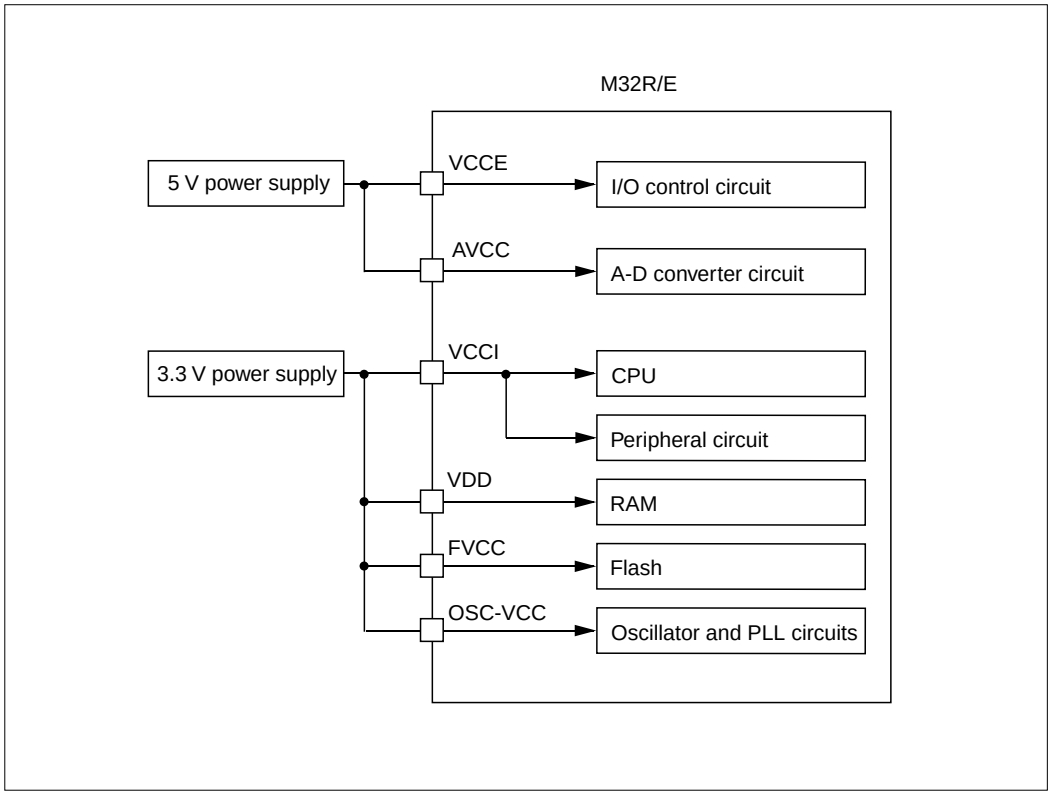


Figure 22.1.1 Configuration of the Power Supply Circuit

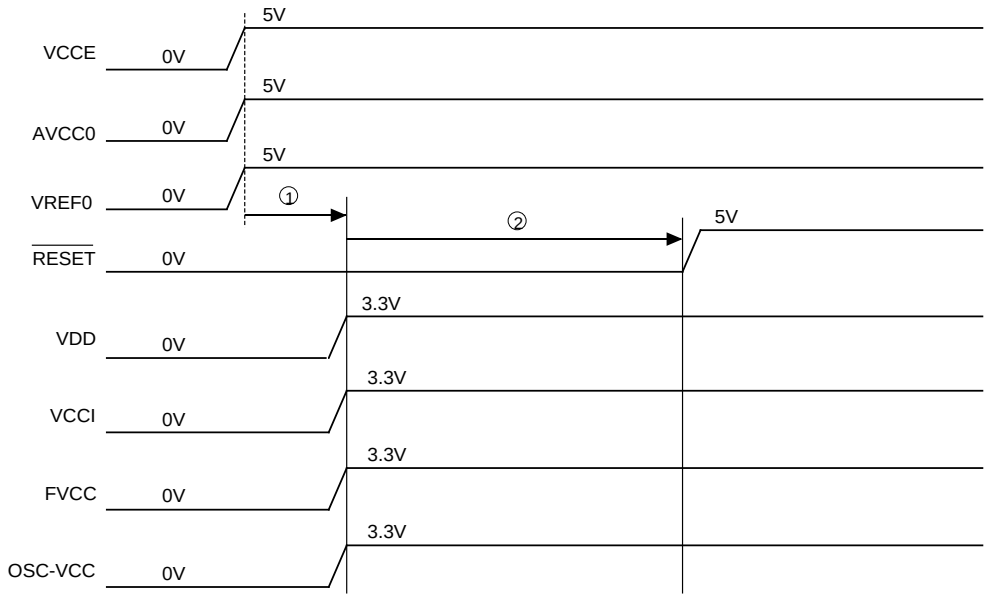
Table 22.1.1 List of Power Supply Functions

Type of Power Supply	Pin Name	Function
5.0 V system	VCCE	Supplies power to external I/O ports
	AVCC0	Power supply for A-D converter
	VREF0	Reference voltage for A-D converter
3.3 V system	VCCI	Supplies power to internal logic
	FVCC	Power supply for internal flash memory
	VDD	Power supply for internal RAM backup
	OSC-VCC	Power supply for oscillator and PLL circuits

22.2 Power-On Sequence

22.2.1 Power-On Sequence When Not Using RAM Backup

The diagram below shows a power-on sequence (5.0 V, 3.3 V power supply) of the M32R/E when not using RAM backup.



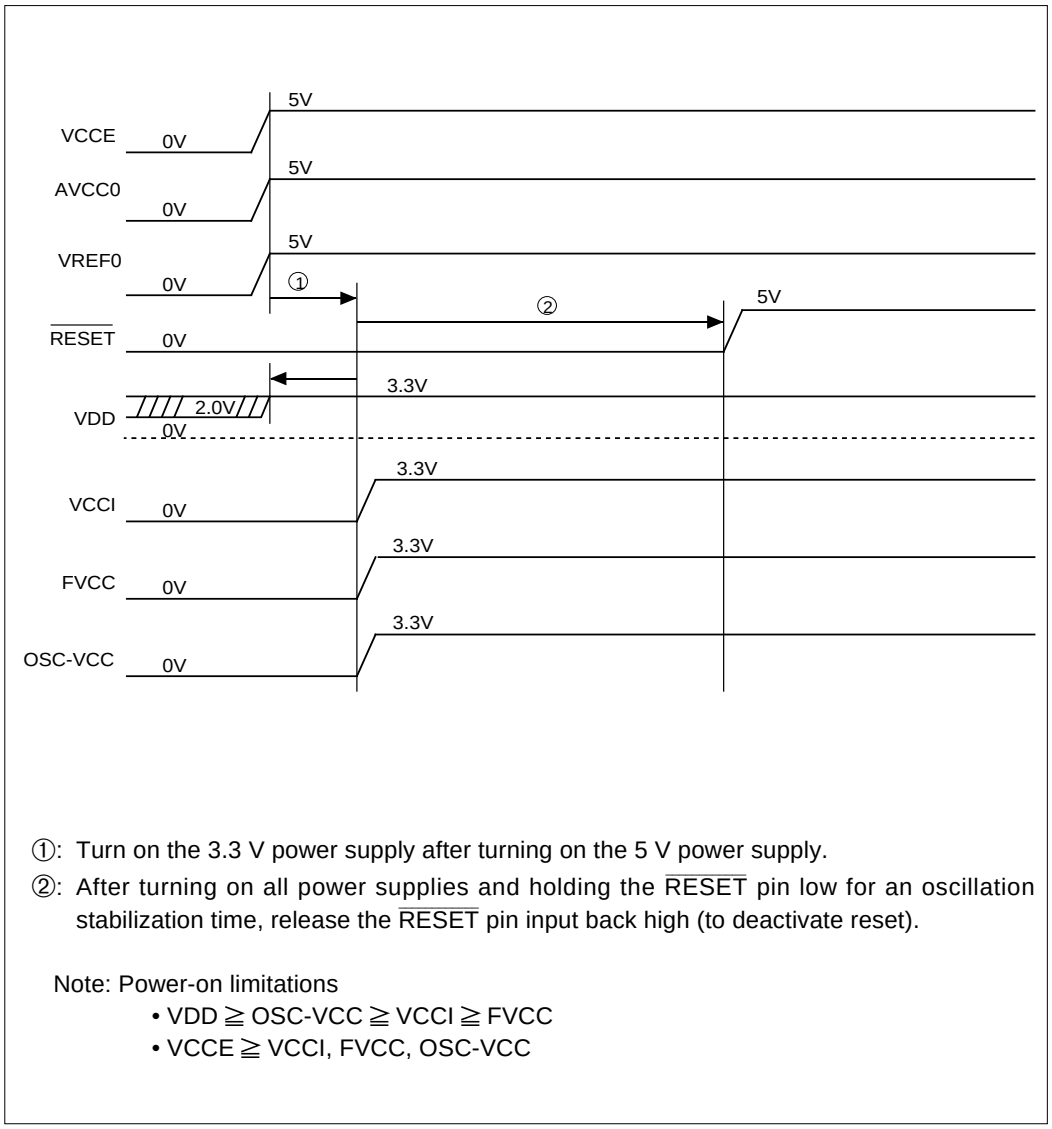
①: Turn on the 3.3 V power supply after turning on the 5 V power supply.
②: After turning on all power supplies and holding the $\overline{\text{RESET}}$ pin low for an oscillation stabilization time, release the $\overline{\text{RESET}}$ pin input back high (to deactivate reset).

Note: Power-on limitations
- VDD ≧ OSC-VCC ≧ VCCI ≧ FVCC
- VCCE ≧ VCCI, FVCC, OSC-VCC

Figure 22.2.1 Power-On Sequence When Not Using RAM Backup

22.2.2 Power-On Sequence When Using RAM Backup

The diagram below shows a power-on sequence (5.0 V, 3.3 V power supply) of the M32R/E when using RAM backup.



①: Turn on the 3.3 V power supply after turning on the 5 V power supply.

②: After turning on all power supplies and holding the $\overline{\text{RESET}}$ pin low for an oscillation stabilization time, release the $\overline{\text{RESET}}$ pin input back high (to deactivate reset).

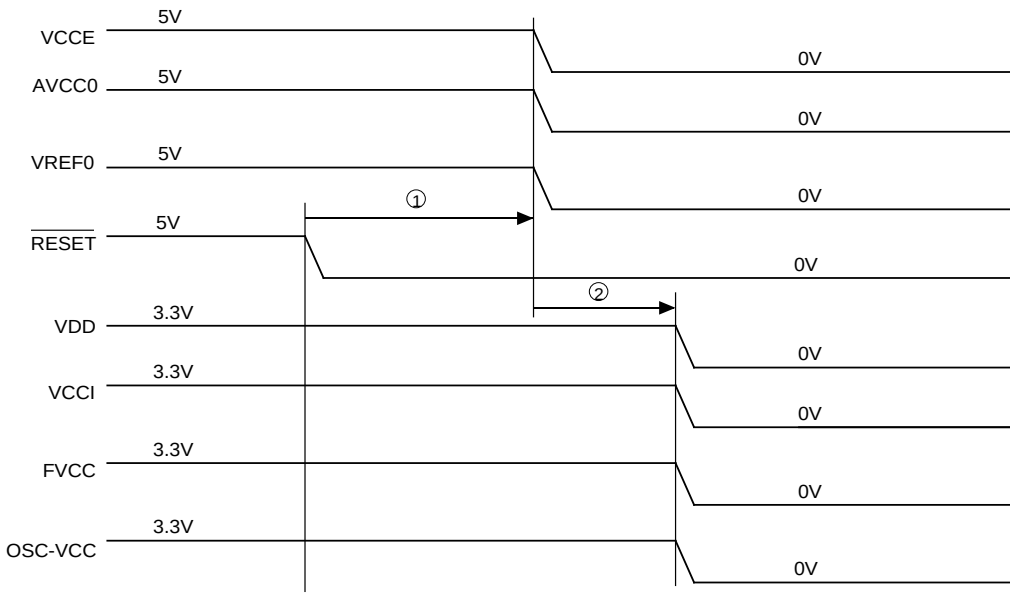
Note: Power-on limitations
- VDD ≧ OSC-VCC ≧ VCCI ≧ FVCC
- VCCE ≧ VCCI, FVCC, OSC-VCC

Figure 22.2.2 Power-On Sequence When Using RAM Backup

22.3 Power-Shutdown Sequence

22.3.1 Power-Shutdown Sequence When Not Using RAM Backup

The diagram below shows a power-shutdown sequence (5.0 V, 3.3 V power supply) of the M32R/E when not using RAM backup.



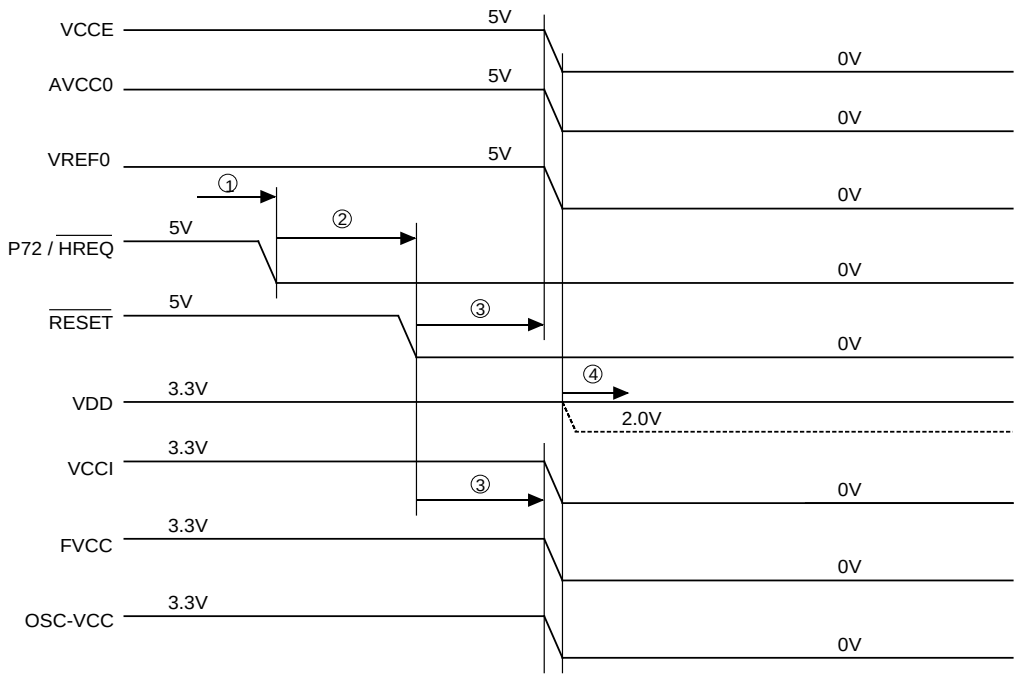
①: Pull the $\overline{\text{RESET}}$ pin input low.
②: Turn off the 5 V and the 3 V power supply after the $\overline{\text{RESET}}$ pin goes low.

Note: Power-shutdown requirements
- VDD ≧ VCCI ≧ FVCC
- OSC-VCC ≧ VCCI

Figure 22.3.1 Power-Shutdown Sequence When Not Using RAM Backup

22.3.2 Power-Shutdown Sequence When Using RAM Backup

The diagram below shows a power-shutdown sequence (5.0 V, 3.3 V power supply) of the M32R/E when using RAM backup.



①: Pull the $\overline{\text{HREQ}}$ pin input low to halt the CPU at end of bus cycle. Or disable RAM access in software. The M32R/E allows P72 to be used as HREQ irrespective of its operation mode.

②: With the CPU halted, pull the $\overline{\text{RESET}}$ pin input low. Or while RAM access is disabled, pull the $\overline{\text{RESET}}$ pin input low.

③: Turn off the 5 V and the 3.3 V power supply after the $\overline{\text{RESET}}$ pin goes low.

④: Reduce the VDD voltage from 3.3 V to 2.0 V as necessary.

Note: Power-shutdown requirements
- VDD ≧ VCCI ≧ FVCC
- OSC-VCC ≧ VCCI

Figure 22.3.2 Power-Shutdown Sequence When Using RAM Backup

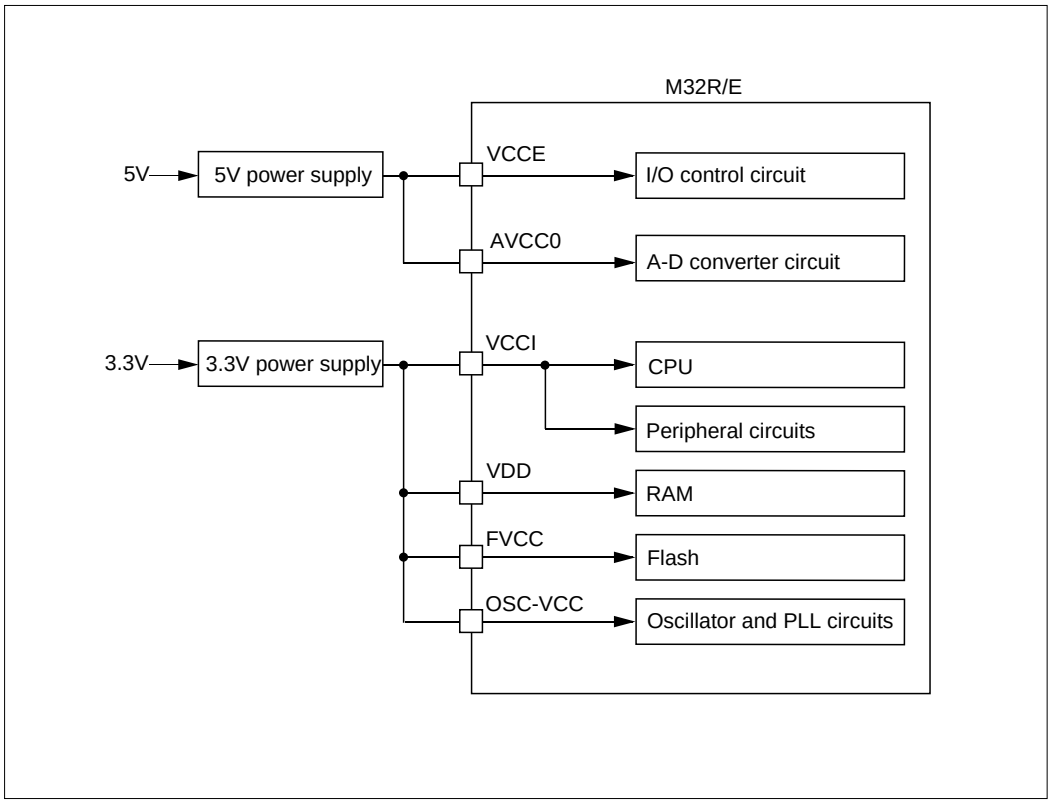


Figure 22.3.3 Microcomputer Ready to Run State (VCCE = 5 V, VCCI system = 3.3 V, VDD = 3.3 V)

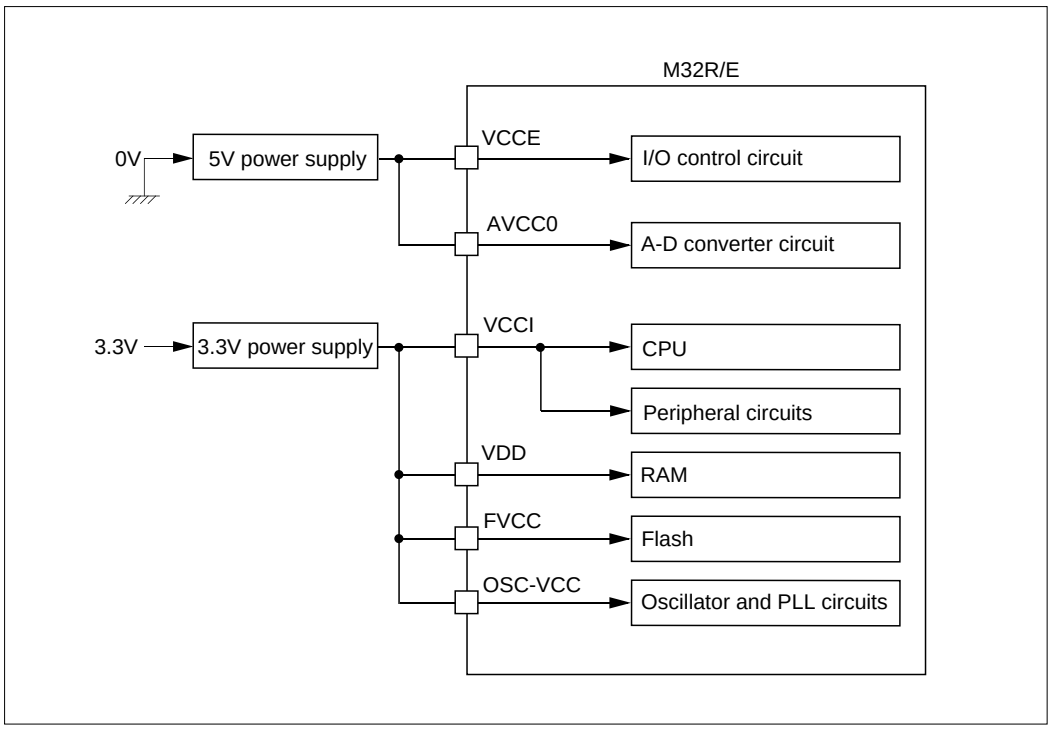


Figure 22.3.4 CPU Reset State

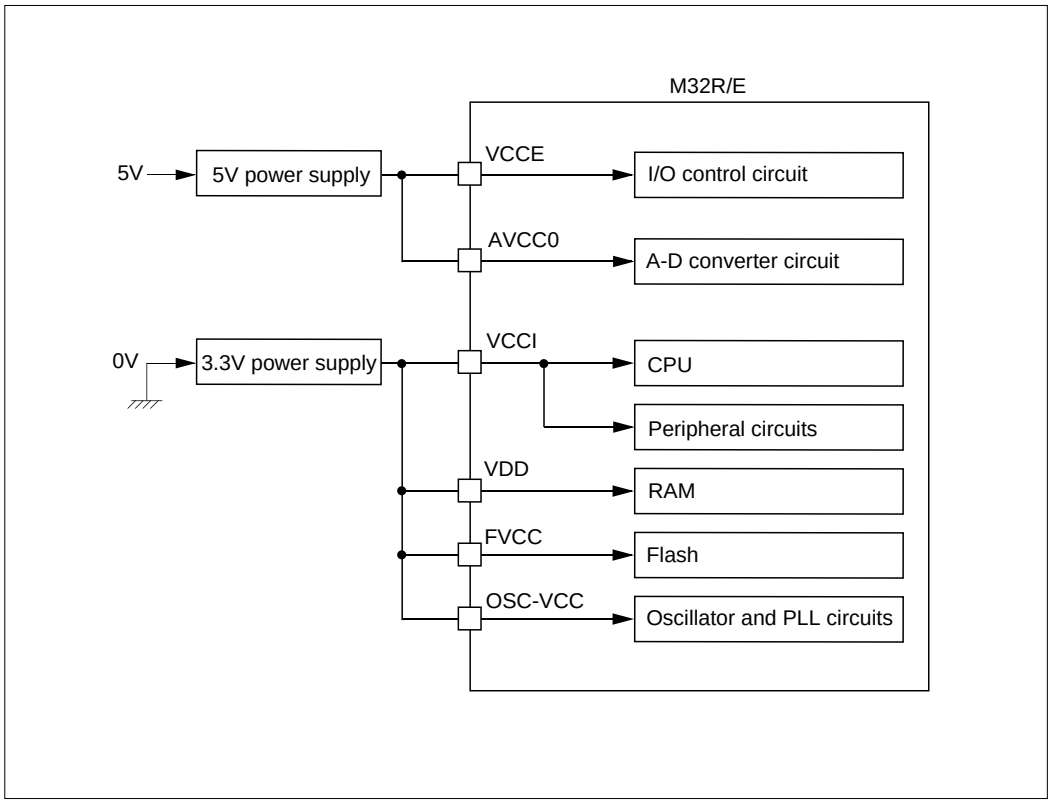


Figure 22.3.5 CPU Halt State

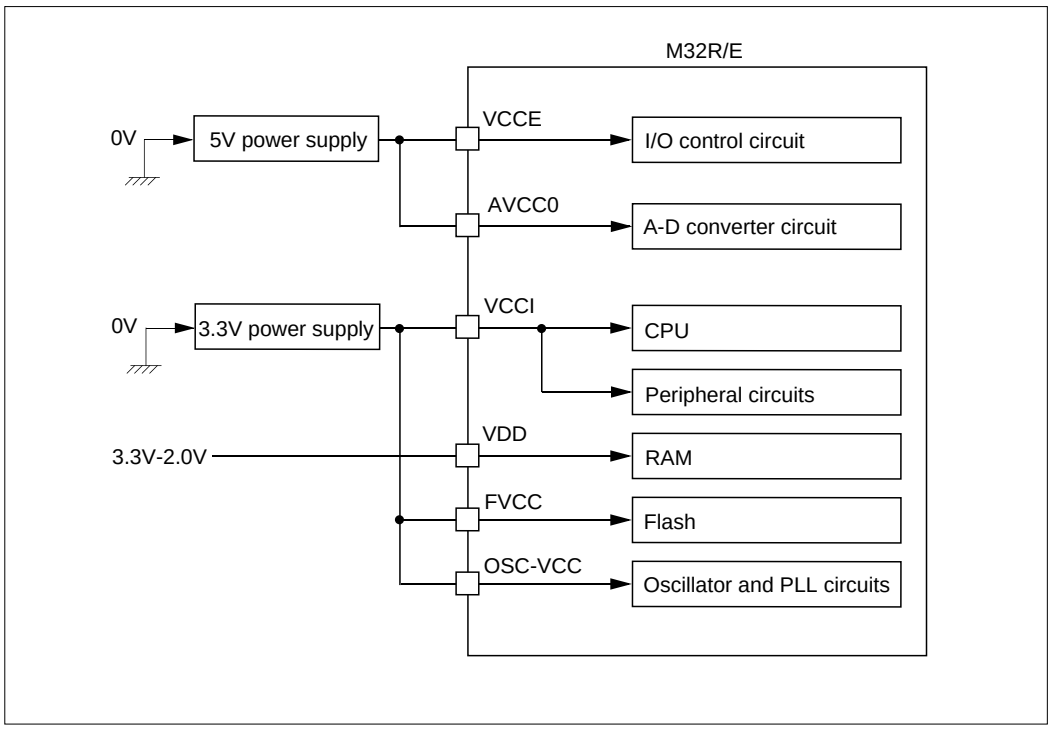


Figure 22.3.6 SRAM Data Backup State

CHAPTER 23

ELECTRICAL CHARACTERISTICS

23.1 Absolute Maximum Ratings

Absolute Maximum Ratings (-40 to 125°C)

Symbol	Parameter	Condition	Rated Value	Unit
VCCI	Internal Logic Power Supply Voltage	VDD≧VCCI≧FVCC=OSC-VCC	-0.3 — 4.2	V
VDD	RAM Power Supply Voltage	VDD≧VCCI≧FVCC=OSC-VCC	-0.3 — 4.2	V
OSC-VCC	PLL Power Supply Voltage	VDD≧VCCI≧FVCC=OSC-VCC	-0.3 — 4.2	V
FVCC	Flash Power Supply Voltage	VDD≧VCCI≧FVCC=OSC-VCC	-0.3 — 4.2	V
VCCE	External I/O Buffer Voltage	VCCE≧AVCC≧VREF	-0.3 — 6.5	V
AVCC	Analog Power Supply Voltage	VCCE≧AVCC≧VREF	-0.3 — 6.5	V
VREF	Analog Reference Voltage	VCCE≧AVCC≧VREF	-0.3 — 6.5	V
VI	Xin, VCNT		-0.3 — OSC-VCC+0.3	V
	Other		-0.3 — VCCE+0.3	
VO	Xout		-0.3 — OSC-VCC+0.3	V
	Other		-0.3 — VCCE+0.3	
Pd	Power Dissipation	Ta=-40 — 85°C	600	mW
		Ta=-40 — 125°C	500	mW
TOPR	Operating Ambient Temperature (Note)		-40 — 125	°C
Tstg	Storage Temperature		-65 — 150	°C

Note: This does not guarantee that the device will operate continuously at 125°C. If your application system is intended to operate at 125°C, please consult Mitsubishi.

23.2 Recommended Operating Conditions

Recommended Operating Conditions (Referenced to VCCE = 5 V±0.5 V, VCCI = 3.3 V±0.3 V, Ta = -40 to 85°C unless otherwise specified)

Symbol	Parameter		Rated Value			Unit
			MIN	TYP	MAX	
VCCE	External I/O Buffer Voltage (Note 1)		4.5	5.0	5.5	V
VCCI	Internal Logic Power Supply Voltage (Note 2)		3.0	3.3	3.6	V
VDD	RAM Power Supply Voltage (Note 2)		3.0	3.3	3.6	V
FVCC	Flash Power Supply Voltage (Note 2)		3.0	3.3	3.6	V
AVCC	Analog Power Supply Voltage (Note 1)		4.5	5.0	5.5	V
OSC-VCC	PLL Power Supply Voltage (Note 2)		3.0	3.3	3.6	V
VREF	Analog Reference Voltage (Note 1)			5.0	5.5	V
VIH	High Level Input Voltage	Ports P0-P22, $\overline{\text{RESET}}$, MOD0, MOD1, FP	0.8VCCE		VCCE	V
		Ports P0, P1 (in only external extended/processor modes), $\overline{\text{WAIT}}$	0.43VCCE		VCCE	V
VIL	Low Level Input Voltage	Ports P0-P22, $\overline{\text{RESET}}$, MOD0, MOD1, FP	0		0.2VCCE	V
		Ports P0, P1 (in only external extended/processor modes), $\overline{\text{WAIT}}$	0		0.16VCCE	V
IOH(peak)	High Level Peak Output Current P0-P11, P14-P22 (Note 3)				-10	mA
IOH(avg)	High Level Mean Output Current P0-P11, P14-P22 (Note 4)				-5	mA
IOL(peak)	Low Level Peak Output Current P0-P11, P14-P22 (Note 3)				10	mA
IOL(avg)	Low Level Mean Output Current P0-P11, P14-P22 (Note 4)				5	mA
CL	Output Load Capacitance	JTCK,JTDI,JTMS, JTDO,JTRST			80	PF
		Other than above	15		50	PF
f(XIN)	External Clock Input Frequency		5		10	MHz

Note 1: Tested under conditions VCCE ≧ AVCC ≧ VREF
Note 2: Tested under conditions VDD ≧ VCCI ≧ FVCC ≧ OSC-VCC
Note 3: Make sure the total output current of ports (peak) meet the conditions below.
| Ports P0+P1+P2 | ≦ 80 mA
| Ports P3+P4+P15+P22 | ≦ 80 mA
| Ports P6+P7+P8+P9+P17 | ≦ 80 mA
| Ports P10+P11 | ≦ 80 mA
Note 4: The mean output current refers to the average current on ports within a 100-ms period.

Recommended Operating Conditions (Referenced to VCCE = 5 V±0.5 V, VCCI = 3.3 V±0.3 V, Ta = -40 to 125°C unless otherwise specified)

Symbol	Parameter		Rated Value			Unit
			MIN	TYP	MAX	
VCCE	External I/O Buffer Voltage (Note 1)		4.5	5.0	5.5	V
VCCI	Internal Logic Power Supply Voltage (Note 2)		3.0	3.3	3.6	V
VDD	RAM Power Supply Voltage (Note 2)		3.0	3.3	3.6	V
FVCC	Flash Power Supply Voltage (Note 2)		3.0	3.3	3.6	V
AVCC	Analog Power Supply Voltage (Note 1)		4.5	5.0	5.5	V
OSC-VCC	PLL Power Supply Voltage (Note 2)		3.0	3.3	3.6	V
VREF	Analog Reference Voltage (Note 1)			5.0	5.5	V
VIH	High Level Input Voltage	Ports P0-P22, $\overline{\text{RESET}}$, MOD0, MOD1, FP	0.8VCCE		VCCE	V
		Ports P0, P1 (in only external extended/processor modes), $\overline{\text{WAIT}}$	0.43VCCE		VCCE	V
VIL	Low Level Input Voltage	Ports P0-P22, $\overline{\text{RESET}}$, MOD0, MOD1, FP	0		0.2VCCE	V
		Ports P0, P1 (in only external extended/processor modes), $\overline{\text{WAIT}}$	0		0.16VCCE	V
IOH(peak)	High Level Peak Output Current P0-P11, P14-P22 (Note 3)				-10	mA
IOH(avg)	High Level Mean Output Current P0-P11, P14-P22 (Note 4)				-5	mA
IOL(peak)	Low Level Peak Output Current P0-P11, P14-P22 (Note 3)				10	mA
IOL(avg)	Low Level Mean Output Current P0-P11, P14-P22 (Note 4)				5	mA
CL	Output Load Capacitance	JTCK,JTDI,JTMS, JTDO,JTRST			80	PF
		Other than above	15		50	PF
f(XIN)	External Clock Input Frequency		5		8	MHz

Note 1: Tested under conditions VCCE ≧ AVCC ≧ VREF
Note 2: Tested under conditions VDD ≧ VCCI ≧ FVCC ≧ OSC-VCC
Note 3: Make sure the total output current of ports (peak) meet the conditions below.
| Ports P0+P1+P2 | ≦ 80 mA
| Ports P3+P4+P15+P22 | ≦ 80 mA
| Ports P6+P7+P8+P9+P17 | ≦ 80 mA
| Ports P10+P11 | ≦ 80 mA
Note 4: The mean output current refers to the average current on ports within a 100-ms period.

23.3 DC Characteristics

23.3.1 Electrical Characteristics

(1) Electrical Characteristics when f(XIN) = 10 MHz
(Referenced to VCCE = 5 V±0.5 V, VCCI = 3.3 V±0.3 V, Ta = -40 to 85°C unless otherwise specified)

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
VOH	High Level Output Voltage	IOH≧-2mA	VCCE + 0.165 × IOH(mA)		VCCE	V
VOL	Low Level Output Voltage	IOL≦2mA	0		0.15 × IOL(mA)	V
VDD	RAM Retention Power Supply Voltage	When operating	3.0		VCCI	V
		During backup	1.5		3.6	
IIH	High Level Input Current	VI=VCCE	-5		5	µA
IIL	Low Level Input Current	VI=0V	-5		5	µA
ICC-5V	5 V System Power Supply Current (Note 1)	f(XIN)=10.0MHz, When reset			1	mA
		f(XIN)=10.0MHz, When operating		1	12	
ICCI-3V	3.3 V System Power Supply Current (Note 2)	f(XIN)=10.0MHz, When reset			75	mA
		f(XIN)=10.0MHz, When operating		75	125	
IDDhold	RAM Retention Power Supply Current	Ta=25°C			100	µA
		Ta=85°C			2000	
V_{T+} —V_{T-}	Hysteresis (Note 3) RTDCLK, RTDRXD, SCLKI0,1,4,5 RXD0,1,2,3,4,5,6,7, TIN0—11, TINA0,1 TINB0,1, $\overline{RESET}$, FP, MOD0,1, JTMS, JTRST, JTDI	VCCE=5V	1.0			V
V_{T+} —V_{T-}	Hysteresis (Note 4) $\overline{SBI}$, $\overline{HREQ}$	VCCE=5V	0.3			V

Note 1: Total current of VCCE = AVCC = VREF in single-chip mode. See the next page for the rated value of power supply current on each power supply pin.

Note 2: Total current of VCCI = VDD = FVCC = OSC-VCC in single-chip mode. See the next page for the rated value of power supply current on each power supply pin.

Note 3: All these pins, except the $\overline{RESET}$ pin, are shared with other functions serving dual or triple purposes.

Note 4: The $\overline{HREQ}$ pin is shared with another function serving dual purposes.

(2) Electrical Characteristics of Each Power Supply Pin when f(XIN) = 10 MHz
(Referenced to VCCE = 5 V±0.5 V, VCCI = 3.3 V±0.3 V, Ta = -40 to 85°C unless otherwise specified)

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
ICCE	VCCE Power Supply Current when Operating	f(XIN)=10.0MHz			10	mA
ICCI	VCCI Power Supply Current when Operating	f(XIN)=10.0MHz			120	
IOSC-ICC	OSC-VCC Power Supply Current when Operating	f(XIN)=10.0MHz			20	mA
FICC	FVCC Power Supply Current when Operating (Note 1)	f(XIN)=10.0MHz			50	mA
IDD	VDD Power Supply Current when Operating (Note 2)	f(XIN)=10.0MHz			35	mA
IAVCC	AVCC Power Supply Current when Operating	f(XIN)=10.0MHz			3	mA
IREF	VREF Power Supply Current when Operating	f(XIN)=10.0MHz			3	mA

Note 1: This refers to the maximum value including programming and erase currents.
Note 2: This refers to the maximum value including currents needed when running a program in RAM.

(3) Backup RAM retention power supply current when f(XIN) = 10 MHz

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
IDDhold	RAM retention power supply current	Ta=85°C, VDD=1.5V			250	μA

(4) Electrical Characteristics when f(XIN) = 8 MHz
(Referenced to VCCE = 5 V±0.5 V, VCCI = 3.3 V±0.3 V, Ta = -40 to 125°C unless otherwise specified)

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
VOH	High Level Output Voltage	IOH≧-2mA	VCCE + 0.165 × IOH(mA)		VCCE	V
VOL	Low Level Output Voltage	IOL≦2mA	0		0.15 × IOL(mA)	V
VDD	RAM Retention Power Supply Voltage	When operating	3.0		VCCI	V
		During backup	1.5		3.6	
IIH	High Level Input Current	VI=VCCE	-5		5	µA
IIL	Low Level Input Current	VI=0V	-5		5	µA
ICC-5V	5 V System Power Supply Current (Note 1)	f(XIN)=8.0MHz, When reset			1	mA
		f(XIN)=8.0MHz, When operating		1	12	
ICCI-3V	3.3V System Power Supply Current (Note 2)	f(XIN)=8.0MHz, When reset			70	mA
		f(XIN)=8.0MHz, When operating		60	110	
IDDhold	RAM Retention Power Supply Current	Ta=25°C			100	µA
		Ta=125°C			7500	
V_{T+} — V_{T-}	Hysteresis (Note 3) RTDCLK, RTDRXD, SCLKI0,1,4,5 RXD0,1,2,3,4,5,6,7, TIN0—11, TINA0,1 TINB0,1, $\overline{\text{RESET}}$, FP, MOD0,1, JTMS, JTRST, JTDI	VCCE=5V	1.0			V
V_{T+} — V_{T-}	Hysteresis (Note 4) $\overline{\text{SBI}}$, $\overline{\text{HREQ}}$	VCCE=5V	0.3			V

Note 1: Total current of VCCE = AVCC = VREF in single-chip mode. See the next page for the rated value of power supply current on each power supply pin.
Note 2: Total current of VCCI = VDD = FVCC = OSC-VCC in single-chip mode. See the next page for the rated value of power supply current on each power supply pin.
Note 3: All these pins, except the $\overline{\text{RESET}}$ pin, are shared with other functions serving dual or triple purposes.
Note 4: The $\overline{\text{HREQ}}$ pin is shared with another function serving dual purposes.

(5) Electrical Characteristics of Each Power Supply Pin when f(XIN) = 8 MHz
(Referenced to VCCE = 5 V±0.5 V, VCCI = 3.3 V±0.3 V, Ta = -40 to 125°C unless otherwise specified)

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
ICCE	VCCE Power Supply Current when Operating	f(XIN)=8.0MH z			10	mA
ICCI	VCCI Power Supply Current when Operating	f(XIN)=8.0MH z			105	
OSC-ICC	OSC-VCC Power Supply Current when Operating	f(XIN)=8.0MH z			16	mA
FICC	FVCC Power Supply Current when Operating (Note 1)	f(XIN)=8.0MH z			50	mA
IDD	VDD Power Supply Current when Operating (Note 2)	f(XIN)=8.0MH z			30	mA
AICC	AVCC Power Supply Current when Operating	f(XIN)=8.0MH z			3	mA
IREF	VREF Power Supply Current when Operating	f(XIN)=8.0MH z			3	mA

Note 1: This refers to the maximum value including programming and erase currents.
Note 2: This refers to the maximum value including currents needed when running a program in RAM.

(6) Backup RAM retention power supply current when f(XIN) = 8 MHz

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
IDDhold	RAM retention power supply current	Ta=125ºC, VDD=1.5V			500	μA
		Ta=105ºC, DD=1.5V			300	μA

23.3.2 Flash Related Electrical Characteristics

Flash Related Electrical Characteristics (Referenced to VCCE = 5 V±0.5 V, VCCI = 3.3 V±0.3 V unless otherwise specified)

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
Ifvcc1	FVCC Power Supply Current (when Programming)				50	mA
Ifvcc2	FVCC Power Supply Current (when Erasing)				40	mA
Topr	Flash Rewrite Ambient Temperature		0		70	°C
cycle	Rewrite Durability				100	

23.4 A-D Conversion Characteristics

A-D Conversion Characteristics (Referenced to AVCC = VREF = VCCE = 5.12 V, Ta = -40 to 85°C, f(XIN) = 10.0 MHz unless otherwise specified)

Symbol	Parameter			Test Condition	Rated Value			Unit
					MIN	TYP	MAX	
—	Resolution			VREF=VCC			10	Bits
—	Absolute Accuracy (Note 1)	Low speed mode					±2	LSB
		High speed mode					±3	LSB
TCONV	Conversion Time	Low speed mode	Normal		14950			ns
			Double speed		8650			
		High speed mode	x2 speed		6550			
			x4 speed		4450			
IIAN	Analog Input Leakage Current			(Note 2)	-200		200	nA

Note 1: The absolute accuracy indicates the accuracy of output codes produced by the A-D converter with respect to analog inputs including all error sources (e.g., quantization and other errors), and is calculated by the equation below.

Absolute accuracy = output code - (analog input voltage Ni / 1 LSB)

where 1 LSB = 5 mV when AVCC = VREF = 5.12 V

Note 2: This refers to the input leakage current on AN0-AN15 when the A-D converter is inactive. Input voltage conditions are 0≦ANi≦AVCC. Temperature conditions are Ta =-40 to 85°C.

A-D Conversion Characteristics (Referenced to AVCC = VREF = VCCE = 5.12 V, Ta = -40 to 125°C, f(XIN) = 8.0 MHz unless otherwise specified)

Symbol	Parameter			Test Condition	Rated Value			Unit
					MIN	TYP	MAX	
—	Resolution			VREF=VCC			10	Bits
—	Absolute Accuracy (Note 1)	Low speed mode					±2	LSB
		High speed mode					±3	LSB
TCONV	Conversion Time	Low speed mode	Normal		18687.5			ns
			Double speed		10812.5			
		High speed mode	x2 speed		8187.5			
			x4 speed		5562.5			
IIAN	Analog Input Leakage Current			(Note 2)	-200		200	nA

Note 1: The absolute accuracy indicates the accuracy of output codes produced by the A-D converter with respect to analog inputs including all error sources (e.g., quantization and other errors), and is calculated by the equation below.

Absolute accuracy = output code - (analog input voltage Ni / 1 LSB)

where 1 LSB = 5 mV when AVCC = VREF = 5.12 V

Note 2: This refers to the input leakage current on AN0-AN15 when the A-D converter is inactive. Input voltage conditions are 0≦ANi≦AVCC. Temperature conditions are Ta =-40 to 85°C.

23.5 D-A Conversion Characteristics

23.5.1 D-A Conversion Characteristics

D-A Conversion Characteristics (Referenced to AVCC = VREF = VCCF = 5.12 V, Ta = 25°C, f(XIN) = 10/8 MHz unless otherwise specified)

Symbol	Parameter	Test Condition	Rated Value			Unit
			MIN	TYP	MAX	
—	Resolution				8	Bits
—	Absolute Accuracy				1	%
tsu	Setup Time				3	μS
Ro	Output Resistance		4	10	20	KΩ
IVREF	Reference Power Supply Input Current	(Note)			1.5	mA

Note: This applies to the case where the device is using one D-A converter and the D-A register value for the u nused D-A converter is H'00.
The A-D converter's ladder resistance is not included.

23.6 AC Characteristics

23.6.1 Timing Requirements

- Unless otherwise noted, timing conditions are VCCE = 5 V ± 0.5 V, VCCI = 3.3 V ± 0.3 V, Ta = -40 to 125°C
- The characteristic values apply to the case of concentrated capacitance with an output load capacitance of 15 to 50 pF (however, 80 pF for JTAG-related). In cases where the output load capacitance varies, they may deviate from the rated switching characteristics.

(1) Input/output ports

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.1
			MIN	MAX		
t_{su}(P-E)	Port Input Setup Time		100		ns	①
t_{h}(E-P)	Port Input Hold Time		0		ns	②

(2) Serial I/O

a) CSIO mode, with internal clock selected

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.2
			MIN	MAX		
$t_{su(D-CLK)}$	RxD Input Setup Time		150		ns	④
$t_{h(CLK-D)}$	RxD Input Hold Time		50		ns	⑤

b) CSIO mode, with external clock selected

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.2
			MIN	MAX		
$t_{c(CLK)}$	CLK Input Cycle Time		640		ns	⑦
$t_{w(CLKH)}$	CLK Input High Pulse Width		300		ns	⑧
$t_{w(CLKL)}$	CLK Input Low Pulse Width		300		ns	⑨
$t_{su(D-CLK)}$	RxD Input Setup Tim		60		ns	⑩
$t_{h(CLK-D)}$	RxD Input Hold Time		100		ns	⑪

(3) SBI

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.3
			MIN	MAX		
$t_{w(SBIL)}$	SBI Input Pulse Width		$\frac{5}{2}$ tc(BCLK)		ns	⑬

(4) TINi (i=0, 3, 16-23)

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.5
			MIN	MAX		
$t_{w(TINi)}$	TINi Input Pulse Width		$\frac{7}{2}$ tc(BCLK)		ns	(14)

(5) TIN8-11 (When multiply-by-4 event count or up/down event count mode)

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.6
			MIN	MAX		
tw(TIN8-11)	TIN8-11 Input Pulse Width (When multiply-by-4 event count or up/down event count mode)		$\frac{7}{2}$ tc(BCLK)		ns	(91)
tw(TIN8-11)	TIN8-11 Input Phase (When multiply-by-4 event count or up/down event count mode)		$\frac{7}{2}$ tc(BCLK)		ns	(92)

(6) Read and write timing

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.7 23.6.8 23.6.9
			MIN	MAX		
$t_{su(D-BCLKH)}$	Data Input Setup Time before BCLK		26		ns	(31)
$t_{h(BCLKH-D)}$	Data Input Hold Time after BCLK		0		ns	(32)
$t_{su(WAITL-BCLKH)}$	$\overline{WAIT}$ Input Setup Time before BCLK		26		ns	(33)
$t_{h(BCLKH-WAITL)}$	$\overline{WAIT}$ Input Hold Time after BCLK		0		ns	(34)
$t_{su(WAITH-BCLKH)}$	$\overline{WAIT}$ Input Setup Time before BCLK		26		ns	(78)
$t_{h(BCLKH-WAITH)}$	$\overline{WAIT}$ Input Hold Time after BCLK		0		ns	(79)
$t_{w(RDL)}$	Read Low Pulse Width		$\frac{3}{2}$ tc(BCLK)-23		ns	(43)
$t_{su(D-RDH)}$	Data Input Setup Time before Read		30		ns	(44)
$t_{h(RDH-D)}$	Data Input Hold Time after Read		0		ns	(45)
$t_{w(BLWL)}$ $t_{w(BHWL)}$	Write Low Pulse Width (Byte write mode)		tc(BCLK) -25		ns	(51)
$t_{d(RDH-BLWL)}$ $t_{d(RDH-BHWL)}$	Write Delay Time after Read		$\frac{tc(BCLK)}{2}$ -10		ns	(56)
$t_{d(BLWH-RDL)}$ $t_{d(BHWH-RDL)}$	Read Delay Time after Write		$\frac{tc(BCLK)}{2}$ -10		ns	(57)
$t_{w(WRL)}$	Write Low Pulse Width (Byte enable mode)		tc(BCLK) -25		ns	(68)
$t_{d(RDH-BLEL)}$ $t_{d(RDH-BHEL)}$	Write Delay Time after Read (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -10		ns	(80)
$t_{d(BLEH-RDL)}$ $t_{d(BHEH-RDL)}$	Read Delay Time after Write (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -10		ns	(81)

(7) Bus arbitration timing

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.10
			MIN	MAX		
$t_{su(HREQL-BCLKH)}$	$\overline{HREQ}$ Input Setup Time before BCLK		27		ns	(35)
$t_{h(BCLKH-HREQL)}$	$\overline{HREQ}$ Input Hold Time after BCLK		0		ns	(36)

(8) Input transition time on JTAG pin

Symbol	Condition			Rated Value		Unit	See Figure 23.6.11
				MIN	MAX		
tr	Input Rising Transition Time	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	Input Falling Transition Time	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: Stipulated values are guaranteed values when the test pin load capacitance CL=80pF.

(9) JTAG interface timing

Symbol	Condition	Rated Value		Unit	See Figure 23.6.12
		MIN	MAX		
$t_{c(JTCK)}$	JTCK Input Cycle Time	100		ns	(60)
$t_{w(JTCKH)}$	JTCK Input High Pulse Width	40		ns	(61)
$t_{w(JTCKL)}$	JTCK Input Low Pulse Width	40		ns	(62)
$t_{su(JTDI-JTCK)}$	JTDI, JTMS Input Setup Time	15		ns	(63)
$t_{h(JTCK-JTDI)}$	JTDI, JTMS Input Hold Time	20		ns	(64)
$t_{d(JTCK-JTDOV)}$	JTDO Output Delay Time after JTCK Fall		40	ns	(65)
$t_{d(JTCK-JTDOX)}$	JTDO Output Hi-Z Delay Time after JTCK Fall		40	ns	(66)
$t_{W(JTRST)}$	TRST Input Low Pulse Width	tc(JTCK)		ns	(67)

Note: Stipulated values are guaranteed values when the test pin load capacitance CL=80pF.

(10) RTD Timing

Symbol	Parameter	Rated Value		Unit	See Figure 23.6.13
		MIN	MAX		
tc(RTDCLK)	RTDCLK Input Cycle Time	500		ns	(90)
tw(RTDCLKH)	RTDCLK Input High Pulse Width	230		ns	(83)
tw(RTDCLKL)	RTDCLK Input Low Pulse Width	230		ns	(84)
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)
tv(RTDRXD-RTDCLKL)	RTDRXD Input Setup Time	60		ns	(89)
th(RTDCLKH-RTDRXD)	RTDRXD Input Hold Time	100		ns	(88)

23.6.2 Switching Characteristics

(1) Input/output ports

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.1
			MIN	MAX		
$t_{d(E-P)}$	Port Data Output Delay Time			100	ns	③

(2) Serial I/O

a) CSIO mode, with internal clock selected

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.2
			MIN	MAX		
$t_{d(CLK-D)}$	TxD Output Delay Time			60	ns	⑥
$t_{h(CLK-D)}$	TxD Hold Time		0		ns	⑧②

b) CSIO mode, with external clock selected

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.2
			MIN	MAX		
$t_{d(CLK-D)}$	TxD Output Delay Time			160	ns	⑫

(3) TOi (i=0-20)

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.4
			MIN	MAX		
$t_{d(BCLK-TOi)}$	TOi Output Delay Time			100	ns	⑮

(4) Read and write timing

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.7 23.6.8 23.6.9
			MIN	MAX		
$t_{c(BCLK)}$	BCLK Output Cycle Time			$\frac{tc(Xin)}{2}$	ns	16
$t_{w(BCLKH)}$	BCLK Output High Pulse Width		$\frac{tc(BCLK)}{2}$ - 5		ns	17
$t_{w(BCLKL)}$	BCLK Output Low Pulse Width		$\frac{tc(BCLK)}{2}$ - 5		ns	18
$t_{d(BCLKH-A)}$	Address Delay Time after BCLK			24	ns	19
$t_{d(BCLKH-CS)}$	Chip Select Delay Time after BCLK			24	ns	20
$t_{v(BCLKH-A)}$	Valid Address Time after BCLK		-11		ns	21
$t_{v(BCLKH-CS)}$	Valid Chip Select Time after BCLK		-11		ns	22
$t_{d(BCLKL-RDL)}$	Read Delay Time after BCLK			10	ns	23
$t_{v(BCLKH-RDL)}$	Valid Read Time after BCLK		-12		ns	24
$t_{d(BCLKL-BLWL)}$ $t_{d(BCLKL-BHWL)}$	Write Delay Time after BCLK			11	ns	25
$t_{v(BCLKL-BLWL)}$ $t_{v(BCLKL-BHWL)}$	Valid Write Time after BCLK		-12		ns	26
$t_{v(BCLKL-D)}$	Data Output Delay Time after BCLK			18	ns	27
$t_{v(BCLKH-D)}$	Valid Data Output Time after BCLK		-16		ns	28
$t_{pzx(BCLKL-DZ)}$	Data Output Enable Time after BCLK		-19		ns	29
$t_{pxz(BCLKH-DZ)}$	Data Output Disable Time after BCLK			5	ns	30
$t_{d(A-RDL)}$	Address Delay Time before Read		$\frac{tc(BCLK)}{2}$ -15		ns	39
$t_{d(CS-RDL)}$	Chip Select Delay Time before Read		$\frac{tc(BCLK)}{2}$ -15		ns	40
$t_{v(RDH-A)}$	Valid Address Time after Read		0		ns	41
$t_{v(RDH-CS)}$	Valid Chip Select Time after Read		0		ns	42
$t_{pzx(RDH-DZ)}$	Data Output Enable Time after Read		$\frac{tc(BCLK)}{2}$		ns	46
$t_{d(A-BLWL)}$ $t_{d(A-BHWL)}$	Address Delay Time before Write (Byte write mode)		$\frac{tc(BCLK)}{2}$ -15		ns	47
$t_{d(CS-BLWL)}$ $t_{d(CS-BHWL)}$	Chip Select Delay Time before Write (Byte write mode)		$\frac{tc(BCLK)}{2}$ -15		ns	48
$t_{v(BLWH-A)}$ $t_{v(BHWH-A)}$	Valid Address Time after Write (Byte write mode)		$\frac{tc(BCLK)}{2}$ -15		ns	49
$t_{v(BLWH-CS)}$ $t_{v(BHWH-CS)}$	Valid Chip Select Time after Write (Byte write mode)		$\frac{tc(BCLK)}{2}$ -15		ns	50

Read and write timing (continued from the preceding page)

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.7 23.6.8 23.6.9
			MIN	MAX		
$t_{d(BLWL-D)}$ $t_{d(BHWL-D)}$	Data Output Delay Time after Write (Byte write mode)			15	ns	52
$t_{v(BLWH-D)}$ $t_{v(BHWH-D)}$	Valid Data Output Time after Write (Byte write mode)		$\frac{tc(BCLK)}{2}$ -13		ns	53
$t_{pxz(BLWH-DZ)}$ $t_{pxz(BHWH-DZ)}$	Data Output Disable Time after Write (Byte write mode)			$\frac{tc(BCLK)}{2}$ +5	ns	54
$t_{d(A-WRL)}$	Address Delay Time before Write (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -15		ns	69
$t_{d(CS-WRL)}$	Chip Select Delay Time before Write (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -15		ns	70
$t_{v(WRH-A)}$	Valid Address Time after Write (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -15		ns	71
$t_{v(WRH-CS)}$	Valid Chip Select Time after Write (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -15		ns	72
$t_{d(BLE-WRL)}$ $t_{d(BHE-WRL)}$	Byte enable delay time before write (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -15		ns	73
$t_{v(WRH-BLE)}$ $t_{v(WRH-BHE)}$	Byte enable delay time after write (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -15		ns	74
$t_{d(WRL-D)}$	Data Output Delay Time after Write (Byte enable mode)			15	ns	75
$t_{v(WRH-D)}$	Valid Data Output Time after Write (Byte enable mode)		$\frac{tc(BCLK)}{2}$ -13		ns	76
$t_{pxz(WRH-DZ)}$	Data output disable time after write (Byte enable mode)			$\frac{tc(BCLK)}{2}$ +5	ns	77
$t_{w(RDH)}$	Read high-level pulse width		$\frac{tc(BCLK)}{2}$ -3		ns	55

(5) Bus arbitration

Symbol	Parameter	Condition	Rated Value		Unit	See Figure 23.6.10
			MIN	MAX		
$t_{d(BCLKL-HACKL)}$	$\overline{HACK}$ Delay Time after BCLK			29	ns	37
$t_{v(BCLKL-HACKL)}$	Valid $\overline{HACK}$ Time after BCLK		-11		ns	38

23.6.3 AC Characteristics

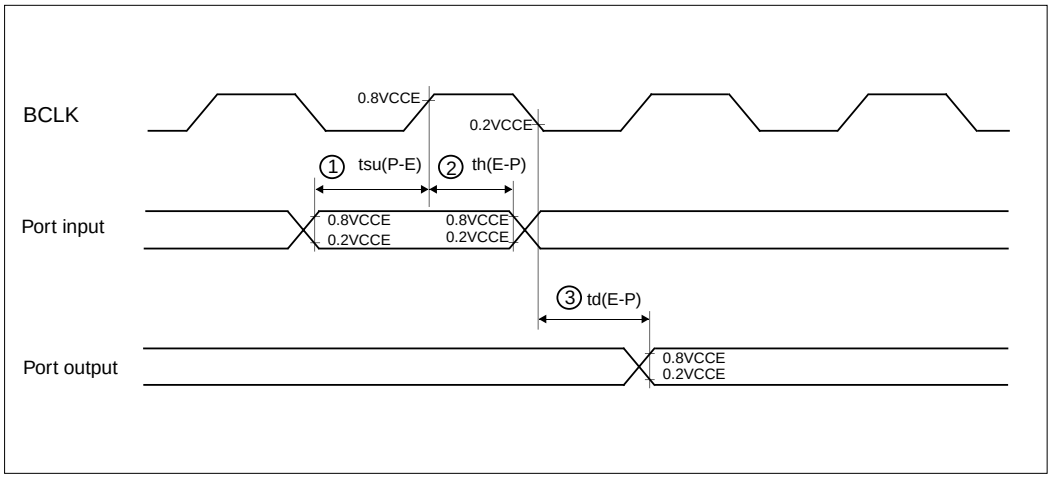


Figure 23.6.1 Input/Output Port Timing

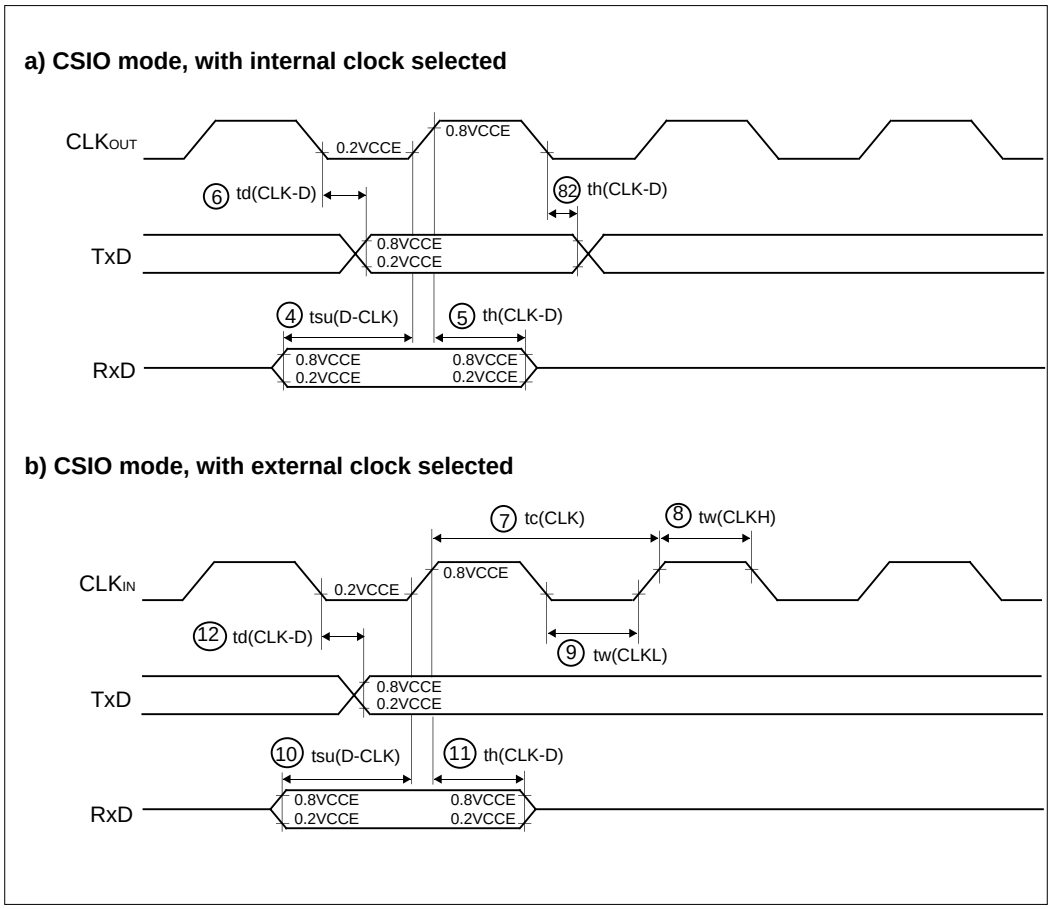


Figure 23.6.2 Serial I/O Timing

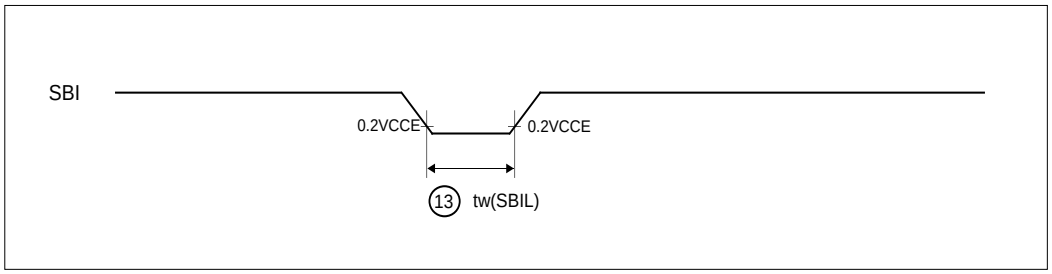


Figure 23.6.3 SBI Timing

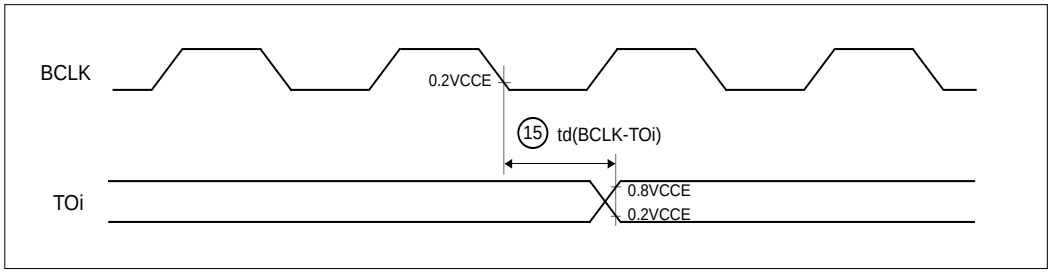


Figure 23.6.4 TOi Timing

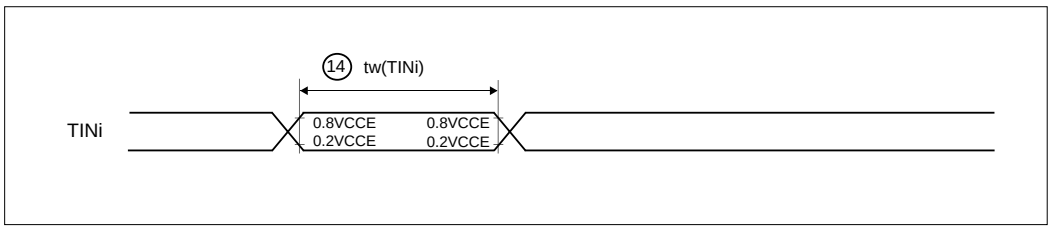


Figure 23.6.5 TINi Timing

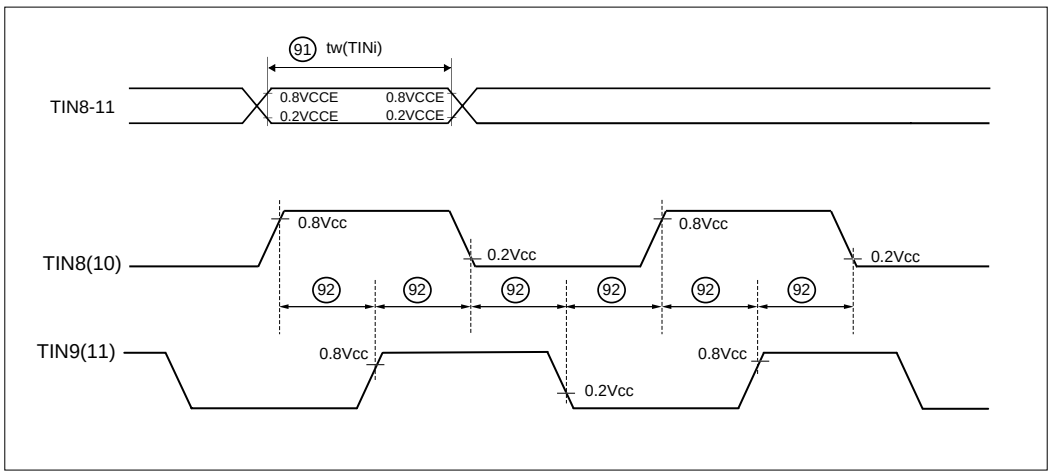
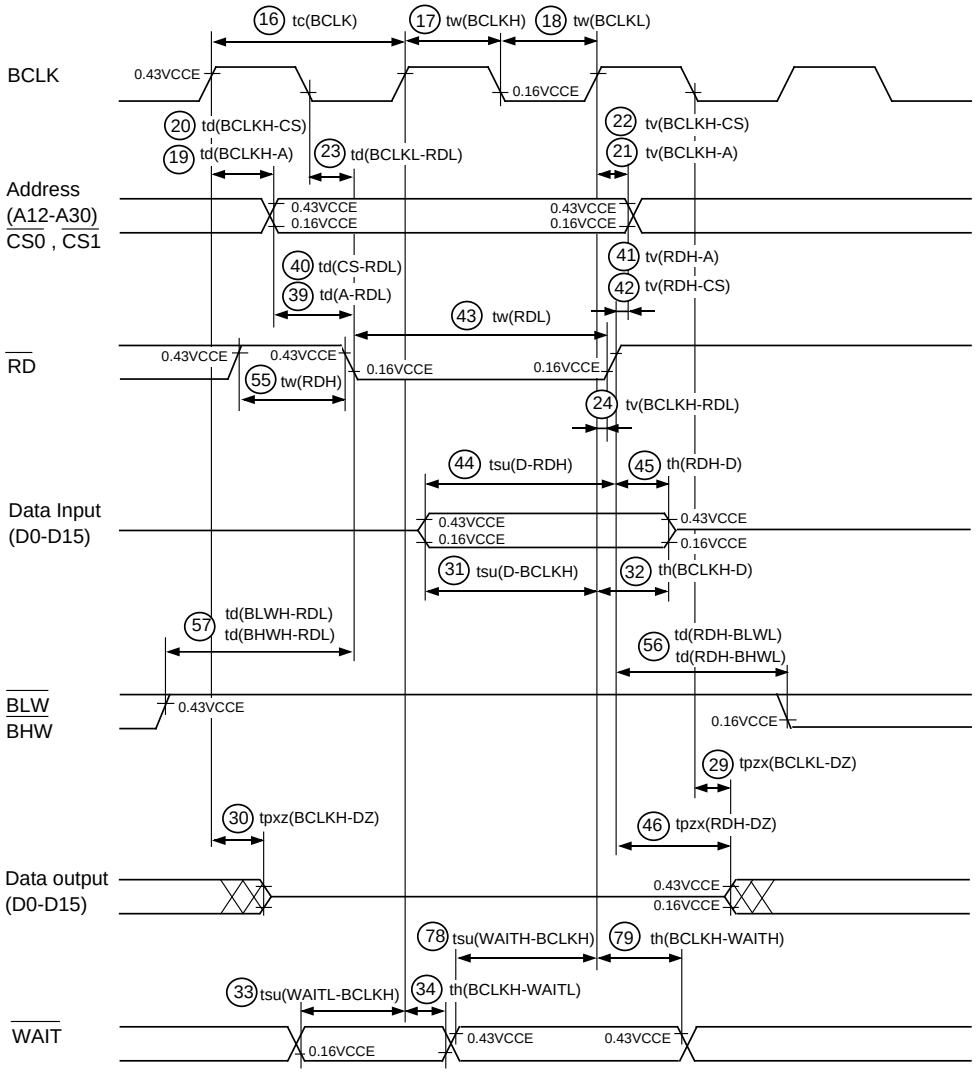


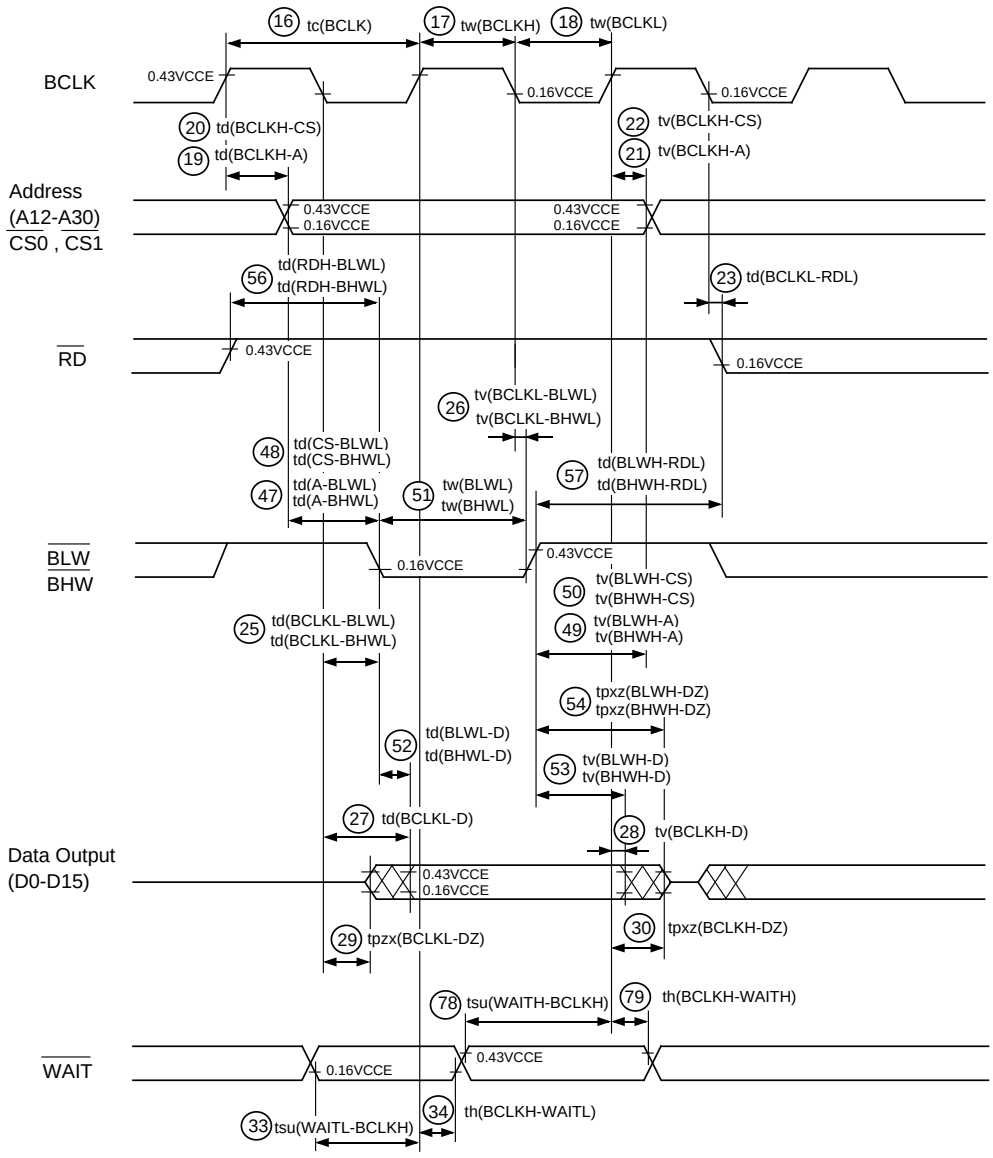
Figure 23.6.6 TIN8-11 Input Timing (When multiply-by-4 event count or up/down event count mode)



Note 1: Stipulated values are guaranteed values when the test pin load capacitance CL = 15 to 50 pF.

Note 2: Input and output signals are determined high or low with respect to TTL level.

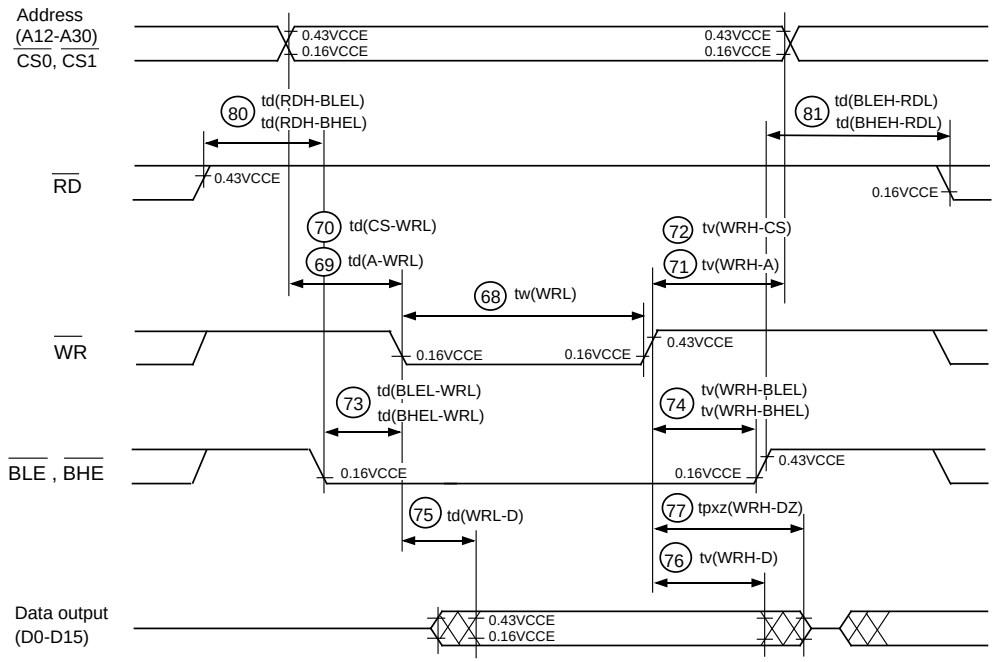
Figure 23.6.7 Read Timing



Note 1: Stipulated values are guaranteed values when the test pin load capacitance CL = 15 to 50 pF.

Note 2: Input and output signals are determined high or low with respect to TTL level.

Figure 23.6.8 Write Timing



Note 1: Stipulated values are guaranteed values when the test pin load capacitance CL = 15 to 50 pF.

Note 2: Input and output signals are determined high or low with respect to TTL level.

Figure 23.6.9 Write Timing (Byte enable mode)

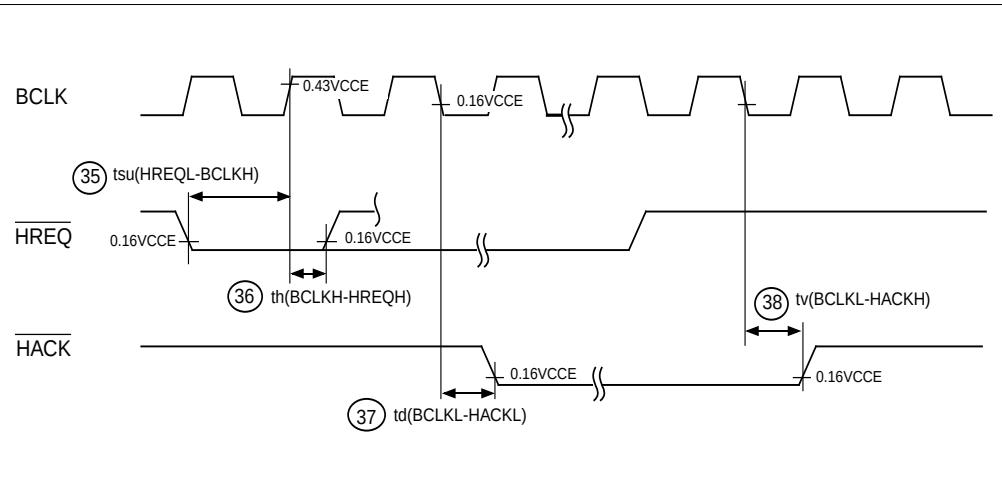
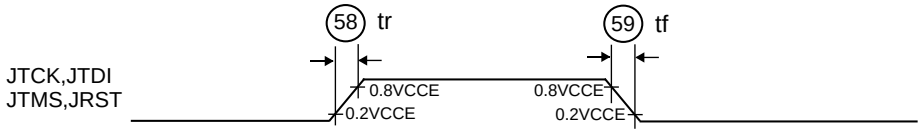
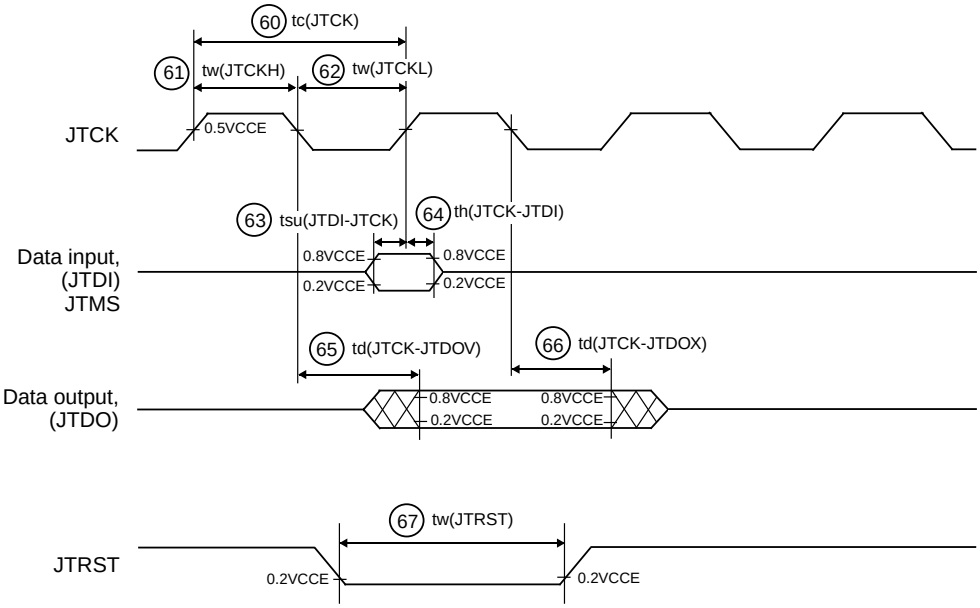


Figure 23.6.10 Bus Arbitration Timing



Note: Stipulated values are guaranteed values when the test pin load capacitance CL = 80 pF.

Figure 23.6.11 Input Transition Time on JTAG pins



Note: Stipulated values are guaranteed values when the test pin load capacitance CL = 80 pF.

Figure 23.6.12 JTAG Interface Timing

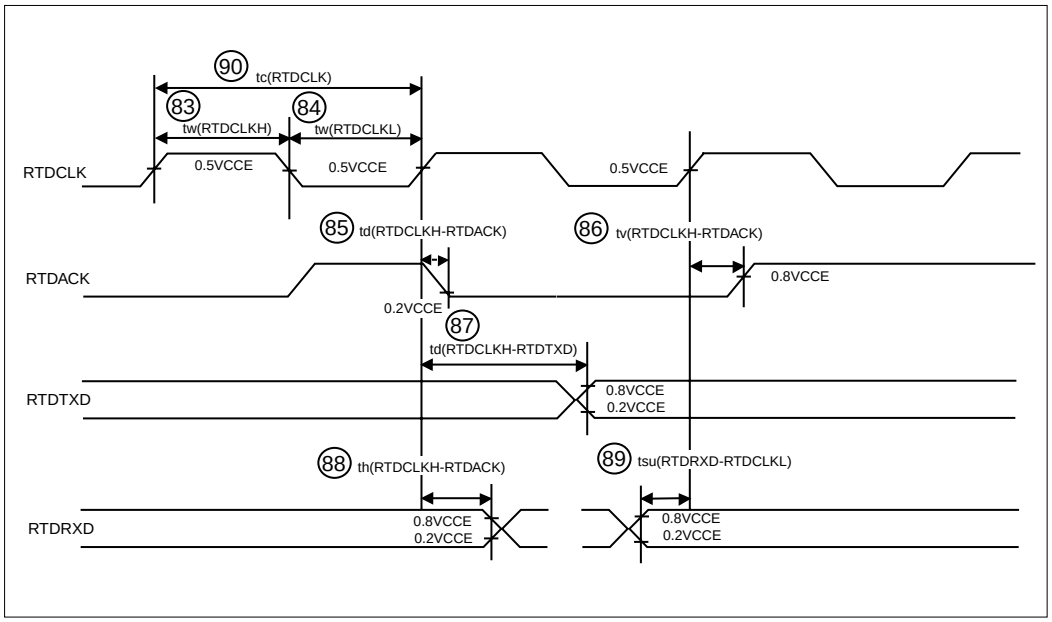


Figure 23.6.13 RTD Timing

*** This is a blank page ***

CHAPTER 24
STANDARD CHARACTERISTICS

24.1 A-D Conversion Characteristics

To be written at a later time

APPENDIX 1
MECHANICAL SPECIFICATIONS

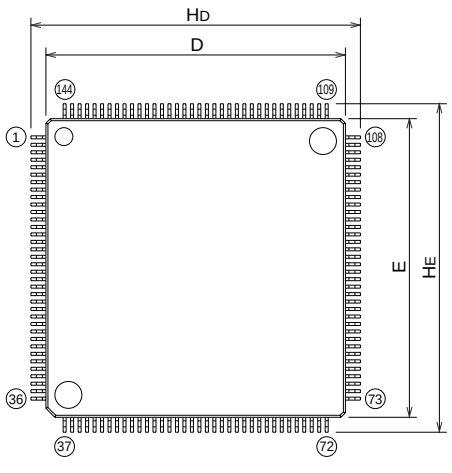
Appendix 1.1 Dimensional Outline Drawing

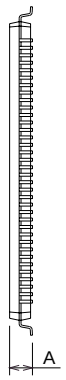
Appendix 1.1 Dimensional Outline Drawing

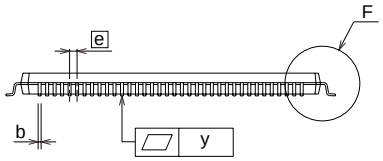
(1) 144 pin LQFP

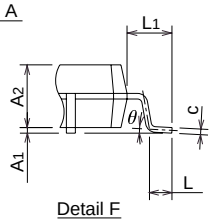
144P6Q-A **Plastic 144pin 20×20mm body LQFP**

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP144-P-2020-0.50	—		Cu Alloy

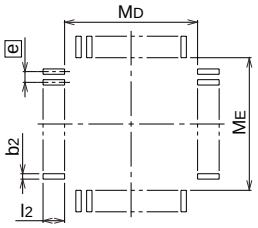








Detail F



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	1.7
A_1	0.05	0.125	0.2
A_2	—	1.4	—
b	0.17	0.22	0.27
c	0.105	0.125	0.175
D	19.9	20.0	20.1
E	19.9	20.0	20.1
e	—	0.5	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
L	0.35	0.5	0.65
L_1	—	1.0	—
y	—	—	0.1
θ	0_i	—	8_i
b_2	—	0.225	—
l_2	1.0	—	—
M_D	—	20.4	—
M_E	—	20.4	—

APPENDIX 2
INSTRUCTION PROCESSING TIME

Appendix 2.1 M32R/E Instruction Processing Time

Appendix 2.1 M32R/E Instruction Processing Time

For the M32R, the number of instruction execution cycles in E stage normally represents its 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 IF (instruction fetch) and D (decode) stages, not just E (execution) stage, must also be taken into account.

The table below shows the instruction processing time in each pipelined stage of the M32R.

Table 2.1.1 Instruction Processing Time of Each Pipeline Stage

	Number of execution cycles in each stage (Note)				
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 instruction (MUL)	R	1	3	–	1
Divide/remainder instructions (DIV, DIVU,REM,REMU)	R	1	37	–	1
Other instructions (including those for DSP function)	R	1	1	–	1

Note: For R and W, refer to the calculation methods described in the next page.

The following shows the number of memory access cycles in 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 required for actual 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 requested a memory access, the instruction processing may take more time than the calculated value.

■ **R (read cycle)** Cycles

When existing in instruction queue	1
When reading internal resource (ROM, RAM)	1
When reading internal resource (SFR)(byte, halfword)	2
When reading internal resource (SFR)(word)	4
When reading external memory (byte, halfword)	5 (Note)
When reading external memory (word)	9 (Note)
When successively fetching instructions from external memory	8 (Note)

■ **W (write cycle)** Cycles

When writing to internal resource (RAM)	1
When writing to internal resource (SFR)(byte, halfword)	2
When writing to internal resource (SFR)(word)	4
When writing to external memory (byte, halfword)	4 (Note)
When writing to external memory (word)	8 (Note)

Note: This applies for external access with one wait cycle. (When the 32171 accesses external circuits, it requires at least one wait cycle inserted.)

❊ This is a blank page. ❊

APPENDIX 3

PRECAUTIONS ABOUT NOISE

Appendix 3.1 Precautions about 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 with these measures incorporated be fully evaluated before it can actually be put to use.

Appendix 3.1.1 Reduction of Wiring Length

Wiring on the board may serve as an antenna to draws noise into the microcomputer. Shorter the total wiring length, the smaller the possibility of drawing noise into the microcomputer.

(1) Wiring of the $\overline{\text{RESET}}$ pin

Reduce the length of wiring connecting to the $\overline{\text{RESET}}$ pin. Especially when connecting a capacitor between the $\overline{\text{RESET}}$ and VSS pins, make sure it is connected to each pin with the shortest possible wiring (within 20 mm).

<Reasons>

Reset is a function to initialize the internal logic of the microcomputer. The pulse width applied to the $\overline{\text{RESET}}$ pin is important and is therefore stipulated as part of timing requirements. If noise in pulse width shorter than the stipulated duration is applied to the $\overline{\text{RESET}}$ pin, reset will be negated before the internal logic of the microcomputer is fully initialized, causing the program to go wild.

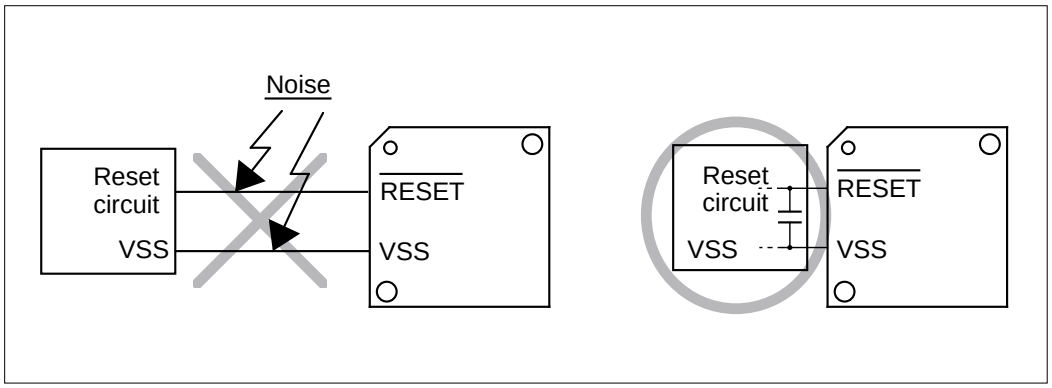


Figure 3.1.1 Wiring of the $\overline{\text{RESET}}$ Pin

(2) Wiring of clock input/output pins

Reduce the length of wiring connecting to the clock input/output pins. When connecting a capacitor to the oscillator, make sure its ground lead wire and the VSS pin on the microcomputer are connected with the shortest possible wiring (within 20 mm). Also, make sure the VSS pattern for clock oscillation is used for only the oscillator circuit and is separated from other VSS patterns.

<Reasons>

The microcomputer operates synchronously with the clock generated by the oscillator circuit. Inclusion of noise on clock input/output pins causes the clock waveform to become distorted, which may result in the microcomputer operating erratically or getting out of control. Also, if a noise-induced potential difference exists between the microcomputer's VSS level and the oscillator's VSS level, the clock fed into the microcomputer may not be an exact clock.

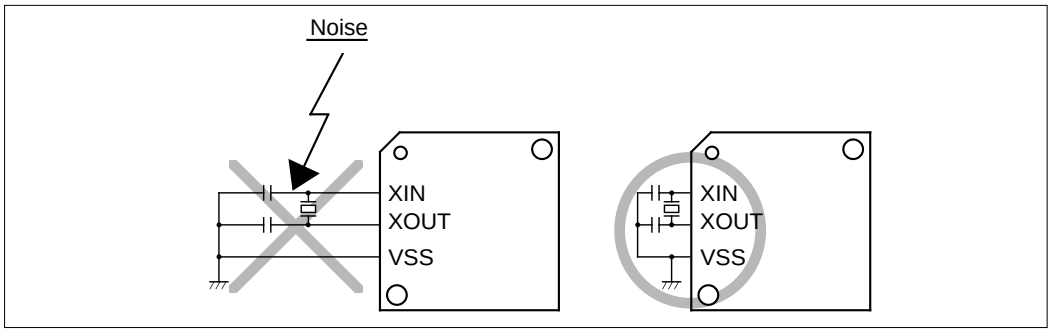


Figure 3.1.2 Wiring of Clock Input/Output Pins

(3) Wiring of operation mode setup pins

When connecting operation mode setup pins and the VCC or VSS pin, make sure they are connected with the shortest possible wiring.

<Reasons>

The levels of operation mode setup pins affect the microcomputer's operation mode. When connecting operation mode setup pins and the VCC or VSS pin, be careful that no noise-induced potential difference will exist between 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.

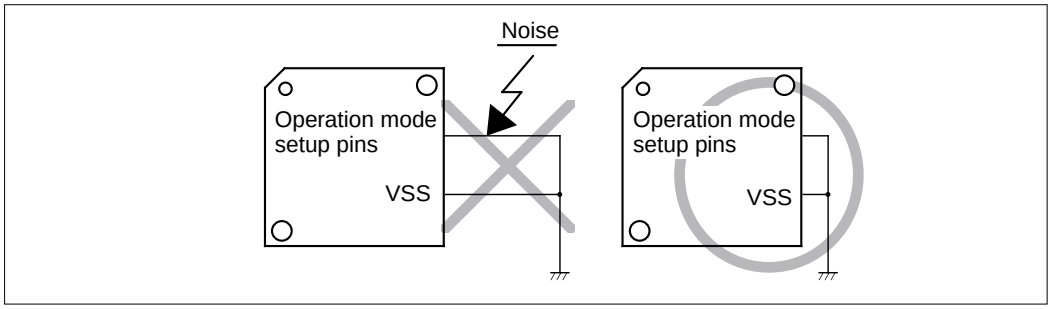


Figure 3.1.3 Example for Wiring of MOD0 and MOD1 Pins

Appendix 3.1.2 Inserting a Bypass Capacitor between VSS and VCC Lines

Insert a bypass capacitor of about 0.1 mF between VSS and VCC lines in such a way as to meet the requirements described below.

- The wiring length between VSS pin and bypass capacitor and that between VCC pin and bypass capacitor are equal.
- The wiring length between VSS pin and bypass capacitor and that between VCC pin and bypass capacitor are the shortest possible.
- The VSS and VCC lines are comprised of wiring in greater width than that of other signal lines.

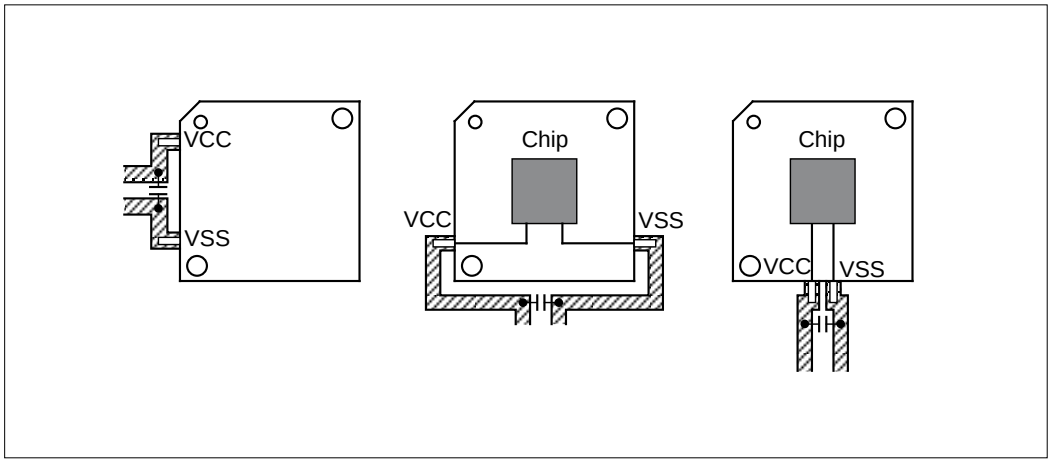


Figure 3.1.4 Bypass Capacitor between VSS and VCC Lines

Appendix 3.1.3 Processing Analog Input Pin Wiring

Connect a resistor of about 100 to 500 W (in series to the analog signal wire 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 inevitably is 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 ridding on the ground line may penetrate into the microcomputer via the capacitor.

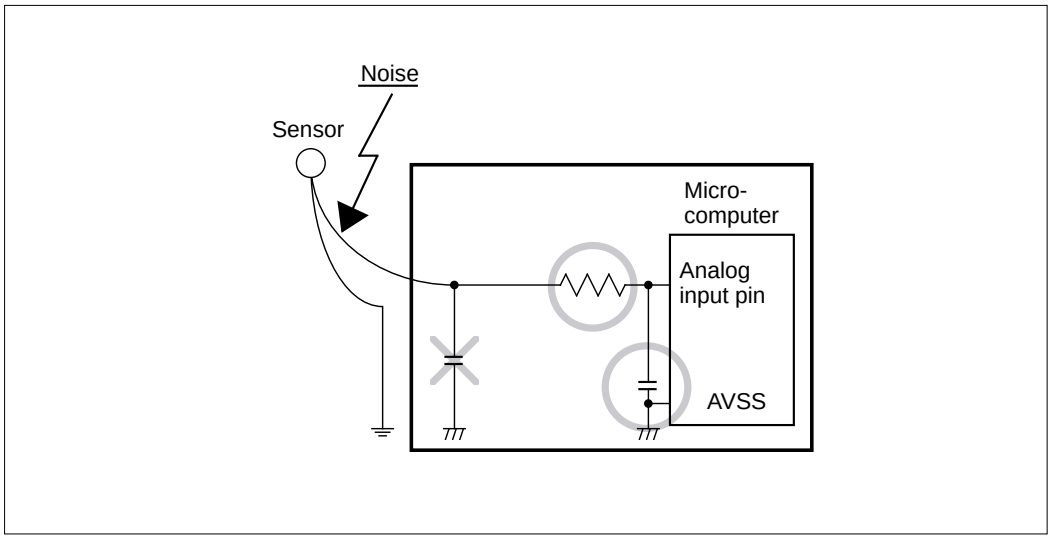


Figure 3.1.5 Resistor and Capacitor for Analog Signal Line

Appendix 3.1.4 Consideration about the Oscillator

The oscillator that generates the fundamental clock for microcomputer operation requires consideration to make it less susceptible to influences from other signals.

(1) Avoidance from large-current signal lines

Signal lines in which a large current flows 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.

<Reasons>

Systems using the microcomputer contain signal lines to control, for example, a motor, LED, and thermal head. When a large current flows in these signal lines, it generates noise due to mutual inductance.

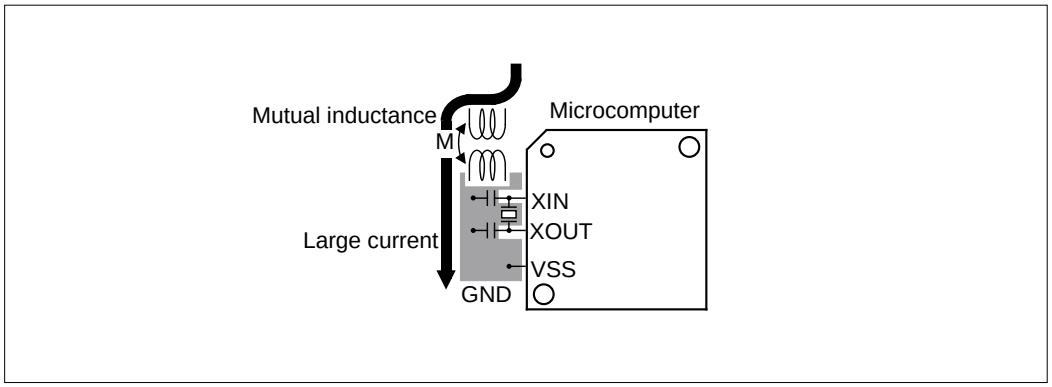


Figure 3.1.6 Wiring of Large-current Signal Lines

(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 rapidly level-changing signal lines will not intersect 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 they intersect 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.

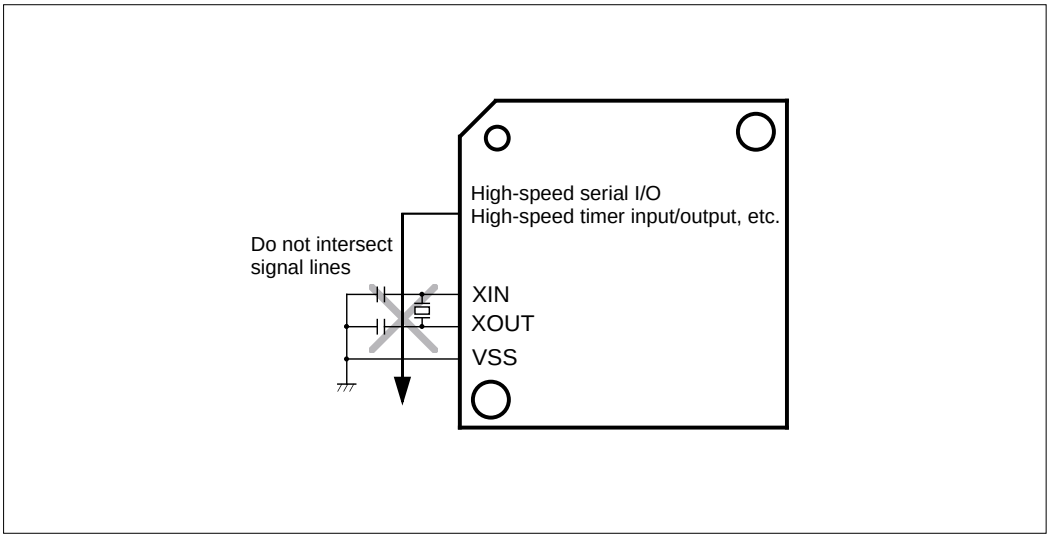


Figure 3.1.7 Wiring of Rapidly Level-changing Signal Lines (i = 0 to 3)

Appendix 3.1.5 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 100 W (or more) in series to input/output ports.

Software measures

- For input ports, read out data in a program two or more times to verify that levels match.
- 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.

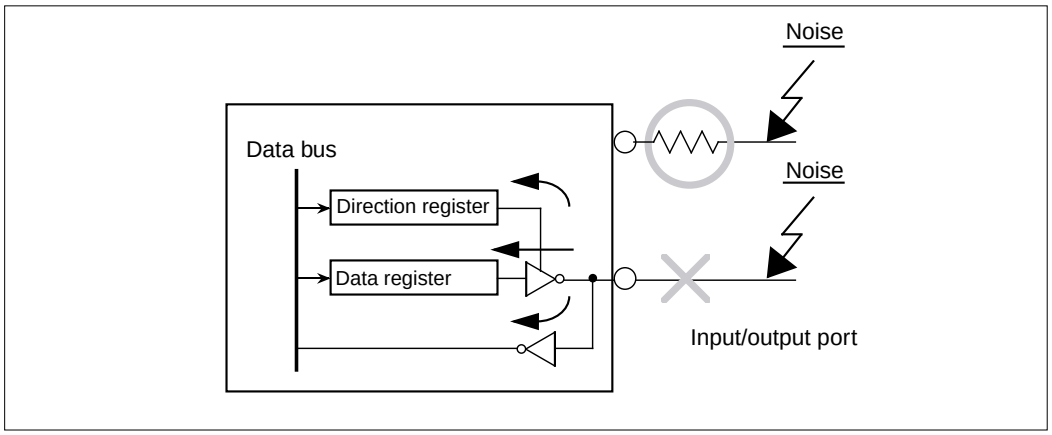


Figure 3.1.8 Processing Input/Output Ports

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
32172/32173 Group

Oct. First Edition 2001

Editioned by
Committee of editing of Mitsubishi Semiconductor User's Manual

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

32172 32173 Group
User's Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan