User's Manual

32

R32C/116A Group

User's Manual: Hardware

RENESAS MCU

M16C Family / R32C/100 Series

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

Renesas Electronics
www.renesas.com

Rev.1.10 Sep 2012

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

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

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

(2012.4)

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

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

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

About This Manual

1. Purpose and Target User

This manual is designed to be read primarily by application developers who have an understanding of this microcomputer (MCU) including its hardware functions and electrical characteristics. The user should have a basic understanding of electric circuits, logic circuits and, MCUs.
This manual consists of 29 chapters covering six main categories: Overview, CPU, System Control, Peripherals, Electrical Characteristics, and Usage Notes.

Carefully read all notes in this document prior to use. Notes are found throughout each chapter, at the end of each chapter, and in the dedicated Usage Notes chapter.

The Revision History at the end of this manual summarizes primary modifications and additions to the previous versions. For details, please refer to the relative chapters or sections of this manual.

The R32C/116A Group includes the documents listed below. Verify this manual is the latest version by visiting the Renesas Electronics website.

Type of Document	Contents	Document Name	Document Number
Datasheet	Overview of Hardware and Electrical Characteristics	R32C/116A Group Datasheet	R01DS0066EJ0110
User’s Manual: Hardware	Specifications and detailed descriptions of: -pin layout -memory map -peripherals -electrical characteristics -timing characteristics Refer to the Application Manual for peripheral usage.	R32C/116A Group User’s Manual: Hardware	This publication
User’s Manual: Software/Software Manual	Descriptions of instruction set	R32C/100 Series Software Manual	REJ09B0267-0100
Application Note	-Usages -Applications -Sample programs -Programing technics using Assembly language or C programming language	Available on the Renesas Electronics website.	
Renesas Technical Update	Bulletins on product specifications, documents, etc.		

2. Numbers and Symbols

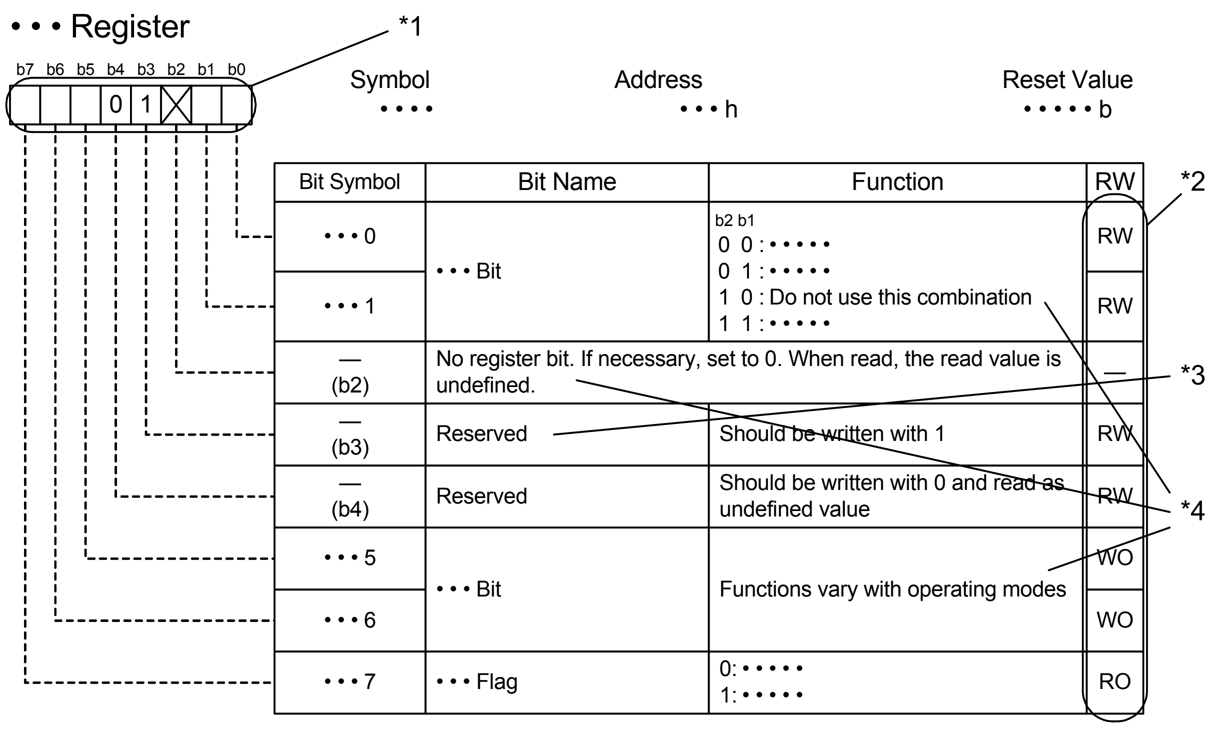
The following explains the denotations used in this manual for registers, bits, pins and various numbers.

(1) Registers, bits, and pins
Registers, bits, and pins are indicated by symbols. Each symbol has a register/bit/pin identifier after the symbol.
Example: PM03 bit in the PM0 register
P3_5 pin, VCC pin

(2) Numbers
A binary number has the suffix “b” except for a 1-bit value.
A hexadecimal number has the suffix “h”.
A decimal number has no suffix.
Example: Binary notation: 11b
Hexadecimal notation: EFA0h
Decimal notation: 1234

3. Registers

The following illustration describes registers used throughout this manual.



• • • Register

Symbol	Address	Reset Value
• • • •	• • • h	• • • • • b

Bit Symbol	Bit Name	Function	RW
• • • 0	• • • Bit	b2 b1 0 0 : • • • • • 0 1 : • • • • • 1 0 : Do not use this combination 1 1 : • • • • •	RW
• • • 1			RW
— (b2)	No register bit. If necessary, set to 0. When read, the read value is undefined.		—
— (b3)	Reserved	Should be written with 1	RW
— (b4)	Reserved	Should be written with 0 and read as undefined value	RW
• • • 5	• • • Bit	Functions vary with operating modes	WO
• • • 6			WO
• • • 7	• • • Flag	0: • • • • • 1: • • • • •	RO

*1
Blank box: Set this bit to 0 or 1 according to the function.
0: Set this bit to 0.
1: Set this bit to 1.
X: Nothing is assigned to this bit.

*2
RW: Read and write
RO: Read only
WO: Write only (the read value is undefined)
—: Not applicable

*3
- Reserved bit: This bit field is reserved. Set this bit to a specified value. For RW bits, the written value is read unless otherwise noted.

*4
- No register bit(s): No register bit(s) is/are assigned to this field. If necessary, set to 0 for possible future implementation.
- Do not use this combination: Proper operation is not guaranteed when this value is set.
- Functions vary with operating modes: Functions vary with peripheral operating modes. Refer to register illustrations of the respective mode.

4. Abbreviations and Acronyms

The following acronyms and terms are used throughout this manual.

Abbreviation/Acronym	Meaning
ACIA	Asynchronous Communications Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connection
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

TABLE OF CONTENTS

R01UH0213EJ0110
Rev.1.10
Sep 28, 2012

1. Overview

1.1 Features

The M16C Family offers a robust platform of 32-/16-bit CISC microcomputers (MCUs) featuring high ROM code efficiency, extensive EMI/EMS noise immunity, ultra-low power consumption, high-speed processing in actual applications, and numerous and varied integrated peripherals. Extensive device scalability from low- to high-end, featuring a single architecture as well as compatible pin assignments and peripheral functions, provides support for a vast range of application fields.

The R32C/100 Series is a high-end microcontroller series in the M16C Family. With a 4-Gbyte memory space, it achieves maximum code efficiency and high-speed processing with 32-bit CISC architecture, multiplier, multiply-accumulate unit, and floating point unit. The selection from the broadest choice of on-chip peripheral devices — UART, CRC, DMAC, A/D and D/A converters, timers, I^2C, and watchdog timer enables to minimize external components.

The R32C/100 Series, in particular, provides the R32C/116A Group as a standard product. This product, provided as a 144/176-pin plastic molded LQFP package, has 11 channels of serial interface and one channel of multi-master I^2C-bus interface.

1.1.1 Applications

Car audio, audio, cameras, television, home appliance, printer, office/industrial equipment, communication/portable devices, etc.

1.1.2 Performance Overview

Tables 1.1 to 1.4 show the performance overview of the R32C/116A Group.

Table 1.1 Performance Overview for the 176 pin-Package (1/2)

Unit	Function	Explanation
CPU	Central processing unit	R32C/100 Series CPU Core • Basic instructions: 108 • Minimum instruction execution time: 15.625 ns (f(CPU) = 64 MHz) • Multiplier: 32-bit × 32-bit → 64-bit • Multiply-accumulate unit: 32-bit × 32-bit + 64-bit → 64-bit • IEEE-754 compatible FPU: Single precision • 32-bit barrel shifter • Operating mode: Single-chip mode, memory expansion mode, microprocessor mode (optional (1))
Memory		Flash memory: 512 Kbytes to 1 Mbyte RAM: 96 Kbytes Data flash: 4 Kbytes × 2 blocks Refer to Table 1.5 for memory size of each product group
Voltage Detector	Low voltage detector	Optional (1) Low voltage detection interrupt
Clock	Clock generator	• 4 circuits (main clock, sub clock, PLL, on-chip oscillator) • Oscillation stop detector: Main clock oscillator stop/restart detection • Frequency divide circuit: Divide-by-2 to divide-by-24 selectable • Low power modes: Wait mode, stop mode
External Bus Expansion	Bus and memory expansion	• Address space: 4 Gbytes (of which up to 64 Mbytes is user accessible) • External bus Interface: Support for wait-state insertion, 4 chip select outputs • Bus format: Separate bus/Multiplexed bus selectable, data bus width selectable (8/16/32 bits)
Interrupts		Interrupt vectors: 261 External interrupt inputs: $\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 9, key input × 4 Interrupt priority levels: 7
Watchdog Timer		15 bits × 1 (selectable input frequency from prescaler output) • Automatic timer start function is available
DMA	DMAC	4 channels • Cycle-steal transfer mode • Request sources: 61 • 2 transfer modes: Single transfer, repeat transfer
	DMAC II	• Triggered by an interrupt request of any peripheral • 3 characteristic transfer functions: Immediate data transfer, calculation result transfer, chain transfer
I/O Ports	Programmable I/O ports	• 2 input-only ports • 156 CMOS I/O ports • 52 ports are 5 V tolerant • A pull-up resistor is selectable for every 4 input ports (except 5 V tolerant inputs)

Note:

1. Contact a Renesas Electronics sales office to use the optional features.

Table 1.2 **Performance Overview for the 176-pin Package (2/2)**

Unit	Function	Explanation
Timer	Timer A	16-bit timer × 5 Timer mode, event counter mode, one-shot timer mode, pulse-width modulation (PWM) mode Two-phase pulse signal processing in event counter mode (two-phase encoder input) × 3
	Timer B	16-bit timer × 6 Timer mode, event counter mode, pulse frequency measurement mode, pulse-width measurement mode
	Three-phase motor control timer	Three-phase motor control timer × 1 (timers A1, A2, A4, and B2 used) 8-bit programmable dead time timer
Serial Interface	UART0 to UART10	Asynchronous/synchronous serial interface × 11 channels • I^{2}C-bus (UART0 to UART6) • Special mode 2 (UART0 to UART6) • IEBus (optional [1]) (UART0 to UART6)
A/D Converter		10-bit resolution × 34 channels Sample and hold functionality integrated Self test/Open-circuit detection assist
D/A Converter		8-bit resolution × 2
CRC Calculator		CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$)
X-Y Converter		16 bits × 16 bits
Intelligent I/O		Time measurement (input capture): 16 bits × 16 Waveform generation (output compare): 16 bits × 24 Serial interface: Variable-length synchronous serial I/O mode, IEBus mode (optional [1])
Multi-master I^{2}C-bus Interface		1 channel
Flash Memory		Programming and erasure supply voltage: VCC = 3.0 to 5.5 V Minimum endurance: 1,000 program/erase cycles Suspend/resume function available Security protection: ROM code protect, ID code protect Debugging: On-chip debug, on-board flash programming
Operating Frequency/Supply Voltage		64 MHz/VCC = 3.0 to 5.5 V
Operating Temperature		-20°C to 85°C (N version) -40°C to 85°C (D version) -40°C to 85°C (P version)
Current Consumption		45 mA (VCC = 5.0 V, f(CPU) = 64 MHz) 8 µA (VCC = 3.3 V, f(XCIN) = 32.768 kHz, in wait mode)
Package		176-pin plastic molded LQFP (PLQP0176KB-A)

Note:
1. Contact a Renesas Electronics sales office to use the optional features.

Table 1.3 Performance Overview for the 144-pin Package (1/2)

Unit	Function	Explanation
CPU	Central processing unit	R32C/100 Series CPU Core • Basic instructions: 108 • Minimum instruction execution time: 15.625 ns (f(CPU) = 64 MHz) • Multiplier: 32-bit × 32-bit → 64-bit • Multiply-accumulate unit: 32-bit × 32-bit + 64-bit → 64-bit • IEEE-754 compatible FPU: Single precision • 32-bit barrel shifter • Operating mode: Single-chip mode, memory expansion mode, microprocessor mode (optional [1])
Memory		Flash memory: 512 Kbytes to 1 Mbyte RAM: 96 Kbytes Data flash: 4 Kbytes × 2 blocks Refer to Table 1.5 for memory size of each product group
Voltage Detector	Low voltage detector	Optional [1] Low voltage detection interrupt
Clock	Clock generator	• 4 circuits (main clock, sub clock, PLL, on-chip oscillator) • Oscillation stop detector: Main clock oscillator stop/restart detection • Frequency divide circuit: Divide-by-2 to divide-by-24 selectable • Low power modes: Wait mode, stop mode
External Bus Expansion	Bus and memory expansion	• Address space: 4 Gbytes (of which up to 64 Mbytes is user accessible) • External bus Interface: Support for wait-state insertion, 4 chip select outputs • Bus format: Separate bus/Multiplexed bus selectable, data bus width selectable (8/16/32 bits)
Interrupts		Interrupt vectors: 261 External interrupt inputs: $\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 9, key input × 4 Interrupt priority levels: 7
Watchdog Timer		15 bits × 1 (selectable input frequency from prescaler output) Automatic timer start function is available
DMA	DMAC	4 channels • Cycle-steal transfer mode • Request sources: 61 • 2 transfer modes: Single transfer, repeat transfer
	DMAC II	• Triggered by an interrupt request of any peripheral • 3 characteristic transfer functions: Immediate data transfer, calculation result transfer, chain transfer
I/O Ports	Programmable I/O ports	• 2 input-only ports • 124 CMOS I/O ports • 40 ports are 5 V tolerant • A pull-up resistor is selectable for every 4 input ports (except 5 V tolerant inputs)

Note:

1. Contact a Renesas Electronics sales office to use the optional features.

Table 1.4 Performance Overview for the 144-pin Package (2/2)

Unit	Function	Explanation
Timer	Timer A	16-bit timer × 5 Timer mode, event counter mode, one-shot timer mode, pulse-width modulation (PWM) mode Two-phase pulse signal processing in event counter mode (two-phase encoder input) × 3
	Timer B	16-bit timer × 6 Timer mode, event counter mode, pulse frequency measurement mode, pulse-width measurement mode
	Three-phase motor control timer	Three-phase motor control timer × 1 (timers A1, A2, A4, and B2 used) 8-bit programmable dead time timer
Serial Interface	UART0 to UART10	Asynchronous/synchronous serial interface × 11 channels • I^{2}C-bus (UART0 to UART6) • Special mode 2 (UART0 to UART6) • IEBus (optional [1]) (UART0 to UART6)
A/D Converter		10-bit resolution × 34 channels Sample and hold functionality integrated Self test/Open-circuit detection assist
D/A Converter		8-bit resolution × 2
CRC Calculator		CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$)
X-Y Converter		16 bits × 16 bits
Intelligent I/O		Time measurement (input capture): 16 bits × 16 Waveform generation (output compare): 16 bits × 24 Serial interface: Variable-length synchronous serial I/O mode, IEBus mode (optional [1])
Multi-master I^{2}C-bus Interface		1 channel
Flash Memory		Programming and erasure supply voltage: VCC = 3.0 to 5.5 V Minimum endurance: 1,000 program/erase cycles Suspend/resume function available Security protection: ROM code protect, ID code protect Debugging: On-chip debug, on-board flash programming
Operating Frequency/Supply Voltage		64 MHz/VCC = 3.0 to 5.5 V
Operating Temperature		-20°C to 85°C (N version) -40°C to 85°C (D version) -40°C to 85°C (P version)
Current Consumption		45 mA (VCC = 5.0 V, f(CPU) = 64 MHz) 8 µA (VCC = 3.3 V, f(XCIN) = 32.768 kHz, in wait mode)
Package		144-pin plastic molded LQFP (PLQP0144KA-A)

Note:

1. Contact a Renesas Electronics sales office to use the optional features.

1.2 Product Information

Table 1.5 lists the product information and Figure 1.1 shows the details of the part number.

Table 1.5 R32C/116A Group Product List **As of September, 2012**

Part Number	Package Code [1]	ROM Capacity [2]	RAM Capacity	Remarks
R5F6416JANFE (P)	PLQP0176KB-A	512 Kbytes + 8 Kbytes	96 Kbytes	-20°C to 85°C (N version)
R5F6416JADFE				-40°C to 85°C (D version)
R5F6416JAPFE				-40°C to 85°C (P version)
R5F6416JANFD (P)	PLQP0144KA-A			-20°C to 85°C (N version)
R5F6416JADFD				-40°C to 85°C (D version)
R5F6416JAPFD				-40°C to 85°C (P version)
R5F6416KANFE (P)	PLQP0176KB-A	640 Kbytes + 8 Kbytes	96 Kbytes	-20°C to 85°C (N version)
R5F6416KADFE				-40°C to 85°C (D version)
R5F6416KAPFE				-40°C to 85°C (P version)
R5F6416KANFD (P)	PLQP0144KA-A			-20°C to 85°C (N version)
R5F6416KADFD				-40°C to 85°C (D version)
R5F6416KAPFD				-40°C to 85°C (P version)
R5F6416LANFE (P)	PLQP0176KB-A	768 Kbytes + 8 Kbytes	96 Kbytes	-20°C to 85°C (N version)
R5F6416LADFE				-40°C to 85°C (D version)
R5F6416LAPFE				-40°C to 85°C (P version)
R5F6416LANFD (P)	PLQP0144KA-A			-20°C to 85°C (N version)
R5F6416LADFD				-40°C to 85°C (D version)
R5F6416LAPFD				-40°C to 85°C (P version)
R5F6416MANFE (P)	PLQP0176KB-A	1 Mbyte + 8 Kbytes	96 Kbytes	-20°C to 85°C (N version)
R5F6416MADFE				-40°C to 85°C (D version)
R5F6416MAPFE				-40°C to 85°C (P version)
R5F6416MANFD (P)	PLQP0144KA-A			-20°C to 85°C (N version)
R5F6416MADFD				-40°C to 85°C (D version)
R5F6416MAPFD				-40°C to 85°C (P version)

(P): On planning phase

Notes:

1. The old package codes are as follows:
   PLQP0144KA-A: 144P6Q-A, PLQP0176KB-A: 176P6Q-A
2. “+ 8 Kbytes” in the ROM capacity column indicates the data flash capacity.

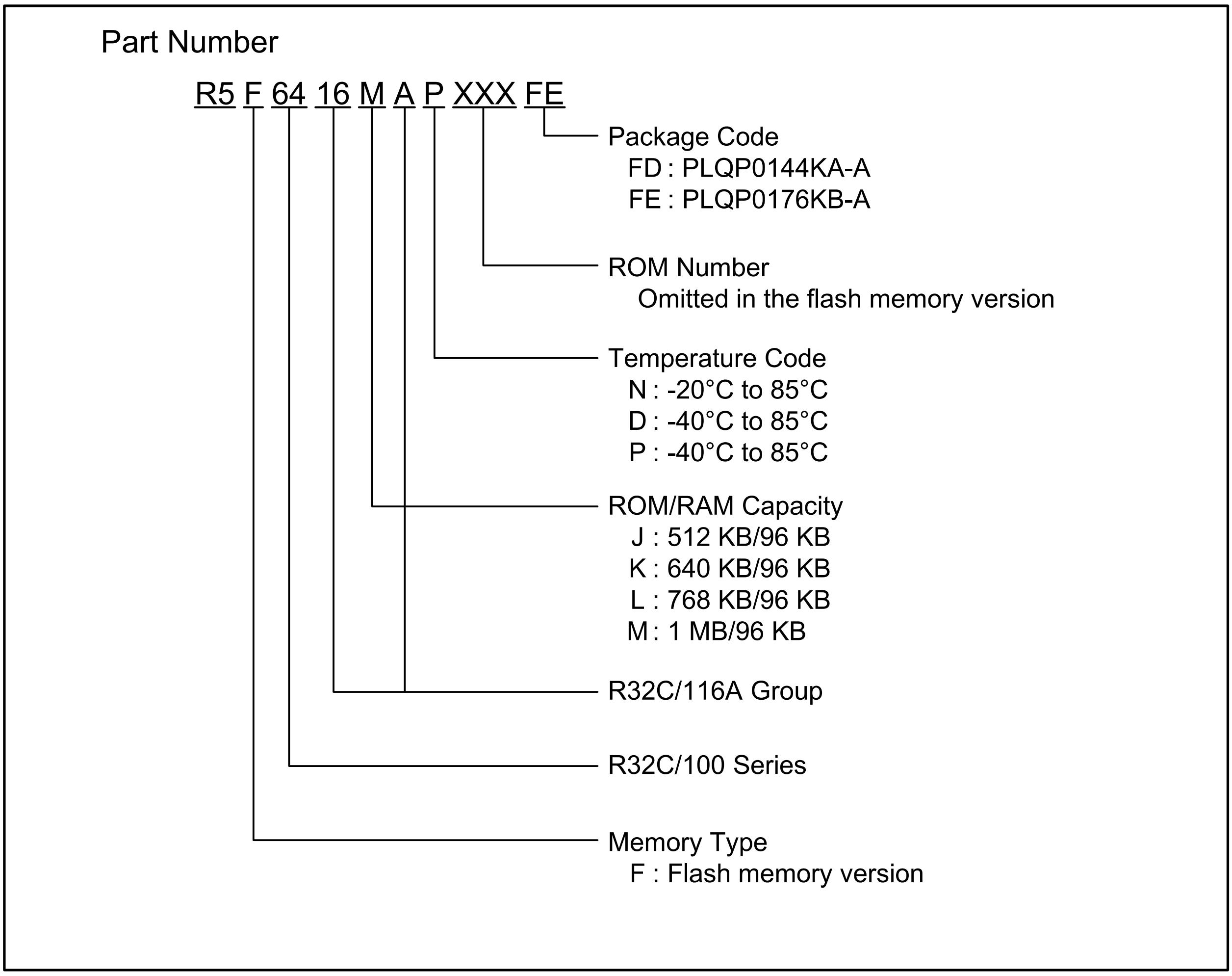
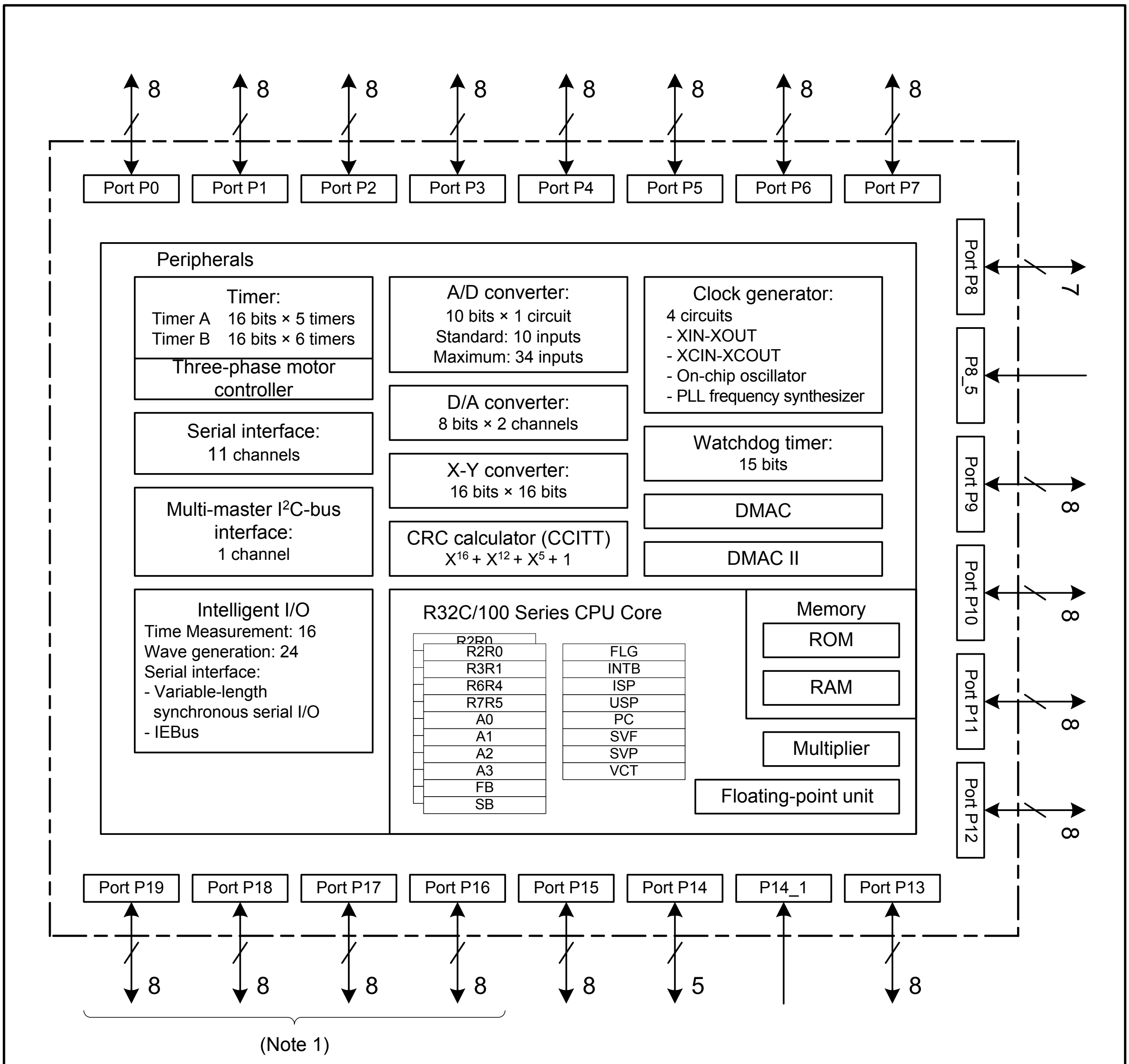


Figure 1.1 Part Numbering

1.3 Block Diagram

Figure 1.2 shows a block diagram of the R32C/116A Group.



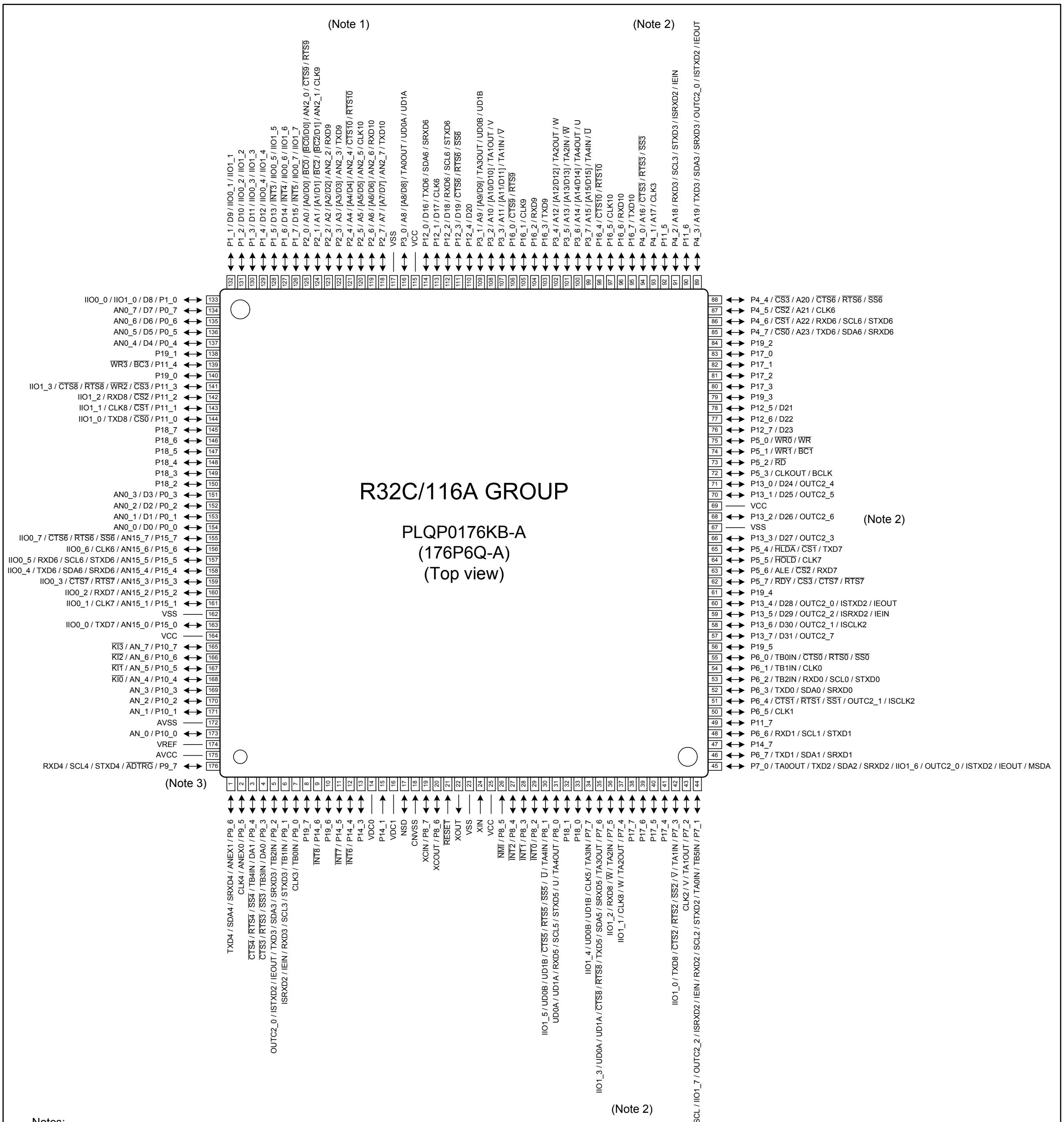
Note:

1. Ports P16 to P19 are available in the 176-pin package only.

Figure 1.2 R32C/116A Group Block Diagram

1.4 Pin Assignments

Figures 1.3 and 1.4 show the pin assignments (top view) and Tables 1.6 to 1.13 show the pin characteristics.


(Note 1)
(Note 2)
132 P1_1 / D9 / IIO0_1 / IIO1_1
131 P1_2 / D10 / IIO0_2 / IIO1_2
130 P1_3 / D11 / IIO0_3 / IIO1_3
129 P1_4 / D12 / IIO0_4 / IIO1_4
128 P1_5 / D13 / INT3 / IIO0_5 / IIO1_5
127 P1_6 / D14 / INT4 / IIO0_6 / IIO1_6
126 P1_7 / D15 / INT5 / IIO0_7 / IIO1_7
125 P2_0 / A0 / [A0/D0] / [BC0/D0] / BC0 / AN2_0 / CTS9 / RTS9
124 P2_1 / A1 / [A1/D1] / [BC2/D1] / BC2 / AN2_1 / CLK9
123 P2_2 / A2 / [A2/D2] / AN2_2 / RXD9
122 P2_3 / A3 / [A3/D3] / AN2_3 / TXD9
121 P2_4 / A4 / [A4/D4] / AN2_4 / CTS10 / RTS10
120 P2_5 / A5 / [A5/D5] / AN2_5 / CLK10
119 P2_6 / A6 / [A6/D6] / AN2_6 / RXD10
118 P2_7 / A7 / [A7/D7] / AN2_7 / TXD10
117 VSS
116 P3_0 / A8 / [A8/D8] / TA0OUT / UD0A / UD1A
115 VCC
114 P12_0 / D16 / TXD6 / SDA6 / SRXD6
113 P12_1 / D17 / CLK6
112 P12_2 / D18 / RXD6 / SCL6 / STXD6
111 P12_3 / D19 / CTS6 / RTS6 / SS6
110 P12_4 / D20
109 P3_1 / A9 / [A9/D9] / TA3OUT / UD0B / UD1B
108 P3_2 / A10 / [A10/D10] / TA1OUT / V
107 P3_3 / A11 / [A11/D11] / TA1IN / V
106 P16_0 / CTS9 / RTS9
105 P16_1 / CLK9
104 P16_2 / RXD9
103 P16_3 / TXD9
102 P3_4 / A12 / [A12/D12] / TA2OUT / W
101 P3_5 / A13 / [A13/D13] / TA2IN / W
100 P3_6 / A14 / [A14/D14] / TA4OUT / U
99 P3_7 / A15 / [A15/D15] / TA4IN / U
98 P16_4 / CTS10 / RTS10
97 P16_5 / CLK10
96 P16_6 / RXD10
95 P16_7 / TXD10
94 P4_0 / A16 / CTS3 / RTS3 / SS3
93 P4_1 / A17 / CLK3
92 P11_5
91 P4_2 / A18 / RXD3 / SCL3 / STXD3 / ISRXD2 / IEIN
90 P11_6
89 P4_3 / A19 / TXD3 / SDA3 / SRXD3 / OUTC2_0 / ISTXD2 / IEOUT
IIO0_0 / IIO1_0 / D8 / P1_0 133
AN0_7 / D7 / P0_7 134
AN0_6 / D6 / P0_6 135
AN0_5 / D5 / P0_5 136
AN0_4 / D4 / P0_4 137
P19_1 138
WR3 / BC3 / P11_4 139
P19_0 140
IIO1_3 / CTS8 / RTS8 / WR2 / CS3 / P11_3 141
IIO1_2 / RXD8 / CS2 / P11_2 142
IIO1_1 / CLK8 / CS1 / P11_1 143
IIO1_0 / TXD8 / CS0 / P11_0 144
P18_7 145
P18_6 146
P18_5 147
P18_4 148
P18_3 149
P18_2 150
AN0_3 / D3 / P0_3 151
AN0_2 / D2 / P0_2 152
AN0_1 / D1 / P0_1 153
AN0_0 / D0 / P0_0 154
IIO0_7 / CTS6 / RTS6 / SS6 / AN15_7 / P15_7 155
IIO0_6 / CLK6 / AN15_6 / P15_6 156
IIO0_5 / RXD6 / SCL6 / STXD6 / AN15_5 / P15_5 157
IIO0_4 / TXD6 / SDA6 / SRXD6 / AN15_4 / P15_4 158
IIO0_3 / CTS7 / RTS7 / AN15_3 / P15_3 159
IIO0_2 / RXD7 / AN15_2 / P15_2 160
IIO0_1 / CLK7 / AN15_1 / P15_1 161
VSS 162
IIO0_0 / TXD7 / AN15_0 / P15_0 163
VCC 164
KI3 / AN_7 / P10_7 165
KI2 / AN_6 / P10_6 166
KI1 / AN_5 / P10_5 167
KI0 / AN_4 / P10_4 168
AN_3 / P10_3 169
AN_2 / P10_2 170
AN_1 / P10_1 171
AVSS 172
AN_0 / P10_0 173
VREF 174
AVCC 175
RXD4 / SCL4 / STXD4 / ADTRG / P9_7 176
(Note 3)
R32C/116A GROUP
PLQP0176KB-A
(176P6Q-A)
(Top view)
88 P4_4 / CS3 / A20 / CTS6 / RTS6 / SS6
87 P4_5 / CS2 / A21 / CLK6
86 P4_6 / CS1 / A22 / RXD6 / SCL6 / STXD6
85 P4_7 / CS0 / A23 / TXD6 / SDA6 / SRXD6
84 P19_2
83 P17_0
82 P17_1
81 P17_2
80 P17_3
79 P19_3
78 P12_5 / D21
77 P12_6 / D22
76 P12_7 / D23
75 P5_0 / WR0 / WR
74 P5_1 / WR1 / BC1
73 P5_2 / RD
72 P5_3 / CLKOUT / BCLK
71 P13_0 / D24 / OUTC2_4
70 P13_1 / D25 / OUTC2_5
69 VCC
68 P13_2 / D26 / OUTC2_6
(Note 2)
67 VSS
66 P13_3 / D27 / OUTC2_3
65 P5_4 / HLDA / CS1 / TXD7
64 P5_5 / HOLD / CLK7
63 P5_6 / ALE / CS2 / RXD7
62 P5_7 / RDY / CS3 / CTS7 / RTS7
61 P19_4
60 P13_4 / D28 / OUTC2_0 / ISTXD2 / IEOUT
59 P13_5 / D29 / OUTC2_2 / ISRXD2 / IEIN
58 P13_6 / D30 / OUTC2_1 / ISCLK2
57 P13_7 / D31 / OUTC2_7
56 P19_5
55 P6_0 / TB0IN / CTS0 / RTS0 / SS0
54 P6_1 / TB1IN / CLK0
53 P6_2 / TB2IN / RXD0 / SCL0 / STXD0
52 P6_3 / TXD0 / SDA0 / SRXD0
51 P6_4 / CTS1 / RTS1 / SS1 / OUTC2_1 / ISCLK2
50 P6_5 / CLK1
49 P11_7
48 P6_6 / RXD1 / SCL1 / STXD1
47 P14_7
46 P6_7 / TXD1 / SDA1 / SRXD1
45 P7_0 / TA0OUT / TXD2 / SDA2 / SRXD2 / IIO1_6 / OUTC2_0 / ISTXD2 / IEOUT / MSDA
1 TXD4 / SDA4 / SRXD4 / ANEX1 / P9_6
2 CLK4 / ANEX0 / P9_5
3 CTS4 / RTS4 / SS4 / TB4IN / DA1 / P9_4
4 CTS3 / RTS3 / SS3 / TB3IN / DA0 / P9_3
5 OUTC2_0 / ISTXD2 / IEOUT / TXD3 / SDA3 / SRXD3 / TB2IN / P9_2
6 ISRXD2 / IEIN / RXD3 / SCL3 / STXD3 / TB1IN / P9_1
7 CLK3 / TB0IN / P9_0
8 P19_7
9 INT8 / P14_6
10 P19_6
11 INT7 / P14_5
12 INT6 / P14_4
13 P14_3
14 VDC0
15 P14_1
16 VDC1
17 NSD
18 CNVSS
19 XCIN / P8_7
20 XCOUT / P8_6
21 RESET
22 XOUT
23 VSS
24 XIN
25 VCC
26 NMI / P8_5
27 INT2 / P8_4
28 INT1 / P8_3
29 INT0 / P8_2
30 IIO1_5 / UD0B / UD1B / CTS5 / RTS5 / SS5 / U / TA4IN / P8_1
31 UD0A / UD1A / RXD5 / SCL5 / STXD5 / U / TA4OUT / P8_0
32 P18_1
33 P18_0
34 IIO1_4 / UD0B / UD1B / CLK5 / TA3IN / P7_7
35 IIO1_3 / UD0A / UD1A / CTS8 / RTS8 / TXD5 / SDA5 / SRXD5 / TA3OUT / P7_6
36 IIO1_2 / RXD8 / W / TA2IN / P7_5
37 IIO1_1 / CLK8 / W / TA2OUT / P7_4
38 P17_7
39 P17_6
40 P17_5
41 P17_4
42 IIO1_0 / TXD8 / CTS2 / RTS2 / SS2 / V / TA1IN / P7_3
43 CLK2 / V / TA1OUT / P7_2
44 MSCL / IIO1_7 / OUTC2_2 / ISRXD2 / IEIN / RXD2 / SCL2 / STXD2 / TA0IN / TB5IN / P7_1
(Note 2)

Notes:
1. Pin names in brackets [ ] represent a functional signal as a whole and should not be considered as two separate pins.
2. The following pins are 5 V tolerant inputs: P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_3, P12_0 to P12_7, P16_0 to P16_7, and P17_0 to P17_3.
3. The position of pin number 1 varies by product. Refer to the index mark in attached “Package Dimensions”.

Figure 1.3 Pin Assignment for the 176-pin Package (top view)

Table 1.6 Pin Characteristics for the 176-pin Package (1/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
1		P9_6			TXD4/SDA4/SRXD4		ANEX1	
2		P9_5			CLK4		ANEX0	
3		P9_4		TB4IN	CTS4/RTS4/SS4		DA1	
4		P9_3		TB3IN	CTS3/RTS3/SS3		DA0	
5		P9_2		TB2IN	TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/IEOUT		
6		P9_1		TB1IN	RXD3/SCL3/STXD3	ISRXD2/IEIN		
7		P9_0		TB0IN	CLK3			
8		P19_7						
9		P14_6	INT8					
10		P19_6						
11		P14_5	INT7					
12		P14_4	INT6					
13		P14_3						
14	VDC0							
15		P14_1						
16	VDC1							
17	NSD							
18	CNVSS							
19	XCIN	P8_7						
20	XCOUT	P8_6						
21	RESET							
22	XOUT							
23	VSS							
24	XIN							
25	VCC							
26		P8_5	NMI					
27		P8_4	INT2					
28		P8_3	INT1					
29		P8_2	INT0					
30		P8_1		TA4IN/U	CTS5/RTS5/SS5	IIO1_5/UD0B/UD1B		
31		P8_0		TA4OUT/U	RXD5/SCL5/STXD5	UD0A/UD1A		
32		P18_1						
33		P18_0						
34		P7_7		TA3IN	CLK5	IIO1_4/UD0B/UD1B		
35		P7_6		TA3OUT	TXD5/SDA5/SRXD5/CTS8/RTS8	IIO1_3/UD0A/UD1A		
36		P7_5		TA2IN/W	RXD8	IIO1_2		

Table 1.7 Pin Characteristics for the 176-pin Package (2/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
37		P7_4		TA2OUT/W	CLK8	IIO1_1		
38		P17_7						
39		P17_6						
40		P17_5						
41		P17_4						
42		P7_3		TA1IN/V	CTS2/RTS2/SS2/TXD8	IIO1_0		
43		P7_2		TA1OUT/V	CLK2			
44		P7_1		TA0IN/ TB5IN	RXD2/SCL2/STXD2/ MSCL	IIO1_7/OUTC2_2/ ISRXD2/IEIN		
45		P7_0		TA0OUT	TXD2/SDA2/SRXD2/ MSDA	IIO1_6/OUTC2_0/ ISTXD2/IEOUT		
46		P6_7			TXD1/SDA1/SRXD1			
47		P14_7						
48		P6_6			RXD1/SCL1/STXD1			
49		P11_7						
50		P6_5			CLK1			
51		P6_4			CTS1/RTS1/SS1	OUTC2_1/ISCLK2		
52		P6_3			TXD0/SDA0/SRXD0			
53		P6_2		TB2IN	RXD0/SCL0/STXD0			
54		P6_1		TB1IN	CLK0			
55		P6_0		TB0IN	CTS0/RTS0/SS0			
56		P19_5						
57		P13_7				OUTC2_7		D31
58		P13_6				OUTC2_1/ISCLK2		D30
59		P13_5				OUTC2_2/ISRXD2/ IEIN		D29
60		P13_4				OUTC2_0/ISTXD2/ IEOUT		D28
61		P19_4						
62		P5_7			CTS7/RTS7			RDY/CS3
63		P5_6			RXD7			ALE/CS2
64		P5_5			CLK7			HOLD
65		P5_4			TXD7			HLDA/CS1
66		P13_3				OUTC2_3		D27
67	VSS							
68		P13_2				OUTC2_6		D26
69	VCC							
70		P13_1				OUTC2_5		D25
71		P13_0				OUTC2_4		D24
72		P5_3						CLKOUT/ BCLK
73		P5_2						RD
74		P5_1						WR1/BC1

Table 1.8 Pin Characteristics for the 176-pin Package (3/5)

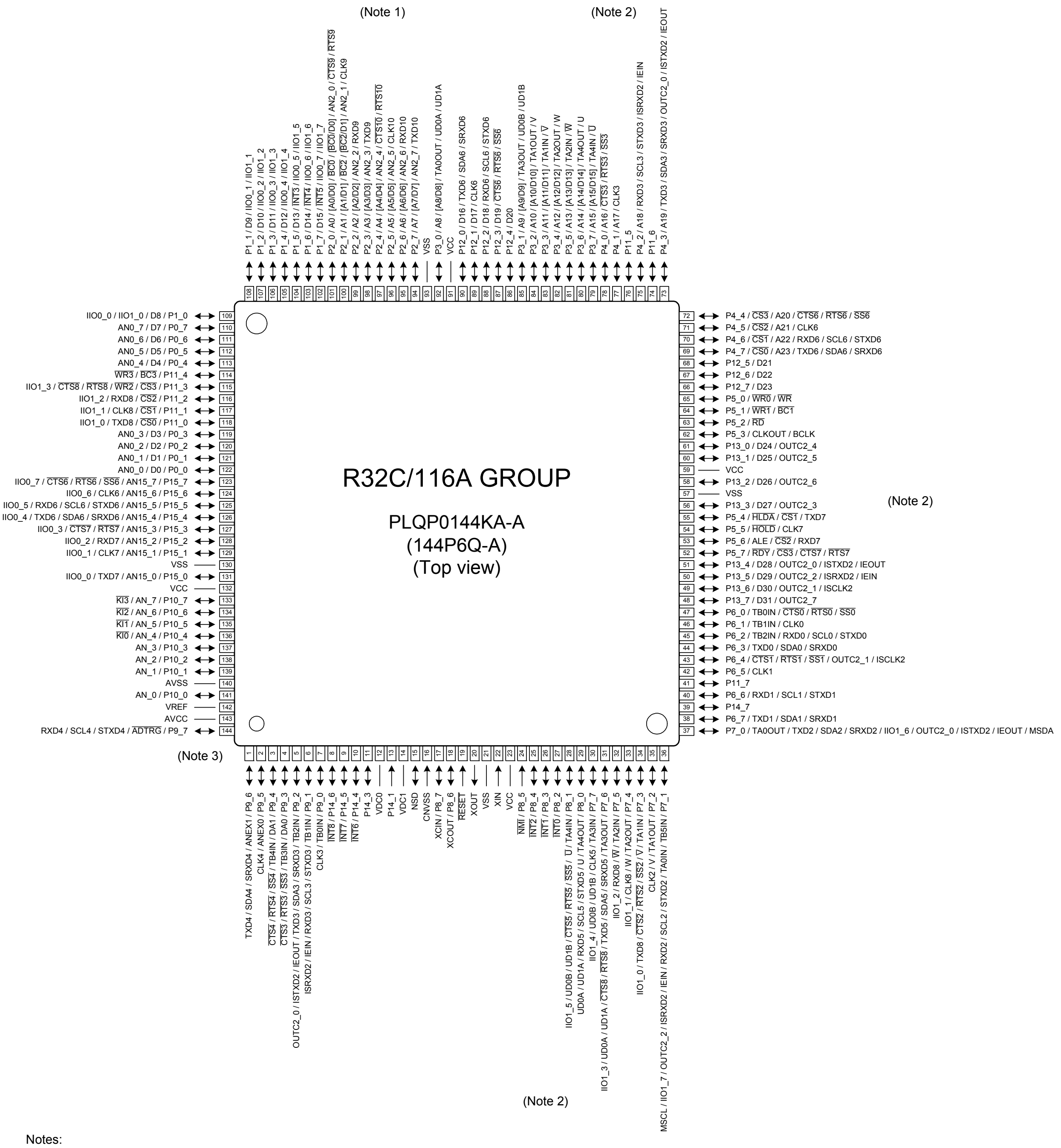
Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
75		P5_0						$\overline{\text{WR0}}$/$\overline{\text{WR}}$
76		P12_7						D23
77		P12_6						D22
78		P12_5						D21
79		P19_3						
80		P17_3						
81		P17_2						
82		P17_1						
83		P17_0						
84		P19_2						
85		P4_7			TXD6/SDA6/SRXD6			$\overline{\text{CS0}}$/A23
86		P4_6			RXD6/SCL6/STXD6			$\overline{\text{CS1}}$/A22
87		P4_5			CLK6			$\overline{\text{CS2}}$/A21
88		P4_4			$\overline{\text{CTS6}}$/$\overline{\text{RTS6}}$/$\overline{\text{SS6}}$			$\overline{\text{CS3}}$/A20
89		P4_3			TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/IEOUT		A19
90		P11_6						
91		P4_2			RXD3/SCL3/STXD3	ISRXD2/IEIN		A18
92		P11_5						
93		P4_1			CLK3			A17
94		P4_0			$\overline{\text{CTS3}}$/$\overline{\text{RTS3}}$/$\overline{\text{SS3}}$			A16
95		P16_7			TXD10			
96		P16_6			RXD10			
97		P16_5			CLK10			
98		P16_4			$\overline{\text{CTS10}}$/$\overline{\text{RTS10}}$			
99		P3_7		TA4IN/$\overline{\text{U}}$				A15(/D15)
100		P3_6		TA4OUT/U				A14(/D14)
101		P3_5		TA2IN/$\overline{\text{W}}$				A13(/D13)
102		P3_4		TA2OUT/W				A12(/D12)
103		P16_3			TXD9			
104		P16_2			RXD9			
105		P16_1			CLK9			
106		P16_0			$\overline{\text{CTS9}}$/$\overline{\text{RTS9}}$			
107		P3_3		TA1IN/$\overline{\text{V}}$				A11(/D11)
108		P3_2		TA1OUT/V				A10(/D10)
109		P3_1		TA3OUT		UD0B/UD1B		A9(/D9)
110		P12_4						D20
111		P12_3			$\overline{\text{CTS6}}$/$\overline{\text{RTS6}}$/$\overline{\text{SS6}}$			D19
112		P12_2			RXD6/SCL6/STXD6			D18
113		P12_1			CLK6			D17
114		P12_0			TXD6/SDA6/SRXD6			D16

Table 1.9 Pin Characteristics for the 176-pin Package (4/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
115	VCC							
116		P3_0		TA0OUT		UD0A/UD1A		A8(/D8)
117	VSS							
118		P2_7			TXD10		AN2_7	A7(/D7)
119		P2_6			RXD10		AN2_6	A6(/D6)
120		P2_5			CLK10		AN2_5	A5(/D5)
121		P2_4			CTS10/RTS10		AN2_4	A4(/D4)
122		P2_3			TXD9		AN2_3	A3(/D3)
123		P2_2			RXD9		AN2_2	A2(/D2)
124		P2_1			CLK9		AN2_1	A1(/D1)/ BC2(/D1)
125		P2_0			CTS9/RTS9		AN2_0	A0(/D0)/ BC0(/D0)
126		P1_7	INT5			IIO0_7/IIO1_7		D15
127		P1_6	INT4			IIO0_6/IIO1_6		D14
128		P1_5	INT3			IIO0_5/IIO1_5		D13
129		P1_4				IIO0_4/IIO1_4		D12
130		P1_3				IIO0_3/IIO1_3		D11
131		P1_2				IIO0_2/IIO1_2		D10
132		P1_1				IIO0_1/IIO1_1		D9
133		P1_0				IIO0_0/IIO1_0		D8
134		P0_7					AN0_7	D7
135		P0_6					AN0_6	D6
136		P0_5					AN0_5	D5
137		P0_4					AN0_4	D4
138		P19_1						
139		P11_4						BC3/WR3
140		P19_0						
141		P11_3			CTS8/RTS8	IIO1_3		CS3/WR2
142		P11_2			RXD8	IIO1_2		CS2
143		P11_1			CLK8	IIO1_1		CS1
144		P11_0			TXD8	IIO1_0		CS0
145		P18_7						
146		P18_6						
147		P18_5						
148		P18_4						
149		P18_3						
150		P18_2						
151		P0_3					AN0_3	D3
152		P0_2					AN0_2	D2
153		P0_1					AN0_1	D1
154		P0_0					AN0_0	D0
155		P15_7			CTS6/RTS6/SS6	IIO0_7	AN15_7	

Table 1.10 Pin Characteristics for the 176-pin Package (5/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
156		P15_6			CLK6	IIO0_6	AN15_6	
157		P15_5			RXD6/SCL6/STXD6	IIO0_5	AN15_5	
158		P15_4			TXD6/SDA6/SRXD6	IIO0_4	AN15_4	
159		P15_3			$\overline{\text{CTS7}}$/$\overline{\text{RTS7}}$	IIO0_3	AN15_3	
160		P15_2			RXD7	IIO0_2	AN15_2	
161		P15_1			CLK7	IIO0_1	AN15_1	
162	VSS							
163		P15_0			TXD7	IIO0_0	AN15_0	
164	VCC							
165		P10_7	$\overline{\text{KI3}}$				AN_7	
166		P10_6	$\overline{\text{KI2}}$				AN_6	
167		P10_5	$\overline{\text{KI1}}$				AN_5	
168		P10_4	$\overline{\text{KI0}}$				AN_4	
169		P10_3					AN_3	
170		P10_2					AN_2	
171		P10_1					AN_1	
172	AVSS							
173		P10_0					AN_0	
174	VREF							
175	AVCC							
176		P9_7			RXD4/SCL4/STXD4		$\overline{\text{ADTRG}}$	


(Note 1)
(Note 2)
P1_1 / D9 / IIO0_1 / IIO1_1
P1_2 / D10 / IIO0_2 / IIO1_2
P1_3 / D11 / IIO0_3 / IIO1_3
P1_4 / D12 / IIO0_4 / IIO1_4
P1_5 / D13 / INT3 / IIO0_5 / IIO1_5
P1_6 / D14 / INT4 / IIO0_6 / IIO1_6
P1_7 / D15 / INT5 / IIO0_7 / IIO1_7
P2_0 / A0 / [A0/D0] / BC0 / [BC0/D0] / AN2_0 / CTS9 / RTS9
P2_1 / A1 / [A1/D1] / BC2 / [BC2/D1] / AN2_1 / CLK9
P2_2 / A2 / [A2/D2] / AN2_2 / RXD9
P2_3 / A3 / [A3/D3] / AN2_3 / TXD9
P2_4 / A4 / [A4/D4] / AN2_4 / CTS10 / RTS10
P2_5 / A5 / [A5/D5] / AN2_5 / CLK10
P2_6 / A6 / [A6/D6] / AN2_6 / RXD10
P2_7 / A7 / [A7/D7] / AN2_7 / TXD10
VSS
P3_0 / A8 / [A8/D8] / TA0OUT / UD0A / UD1A
VCC
P12_0 / D16 / TXD6 / SDA6 / SRXD6
P12_1 / D17 / CLK6
P12_2 / D18 / RXD6 / SCL6 / STXD6
P12_3 / D19 / CTS6 / RTS6 / SS6
P12_4 / D20
P3_1 / A9 / [A9/D9] / TA3OUT / UD0B / UD1B
P3_2 / A10 / [A10/D10] / TA1OUT / V
P3_3 / A11 / [A11/D11] / TA1IN / V
P3_4 / A12 / [A12/D12] / TA2OUT / W
P3_5 / A13 / [A13/D13] / TA2IN / W
P3_6 / A14 / [A14/D14] / TA4OUT / U
P3_7 / A15 / [A15/D15] / TA4IN / U
P4_0 / A16 / CTS3 / RTS3 / SS3
P4_1 / A17 / CLK3
P11_5
P4_2 / A18 / RXD3 / SCL3 / STXD3 / ISRXD2 / IEIN
P11_6
P4_3 / A19 / TXD3 / SDA3 / SRXD3 / OUTC2_0 / ISTXD2 / IEOUT
108
107
106
105
104
103
102
101
100
99
98
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
IIO0_0 / IIO1_0 / D8 / P1_0 109
AN0_7 / D7 / P0_7 110
AN0_6 / D6 / P0_6 111
AN0_5 / D5 / P0_5 112
AN0_4 / D4 / P0_4 113
WR3 / BC3 / P11_4 114
IIO1_3 / CTS8 / RTS8 / WR2 / CS3 / P11_3 115
IIO1_2 / RXD8 / CS2 / P11_2 116
IIO1_1 / CLK8 / CS1 / P11_1 117
IIO1_0 / TXD8 / CS0 / P11_0 118
AN0_3 / D3 / P0_3 119
AN0_2 / D2 / P0_2 120
AN0_1 / D1 / P0_1 121
AN0_0 / D0 / P0_0 122
IIO0_7 / CTS6 / RTS6 / SS6 / AN15_7 / P15_7 123
IIO0_6 / CLK6 / AN15_6 / P15_6 124
IIO0_5 / RXD6 / SCL6 / STXD6 / AN15_5 / P15_5 125
IIO0_4 / TXD6 / SDA6 / SRXD6 / AN15_4 / P15_4 126
IIO0_3 / CTS7 / RTS7 / AN15_3 / P15_3 127
IIO0_2 / RXD7 / AN15_2 / P15_2 128
IIO0_1 / CLK7 / AN15_1 / P15_1 129
VSS 130
IIO0_0 / TXD7 / AN15_0 / P15_0 131
VCC 132
KI3 / AN_7 / P10_7 133
KI2 / AN_6 / P10_6 134
KI1 / AN_5 / P10_5 135
KI0 / AN_4 / P10_4 136
AN_3 / P10_3 137
AN_2 / P10_2 138
AN_1 / P10_1 139
AVSS 140
AN_0 / P10_0 141
VREF 142
AVCC 143
RXD4 / SCL4 / STXD4 / ADTRG / P9_7 144
R32C/116A GROUP
PLQP0144KA-A
(144P6Q-A)
(Top view)
72 P4_4 / CS3 / A20 / CTS6 / RTS6 / SS6
71 P4_5 / CS2 / A21 / CLK6
70 P4_6 / CS1 / A22 / RXD6 / SCL6 / STXD6
69 P4_7 / CS0 / A23 / TXD6 / SDA6 / SRXD6
68 P12_5 / D21
67 P12_6 / D22
66 P12_7 / D23
65 P5_0 / WR0 / WR
64 P5_1 / WR1 / BC1
63 P5_2 / RD
62 P5_3 / CLKOUT / BCLK
61 P13_0 / D24 / OUTC2_4
60 P13_1 / D25 / OUTC2_5
59 VCC
58 P13_2 / D26 / OUTC2_6
57 VSS
56 P13_3 / D27 / OUTC2_3
55 P5_4 / HLDA / CS1 / TXD7
54 P5_5 / HOLD / CLK7
53 P5_6 / ALE / CS2 / RXD7
52 P5_7 / RDY / CS3 / CTS7 / RTS7
51 P13_4 / D28 / OUTC2_0 / ISTXD2 / IEOUT
50 P13_5 / D29 / OUTC2_2 / ISRXD2 / IEIN
49 P13_6 / D30 / OUTC2_1 / ISCLK2
48 P13_7 / D31 / OUTC2_7
47 P6_0 / TB0IN / CTS0 / RTS0 / SS0
46 P6_1 / TB1IN / CLK0
45 P6_2 / TB2IN / RXD0 / SCL0 / STXD0
44 P6_3 / TXD0 / SDA0 / SRXD0
43 P6_4 / CTS1 / RTS1 / SS1 / OUTC2_1 / ISCLK2
42 P6_5 / CLK1
41 P11_7
40 P6_6 / RXD1 / SCL1 / STXD1
39 P14_7
38 P6_7 / TXD1 / SDA1 / SRXD1
37 P7_0 / TA0OUT / TXD2 / SDA2 / SRXD2 / IIO1_6 / OUTC2_0 / ISTXD2 / IEOUT / MSDA
(Note 2)
(Note 3)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
TXD4 / SDA4 / SRXD4 / ANEX1 / P9_6
CLK4 / ANEX0 / P9_5
CTS4 / RTS4 / SS4 / TB4IN / DA1 / P9_4
CTS3 / RTS3 / SS3 / TB3IN / DA0 / P9_3
OUTC2_0 / ISTXD2 / IEOUT / TXD3 / SDA3 / SRXD3 / TB2IN / P9_2
ISRXD2 / IEIN / RXD3 / SCL3 / STXD3 / TB1IN / P9_1
CLK3 / TB0IN / P9_0
INT8 / P14_6
INT7 / P14_5
INT6 / P14_4
P14_3
VDC0
P14_1
VDC1
NSD
CNVSS
XCIN / P8_7
XCOUT / P8_6
RESET
XOUT
VSS
XIN
VCC
NMI / P8_5
INT2 / P8_4
INT1 / P8_3
INT0 / P8_2
IIO1_5 / UD0B / UD1B / CTS5 / RTS5 / SS5 / U / TA4IN / P8_1
UD0A / UD1A / RXD5 / SCL5 / STXD5 / U / TA4OUT / P8_0
IIO1_4 / UD0B / UD1B / CLK5 / TA3IN / P7_7
IIO1_3 / UD0A / UD1A / CTS8 / RTS8 / TXD5 / SDA5 / SRXD5 / TA3OUT / P7_6
IIO1_2 / RXD8 / W / TA2IN / P7_5
IIO1_1 / CLK8 / W / TA2OUT / P7_4
IIO1_0 / TXD8 / CTS2 / RTS2 / SS2 / V / TA1IN / P7_3
CLK2 / V / TA1OUT / P7_2
MSCL / IIO1_7 / OUTC2_2 / ISRXD2 / IEIN / RXD2 / SCL2 / STXD2 / TA0IN / TB5IN / P7_1
(Note 2)

Notes:

1. Pin names in brackets [ ] represent a functional signal as a whole and should not be considered as two separate pins.
2. The following pins are 5 V tolerant inputs: P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_3, P12_0 to P12_7.
3. The position of pin number 1 varies by product. Refer to the index mark in attached “Package Dimensions”.

Figure 1.4 Pin Assignment for the 144-pin Package (top view)

Table 1.11 Pin Characteristics for the 144-pin Package (1/4)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
1		P9_6			TXD4/SDA4/SRXD4		ANEX1	
2		P9_5			CLK4		ANEX0	
3		P9_4		TB4IN	CTS4/RTS4/SS4		DA1	
4		P9_3		TB3IN	CTS3/RTS3/SS3		DA0	
5		P9_2		TB2IN	TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/IEOUT		
6		P9_1		TB1IN	RXD3/SCL3/STXD3	ISRXD2/IEIN		
7		P9_0		TB0IN	CLK3			
8		P14_6	INT8					
9		P14_5	INT7					
10		P14_4	INT6					
11		P14_3						
12	VDC0							
13		P14_1						
14	VDC1							
15	NSD							
16	CNVSS							
17	XCIN	P8_7						
18	XCOUT	P8_6						
19	RESET							
20	XOUT							
21	VSS							
22	XIN							
23	VCC							
24		P8_5	NMI					
25		P8_4	INT2					
26		P8_3	INT1					
27		P8_2	INT0					
28		P8_1		TA4IN/U	CTS5/RTS5/SS5	IIO1_5/UD0B/UD1B		
29		P8_0		TA4OUT/U	RXD5/SCL5/STXD5	UD0A/UD1A		
30		P7_7		TA3IN	CLK5	IIO1_4/UD0B/UD1B		
31		P7_6		TA3OUT	TXD5/SDA5/SRXD5/CTS8/RTS8	IIO1_3/UD0A/UD1A		
32		P7_5		TA2IN/W	RXD8	IIO1_2		
33		P7_4		TA2OUT/W	CLK8	IIO1_1		
34		P7_3		TA1IN/V	CTS2/RTS2/SS2/TXD8	IIO1_0		
35		P7_2		TA1OUT/V	CLK2			
36		P7_1		TA0IN/TB5IN	RXD2/SCL2/STXD2/MSCL	IIO1_7/OUTC2_2/ISRXD2/IEIN		

Table 1.12 Pin Characteristics for the 144-pin Package (2/4)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
37		P7_0		TA0OUT	TXD2/SDA2/SRXD2/ MSDA	IIO1_6/OUTC2_0/ ISTXD2/IEOUT		
38		P6_7			TXD1/SDA1/SRXD1			
39		P14_7						
40		P6_6			RXD1/SCL1/STXD1			
41		P11_7						
42		P6_5			CLK1			
43		P6_4			CTS1/RTS1/SS1	OUTC2_1/ISCLK2		
44		P6_3			TXD0/SDA0/SRXD0			
45		P6_2		TB2IN	RXD0/SCL0/STXD0			
46		P6_1		TB1IN	CLK0			
47		P6_0		TB0IN	CTS0/RTS0/SS0			
48		P13_7				OUTC2_7		D31
49		P13_6				OUTC2_1/ISCLK2		D30
50		P13_5				OUTC2_2/ISRXD2/ IEIN		D29
51		P13_4				OUTC2_0/ISTXD2/ IEOUT		D28
52		P5_7			CTS7/RTS7			RDY/CS3
53		P5_6			RXD7			ALE/CS2
54		P5_5			CLK7			HOLD
55		P5_4			TXD7			HLDA/CS1
56		P13_3				OUTC2_3		D27
57	VSS							
58		P13_2				OUTC2_6		D26
59	VCC							
60		P13_1				OUTC2_5		D25
61		P13_0				OUTC2_4		D24
62		P5_3						CLKOUT/ BCLK
63		P5_2						RD
64		P5_1						WR1/BC1
65		P5_0						WR0/WR
66		P12_7						D23
67		P12_6						D22
68		P12_5						D21
69		P4_7			TXD6/SDA6/SRXD6			CS0/A23
70		P4_6			RXD6/SCL6/STXD6			CS1/A22
71		P4_5			CLK6			CS2/A21
72		P4_4			CTS6/RTS6/SS6			CS3/A20
73		P4_3			TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/ IEOUT		A19
74		P11_6						

Table 1.13 Pin Characteristics for the 144-pin Package (3/4)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
75		P4_2			RXD3/SCL3/STXD3	ISRXD2/IEIN		A18
76		P11_5						
77		P4_1			CLK3			A17
78		P4_0			$\overline{\mathrm{CTS3}}$/$\overline{\mathrm{RTS3}}$/$\overline{\mathrm{SS3}}$			A16
79		P3_7		TA4IN/$\overline{\mathrm{U}}$				A15(/D15)
80		P3_6		TA4OUT/U				A14(/D14)
81		P3_5		TA2IN/$\overline{\mathrm{W}}$				A13(/D13)
82		P3_4		TA2OUT/W				A12(/D12)
83		P3_3		TA1IN/$\overline{\mathrm{V}}$				A11(/D11)
84		P3_2		TA1OUT/V				A10(/D10)
85		P3_1		TA3OUT		UD0B/UD1B		A9(/D9)
86		P12_4						D20
87		P12_3			$\overline{\mathrm{CTS6}}$/$\overline{\mathrm{RTS6}}$/$\overline{\mathrm{SS6}}$			D19
88		P12_2			RXD6/SCL6/STXD6			D18
89		P12_1			CLK6			D17
90		P12_0			TXD6/SDA6/SRXD6			D16
91	VCC							
92		P3_0		TA0OUT		UD0A/UD1A		A8(/D8)
93	VSS							
94		P2_7			TXD10		AN2_7	A7(/D7)
95		P2_6			RXD10		AN2_6	A6(/D6)
96		P2_5			CLK10		AN2_5	A5(/D5)
97		P2_4			$\overline{\mathrm{CTS10}}$/$\overline{\mathrm{RTS10}}$		AN2_4	A4(/D4)
98		P2_3			TXD9		AN2_3	A3(/D3)
99		P2_2			RXD9		AN2_2	A2(/D2)
100		P2_1			CLK9		AN2_1	A1(/D1)/$\overline{\mathrm{BC2}}$(/D1)
101		P2_0			$\overline{\mathrm{CTS9}}$/$\overline{\mathrm{RTS9}}$		AN2_0	A0(/D0)/$\overline{\mathrm{BC0}}$(/D0)
102		P1_7	$\overline{\mathrm{INT5}}$			IIO0_7/IIO1_7		D15
103		P1_6	$\overline{\mathrm{INT4}}$			IIO0_6/IIO1_6		D14
104		P1_5	$\overline{\mathrm{INT3}}$			IIO0_5/IIO1_5		D13
105		P1_4				IIO0_4/IIO1_4		D12
106		P1_3				IIO0_3/IIO1_3		D11
107		P1_2				IIO0_2/IIO1_2		D10
108		P1_1				IIO0_1/IIO1_1		D9
109		P1_0				IIO0_0/IIO1_0		D8
110		P0_7					AN0_7	D7
111		P0_6					AN0_6	D6

Table 1.14 Pin Characteristics for the 144-pin Package (4/4)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
112		P0_5					AN0_5	D5
113		P0_4					AN0_4	D4
114		P11_4						$\overline{\mathrm{BC3}}$/$\overline{\mathrm{WR3}}$
115		P11_3			$\overline{\mathrm{CTS8}}$/$\overline{\mathrm{RTS8}}$	IIO1_3		$\overline{\mathrm{CS3}}$/$\overline{\mathrm{WR2}}$
116		P11_2			RXD8	IIO1_2		CS2
117		P11_1			CLK8	IIO1_1		CS1
118		P11_0			TXD8	IIO1_0		CS0
119		P0_3					AN0_3	D3
120		P0_2					AN0_2	D2
121		P0_1					AN0_1	D1
122		P0_0					AN0_0	D0
123		P15_7			$\overline{\mathrm{CTS6}}$/$\overline{\mathrm{RTS6}}$/$\overline{\mathrm{SS6}}$	IIO0_7	AN15_7	
124		P15_6			CLK6	IIO0_6	AN15_6	
125		P15_5			RXD6/SCL6/STXD6	IIO0_5	AN15_5	
126		P15_4			TXD6/SDA6/SRXD6	IIO0_4	AN15_4	
127		P15_3			$\overline{\mathrm{CTS7}}$/$\overline{\mathrm{RTS7}}$	IIO0_3	AN15_3	
128		P15_2			RXD7	IIO0_2	AN15_2	
129		P15_1			CLK7	IIO0_1	AN15_1	
130	VSS							
131		P15_0			TXD7	IIO0_0	AN15_0	
132	VCC							
133		P10_7	$\overline{\mathrm{KI3}}$				AN_7	
134		P10_6	$\overline{\mathrm{KI2}}$				AN_6	
135		P10_5	$\overline{\mathrm{KI1}}$				AN_5	
136		P10_4	$\overline{\mathrm{KI0}}$				AN_4	
137		P10_3					AN_3	
138		P10_2					AN_2	
139		P10_1					AN_1	
140	AVSS							
141		P10_0					AN_0	
142	VREF							
143	AVCC							
144		P9_7			RXD4/SCL4/STXD4		$\overline{\mathrm{ADTRG}}$	

1.5 Pin Definitions and Functions

Tables 1.15 to 1.19 show the pin definitions and functions.

Table 1.15 Pin Definitions and Functions (1/4)

Function	Symbol	I/O	Description
Power supply	VCC, VSS	I	Applicable as follows: VCC = 3.0 to 5.5 V, VSS = 0 V
Connecting pins for decoupling capacitor	VDC0, VDC1	—	A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1
Analog power supply	AVCC, AVSS	I	Power supply for the A/D converter. AVCC and AVSS should be connected to VCC and VSS, respectively
Reset input	$\overline{\text{RESET}}$	I	The MCU is reset when this pin is driven low
CNVSS	CNVSS	I	This pin should be connected to VSS via a resistor
Debug port	NSD	I/O	This pin is to communicate with a debugger. It should be connected to VCC via a resistor of 1 to 4.7 kΩ
Main clock input	XIN	I	Input/output for the main clock oscillator. A crystal, or a ceramic resonator should be connected between pins XIN and XOUT. An external clock should be input at the XIN while leaving the XOUT open
Main clock output	XOUT	O	
Sub clock input	XCIN	I	Input/output for the sub clock oscillator. A crystal oscillator should be connected between pins XCIN and XCOUT. An external clock should be input at the XCIN while leaving the XCOUT open
Sub clock output	XCOUT	O	
BCLK output	BCLK	O	BCLK output
Clock output	CLKOUT	O	Output of the clock with the same frequency as low speed clocks, f8, or f32
External interrupt input	$\overline{\text{INT0}}$ to $\overline{\text{INT8}}$	I	Input for external interrupts
NMI input	P8_5/$\overline{\text{NMI}}$	I	Input for NMI
Key input interrupt	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	Input for the key input interrupt
Bus control pins	D0 to D7	I/O	Input/output of data (D0 to D7) while accessing an external memory space with a separate bus
	D8 to D15	I/O	Input/output of data (D8 to D15) while accessing an external memory space with 16-bit or 32-bit separate bus
	D16 to D31	I/O	Input/output of data (D16 to D31) while accessing an external memory space with 32-bit separate bus
	A0 to A23	O	Output of address bits A0 to A23
	A0/D0 to A7/D7	I/O	Output of address bits (A0 to A7) and input/output of data (D0 to D7) by time-division while accessing an external memory space with multiplexed bus
	A8/D8 to A15/D15	I/O	Output of address bits (A8 to A15) and input/output of data (D8 to D15) by time-division while accessing an external memory space with 16-bit or 32-bit multiplexed bus

Table 1.16 Pin Definitions and Functions (2/4)

Function	Symbol	I/O	Description
Bus control pins	$\overline{\text{BC0}}$/D0, $\overline{\text{BC2}}$/D1	I/O	Output of byte control ($\overline{\text{BC0}}$ and $\overline{\text{BC2}}$) and input/output of data (D0 and D1) by time-division while accessing an external memory space with multiplexed bus
	$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	O	Chip select output
	$\overline{\text{WR0}}$/$\overline{\text{WR1}}$/$\overline{\text{WR2}}$/ $\overline{\text{WR3}}$, $\overline{\text{WR}}$/$\overline{\text{BC0}}$/$\overline{\text{BC1}}$/ $\overline{\text{BC2}}$/$\overline{\text{BC3}}$, $\overline{\text{RD}}$	O	Output of write, byte control, and read signals. Either $\overline{\text{WRx}}$ or $\overline{\text{WR}}$ and $\overline{\text{BCx}}$ can be selected by a program. Data is read when $\overline{\text{RD}}$ is low. • When $\overline{\text{WR0}}$, $\overline{\text{WR1}}$, $\overline{\text{WR2}}$, $\overline{\text{WR3}}$, and $\overline{\text{RD}}$ are selected, data is written to the following address: 4n+0, when $\overline{\text{WR0}}$ is low 4n+1, when $\overline{\text{WR1}}$ is low 4n+2, when $\overline{\text{WR2}}$ is low 4n+3, when $\overline{\text{WR3}}$ is low on 32-bit external data bus or an even address, when $\overline{\text{WR0}}$ is low an odd address, when $\overline{\text{WR1}}$ is low on 16-bit external data bus • When $\overline{\text{WR}}$, $\overline{\text{BC0}}$, $\overline{\text{BC1}}$, $\overline{\text{BC2}}$, $\overline{\text{BC3}}$, and $\overline{\text{RD}}$ are selected, data is written, when $\overline{\text{WR}}$ is low and the following address is accessed: 4n+0, when $\overline{\text{BC0}}$ is low 4n+1, when $\overline{\text{BC1}}$ is low 4n+2, when $\overline{\text{BC2}}$ is low 4n+3, when $\overline{\text{BC3}}$ is low on 32-bit external data bus or an even address, when $\overline{\text{BC0}}$ is low an odd address, when $\overline{\text{BC1}}$ is low on 16-bit external data bus
	ALE	O	Latch enable signal in multiplexed bus format
	$\overline{\text{HOLD}}$	I	The MCU is in a hold state while this pin is held low
	$\overline{\text{HLDA}}$	O	This pin is driven low while the MCU is held in a hold state
	$\overline{\text{RDY}}$	I	Bus cycle is extended by the CPU if this pin is low on the falling edge of BCLK

Table 1.17 Pin Definitions and Functions (3/4)

Function	Symbol	I/O	Description
I/O port [1]	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7	I/O	I/O ports in CMOS. Each port can be programmed to input or output under the control of the direction register. Some ports are 5 V tolerant inputs. Pull-up resistors and N-channel open drain setting can be enabled on some ports. Refer to Table 1.19 “Pin Specifications” for details
Input port	P14_1	I	Input port in CMOS Pull-up resistor is selectable. Refer to Table 1.19 “Pin Specifications” for details
Timer A	TA0OUT to TA4OUT	I/O	Timers A0 to A4 input/output
	TA0IN to TA4IN	I	Timers A0 to A4 input
Timer B	TB0IN to TB5IN	I	Timers B0 to B5 input
Three-phase motor control timer output	U, $\overline{\text{U}}$, V, $\overline{\text{V}}$, W, $\overline{\text{W}}$	O	Three-phase motor control timer output
Serial interface	$\overline{\text{CTS0}}$ to $\overline{\text{CTS10}}$	I	Handshake input
	$\overline{\text{RTS0}}$ to $\overline{\text{RTS10}}$	O	Handshake output
	CLK0 to CLK10	I/O	Transmit/receive clock input/output
	RXD0 to RXD10	I	Serial data input
	TXD0 to TXD10	O	Serial data output
I^{2}C-bus (simplified)	SDA0 to SDA6	I/O	Serial data input/output
	SCL0 to SCL6	I/O	Transmit/receive clock input/output
Serial interface special functions	STXD0 to STXD6	O	Serial data output in slave mode
	SRXD0 to SRXD6	I	Serial data input in slave mode
	$\overline{\text{SS0}}$ to $\overline{\text{SS6}}$	I	Input to control serial interface special functions

Note:

1. Ports P16 to P19 are available in the 176-pin package only.

Table 1.18 Pin Definitions and Functions (4/4)

Function	Symbol	I/O	Description
A/D converter	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7	I	Analog input for the A/D converter
	$\overline{\text{ADTRG}}$	I	External trigger input for the A/D converter
	ANEX0	I/O	Expanded analog input for the A/D converter and output in external op-amp connection mode
	ANEX1	I	Expanded analog input for the A/D converter
D/A converter	DA0, DA1	O	Output for the D/A converter
Reference voltage input	VREF	I	Reference voltage input for the A/D converter and D/A converter
Intelligent I/O	IIO0_0 to IIO0_7	I/O	Input/output for Intelligent I/O group 0. Either input capture or output compare is selectable
	IIO1_0 to IIO1_7	I/O	Input/output for Intelligent I/O group 1. Either input capture or output compare is selectable
	UD0A, UD0B, UD1A, UD1B	I	Input for the two-phase encoder
	OUTC2_0 to OUTC2_7	O	Output for OC (output compare) of Intelligent I/O group 2
	ISCLK2	I/O	Clock input/output for the serial interface
	ISRXD2	I	Receive data input for the serial interface
	ISTXD2	O	Transmit data output for the serial interface
	IEIN	I	Receive data input for the serial interface
	IEOUT	O	Transmit data output for the serial interface
Multi-master I^{2}C-bus	MSDA	I/O	Serial data input/output
	MSCL	I/O	Transmit/receive clock input/output

Table 1.19 Pin Specifications

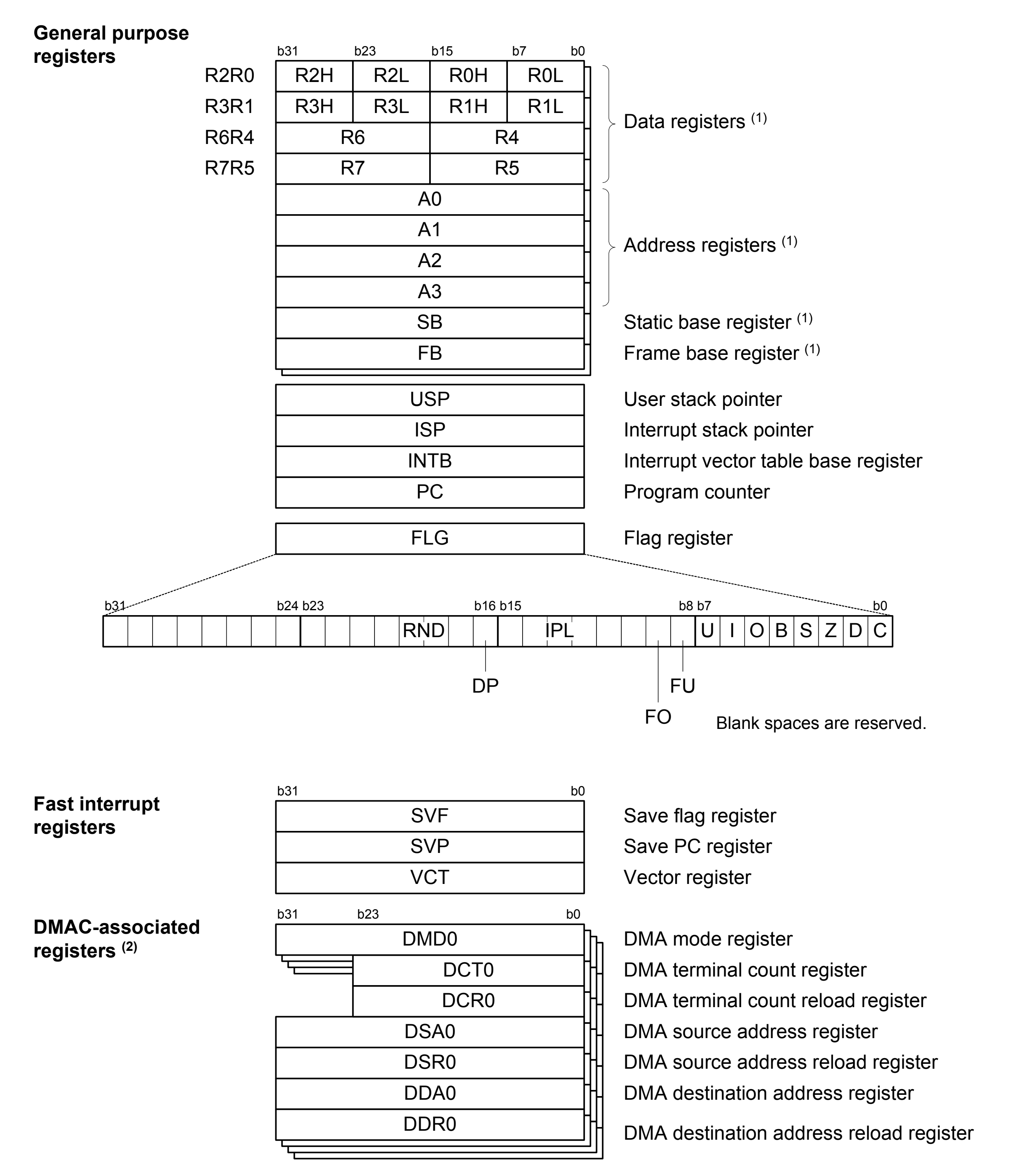
Pin names	Package		Selectable Functions		5 V tolerant input (3)
	176-pin	144-pin	Pull-up resistor (1)	N-channel open drain (2)	
P0_0 to P0_7	✓	✓	✓		
P1_0 to P1_7	✓	✓	✓		
P2_0 to P2_7	✓	✓	✓	✓	
P3_0 to P3_7	✓	✓	✓		
P4_0 to P4_7	✓	✓		✓	✓
P5_0 to P5_3	✓	✓	✓		
P5_4 to P5_7	✓	✓		✓	✓
P6_0 to P6_7	✓	✓		✓	✓
P7_0 to P7_7	✓	✓		✓	✓
P8_0 to P8_3	✓	✓		✓	✓
P8_4, P8_6, P8_7	✓	✓	✓		
P9_0 to P9_7	✓	✓	✓	✓	
P10_0 to P10_7	✓	✓	✓		
P11_0 to P11_3	✓	✓	✓	✓	
P11_4 to P11_7	✓	✓	✓		
P12_0 to P12_7	✓	✓		✓	✓
P13_0 to P13_7	✓	✓	✓		
P14_1, P14_3	✓	✓	✓		
P14_4 to P14_7	✓	✓	✓		
P15_0 to P15_7	✓	✓	✓	✓	
P16_0 to P16_7	✓			✓	✓
P17_0 to P17_3	✓			✓	✓
P17_4 to P17_7	✓		✓		
P18_0 to P18_7	✓		✓		
P19_0 to P19_7	✓		✓		

Notes:

1. Pull-up resistors are selected for the following 4-pin units, but are enabled only for the input pins: Pi_0 to Pi_3 and Pi_4 to Pi_7 (i = 0 to 19).
2. N-channel open drain output can be enabled on the applicable pins on a discrete pin basis.
3. 5 V tolerant input is enabled when an applicable pin is set as an input port. When it is set as an I/O port, to enable 5 V tolerant input, this pin should be set as N-channel open drain output.

2. Central Processing Unit (CPU)

The CPU contains the registers shown below. There are two register banks each consisting of registers R2R0, R3R1, R6R4, R7R5, A0 to A3, SB, and FB.



Notes:

1. There are two banks of these registers.
2. There are four identical sets of DMAC-associated registers.

Figure 2.1 CPU Registers

2.1 General Purpose Registers

2.1.1 Data Registers (R2R0, R3R1, R6R4, and R7R5)

These 32-bit registers are primarily used for transfers and arithmetic/logic operations.
Each of the registers can be divided into upper and lower 16-bit registers, e.g. R2R0 can be divided into R2 and R0, R3R1 can be divided into R3 and R1, etc.
Moreover, data registers R2R0 and R3R1 can be divided into four 8-bit data registers: upper (R2H and R3H), mid-upper (R2L and R3L), mid-lower (R0H and R1H), and lower (R0L and R1L).

2.1.2 Address Registers (A0, A1, A2, and A3)

These 32-bit registers have functions similar to data registers. They are also used for address register indirect addressing and address register relative addressing.

2.1.3 Static Base Register (SB)

This 32-bit register is used for SB relative addressing.

2.1.4 Frame Base Register (FB)

This 32-bit register is used for FB relative addressing.

2.1.5 Program Counter (PC)

This 32-bit counter indicates the address of the instruction to be executed next.

2.1.6 Interrupt Vector Table Base Register (INTB)

This 32-bit register indicates the start address of a relocatable vector table.

2.1.7 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP)

Two types of 32-bit stack pointers (SPs) are provided: user stack pointer (USP) and interrupt stack pointer (ISP).
Use the stack pointer select flag (U flag) to select either the user stack pointer (USP) or the interrupt stack pointer (ISP). The U flag is bit 7 in the flag register (FLG). Refer to 2.1.8 “Flag Register (FLG)” for details.
To minimize the overhead of interrupt sequence due to less memory access, set the user stack pointer (USP) or the interrupt stack pointer (ISP) to a multiple of 4.

2.1.8 Flag Register (FLG)

This 32-bit register indicates the CPU status.

2.1.8.1 Carry Flag (C flag)

This flag retains a carry, borrow, or shifted-out bit generated by the arithmetic logic unit (ALU).

2.1.8.2 Debug Flag (D flag)

This flag is only for debugging. Only set this bit to 0.

2.1.8.3 Zero Flag (Z flag)

This flag becomes 1 when the result of an operation is 0; otherwise it is 0.

2.1.8.4 Sign Flag (S flag)

This flag becomes 1 when the result of an operation is a negative value; otherwise it is 0.

2.1.8.5 Register Bank Select Flag (B flag)

This flag selects a register bank. It indicates 0 when register bank 0 is selected, and 1 when register bank 1 is selected.

2.1.8.6 Overflow Flag (O flag)

This flag becomes 1 when the result of an operation overflows; otherwise it is 0.

2.1.8.7 Interrupt Enable Flag (I flag)

This flag enables maskable interrupts. To disable maskable interrupts, set this flag to 0. To enable them, set this flag to 1. When an interrupt is accepted, the flag becomes 0.

2.1.8.8 Stack Pointer Select Flag (U flag)

To select the interrupt stack pointer (ISP), set this flag to 0. To select the user stack pointer (USP), set this flag to 1.

It becomes 0 when a hardware interrupt is accepted or when an INT instruction designated by a software interrupt number from 0 to 127 is executed.

2.1.8.9 Floating-point Underflow Flag (FU flag)

This flag becomes 1 when an underflow occurs in a floating-point operation; otherwise it is 0. It also becomes 1 when the operand contains invalid numbers (subnormal numbers).

2.1.8.10 Floating-point Overflow Flag (FO flag)

This flag becomes 1 when an overflow occurs in a floating-point operation; otherwise it is 0. It also becomes 1 when the operand contains invalid numbers (subnormal numbers).

2.1.8.11 Processor Interrupt Priority Level (IPL)

The processor interrupt priority level (IPL), consisting of 3 bits, selects a processor interrupt priority level from level 0 to 7. An interrupt is enabled when the interrupt request level is higher than the selected IPL.

When the processor interrupt priority level (IPL) is set to 111b (level 7), all interrupts are disabled.

2.1.8.12 Fixed-point Radix Point Designation Bit (DP bit)

This bit designates the radix point. It also specifies which portion of the fixed-point multiplication result to extract. It is used for the MULX instruction.

2.1.8.13 Floating-point Rounding Mode (RND)

The 2-bit floating-point rounding mode selects a rounding mode for floating-point calculation results.

2.1.8.14 Reserved

Only set this bit to 0. The read value is undefined.

2.2 Fast Interrupt Registers

The following three registers are provided to minimize the overhead of the interrupt sequence. Refer to 11.4 “Fast Interrupt” for details.

2.2.1 Save Flag Register (SVF)

This 32-bit register is used to save the flag register when a fast interrupt occurs.

2.2.2 Save PC Register (SVP)

This 32-bit register is used to save the program counter when a fast interrupt occurs.

2.2.3 Vector Register (VCT)

This 32-bit register is used to indicate a jump address when a fast interrupt occurs.

2.3 DMAC-associated Registers

There are seven types of DMAC-associated registers. Refer to 13. “DMAC” for details.

2.3.1 DMA Mode Registers (DMD0, DMD1, DMD2, and DMD3)

These 32-bit registers are used to set DMA transfer mode, bit rate, etc.

2.3.2 DMA Terminal Count Registers (DCT0, DCT1, DCT2, and DCT3)

These 24-bit registers are used to set the number of DMA transfers.

2.3.3 DMA Terminal Count Reload Registers (DCR0, DCR1, DCR2, and DCR3)

These 24-bit registers are used to set the reloaded values for DMA terminal count registers.

2.3.4 DMA Source Address Registers (DSA0, DSA1, DSA2, and DSA3)

These 32-bit registers are used to set DMA source addresses.

2.3.5 DMA Source Address Reload Registers (DSR0, DSR1, DSR2, and DSR3)

These 32-bit registers are used to set the reloaded values for DMA source address registers.

2.3.6 DMA Destination Address Registers (DDA0, DDA1, DDA2, and DDA3)

These 32-bit registers are used to set DMA destination addresses.

2.3.7 DMA Destination Address Reload Registers (DDR0, DDR1, DDR2, and DDR3)

These 32-bit registers are used to set reloaded values for DMA destination address registers.

3. Memory

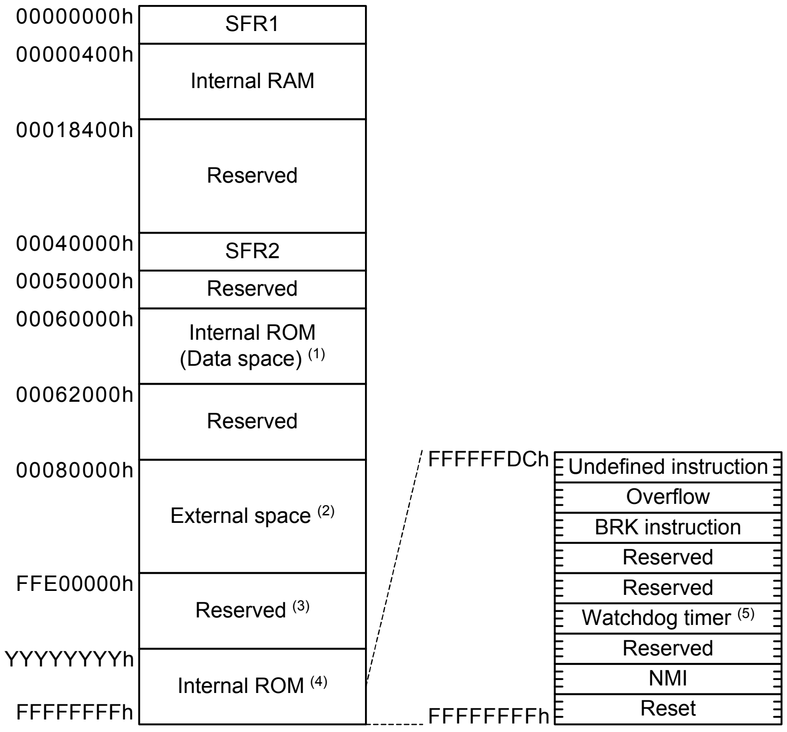
Figure 3.1 shows the memory map of the R32C/116A Group.
The R32C/116A Group provides a 4-Gbyte address space from 00000000h to FFFFFFFFh.

The internal ROM is mapped from address FFFFFFFFh in the inferior direction. For example, the 1-Mbyte internal ROM is mapped from FFF00000h to FFFFFFFFh.
The fixed interrupt vector table contains the start address of interrupt handlers and is mapped from FFFFFFDCh to FFFFFFFFh.
The internal RAM is mapped from address 00000400h in the superior direction. For example, the 96-Kbyte internal RAM is mapped from 00000400h to 000183FFh. Besides being used for data storage, the internal RAM functions as a stack(s) for subroutine calls and/or interrupt handlers.

Special function registers (SFRs), which are control registers for peripheral functions, are mapped from 00000000h to 000003FFh, and from 00040000h to 0004FFFFh. Unoccupied SFR locations are reserved, and no access is allowed.

In memory expansion mode or microprocessor mode, some spaces are reserved for internal use and should not be accessed.

Internal ROM	
Capacity	YYYYYYYYh
512 Kbytes	FFF80000h
640 Kbytes	FFF60000h
768 Kbytes	FFF40000h
1 Mbyte	FFF00000h



Notes:
1. The flash memory version provides two additional 4-Kbyte spaces (blocks A and B) for storing data.
2. This space can be used in memory expansion mode or microprocessor mode. Addresses from 02000000h to FDFFFFFFh are inaccessible.
3. This space is reserved in memory expansion mode. It becomes an external space in microprocessor mode.
4. This space can be used in single-chip mode or memory expansion mode. It becomes an external space in microprocessor mode.
5. The watchdog timer interrupt shares a vector with the oscillator stop detection interrupt and low voltage detection interrupt.

Figure 3.1 Memory Map

4. Special Function Registers (SFRs)

SFRs are memory-mapped peripheral registers that control the operation of peripherals. Table 4.1 SFR List (1) to Table 4.27 SFR List (27) list the SFR details.

Table 4.1 SFR List (1)

Address	Register	Symbol	Reset Value
000000h			
000001h			
000002h			
000003h			
000004h	Clock Control Register	CCR	0001 1000b
000005h			
000006h	Flash Memory Control Register	FMCR	0000 0001b
000007h	Protect Release Register	PRR	00h
000008h	Flash Memory Rewrite Bus Control Register	FEBC	0000h
000009h			
00000Ah			
00000Bh			
00000Ch			
00000Dh			
00000Eh			
00000Fh			
000010h	External Bus Control Register 3	EBC3	0000h
000011h			
000012h	Chip Selects 2 and 3 Boundary Setting Register	CB23	00h
000013h			
000014h	External Bus Control Register 2	EBC2	0000h
000015h			
000016h	Chip Selects 1 and 2 Boundary Setting Register	CB12	00h
000017h			
000018h	External Bus Control Register 1	EBC1	0000h
000019h			
00001Ah	Chip Selects 0 and 1 Boundary Setting Register	CB01	00h
00001Bh			
00001Ch	External Bus Control Register 0	EBC0	0000h
00001Dh			
00001Eh	Peripheral Bus Control Register	PBC	0504h
00001Fh			
000020h to 00005Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.2 SFR List (2)

Address	Register	Symbol	Reset Value
000060h			
000061h	Timer B5 Interrupt Control Register	TB5IC	XXXX X000b
000062h	UART5 Transmit/NACK Interrupt Control Register	S5TIC	XXXX X000b
000063h	UART2 Receive/ACK Interrupt Control Register/I^{2}C-bus Line Interrupt Control Register	S2RIC/I2CLIC	XXXX X000b
000064h	UART6 Transmit/NACK Interrupt Control Register	S6TIC	XXXX X000b
000065h	UART3 Receive/ACK Interrupt Control Register	S3RIC	XXXX X000b
000066h	UART5/6 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register	BCN5IC/BCN6IC	XXXX X000b
000067h	UART4 Receive/ACK Interrupt Control Register	S4RIC	XXXX X000b
000068h	DMA0 Transfer Complete Interrupt Control Register	DM0IC	XXXX X000b
000069h	UART0/3 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register	BCN0IC/BCN3IC	XXXX X000b
00006Ah	DMA2 Transfer Complete Interrupt Control Register	DM2IC	XXXX X000b
00006Bh	A/D Converter 0 Convert Completion Interrupt Control Register	AD0IC	XXXX X000b
00006Ch	Timer A0 Interrupt Control Register	TA0IC	XXXX X000b
00006Dh	Intelligent I/O Interrupt Control Register 0	IIO0IC	XXXX X000b
00006Eh	Timer A2 Interrupt Control Register	TA2IC	XXXX X000b
00006Fh	Intelligent I/O Interrupt Control Register 2	IIO2IC	XXXX X000b
000070h	Timer A4 Interrupt Control Register	TA4IC	XXXX X000b
000071h	Intelligent I/O Interrupt Control Register 4	IIO4IC	XXXX X000b
000072h	UART0 Receive/ACK Interrupt Control Register	S0RIC	XXXX X000b
000073h	Intelligent I/O Interrupt Control Register 6	IIO6IC	XXXX X000b
000074h	UART1 Receive/ACK Interrupt Control Register	S1RIC	XXXX X000b
000075h	Intelligent I/O Interrupt Control Register 8	IIO8IC	XXXX X000b
000076h	Timer B1 Interrupt Control Register	TB1IC	XXXX X000b
000077h	Intelligent I/O Interrupt Control Register 10	IIO10IC	XXXX X000b
000078h	Timer B3 Interrupt Control Register	TB3IC	XXXX X000b
000079h			
00007Ah	INT5 Interrupt Control Register	INT5IC	XX00 X000b
00007Bh			
00007Ch	INT3 Interrupt Control Register	INT3IC	XX00 X000b
00007Dh			
00007Eh	INT1 Interrupt Control Register	INT1IC	XX00 X000b
00007Fh			
000080h			
000081h	UART2 Transmit/NACK Interrupt Control Register/I^{2}C-bus Interrupt Control Register	S2TIC/I2CIC	XXXX X000b
000082h	UART5 Receive/ACK Interrupt Control Register	S5RIC	XXXX X000b
000083h	UART3 Transmit/NACK Interrupt Control Register	S3TIC	XXXX X000b
000084h	UART6 Receive/ACK Interrupt Control Register	S6RIC	XXXX X000b
000085h	UART4 Transmit/NACK Interrupt Control Register	S4TIC	XXXX X000b
000086h			
000087h	UART2 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register	BCN2IC	XXXX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.3 SFR List (3)

Address	Register	Symbol	Reset Value
000088h	DMA1 Transfer Complete Interrupt Control Register	DM1IC	XXXX X000b
000089h	UART1/4 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register	BCN1IC/BCN4IC	XXXX X000b
00008Ah	DMA3 Transfer Complete Interrupt Control Register	DM3IC	XXXX X000b
00008Bh	Key Input Interrupt Control Register	KUPIC	XXXX X000b
00008Ch	Timer A1 Interrupt Control Register	TA1IC	XXXX X000b
00008Dh	Intelligent I/O Interrupt Control Register 1	IIO1IC	XXXX X000b
00008Eh	Timer A3 Interrupt Control Register	TA3IC	XXXX X000b
00008Fh	Intelligent I/O Interrupt Control Register 3	IIO3IC	XXXX X000b
000090h	UART0 Transmit/NACK Interrupt Control Register	S0TIC	XXXX X000b
000091h	Intelligent I/O Interrupt Control Register 5	IIO5IC	XXXX X000b
000092h	UART1 Transmit/NACK Interrupt Control Register	S1TIC	XXXX X000b
000093h	Intelligent I/O Interrupt Control Register 7	IIO7IC	XXXX X000b
000094h	Timer B0 Interrupt Control Register	TB0IC	XXXX X000b
000095h	Intelligent I/O Interrupt Control Register 9	IIO9IC	XXXX X000b
000096h	Timer B2 Interrupt Control Register	TB2IC	XXXX X000b
000097h	Intelligent I/O Interrupt Control Register 11	IIO11IC	XXXX X000b
000098h	Timer B4 Interrupt Control Register	TB4IC	XXXX X000b
000099h			
00009Ah	INT4 Interrupt Control Register	INT4IC	XX00 X000b
00009Bh			
00009Ch	INT2 Interrupt Control Register	INT2IC	XX00 X000b
00009Dh			
00009Eh	INT0 Interrupt Control Register	INT0IC	XX00 X000b
00009Fh			
0000A0h	Intelligent I/O Interrupt Request Register 0	IIO0IR	0000 0XX1b
0000A1h	Intelligent I/O Interrupt Request Register 1	IIO1IR	0000 0XX1b
0000A2h	Intelligent I/O Interrupt Request Register 2	IIO2IR	0000 0X01b
0000A3h	Intelligent I/O Interrupt Request Register 3	IIO3IR	0000 XXX1b
0000A4h	Intelligent I/O Interrupt Request Register 4	IIO4IR	000X 0XX1b
0000A5h	Intelligent I/O Interrupt Request Register 5	IIO5IR	000X 0XX1b
0000A6h	Intelligent I/O Interrupt Request Register 6	IIO6IR	000X 0XX1b
0000A7h	Intelligent I/O Interrupt Request Register 7	IIO7IR	X00X 0XX1b
0000A8h	Intelligent I/O Interrupt Request Register 8	IIO8IR	XX0X 0XX1b
0000A9h	Intelligent I/O Interrupt Request Register 9	IIO9IR	0X00 0XX1b
0000AAh	Intelligent I/O Interrupt Request Register 10	IIO10IR	0X00 0XX1b
0000ABh	Intelligent I/O Interrupt Request Register 11	IIO11IR	0X00 0XX1b
0000ACh			
0000ADh			
0000AEh			
0000AFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.4 **SFR List (4)**

Address	Register	Symbol	Reset Value
0000B0h	Intelligent I/O Interrupt Enable Register 0	IIO0IE	00h
0000B1h	Intelligent I/O Interrupt Enable Register 1	IIO1IE	00h
0000B2h	Intelligent I/O Interrupt Enable Register 2	IIO2IE	00h
0000B3h	Intelligent I/O Interrupt Enable Register 3	IIO3IE	00h
0000B4h	Intelligent I/O Interrupt Enable Register 4	IIO4IE	00h
0000B5h	Intelligent I/O Interrupt Enable Register 5	IIO5IE	00h
0000B6h	Intelligent I/O Interrupt Enable Register 6	IIO6IE	00h
0000B7h	Intelligent I/O Interrupt Enable Register 7	IIO7IE	00h
0000B8h	Intelligent I/O Interrupt Enable Register 8	IIO8IE	00h
0000B9h	Intelligent I/O Interrupt Enable Register 9	IIO9IE	00h
0000BAh	Intelligent I/O Interrupt Enable Register 10	IIO10IE	00h
0000BBh	Intelligent I/O Interrupt Enable Register 11	IIO11IE	00h
0000BCh			
0000BDh			
0000BEh			
0000BFh			
0000C0h			
0000C1h			
0000C2h			
0000C3h			
0000C4h			
0000C5h			
0000C6h			
0000C7h			
0000C8h			
0000C9h			
0000CAh			
0000CBh			
0000CCh			
0000CDh			
0000CEh			
0000CFh			
0000D0h			
0000D1h			
0000D2h			
0000D3h			
0000D4h			
0000D5h			
0000D6h			
0000D7h			
0000D8h			
0000D9h	UART9 Transmit Interrupt Control Register	S9TIC	XXXX X000b
0000DAh			
0000DBh	UART10 Transmit Interrupt Control Register	S10TIC	XXXX X000b
0000DCh			
0000DDh	UART7 Transmit Interrupt Control Register	S7TIC	XXXX X000b
0000DEh	INT7 Interrupt Control Register	INT7IC	XX00 X000b
0000DFh	UART8 Transmit Interrupt Control Register	S8TIC	XXXX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.5 SFR List (5)

Address	Register	Symbol	Reset Value
0000E0h			
0000E1h			
0000E2h			
0000E3h			
0000E4h			
0000E5h			
0000E6h			
0000E7h			
0000E8h			
0000E9h			
0000EAh			
0000EBh			
0000ECh			
0000EDh			
0000EEh			
0000EFh			
0000F0h			
0000F1h			
0000F2h			
0000F3h			
0000F4h			
0000F5h			
0000F6h			
0000F7h			
0000F8h			
0000F9h	UART9 Receive Interrupt Control Register	S9RIC	XXXX X000b
0000FAh			
0000FBh	UART10 Receive Interrupt Control Register	S10RIC	XXXX X000b
0000FCh	INT8 Interrupt Control Register	INT8IC	XX00 X000b
0000FDh	UART7 Receive Interrupt Control Register	S7RIC	XXXX X000b
0000FEh	INT6 Interrupt Control Register	INT6IC	XX00 X000b
0000FFh	UART8 Receive Interrupt Control Register	S8RIC	XXXX X000b
000100h	Group 1 Time Measurement/Waveform Generation Register 0	G1TM0/G1PO0	XXXXh
000101h			
000102h	Group 1 Time Measurement/Waveform Generation Register 1	G1TM1/G1PO1	XXXXh
000103h			
000104h	Group 1 Time Measurement/Waveform Generation Register 2	G1TM2/G1PO2	XXXXh
000105h			
000106h	Group 1 Time Measurement/Waveform Generation Register 3	G1TM3/G1PO3	XXXXh
000107h			

X: Undefined

Blanks are reserved. No access is allowed.

Table 4.6 SFR List (6)

Address	Register	Symbol	Reset Value
000108h	Group 1 Time Measurement/Waveform Generation Register 4	G1TM4/G1PO4	XXXXh
000109h			
00010Ah	Group 1 Time Measurement/Waveform Generation Register 5	G1TM5/G1PO5	XXXXh
00010Bh			
00010Ch	Group 1 Time Measurement/Waveform Generation Register 6	G1TM6/G1PO6	XXXXh
00010Dh			
00010Eh	Group 1 Time Measurement/Waveform Generation Register 7	G1TM7/G1PO7	XXXXh
00010Fh			
000110h	Group 1 Waveform Generation Control Register 0	G1POCR0	0000 X000b
000111h	Group 1 Waveform Generation Control Register 1	G1POCR1	0X00 X000b
000112h	Group 1 Waveform Generation Control Register 2	G1POCR2	0X00 X000b
000113h	Group 1 Waveform Generation Control Register 3	G1POCR3	0X00 X000b
000114h	Group 1 Waveform Generation Control Register 4	G1POCR4	0X00 X000b
000115h	Group 1 Waveform Generation Control Register 5	G1POCR5	0X00 X000b
000116h	Group 1 Waveform Generation Control Register 6	G1POCR6	0X00 X000b
000117h	Group 1 Waveform Generation Control Register 7	G1POCR7	0X00 X000b
000118h	Group 1 Time Measurement Control Register 0	G1TMCR0	00h
000119h	Group 1 Time Measurement Control Register 1	G1TMCR1	00h
00011Ah	Group 1 Time Measurement Control Register 2	G1TMCR2	00h
00011Bh	Group 1 Time Measurement Control Register 3	G1TMCR3	00h
00011Ch	Group 1 Time Measurement Control Register 4	G1TMCR4	00h
00011Dh	Group 1 Time Measurement Control Register 5	G1TMCR5	00h
00011Eh	Group 1 Time Measurement Control Register 6	G1TMCR6	00h
00011Fh	Group 1 Time Measurement Control Register 7	G1TMCR7	00h
000120h	Group 1 Base Timer Register	G1BT	XXXXh
000121h			
000122h	Group 1 Base Timer Control Register 0	G1BCR0	0000 0000b
000123h	Group 1 Base Timer Control Register 1	G1BCR1	0000 0000b
000124h	Group 1 Time Measurement Prescaler Register 6	G1TPR6	00h
000125h	Group 1 Time Measurement Prescaler Register 7	G1TPR7	00h
000126h	Group 1 Function Enable Register	G1FE	00h
000127h	Group 1 Function Select Register	G1FS	00h
000128h			
000129h			
00012Ah			
00012Bh			
00012Ch			
00012Dh			
00012Eh			
00012Fh			
000130h to 00013Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.7 SFR List (7)

Address	Register	Symbol	Reset Value
000140h	Group 2 Waveform Generation Register 0	G2PO0	XXXXh
000141h			
000142h	Group 2 Waveform Generation Register 1	G2PO1	XXXXh
000143h			
000144h	Group 2 Waveform Generation Register 2	G2PO2	XXXXh
000145h			
000146h	Group 2 Waveform Generation Register 3	G2PO3	XXXXh
000147h			
000148h	Group 2 Waveform Generation Register 4	G2PO4	XXXXh
000149h			
00014Ah	Group 2 Waveform Generation Register 5	G2PO5	XXXXh
00014Bh			
00014Ch	Group 2 Waveform Generation Register 6	G2PO6	XXXXh
00014Dh			
00014Eh	Group 2 Waveform Generation Register 7	G2PO7	XXXXh
00014Fh			
000150h	Group 2 Waveform Generation Control Register 0	G2POCR0	0000 0000b
000151h	Group 2 Waveform Generation Control Register 1	G2POCR1	0000 0000b
000152h	Group 2 Waveform Generation Control Register 2	G2POCR2	0000 0000b
000153h	Group 2 Waveform Generation Control Register 3	G2POCR3	0000 0000b
000154h	Group 2 Waveform Generation Control Register 4	G2POCR4	0000 0000b
000155h	Group 2 Waveform Generation Control Register 5	G2POCR5	0000 0000b
000156h	Group 2 Waveform Generation Control Register 6	G2POCR6	0000 0000b
000157h	Group 2 Waveform Generation Control Register 7	G2POCR7	0000 0000b
000158h			
000159h			
00015Ah			
00015Bh			
00015Ch			
00015Dh			
00015Eh			
00015Fh			
000160h	Group 2 Base Timer Register	G2BT	XXXXh
000161h			
000162h	Group 2 Base Timer Control Register 0	G2BCR0	0000 0000b
000163h	Group 2 Base Timer Control Register 1	G2BCR1	0000 0000b
000164h	Base Timer Start Register	BTSR	XXXX 0000b
000165h			
000166h	Group 2 Function Enable Register	G2FE	00h
000167h	Group 2 RTP Output Buffer Register	G2RTP	00h
000168h			
000169h			
00016Ah	Group 2 Serial Interface Mode Register	G2MR	00XX X000b
00016Bh	Group 2 Serial Interface Control Register	G2CR	0000 X110b
00016Ch	Group 2 SI/O Transmit Buffer Register	G2TB	XXXXh
00016Dh			
00016Eh	Group 2 SI/O Receive Buffer Register	G2RB	XXXXh
00016Fh			

X: Undefined

Blanks are reserved. No access is allowed.

Table 4.8 SFR List (8)

Address	Register	Symbol	Reset Value
000170h	Group 2 IEBus Address Register	IEAR	XXXXh
000171h			
000172h	Group 2 IEBus Control Register	IECR	00XX X000b
000173h	Group 2 IEBus Transmit Interrupt Source Detect Register	IETIF	XXX0 0000b
000174h	Group 2 IEBus Receive Interrupt Source Detect Register	IERIF	XXX0 0000b
000175h			
000176h			
000177h			
000178h			
000179h			
00017Ah			
00017Bh			
00017Ch			
00017Dh			
00017Eh			
00017Fh			
000180h	Group 0 Time Measurement/Waveform Generation Register 0	G0TM0/G0PO0	XXXXh
000181h			
000182h	Group 0 Time Measurement/Waveform Generation Register 1	G0TM1/G0PO1	XXXXh
000183h			
000184h	Group 0 Time Measurement/Waveform Generation Register 2	G0TM2/G0PO2	XXXXh
000185h			
000186h	Group 0 Time Measurement/Waveform Generation Register 3	G0TM3/G0PO3	XXXXh
000187h			
000188h	Group 0 Time Measurement/Waveform Generation Register 4	G0TM4/G0PO4	XXXXh
000189h			
00018Ah	Group 0 Time Measurement/Waveform Generation Register 5	G0TM5/G0PO5	XXXXh
00018Bh			
00018Ch	Group 0 Time Measurement/Waveform Generation Register 6	G0TM6/G0PO6	XXXXh
00018Dh			
00018Eh	Group 0 Time Measurement/Waveform Generation Register 7	G0TM7/G0PO7	XXXXh
00018Fh			
000190h	Group 0 Waveform Generation Control Register 0	G0POCR0	0000 X000b
000191h	Group 0 Waveform Generation Control Register 1	G0POCR1	0X00 X000b
000192h	Group 0 Waveform Generation Control Register 2	G0POCR2	0X00 X000b
000193h	Group 0 Waveform Generation Control Register 3	G0POCR3	0X00 X000b
000194h	Group 0 Waveform Generation Control Register 4	G0POCR4	0X00 X000b
000195h	Group 0 Waveform Generation Control Register 5	G0POCR5	0X00 X000b
000196h	Group 0 Waveform Generation Control Register 6	G0POCR6	0X00 X000b
000197h	Group 0 Waveform Generation Control Register 7	G0POCR7	0X00 X000b
000198h	Group 0 Time Measurement Control Register 0	G0TMCR0	00h
000199h	Group 0 Time Measurement Control Register 1	G0TMCR1	00h
00019Ah	Group 0 Time Measurement Control Register 2	G0TMCR2	00h
00019Bh	Group 0 Time Measurement Control Register 3	G0TMCR3	00h
00019Ch	Group 0 Time Measurement Control Register 4	G0TMCR4	00h
00019Dh	Group 0 Time Measurement Control Register 5	G0TMCR5	00h
00019Eh	Group 0 Time Measurement Control Register 6	G0TMCR6	00h
00019Fh	Group 0 Time Measurement Control Register 7	G0TMCR7	00h

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.9 SFR List (9)

Address	Register	Symbol	Reset Value
0001A0h	Group 0 Base Timer Register	G0BT	XXXXh
0001A1h			
0001A2h	Group 0 Base Timer Control Register 0	G0BCR0	0000 0000b
0001A3h	Group 0 Base Timer Control Register 1	G0BCR1	0000 0000b
0001A4h	Group 0 Time Measurement Prescaler Register 6	G0TPR6	00h
0001A5h	Group 0 Time Measurement Prescaler Register 7	G0TPR7	00h
0001A6h	Group 0 Function Enable Register	G0FE	00h
0001A7h	Group 0 Function Select Register	G0FS	00h
0001A8h			
0001A9h			
0001AAh			
0001ABh			
0001ACh			
0001ADh			
0001AEh			
0001AFh			
0001B0h			
0001B1h			
0001B2h			
0001B3h			
0001B4h			
0001B5h			
0001B6h			
0001B7h			
0001B8h			
0001B9h			
0001BAh			
0001BBh			
0001BCh			
0001BDh			
0001BEh			
0001BFh			
0001C0h			
0001C1h			
0001C2h			
0001C3h			
0001C4h	UART5 Special Mode Register 4	U5SMR4	00h
0001C5h	UART5 Special Mode Register 3	U5SMR3	00h
0001C6h	UART5 Special Mode Register 2	U5SMR2	00h
0001C7h	UART5 Special Mode Register	U5SMR	00h
0001C8h	UART5 Transmit/Receive Mode Register	U5MR	00h
0001C9h	UART5 Bit Rate Register	U5BRG	XXh
0001CAh	UART5 Transmit Buffer Register	U5TB	XXXXh
0001CBh			
0001CCh	UART5 Transmit/Receive Control Register 0	U5C0	0000 1000b
0001CDh	UART5 Transmit/Receive Control Register 1	U5C1	0000 0010b
0001CEh	UART5 Receive Buffer Register	U5RB	XXXXh
0001CFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.10 SFR List (10)

Address	Register	Symbol	Reset Value
0001D0h			
0001D1h			
0001D2h			
0001D3h			
0001D4h	UART6 Special Mode Register 4	U6SMR4	00h
0001D5h	UART6 Special Mode Register 3	U6SMR3	00h
0001D6h	UART6 Special Mode Register 2	U6SMR2	00h
0001D7h	UART6 Special Mode Register	U6SMR	00h
0001D8h	UART6 Transmit/Receive Mode Register	U6MR	00h
0001D9h	UART6 Bit Rate Register	U6BRG	XXh
0001DAh	UART6 Transmit Buffer Register	U6TB	XXXXh
0001DBh			
0001DCh	UART6 Transmit/Receive Control Register 0	U6C0	0000 1000b
0001DDh	UART6 Transmit/Receive Control Register 1	U6C1	0000 0010b
0001DEh	UART6 Receive Buffer Register	U6RB	XXXXh
0001DFh			
0001E0h	UART7 Transmit/Receive Mode Register	U7MR	00h
0001E1h	UART7 Bit Rate Register	U7BRG	XXh
0001E2h	UART7 Transmit Buffer Register	U7TB	XXXXh
0001E3h			
0001E4h	UART7 Transmit/Receive Control Register 0	U7C0	00X0 1000b
0001E5h	UART7 Transmit/Receive Control Register 1	U7C1	XXXX 0010b
0001E6h	UART7 Receive Buffer Register	U7RB	XXXXh
0001E7h			
0001E8h	UART8 Transmit/Receive Mode Register	U8MR	00h
0001E9h	UART8 Bit Rate Register	U8BRG	XXh
0001EAh	UART8 Transmit Buffer Register	U8TB	XXXXh
0001EBh			
0001ECh	UART8 Transmit/Receive Control Register 0	U8C0	00X0 1000b
0001EDh	UART8 Transmit/Receive Control Register 1	U8C1	XXXX 0010b
0001EEh	UART8 Receive Buffer Register	U8RB	XXXXh
0001EFh			
0001F0h	UART7, UART8 Transmit/Receive Control Register 2	U78CON	X000 0000b
0001F1h			
0001F2h			
0001F3h			
0001F4h			
0001F5h			
0001F6h			
0001F7h			
0001F8h			
0001F9h			
0001FAh			
0001FBh			
0001FCh			
0001FDh			
0001FEh			
0001FFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.11 SFR List (11)

Address	Register	Symbol	Reset Value
000200h to 0002BFh			
0002C0h	X0 Register/Y0 Register	X0R/Y0R	XXXXh
0002C1h			
0002C2h	X1 Register/Y1 Register	X1R/Y1R	XXXXh
0002C3h			
0002C4h	X2 Register/Y2 Register	X2R/Y2R	XXXXh
0002C5h			
0002C6h	X3 Register/Y3 Register	X3R/Y3R	XXXXh
0002C7h			
0002C8h	X4 Register/Y4 Register	X4R/Y4R	XXXXh
0002C9h			
0002CAh	X5 Register/Y5 Register	X5R/Y5R	XXXXh
0002CBh			
0002CCh	X6 Register/Y6 Register	X6R/Y6R	XXXXh
0002CDh			
0002CEh	X7 Register/Y7 Register	X7R/Y7R	XXXXh
0002CFh			
0002D0h	X8 Register/Y8 Register	X8R/Y8R	XXXXh
0002D1h			
0002D2h	X9 Register/Y9 Register	X9R/Y9R	XXXXh
0002D3h			
0002D4h	X10 Register/Y10 Register	X10R/Y10R	XXXXh
0002D5h			
0002D6h	X11 Register/Y11 Register	X11R/Y11R	XXXXh
0002D7h			
0002D8h	X12 Register/Y12 Register	X12R/Y12R	XXXXh
0002D9h			
0002DAh	X13 Register/Y13 Register	X13R/Y13R	XXXXh
0002DBh			
0002DCh	X14 Register/Y14 Register	X14R/Y14R	XXXXh
0002DDh			
0002DEh	X15 Register/Y15 Register	X15R/Y15R	XXXXh
0002DFh			
0002E0h	X-Y Control Register	XYC	XXXX XX00b
0002E1h			
0002E2h			
0002E3h			
0002E4h	UART1 Special Mode Register 4	U1SMR4	00h
0002E5h	UART1 Special Mode Register 3	U1SMR3	00h
0002E6h	UART1 Special Mode Register 2	U1SMR2	00h
0002E7h	UART1 Special Mode Register	U1SMR	00h
0002E8h	UART1 Transmit/Receive Mode Register	U1MR	00h
0002E9h	UART1 Bit Rate Register	U1BRG	XXh
0002EAh	UART1 Transmit Buffer Register	U1TB	XXXXh
0002EBh			
0002ECh	UART1 Transmit/Receive Control Register 0	U1C0	0000 1000b
0002EDh	UART1 Transmit/Receive Control Register 1	U1C1	0000 0010b
0002EEh	UART1 Receive Buffer Register	U1RB	XXXXh
0002EFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.12 SFR List (12)

Address	Register	Symbol	Reset Value
0002F0h			
0002F1h			
0002F2h			
0002F3h			
0002F4h	UART4 Special Mode Register 4	U4SMR4	00h
0002F5h	UART4 Special Mode Register 3	U4SMR3	00h
0002F6h	UART4 Special Mode Register 2	U4SMR2	00h
0002F7h	UART4 Special Mode Register	U4SMR	00h
0002F8h	UART4 Transmit/Receive Mode Register	U4MR	00h
0002F9h	UART4 Bit Rate Register	U4BRG	XXh
0002FAh	UART4 Transmit Buffer Register	U4TB	XXXXh
0002FBh			
0002FCh	UART4 Transmit/Receive Control Register 0	U4C0	0000 1000b
0002FDh	UART4 Transmit/Receive Control Register 1	U4C1	0000 0010b
0002FEh	UART4 Receive Buffer Register	U4RB	XXXXh
0002FFh			
000300h	Count Start Register for Timers B3, B4, and B5	TBSR	000X XXXXb
000301h			
000302h	Timer A1-1 Register	TA11	XXXXh
000303h			
000304h	Timer A2-1 Register	TA21	XXXXh
000305h			
000306h	Timer A4-1 Register	TA41	XXXXh
000307h			
000308h	Three-phase PWM Control Register 0	INVC0	00h
000309h	Three-phase PWM Control Register 1	INVC1	00h
00030Ah	Three-phase Output Buffer Register 0	IDB0	XX11 1111b
00030Bh	Three-phase Output Buffer Register 1	IDB1	XX11 1111b
00030Ch	Dead Time Timer	DTT	XXh
00030Dh	Timer B2 Interrupt Generating Frequency Set Counter	ICTB2	XXh
00030Eh			
00030Fh			
000310h	Timer B3 Register	TB3	XXXXh
000311h			
000312h	Timer B4 Register	TB4	XXXXh
000313h			
000314h	Timer B5 Register	TB5	XXXXh
000315h			
000316h			
000317h			
000318h			
000319h			
00031Ah			
00031Bh	Timer B3 Mode Register	TB3MR	00XX 0000b
00031Ch	Timer B4 Mode Register	TB4MR	00XX 0000b
00031Dh	Timer B5 Mode Register	TB5MR	00XX 0000b
00031Eh			
00031Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.13 SFR List (13)

Address	Register	Symbol	Reset Value
000320h			
000321h			
000322h			
000323h			
000324h	UART3 Special Mode Register 4	U3SMR4	00h
000325h	UART3 Special Mode Register 3	U3SMR3	00h
000326h	UART3 Special Mode Register 2	U3SMR2	00h
000327h	UART3 Special Mode Register	U3SMR	00h
000328h	UART3 Transmit/Receive Mode Register	U3MR	00h
000329h	UART3 Bit Rate Register	U3BRG	XXh
00032Ah	UART3 Transmit Buffer Register	U3TB	XXXXh
00032Bh			
00032Ch	UART3 Transmit/Receive Control Register 0	U3C0	0000 1000b
00032Dh	UART3 Transmit/Receive Control Register 1	U3C1	0000 0010b
00032Eh	UART3 Receive Buffer Register	U3RB	XXXXh
00032Fh			
000330h			
000331h			
000332h			
000333h			
000334h	UART2 Special Mode Register 4	U2SMR4	00h
000335h	UART2 Special Mode Register 3	U2SMR3	00h
000336h	UART2 Special Mode Register 2	U2SMR2	00h
000337h	UART2 Special Mode Register	U2SMR	00h
000338h	UART2 Transmit/Receive Mode Register	U2MR	00h
000339h	UART2 Bit Rate Register	U2BRG	XXh
00033Ah	UART2 Transmit Buffer Register	U2TB	XXXXh
00033Bh			
00033Ch	UART2 Transmit/Receive Control Register 0	U2C0	0000 1000b
00033Dh	UART2 Transmit/Receive Control Register 1	U2C1	0000 0010b
00033Eh	UART2 Receive Buffer Register	U2RB	XXXXh
00033Fh			
000340h	Count Start Register	TABSR	0000 0000b
000341h	Clock Prescaler Reset Register	CPSRF	0XXX XXXXb
000342h	One-shot Start Register	ONSF	0000 0000b
000343h	Trigger Select Register	TRGSR	0000 0000b
000344h	Increment/Decrement Select Register	UDF	0000 0000b
000345h			
000346h	Timer A0 Register	TA0	XXXXh
000347h			
000348h	Timer A1 Register	TA1	XXXXh
000349h			
00034Ah	Timer A2 Register	TA2	XXXXh
00034Bh			
00034Ch	Timer A3 Register	TA3	XXXXh
00034Dh			
00034Eh	Timer A4 Register	TA4	XXXXh
00034Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.14 SFR List (14)

Address	Register	Symbol	Reset Value
000350h	Timer B0 Register	TB0	XXXXh
000351h			
000352h	Timer B1 Register	TB1	XXXXh
000353h			
000354h	Timer B2 Register	TB2	XXXXh
000355h			
000356h	Timer A0 Mode Register	TA0MR	0000 0000b
000357h	Timer A1 Mode Register	TA1MR	0000 0000b
000358h	Timer A2 Mode Register	TA2MR	0000 0000b
000359h	Timer A3 Mode Register	TA3MR	0000 0000b
00035Ah	Timer A4 Mode Register	TA4MR	0000 0000b
00035Bh	Timer B0 Mode Register	TB0MR	00XX 0000b
00035Ch	Timer B1 Mode Register	TB1MR	00XX 0000b
00035Dh	Timer B2 Mode Register	TB2MR	00XX 0000b
00035Eh	Timer B2 Special Mode Register	TB2SC	XXXX XXX0b
00035Fh	Count Source Prescaler Register	TCSPR	0000 0000b
000360h			
000361h			
000362h			
000363h			
000364h	UART0 Special Mode Register 4	U0SMR4	00h
000365h	UART0 Special Mode Register 3	U0SMR3	00h
000366h	UART0 Special Mode Register 2	U0SMR2	00h
000367h	UART0 Special Mode Register	U0SMR	00h
000368h	UART0 Transmit/Receive Mode Register	U0MR	00h
000369h	UART0 Bit Rate Register	U0BRG	XXh
00036Ah	UART0 Transmit Buffer Register	U0TB	XXXXh
00036Bh			
00036Ch	UART0 Transmit/Receive Control Register 0	U0C0	0000 1000b
00036Dh	UART0 Transmit/Receive Control Register 1	U0C1	0000 0010b
00036Eh	UART0 Receive Buffer Register	U0RB	XXXXh
00036Fh			
000370h			
000371h			
000372h			
000373h			
000374h			
000375h			
000376h			
000377h			
000378h			
000379h			
00037Ah			
00037Bh			
00037Ch	CRC Data Register	CRCD	XXXXh
00037Dh			
00037Eh	CRC Input Register	CRCIN	XXh
00037Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.15 SFR List (15)

Address	Register	Symbol	Reset Value
000380h	A/D0 Register 0	AD00	00XXh
000381h			
000382h	A/D0 Register 1	AD01	00XXh
000383h			
000384h	A/D0 Register 2	AD02	00XXh
000385h			
000386h	A/D0 Register 3	AD03	00XXh
000387h			
000388h	A/D0 Register 4	AD04	00XXh
000389h			
00038Ah	A/D0 Register 5	AD05	00XXh
00038Bh			
00038Ch	A/D0 Register 6	AD06	00XXh
00038Dh			
00038Eh	A/D0 Register 7	AD07	00XXh
00038Fh			
000390h			
000391h			
000392h	A/D0 Control Register 4	AD0CON4	XXXX 00XXb
000393h	A/D0 Control Register 5	AD0CON5	00h
000394h	A/D0 Control Register 2	AD0CON2	XX0X X000b
000395h	A/D0 Control Register 3	AD0CON3	XXXX X000b
000396h	A/D0 Control Register 0	AD0CON0	00h
000397h	A/D0 Control Register 1	AD0CON1	00h
000398h	D/A Register 0	DA0	XXh
000399h			
00039Ah	D/A Register 1	DA1	XXh
00039Bh			
00039Ch	D/A Control Register	DACON	XXXX XX00b
00039Dh			
00039Eh			
00039Fh			
0003A0h			
0003A1h			
0003A2h			
0003A3h			
0003A4h			
0003A5h			
0003A6h			
0003A7h			
0003A8h			
0003A9h			
0003AAh			
0003ABh			
0003ACh			
0003ADh			
0003AEh			
0003AFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.16 SFR List (16)

Address	Register	Symbol	Reset Value
0003B0h			
0003B1h			
0003B2h			
0003B3h			
0003B4h			
0003B5h			
0003B6h			
0003B7h			
0003B8h			
0003B9h			
0003BAh			
0003BBh			
0003BCh			
0003BDh			
0003BEh			
0003BFh			
0003C0h	Port P0 Register	P0	XXh
0003C1h	Port P1 Register	P1	XXh
0003C2h	Port P0 Direction Register	PD0	0000 0000b
0003C3h	Port P1 Direction Register	PD1	0000 0000b
0003C4h	Port P2 Register	P2	XXh
0003C5h	Port P3 Register	P3	XXh
0003C6h	Port P2 Direction Register	PD2	0000 0000b
0003C7h	Port P3 Direction Register	PD3	0000 0000b
0003C8h	Port P4 Register	P4	XXh
0003C9h	Port P5 Register	P5	XXh
0003CAh	Port P4 Direction Register	PD4	0000 0000b
0003CBh	Port P5 Direction Register	PD5	0000 0000b
0003CCh	Port P6 Register	P6	XXh
0003CDh	Port P7 Register	P7	XXh
0003CEh	Port P6 Direction Register	PD6	0000 0000b
0003CFh	Port P7 Direction Register	PD7	0000 0000b
0003D0h	Port P8 Register	P8	XXh
0003D1h	Port P9 Register	P9	XXh
0003D2h	Port P8 Direction Register	PD8	00X0 0000b
0003D3h	Port P9 Direction Register	PD9	0000 0000b
0003D4h	Port P10 Register	P10	XXh
0003D5h	Port P11 Register	P11	XXh
0003D6h	Port P10 Direction Register	PD10	0000 0000b
0003D7h	Port P11 Direction Register	PD11	0000 0000b
0003D8h	Port P12 Register	P12	XXh
0003D9h	Port P13 Register	P13	XXh
0003DAh	Port P12 Direction Register	PD12	0000 0000b
0003DBh	Port P13 Direction Register	PD13	0000 0000b
0003DCh	Port P14 Register	P14	XXh
0003DDh	Port P15 Register	P15	XXh
0003DEh	Port P14 Direction Register	PD14	0000 0000b
0003DFh	Port P15 Direction Register	PD15	0000 0000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.17 SFR List (17)

Address	Register	Symbol	Reset Value
0003E0h	Port P16 Register	P16	XXh
0003E1h	Port P17 Register	P17	XXh
0003E2h	Port P16 Direction Register	PD16	0000 0000b
0003E3h	Port P17 Direction Register	PD17	0000 0000b
0003E4h	Port P18 Register	P18	XXh
0003E5h	Port P19 Register	P19	XXh
0003E6h	Port P18 Direction Register	PD18	0000 0000b
0003E7h	Port P19 Direction Register	PD19	0000 0000b
0003E8h			
0003E9h			
0003EAh			
0003EBh			
0003ECh			
0003EDh			
0003EEh			
0003EFh			
0003F0h	Pull-up Control Register 0	PUR0	0000 0000b
0003F1h	Pull-up Control Register 1	PUR1	XXXX X0XXb
0003F2h	Pull-up Control Register 2	PUR2	X00X XXXXb
0003F3h	Pull-up Control Register 3	PUR3	00XX 0000b
0003F4h	Pull-up Control Register 4	PUR4	0XXX 0000b
0003F5h	Pull-up Control Register 5	PUR5	XXXX 0000b
0003F6h			
0003F7h			
0003F8h			
0003F9h			
0003FAh			
0003FBh			
0003FCh			
0003FDh			
0003FEh			
0003FFh	Port Control Register	PCR	?0XX 0XX0b (1)

X: Undefined
Blanks are reserved. No access is allowed.

Note:
1. The bit 7 is 0 in the 144-pin package and 1 in the 176-pin package.

Table 4.18 SFR List (18)

Address	Register	Symbol	Reset Value
040000h	Flash Memory Control Register 0	FMR0	0001 XX00b
040001h	Flash Memory Status Register 0	FMSR0	1000 0000b
040002h			
040003h			
040004h			
040005h			
040006h			
040007h			
040008h	Flash Register Protection Unlock Register 0	FPR0	00h
040009h	Flash Memory Control Register 1	FMR1	0000 0010b
04000Ah	Block Protect Bit Monitor Register 0	FBPM0	??X? ????b (1)
04000Bh	Block Protect Bit Monitor Register 1	FBPM1	XXX? ????b (1)
04000Ch			
04000Dh			
04000Eh			
04000Fh			
040010h			
040011h	Block Protect Bit Monitor Register 2	FBPM2	???? ????b (1)
040012h			
040013h			
040014h			
040015h			
040016h			
040017h			
040018h			
040019h			
04001Ah			
04001Bh			
04001Ch			
04001Dh			
04001Eh			
04001Fh			
040020h	PLL Control Register 0	PLC0	0000 0001b
040021h	PLL Control Register 1	PLC1	0001 1111b
040022h			
040023h			
040024h			
040025h			
040026h			
040027h			
040028h			
040029h			
04002Ah			
04002Bh			
04002Ch			
04002Dh			
04002Eh			
04002Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Note:

1. The reset value reflects the value of the protect bit for each block in the flash memory.

Table 4.19 SFR List (19)

Address	Register	Symbol	Reset Value
040030h to 04003Fh			
040040h			
040041h			
040042h			
040043h			
040044h	Processor Mode Register 0 (1)	PM0	1000 0000b (CNVSS pin = Low) 0000 0011b (CNVSS pin = High)
040045h			
040046h	System Clock Control Register 0	CM0	0000 1000b
040047h	System Clock Control Register 1	CM1	0010 0000b
040048h	Processor Mode Register 3	PM3	00h
040049h			
04004Ah	Protect Register	PRCR	XXXX X000b
04004Bh			
04004Ch	Protect Register 3	PRCR3	0000 0000b
04004Dh	Oscillator Stop Detection Register	CM2	00h
04004Eh			
04004Fh			
040050h			
040051h			
040052h			
040053h	Processor Mode Register 2	PM2	00h
040054h	Chip Select Output Pin Setting Register 0	CSOP0	1000 XXXXb
040055h	Chip Select Output Pin Setting Register 1	CSOP1	01X0 XXXXb
040056h	Chip Select Output Pin Setting Register 2	CSOP2	XXXX 0000b
040057h			
040058h			
040059h			
04005Ah	Low Speed Mode Clock Control Register	CM3	XXXX XX00b
04005Bh			
04005Ch			
04005Dh			
04005Eh			
04005Fh			
040060h	Voltage Regulator Control Register	VRCR	0000 0000b
040061h			
040062h	Low Voltage Detector Control Register	LVDC	0000 XX00b
040063h			
040064h	Detection Voltage Configuration Register	DVCR	0000 XXXXb
040065h			
040066h			
040067h			
040068h to 040093h			

X: Undefined
Blanks are reserved. No access is allowed.

Note:

1. The value in the PM0 register is retained even after a software reset or watchdog timer reset.

Table 4.20 SFR List (20)

Address	Register	Symbol	Reset Value
040094h			
040095h			
040096h			
040097h	Three-phase Output Buffer Control Register	IOBC	0XXX XXXXb
040098h	Input Function Select Register 0	IFS0	X000 0000b
040099h			
04009Ah	Input Function Select Register 2	IFS2	0000 00X0b
04009Bh	Input Function Select Register 3	IFS3	XX00 XX00b
04009Ch			
04009Dh			
04009Eh			
04009Fh			
0400A0h	Port P0_0 Function Select Register	P0_0S	0XXX X000b
0400A1h	Port P1_0 Function Select Register	P1_0S	XXXX X000b
0400A2h	Port P0_1 Function Select Register	P0_1S	0XXX X000b
0400A3h	Port P1_1 Function Select Register	P1_1S	XXXX X000b
0400A4h	Port P0_2 Function Select Register	P0_2S	0XXX X000b
0400A5h	Port P1_2 Function Select Register	P1_2S	XXXX X000b
0400A6h	Port P0_3 Function Select Register	P0_3S	0XXX X000b
0400A7h	Port P1_3 Function Select Register	P1_3S	XXXX X000b
0400A8h	Port P0_4 Function Select Register	P0_4S	0XXX X000b
0400A9h	Port P1_4 Function Select Register	P1_4S	XXXX X000b
0400AAh	Port P0_5 Function Select Register	P0_5S	0XXX X000b
0400ABh	Port P1_5 Function Select Register	P1_5S	XXXX X000b
0400ACh	Port P0_6 Function Select Register	P0_6S	0XXX X000b
0400ADh	Port P1_6 Function Select Register	P1_6S	XXXX X000b
0400AEh	Port P0_7 Function Select Register	P0_7S	0XXX X000b
0400AFh	Port P1_7 Function Select Register	P1_7S	XXXX X000b
0400B0h	Port P2_0 Function Select Register	P2_0S	00XX X000b
0400B1h	Port P3_0 Function Select Register	P3_0S	XXXX X000b
0400B2h	Port P2_1 Function Select Register	P2_1S	00XX X000b
0400B3h	Port P3_1 Function Select Register	P3_1S	XXXX X000b
0400B4h	Port P2_2 Function Select Register	P2_2S	00XX X000b
0400B5h	Port P3_2 Function Select Register	P3_2S	XXXX X000b
0400B6h	Port P2_3 Function Select Register	P2_3S	00XX X000b
0400B7h	Port P3_3 Function Select Register	P3_3S	XXXX X000b
0400B8h	Port P2_4 Function Select Register	P2_4S	00XX X000b
0400B9h	Port P3_4 Function Select Register	P3_4S	XXXX X000b
0400BAh	Port P2_5 Function Select Register	P2_5S	00XX X000b
0400BBh	Port P3_5 Function Select Register	P3_5S	XXXX X000b
0400BCh	Port P2_6 Function Select Register	P2_6S	00XX X000b
0400BDh	Port P3_6 Function Select Register	P3_6S	XXXX X000b
0400BEh	Port P2_7 Function Select Register	P2_7S	00XX X000b
0400BFh	Port P3_7 Function Select Register	P3_7S	XXXX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.21 SFR List (21)

Address	Register	Symbol	Reset Value
0400C0h	Port P4_0 Function Select Register	P4_0S	X0XX X000b
0400C1h	Port P5_0 Function Select Register	P5_0S	XXXX X000b
0400C2h	Port P4_1 Function Select Register	P4_1S	X0XX X000b
0400C3h	Port P5_1 Function Select Register	P5_1S	XXXX X000b
0400C4h	Port P4_2 Function Select Register	P4_2S	X0XX X000b
0400C5h	Port P5_2 Function Select Register	P5_2S	XXXX X000b
0400C6h	Port P4_3 Function Select Register	P4_3S	X0XX X000b
0400C7h	Port P5_3 Function Select Register	P5_3S	XXXX X000b
0400C8h	Port P4_4 Function Select Register	P4_4S	X0XX X000b
0400C9h	Port P5_4 Function Select Register	P5_4S	X0XX X000b
0400CAh	Port P4_5 Function Select Register	P4_5S	X0XX X000b
0400CBh	Port P5_5 Function Select Register	P5_5S	X0XX X000b
0400CCh	Port P4_6 Function Select Register	P4_6S	X0XX X000b
0400CDh	Port P5_6 Function Select Register	P5_6S	X0XX X000b
0400CEh	Port P4_7 Function Select Register	P4_7S	X0XX X000b
0400CFh	Port P5_7 Function Select Register	P5_7S	X0XX X000b
0400D0h	Port P6_0 Function Select Register	P6_0S	X0XX X000b
0400D1h	Port P7_0 Function Select Register	P7_0S	X0XX X000b
0400D2h	Port P6_1 Function Select Register	P6_1S	X0XX X000b
0400D3h	Port P7_1 Function Select Register	P7_1S	X0XX X000b
0400D4h	Port P6_2 Function Select Register	P6_2S	X0XX X000b
0400D5h	Port P7_2 Function Select Register	P7_2S	X0XX X000b
0400D6h	Port P6_3 Function Select Register	P6_3S	X0XX X000b
0400D7h	Port P7_3 Function Select Register	P7_3S	X0XX X000b
0400D8h	Port P6_4 Function Select Register	P6_4S	X0XX X000b
0400D9h	Port P7_4 Function Select Register	P7_4S	X0XX X000b
0400DAh	Port P6_5 Function Select Register	P6_5S	X0XX X000b
0400DBh	Port P7_5 Function Select Register	P7_5S	X0XX X000b
0400DCh	Port P6_6 Function Select Register	P6_6S	X0XX X000b
0400DDh	Port P7_6 Function Select Register	P7_6S	X0XX X000b
0400DEh	Port P6_7 Function Select Register	P6_7S	X0XX X000b
0400DFh	Port P7_7 Function Select Register	P7_7S	X0XX X000b
0400E0h	Port P8_0 Function Select Register	P8_0S	X0XX X000b
0400E1h	Port P9_0 Function Select Register	P9_0S	X0XX X000b
0400E2h	Port P8_1 Function Select Register	P8_1S	X0XX X000b
0400E3h	Port P9_1 Function Select Register	P9_1S	X0XX X000b
0400E4h	Port P8_2 Function Select Register	P8_2S	X0XX X000b
0400E5h	Port P9_2 Function Select Register	P9_2S	X0XX X000b
0400E6h	Port P8_3 Function Select Register	P8_3S	X0XX X000b
0400E7h	Port P9_3 Function Select Register	P9_3S	00XX X000b
0400E8h	Port P8_4 Function Select Register	P8_4S	XXXX X000b
0400E9h	Port P9_4 Function Select Register	P9_4S	00XX X000b
0400EAh			
0400EBh	Port P9_5 Function Select Register	P9_5S	00XX X000b
0400ECh	Port P8_6 Function Select Register	P8_6S	XXXX X000b
0400EDh	Port P9_6 Function Select Register	P9_6S	00XX X000b
0400EEh	Port P8_7 Function Select Register	P8_7S	XXXX X000b
0400EFh	Port P9_7 Function Select Register	P9_7S	X0XX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.22 SFR List (22)

Address	Register	Symbol	Reset Value
0400F0h	Port P10_0 Function Select Register	P10_0S	0XXX X000b
0400F1h	Port P11_0 Function Select Register	P11_0S	X0XX X000b
0400F2h	Port P10_1 Function Select Register	P10_1S	0XXX X000b
0400F3h	Port P11_1 Function Select Register	P11_1S	X0XX X000b
0400F4h	Port P10_2 Function Select Register	P10_2S	0XXX X000b
0400F5h	Port P11_2 Function Select Register	P11_2S	X0XX X000b
0400F6h	Port P10_3 Function Select Register	P10_3S	0XXX X000b
0400F7h	Port P11_3 Function Select Register	P11_3S	X0XX X000b
0400F8h	Port P10_4 Function Select Register	P10_4S	0XXX X000b
0400F9h	Port P11_4 Function Select Register	P11_4S	XXXX X000b
0400FAh	Port P10_5 Function Select Register	P10_5S	0XXX X000b
0400FBh	Port P11_5 Function Select Register	P11_5S	XXXX X000b
0400FCh	Port P10_6 Function Select Register	P10_6S	0XXX X000b
0400FDh	Port P11_6 Function Select Register	P11_6S	XXXX X000b
0400FEh	Port P10_7 Function Select Register	P10_7S	0XXX X000b
0400FFh	Port P11_7 Function Select Register	P11_7S	XXXX X000b
040100h	Port P12_0 Function Select Register	P12_0S	X0XX X000b
040101h	Port P13_0 Function Select Register	P13_0S	XXXX X000b
040102h	Port P12_1 Function Select Register	P12_1S	X0XX X000b
040103h	Port P13_1 Function Select Register	P13_1S	XXXX X000b
040104h	Port P12_2 Function Select Register	P12_2S	X0XX X000b
040105h	Port P13_2 Function Select Register	P13_2S	XXXX X000b
040106h	Port P12_3 Function Select Register	P12_3S	X0XX X000b
040107h	Port P13_3 Function Select Register	P13_3S	XXXX X000b
040108h	Port P12_4 Function Select Register	P12_4S	X0XX X000b
040109h	Port P13_4 Function Select Register	P13_4S	XXXX X000b
04010Ah	Port P12_5 Function Select Register	P12_5S	X0XX X000b
04010Bh	Port P13_5 Function Select Register	P13_5S	XXXX X000b
04010Ch	Port P12_6 Function Select Register	P12_6S	X0XX X000b
04010Dh	Port P13_6 Function Select Register	P13_6S	XXXX X000b
04010Eh	Port P12_7 Function Select Register	P12_7S	X0XX X000b
04010Fh	Port P13_7 Function Select Register	P13_7S	XXXX X000b
040110h			
040111h	Port P15_0 Function Select Register	P15_0S	00XX X000b
040112h			
040113h	Port P15_1 Function Select Register	P15_1S	00XX X000b
040114h			
040115h	Port P15_2 Function Select Register	P15_2S	00XX X000b
040116h	Port P14_3 Function Select Register	P14_3S	XXXX X000b
040117h	Port P15_3 Function Select Register	P15_3S	00XX X000b
040118h	Port P14_4 Function Select Register	P14_4S	XXXX X000b
040119h	Port P15_4 Function Select Register	P15_4S	00XX X000b
04011Ah	Port P14_5 Function Select Register	P14_5S	XXXX X000b
04011Bh	Port P15_5 Function Select Register	P15_5S	00XX X000b
04011Ch	Port P14_6 Function Select Register	P14_6S	XXXX X000b
04011Dh	Port P15_6 Function Select Register	P15_6S	00XX X000b
04011Eh	Port P14_7 Function Select Register	P14_7S	XXXX X000b
04011Fh	Port P15_7 Function Select Register	P15_7S	00XX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.23 SFR List (23)

Address	Register	Symbol	Reset Value
040120h	Port P16_0 Function Select Register	P16_0S	X0XX X000b
040121h	Port P17_0 Function Select Register	P17_0S	X0XX X000b
040122h	Port P16_1 Function Select Register	P16_1S	X0XX X000b
040123h	Port P17_1 Function Select Register	P17_1S	X0XX X000b
040124h	Port P16_2 Function Select Register	P16_2S	X0XX X000b
040125h	Port P17_2 Function Select Register	P17_2S	X0XX X000b
040126h	Port P16_3 Function Select Register	P16_3S	X0XX X000b
040127h	Port P17_3 Function Select Register	P17_3S	X0XX X000b
040128h	Port P16_4 Function Select Register	P16_4S	X0XX X000b
040129h	Port P17_4 Function Select Register	P17_4S	XXXX X000b
04012Ah	Port P16_5 Function Select Register	P16_5S	X0XX X000b
04012Bh	Port P17_5 Function Select Register	P17_5S	XXXX X000b
04012Ch	Port P16_6 Function Select Register	P16_6S	X0XX X000b
04012Dh	Port P17_6 Function Select Register	P17_6S	XXXX X000b
04012Eh	Port P16_7 Function Select Register	P16_7S	X0XX X000b
04012Fh	Port P17_7 Function Select Register	P17_7S	XXXX X000b
040130h	Port P18_0 Function Select Register	P18_0S	XXXX X000b
040131h	Port P19_0 Function Select Register	P19_0S	XXXX X000b
040132h	Port P18_1 Function Select Register	P18_1S	XXXX X000b
040133h	Port P19_1 Function Select Register	P19_1S	XXXX X000b
040134h	Port P18_2 Function Select Register	P18_2S	XXXX X000b
040135h	Port P19_2 Function Select Register	P19_2S	XXXX X000b
040136h	Port P18_3 Function Select Register	P18_3S	XXXX X000b
040137h	Port P19_3 Function Select Register	P19_3S	XXXX X000b
040138h	Port P18_4 Function Select Register	P18_4S	XXXX X000b
040139h	Port P19_4 Function Select Register	P19_4S	XXXX X000b
04013Ah	Port P18_5 Function Select Register	P18_5S	XXXX X000b
04013Bh	Port P19_5 Function Select Register	P19_5S	XXXX X000b
04013Ch	Port P18_6 Function Select Register	P18_6S	XXXX X000b
04013Dh	Port P19_6 Function Select Register	P19_6S	XXXX X000b
04013Eh	Port P18_7 Function Select Register	P18_7S	XXXX X000b
04013Fh	Port P19_7 Function Select Register	P19_7S	XXXX X000b
040140h			
040141h			
040142h			
040143h			
040144h			
040145h			
040146h			
040147h			
040148h			
040149h			
04014Ah			
04014Bh			
04014Ch			
04014Dh			
04014Eh			
04014Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.24 SFR List (24)

Address	Register	Symbol	Reset Value
040150h to 0402FFh			
040300h	UART9 Transmit/Receive Mode Register	U9MR	00h
040301h	UART9 Bit Rate Register	U9BRG	XXh
040302h	UART9 Transmit Buffer Register	U9TB	XXXXh
040303h			
040304h	UART9 Transmit/Receive Control Register 0	U9C0	00X0 1000b
040305h	UART9 Transmit/Receive Control Register 1	U9C1	XXXX 0010b
040306h	UART9 Receive Buffer Register	U9RB	XXXXh
040307h			
040308h	UART10 Transmit/Receive Mode Register	U10MR	00h
040309h	UART10 Bit Rate Register	U10BRG	XXh
04030Ah	UART10 Transmit Buffer Register	U10TB	XXXXh
04030Bh			
04030Ch	UART10 Transmit/Receive Control Register 0	U10C0	00X0 1000b
04030Dh	UART10 Transmit/Receive Control Register 1	U10C1	XXXX 0010b
04030Eh	UART10 Receive Buffer Register	U10RB	XXXXh
04030Fh			
040310h	UART9, UART10 Transmit/Receive Control Register 2	U910CON	X000 0000b
040311h			
040312h			
040313h			
040314h			
040315h			
040316h			
040317h			
040318h to 041FFFh			
042000h to 04201Fh	Protected Area 0	—	Undefined
042020h to 04203Fh	Protected Area 1	—	Undefined
042040h to 04205Fh	Protected Area 2	—	Undefined
042060h to 04207Fh	Protected Area 3	—	Undefined
042080h to 04209Fh	Protected Area 4	—	Undefined
0420A0h to 0420EFh			
0420F0h	Protected Area Protect Release Register	PAPR	XXX0 0000b
0420F1h			
0420F2h	Protected Area Write Access Flag Register	PAWF	XXX0 0000b
0420F3h			
0420F4h			
0420F5h			
0420F6h			
0420F7h			
0420F8h to 043FFFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.25 SFR List (25)

Address	Register	Symbol	Reset Value
044000h to 04403Fh			
044040h			
044041h			
044042h			
044043h			
044044h			
044045h			
044046h			
044047h			
044048h			
044049h			
04404Ah			
04404Bh			
04404Ch	Protect Register 4	PRCR4	0000 0000b
04404Dh	Watchdog Timer Clock Control Register	WDK	0000 000?b (1)
04404Eh	Watchdog Timer Start Register	WDTS	XXXX XXXXb
04404Fh	Watchdog Timer Control Register	WDC	000X XXXXb
044050h			
044051h			
044052h			
044053h			
044054h			
044055h			
044056h			
044057h			
044058h			
044059h			
04405Ah			
04405Bh			
04405Ch			
04405Dh			
04405Eh			
04405Fh	Protect Register 2	PRCR2	0XXX XXXXb

X: Undefined
Blanks are reserved. No access is allowed.

Note:

1. The bit 0 is set to 1 when the most recent reset is caused by the watchdog timer. Otherwise, it is set to 0.

Table 4.26 SFR List (26)

Address	Register	Symbol	Reset Value
044060h			
044061h			
044062h			
044063h			
044064h			
044065h			
044066h			
044067h			
044068h			
044069h			
04406Ah			
04406Bh			
04406Ch			
04406Dh	External Interrupt Request Source Select Register 1	IFSR1	X0XX X000b
04406Eh			
04406Fh	External Interrupt Request Source Select Register 0	IFSR0	0000 0000b
044070h	DMA0 Request Source Select Register 2	DM0SL2	XX00 0000b
044071h	DMA1 Request Source Select Register 2	DM1SL2	XX00 0000b
044072h	DMA2 Request Source Select Register 2	DM2SL2	XX00 0000b
044073h	DMA3 Request Source Select Register 2	DM3SL2	XX00 0000b
044074h			
044075h			
044076h			
044077h			
044078h	DMA0 Request Source Select Register	DM0SL	XXX0 0000b
044079h	DMA1 Request Source Select Register	DM1SL	XXX0 0000b
04407Ah	DMA2 Request Source Select Register	DM2SL	XXX0 0000b
04407Bh	DMA3 Request Source Select Register	DM3SL	XXX0 0000b
04407Ch			
04407Dh	Wake-up IPL Setting Register 2	RIPL2	XX0X 0000b
04407Eh			
04407Fh	Wake-up IPL Setting Register 1	RIPL1	XX0X 0000b
044080h			
044081h			
044082h			
044083h			
044084h			
044085h			
044086h			
044087h			
044088h			
044089h			
04408Ah			
04408Bh			
04408Ch			
04408Dh			
04408Eh			
04408Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.27 SFR List (27)

Address	Register	Symbol	Reset Value
044090h to 0443FFh			
044400h	I²C-bus Transmit/Receive Shift Register	I2CTRSR	XXh
044401h			
044402h	I²C-bus Slave Address Register	I2CSAR	00h
044403h	I²C-bus Control Register 0	I2CCR0	0000 0000b
044404h	I²C-bus Clock Control Register	I2CCCR	0000 0000b
044405h	I²C-bus START and STOP Conditions Control Register	I2CSSCR	0001 1010b
044406h	I²C-bus Control Register 1	I2CCR1	0011 0000b
044407h	I²C-bus Control Register 2	I2CCR2	0X00 0000b
044408h	I²C-bus Status Register	I2CSR	0001 000Xb
044409h			
04440Ah			
04440Bh			
04440Ch			
04440Dh			
04440Eh			
04440Fh			
044410h	I²C-bus Mode Register	I2CMR	XXXX 0000b
044411h			
044412h			
044413h			
044414h			
044415h			
044416h			
044417h			
044418h			
044419h			
04441Ah			
04441Bh			
04441Ch			
04441Dh			
04441Eh			
04441Fh			
044420h to 04FFFFh			

X: Undefined
Blanks are reserved. No access is allowed.

5. Resets

There are three types of operations for resetting the MCU: hardware reset, software reset, and watchdog timer reset.

5.1 Hardware Reset

A hardware reset is generated when a low signal is applied to the $\overline{\text{RESET}}$ pin under the recommended operating conditions of the supply voltage. When the $\overline{\text{RESET}}$ pin is driven low, all pins, and oscillators are reset (refer to Table 5.1 for details), and the main clock starts oscillating. The CPU and SFRs are reset by a low-to-high transition on the $\overline{\text{RESET}}$ pin. Then, the CPU starts executing the program from the address indicated by the reset vector. Internal RAM is not affected by a hardware reset. However, if a hardware reset occurs during a write operation to the internal RAM, the value is undefined.

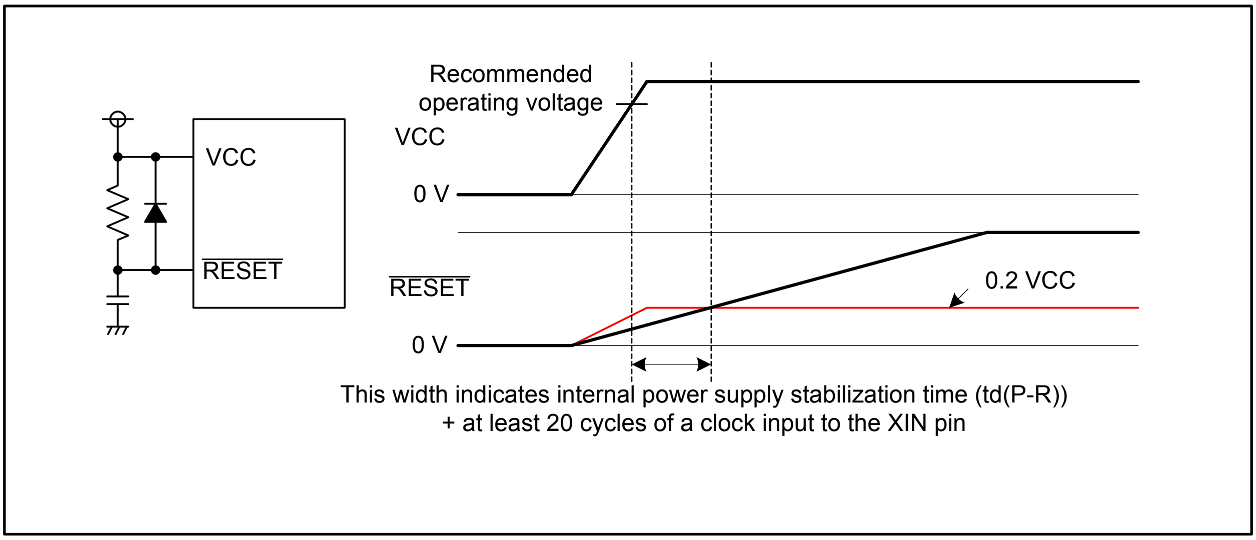
Figure 5.1 shows an example of the reset circuit. Figure 5.2 shows the reset sequence. Table 5.1 lists pin states while the $\overline{\text{RESET}}$ pin is held low. Figure 5.3 shows CPU register states after a reset. Refer to 4. “Special Function Registers (SFRs)” for details on the states of SFRs after a reset.

A. Reset when the supply voltage is stable

(1) Drive the $\overline{\text{RESET}}$ pin low.
(2) Input at least 20 clock cycles to the XIN pin.
(3) Drive the $\overline{\text{RESET}}$ pin high.

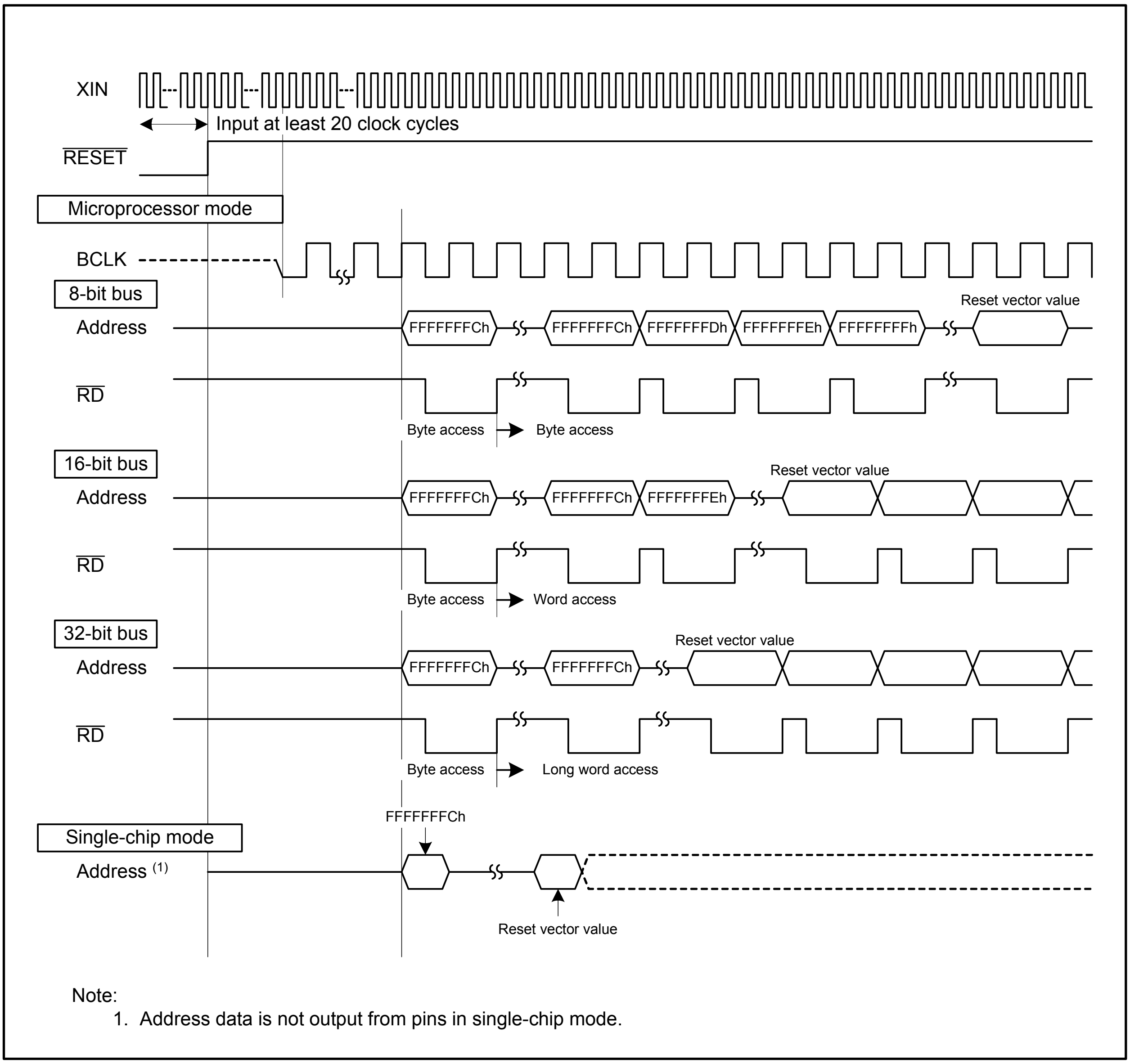
B. Reset when turning on the power

(1) Drive the $\overline{\text{RESET}}$ pin low.
(2) Raise the supply voltage to the recommended operating voltage.
(3) Wait td(P-R) ms until the internal voltage is stabilized.
(4) Input at least 20 clock cycles to the XIN pin.
(5) Drive the $\overline{\text{RESET}}$ pin high.


VCC
$\overline{\text{RESET}}$
Recommended operating voltage
VCC
0 V
$\overline{\text{RESET}}$
0.2 VCC
0 V

This width indicates internal power supply stabilization time (td(P-R)) + at least 20 cycles of a clock input to the XIN pin

Figure 5.1 Reset Circuitry



Note:
1. Address data is not output from pins in single-chip mode.

Figure 5.2 Reset Sequence

Table 5.1 Pin States while $\overline{\text{RESET}}$ Pin is Held Low [1]

Pin Name	Pin States	
	CNVSS = VSS	CNVSS = VCC
P0	Input port (high-impedance)	Inputs data
P1	Input port (high-impedance)	Input port (high-impedance)
P2, P3	Input port (high-impedance)	Output addresses (undefined)
P4_0 to P4_6	Input port (high-impedance)	Output addresses (undefined)
P4_7	Input port (high-impedance)	Outputs the $\overline{\text{CS0}}$ signal (high)
P5_0	Input port (high-impedance)	Outputs the $\overline{\text{WR}}$ signal (high)
P5_1	Input port (high-impedance)	Outputs the $\overline{\text{BC1}}$ signal (undefined)
P5_2	Input port (high-impedance)	Outputs the $\overline{\text{RD}}$ signal (high)
P5_3	Input port (high-impedance)	Outputs BCLK [2]
P5_4	Input port (high-impedance)	Outputs the $\overline{\text{HLDA}}$ signal (output signal depends on an input signal to the $\overline{\text{HOLD}}$ pin) [2]
P5_5	Input port (high-impedance)	Inputs the $\overline{\text{HOLD}}$ signal (high-impedance)
P5_6	Input port (high-impedance)	Outputs the $\overline{\text{CS2}}$ signal (high)
P5_7	Input port (high-impedance)	Inputs the $\overline{\text{RDY}}$ signal (high-impedance)
P6 to P19 [3]	Input port (high-impedance)	Input port (high-impedance)

Notes:
1. Whether a pull-up resistor is enabled or not is undefined until the internal voltage is stabilized.
2. State after power is on and the internal voltage has stabilized. It is undefined until the internal voltage is stabilized
3. Ports P16 to P19 are available in the 176-pin package only.

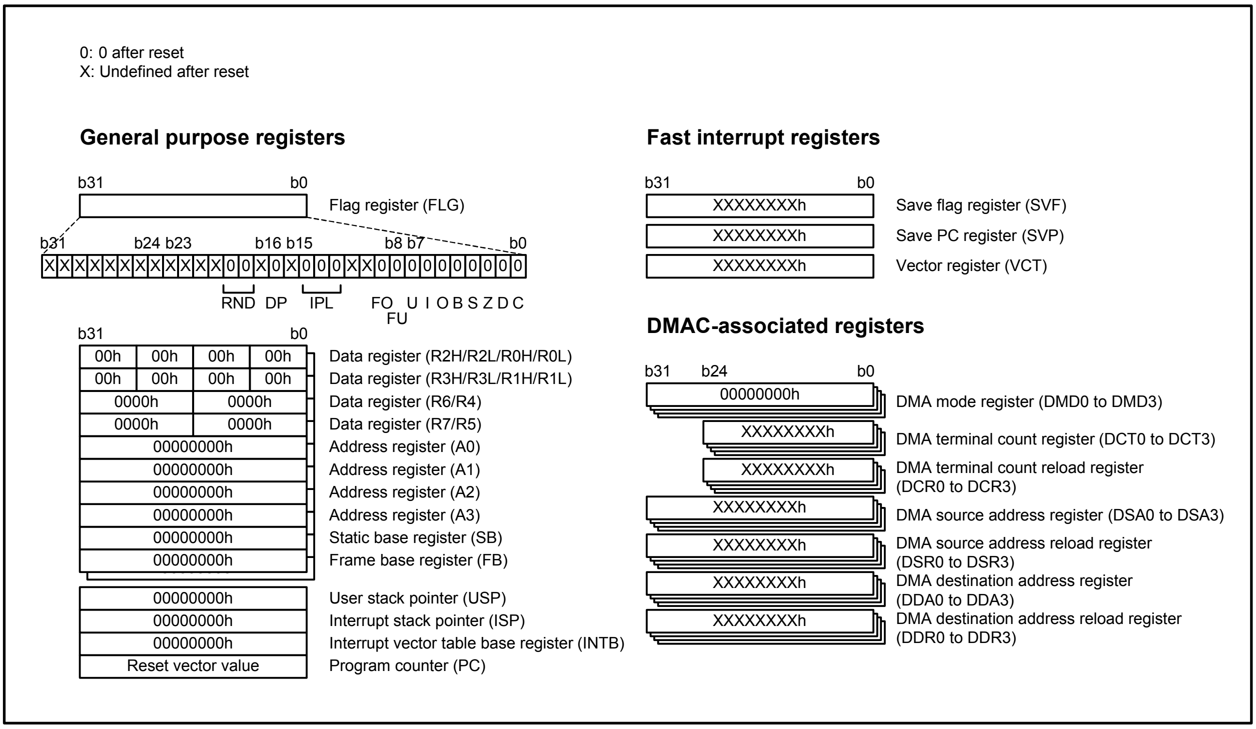


Figure 5.3 CPU Registers after Reset

5.2 Software Reset

The CPU, SFRs, and pins are reset when the PM03 bit in the PM0 register is set to 1 (the MCU is reset). Then, the CPU executes the program from the address indicated by the reset vector.

Set the PM03 bit to 1 while the PLL clock is selected as the CPU clock source and the main clock oscillation is completely stable.

There is no change in processor mode since bits PM01 and PM00 in the PM0 register are not affected by a software reset.

5.3 Watchdog Timer Reset

The CPU, SFRs, and pins are reset when the watchdog timer underflows while the CM06 bit in the CM0 register is 1 (reset when watchdog timer underflows). Then, the CPU executes the program from the address indicated by the reset vector.

There is no change in processor mode since bits PM01 and PM00 in the PM0 register are not affected by a watchdog timer reset.

5.4 Reset Vector

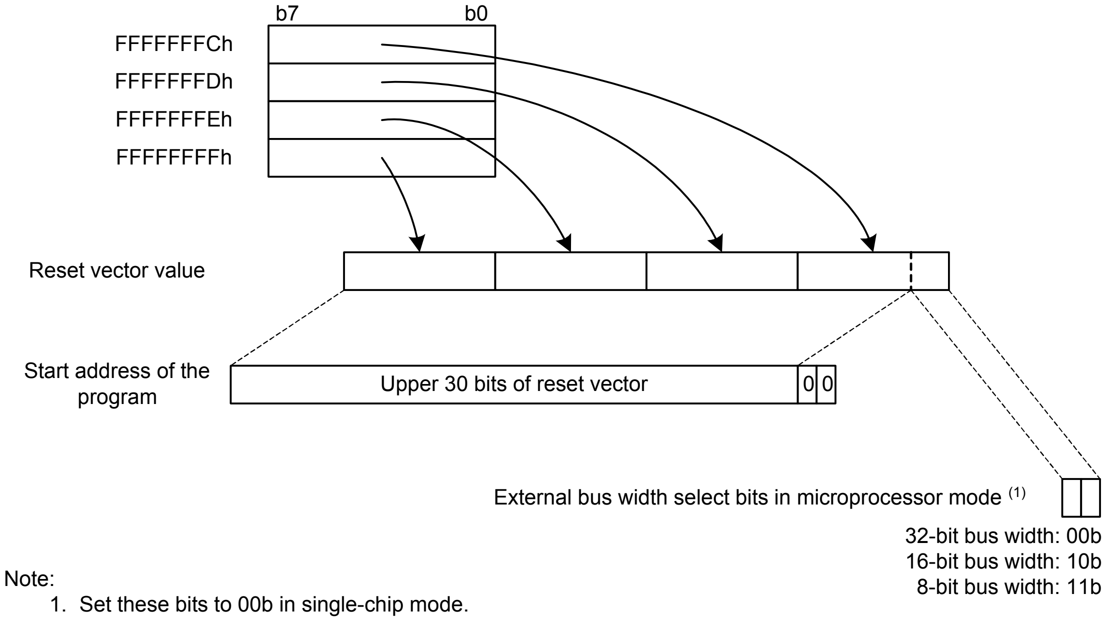
The reset vector in the R32C/100 Series is configured as shown in Figure 5.4.

The start address of a program consists of the upper 30 bits of the reset vector and 00b as lower 2 bits.

The lower 2 bits of the reset vector are bits to select the external bus width in microprocessor mode.

Therefore, the start address of a program requires 4-byte alignment so that the lower 2 bits are 00b.

In single-chip mode, set the external bus width select bits to 00b.



Note:
1. Set these bits to 00b in single-chip mode.

Figure 5.4 Reset Vector Configuration

6. Power Management

6.1 Voltage Regulators for Internal Logic

The supply voltage for internal logic is generated by reducing the input voltage from the VCC pin with the voltage regulators. Figure 6.1 shows a block diagram of the voltage regulators for internal logic, and Figure 6.2 shows the VRCR register.

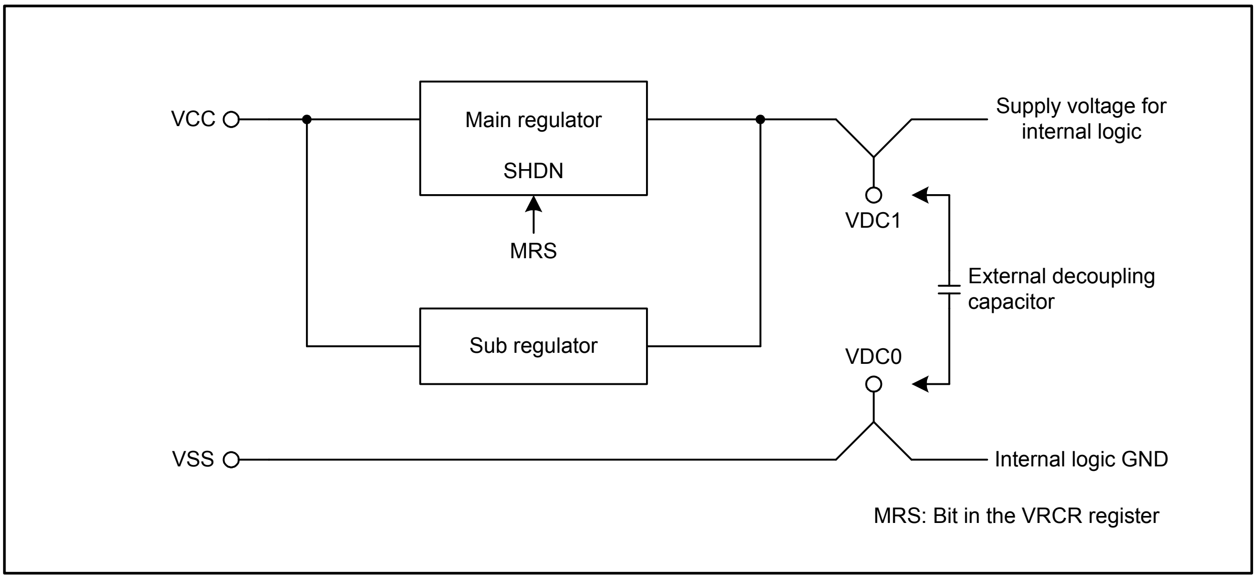
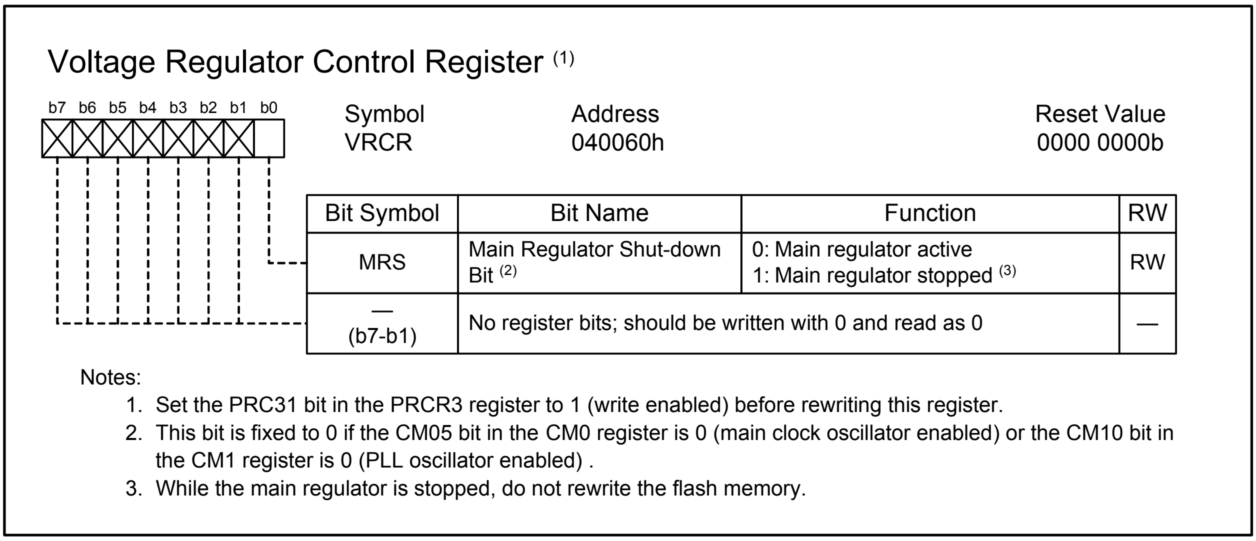

VCC
Main regulator
SHDN
MRS
Sub regulator
Supply voltage for internal logic
VDC1
External decoupling capacitor
VDC0
VSS
Internal logic GND
MRS: Bit in the VRCR register

Figure 6.1 Block Diagram of Voltage Regulators for Internal Logic



Voltage Regulator Control Register (1)

Symbol VRCR

Address 040060h

Reset Value 0000 0000b

Bit Symbol	Bit Name	Function	RW
MRS	Main Regulator Shut-down Bit (2)	0: Main regulator active 1: Main regulator stopped (3)	RW
— (b7-b1)	No register bits; should be written with 0 and read as 0		—

Notes:

1. Set the PRC31 bit in the PRCR3 register to 1 (write enabled) before rewriting this register.
2. This bit is fixed to 0 if the CM05 bit in the CM0 register is 0 (main clock oscillator enabled) or the CM10 bit in the CM1 register is 0 (PLL oscillator enabled) .
3. While the main regulator is stopped, do not rewrite the flash memory.

Figure 6.2 VRCR Register

6.1.1 Decoupling Capacitor

An external decoupling capacitor is required to stabilize internal voltage. The capacitor should be beneficially effective at higher frequencies and maintain a more stable capacitance irrespective of temperature change. In general, ceramic capacitors are recommended. The capacitance varies by conditions such as operating temperature, DC bias, and aging. To select an appropriate capacitor, these conditions should be considered. Also, refer to the recommended capacitor specifications listed in Table 6.1.

The traces between the capacitor and the VDC1/VDC0 pins should be as short and wide as physically possible.

Table 6.1 Recommended Capacitor Specifications

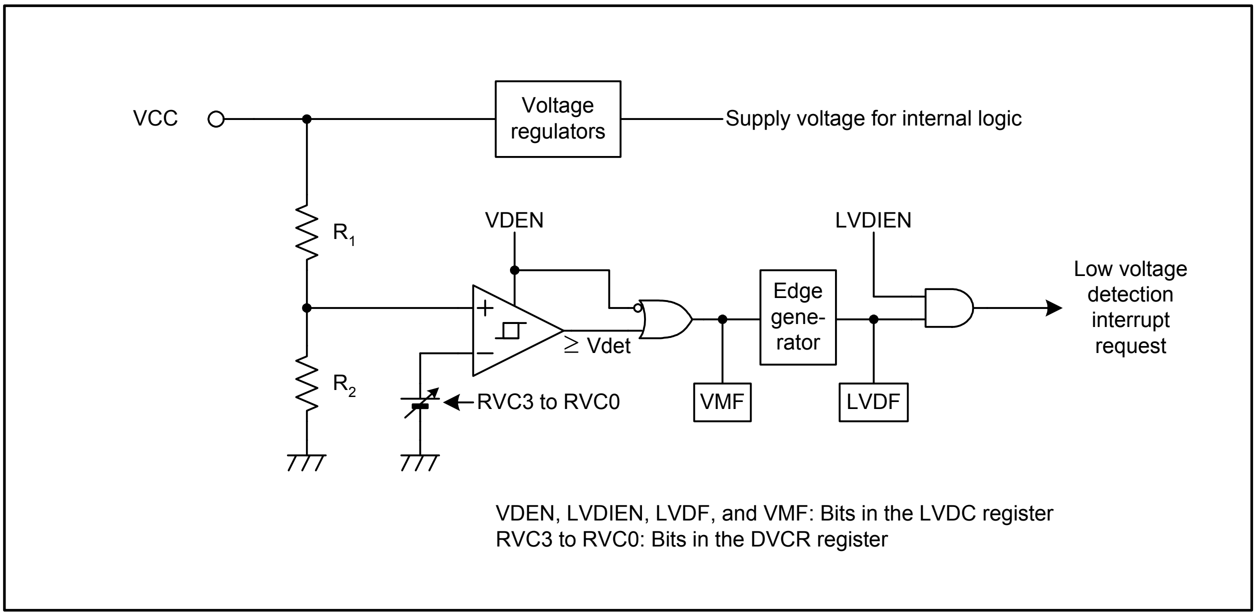
Temperature Characteristics				Rated Voltage (V)	Nominal Capacitance (μF)	Capacitance Tolerance (%)
Applicable standard		Operating temperature range (°C)	Capacitance change (%)			
B	JIS	-25 to 85	±10	6.3 or higher	4.7	±20 or better
R	JIS	-55 to 125	±15	6.3 or higher	4.7	±20 or better
X5R	EIA	-55 to 85	±15	6.3 or higher	4.7	±20 or better
X7R	EIA	-55 to 125	±15	6.3 or higher	4.7	±20 or better
X8R	EIA	-55 to 150	±15	6.3 or higher	4.7	±20 or better
X6S	EIA	-55 to 105	±22	6.3 or higher	4.7	±20 or better
X7S	EIA	-55 to 125	±22	6.3 or higher	4.7	±20 or better

6.2 Low Voltage Detector

The low voltage detector monitors the supply voltage input to the VCC pin.

This circuit is used to monitor the power supply upstream of the voltage regulators for internal logic and provide advanced warning that the power is about to fail. By providing a few milliseconds of advanced warning, the CPU can save any critical parameters to the flash memory and safely shut down.

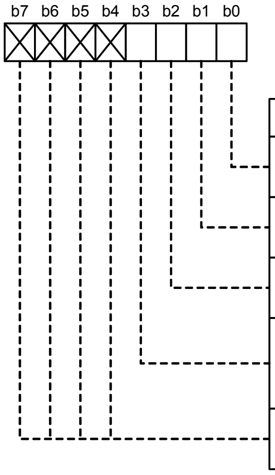
Figure 6.3 shows a block diagram of the low voltage detector, and Figures 6.4 and 6.5 show registers associated with the circuit.



VDEN, LVDIEN, LVDF, and VMF: Bits in the LVDC register
RVC3 to RVC0: Bits in the DVCR register

Figure 6.3 Low Voltage Detector Block Diagram

Low Voltage Detector Control Register [1]



Symbol	Address	Reset Value
LVDC	40062h	0000 XX00b

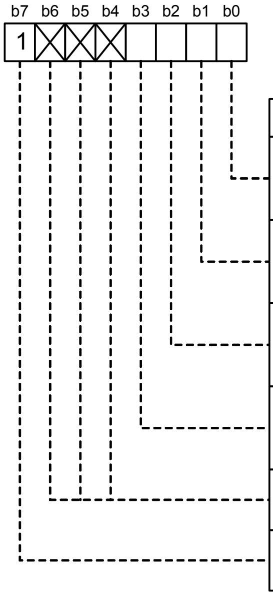
Bit Symbol	Bit Name	Function	RW
VDEN	Low Voltage Detector Enable Bit	0: Low voltage detector disabled 1: Low voltage detector enabled	RW
LVDIEN	Low Voltage Detection Interrupt Enable Bit [2]	0: Interrupt disabled 1: Interrupt enabled	RW
LVDF	Low Voltage Detection Flag [3, 4]	0: Low voltage undetected 1: Low voltage detected (Vdet passed)	RW
VMF	Voltage Monitor Flag [3]	0: VCC < Vdet 1: VCC ≥ Vdet or low voltage detector disabled	RO
— (b7-b4)	No register bits; should be written with 0 and read as 0		—

Notes:

1. Set the PRC31 bit in the PRCR3 register to 1 (write enabled) before rewriting this register.
2. Before setting this bit to 1, set the VDEN bit to 1 first, and wait until the circuit is stabilized.
3. This bit is enabled when the VDEN bit is set to 1.
4. This bit can be set to 0 by a program (Writing 1 to this bit has no effect).

Figure 6.4 LVDC Register

Detection Voltage Configuration Register [1]



Symbol DVCR

Address 40064h

Reset Value 0000 XXXXb

Bit Symbol	Bit Name	Function	RW
RVC0	Reference Voltage Configuration Bit [2]	b3 b2 b1 b0 0 0 0 0 : 3.90 V 0 0 0 1 : 3.75 V 0 0 1 0 : 3.60 V 0 0 1 1 : 3.45 V 0 1 0 0 : 3.30 V 1 0 1 1 : 4.65 V 1 1 0 0 : 4.50 V 1 1 0 1 : 4.35 V 1 1 1 0 : 4.20 V 1 1 1 1 : 4.05 V Only use the combinations listed above	RW
RVC1			RW
RVC2			RW
RVC3			RW
— (b6-b4)	No register bits; should be written with 0 and read as 0		—
— (b7)	Reserved	Should be written with 1	RW

Notes:

1. Set the PRC31 bit in the PRCR3 register to 1 (write enabled) before rewriting this register. Rewrite this register when the VDEN bit in the LVDC register is 0 (low voltage detector disabled).
2. Refer to the following table for detected voltages Vdet(F) and Vdet(R).

Reference Voltage	Low-detection Voltage Vdet(F)	Rise-detection Voltage Vdet(R)
4.65 V	4.55 V	4.77 V
4.50 V	4.40 V	4.62 V
4.35 V	4.24 V	4.46 V
4.20 V	4.09 V	4.31 V
4.05 V	3.95 V	4.17 V
3.90 V	3.80 V	4.02 V
3.75 V	3.65 V	3.87 V
3.60 V	3.50 V	3.72 V
3.45 V	3.35 V	3.57 V
3.30 V	3.20 V	3.42 V

Figure 6.5 DVCR Register

6.2.1 Operational State of Low Voltage Detector

The low voltage detector starts operating stably after td(E-A) when the VDEN bit in the LVDC register is set to 1 (low voltage detector enabled).

When the input voltage to the VCC pin drops below Vdet(F), the VMF bit becomes 0 (VCC < Vdet) and the LVDF bit becomes 1 (low voltage detected (Vdet passed)). At this point an interrupt request is generated when the LVDIEN bit is 1 (low voltage detection interrupt enabled). Set the LVDF bit to 0 (low voltage undetected) by a program.

When the voltage rises to or above Vdet(R) again, the VMF bit becomes 1 (VCC ≥ Vdet) and the LVDF bit becomes 1. At this point an interrupt request is generated when the LVDIEN bit is 1.

Figure 6.6 shows the operation of the low voltage detector.

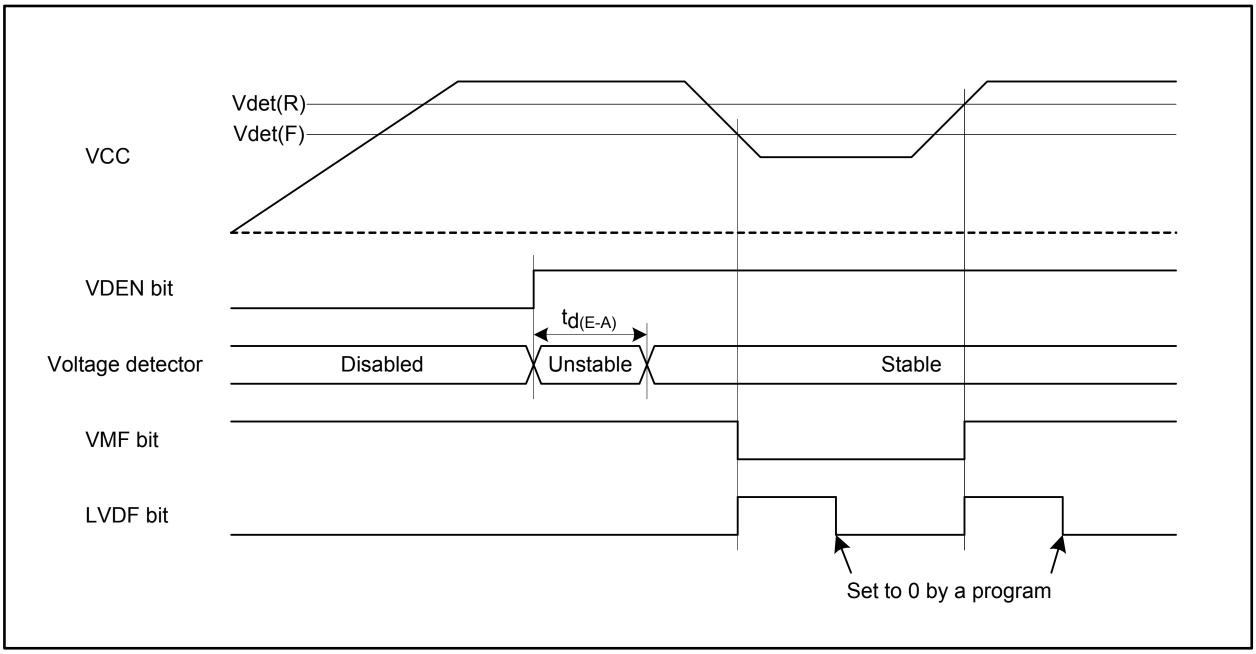


Figure 6.6 Low Voltage Detector Operation

6.2.2 Low Voltage Detection Interrupt

A low voltage detection interrupt request is generated when the input voltage at the VCC pin rises to or above the Vdet(R) level, or falls below the Vdet(F) level while the LVDIEN bit in the LVDC register is 1 (low voltage detection interrupt enabled).

This interrupt shares the interrupt vector with the watchdog timer interrupt and oscillator stop detection interrupt. When using the low voltage detection interrupt with these interrupts at the same time, read the LVDF bit in the LVDC register in the interrupt handler and confirm that the low voltage detection interrupt has been occurred.

The LVDF bit becomes 1 when the input voltage at the VCC pin passes the Vdet(R) level or Vdet(F) level. When the LVDF bit changes from 0 to 1, a low voltage detection interrupt request is generated. Set this bit to 0 (low voltage undetected) by a program.

6.2.3 Application Example of the Low Voltage Detector

Figure 6.7 shows an example of the low voltage detection interrupt.

The supply voltage for internal logic is generated by reducing the input voltage from the VCC pin with the voltage regulators. When the input voltage begins to fall, the internal voltage remains steady. However, as the VCC input voltage continues to fall, the supply voltage for the internal logic also begins to fall, which may affect MCU operation. Consequently, the system can be safely shut down between when the VCC input voltage begins to fall and when the supply voltage for internal logic begins to fall. The low voltage detection interrupt can be applied to detect the falling input voltage.

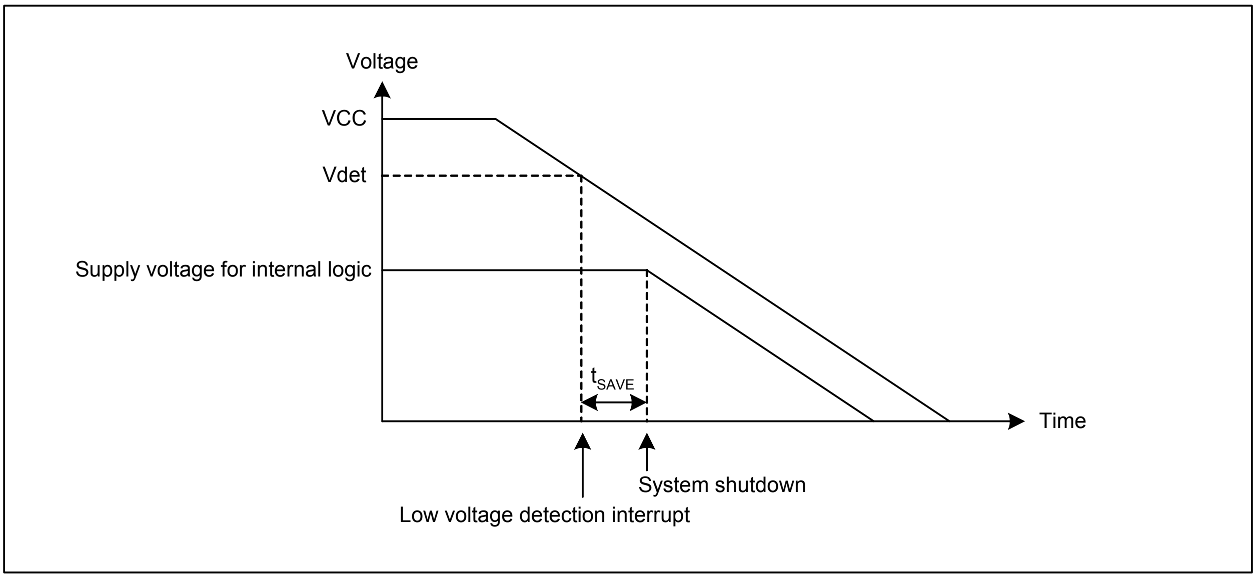

Voltage
VCC
Vdet
Supply voltage for internal logic
t_{SAVE}
Time
System shutdown
Low voltage detection interrupt

Figure 6.7 Example of the Low Voltage Detection Interrupt

7. Processor Mode

7.1 Types of Processor Modes

The R32C/100 Series supports three types of processor modes: single-chip mode, memory expansion mode, and microprocessor mode. Table 7.1 lists the characteristics of each processor mode.

Table 7.1 Processor Mode Characteristics

Processor Mode	Accessible Space	Pin State as I/O Ports
Single-chip mode	SFRs, internal RAM, internal ROM	All pins can be assigned to I/O ports or I/O pins for the peripheral functions
Memory expansion mode	SFRs, internal RAM, internal ROM, external space	Some pins are assigned to bus control pins (1)
Microprocessor mode	SFRs, internal RAM, external space	Some pins are assigned to bus control pins (1)

Note:

1. Refer to 9. “Bus” for details.

The R32C/116A Group supports two standard processor modes: single-chip mode and memory expansion mode. Microprocessor mode is optional. Contact a Renesas Electronics sales office to use this mode.

7.2 Processor Mode Setting

The processor mode to be used is selected by the CNVSS pin state and setting of bits PM01 and PM00 in the PM0 register. After a hardware reset, the operation starts in single-chip mode or microprocessor mode as shown in Table 7.2.

Table 7.2 Processor Mode after Hardware Reset

Input Level into the CNVSS Pin (1)	Processor Mode
Low	Single-chip mode
High	Microprocessor mode

Note:

1. The CNVSS pin should be connected to VCC or VSS via a resistor.

To change to memory expansion mode after starting an operation in single-chip mode, set bits PM01 and PM00 in the PM0 register to 01b (memory expansion mode). Note that the microprocessor mode, selected to start an operation, can be also changed to another mode by setting the bits mentioned above. In this case, however, the internal ROM is inaccessible in every changed mode.

Notes on changing processor mode are as follows:

1. When rewriting bits PM01 and PM00 to 01b (memory expansion mode) or 11b (microprocessor mode), do not change bits PM07 to PM02.
2. When rewriting bits PM07 to PM02, do not change bits PM01 and PM00.
3. Do not change the current mode to microprocessor mode while a program in the internal ROM is being executed.
4. Do not change the current mode to single-chip mode while a program in the external space is being executed.
5. Do not change microprocessor mode to memory expansion mode while a program in the same address as that assigned to the internal ROM is being executed.

Figure 7.1 shows the PM0 register and Figure 7.2 shows the memory map for each processor mode.

Processor Mode Register 0 [1]

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0				

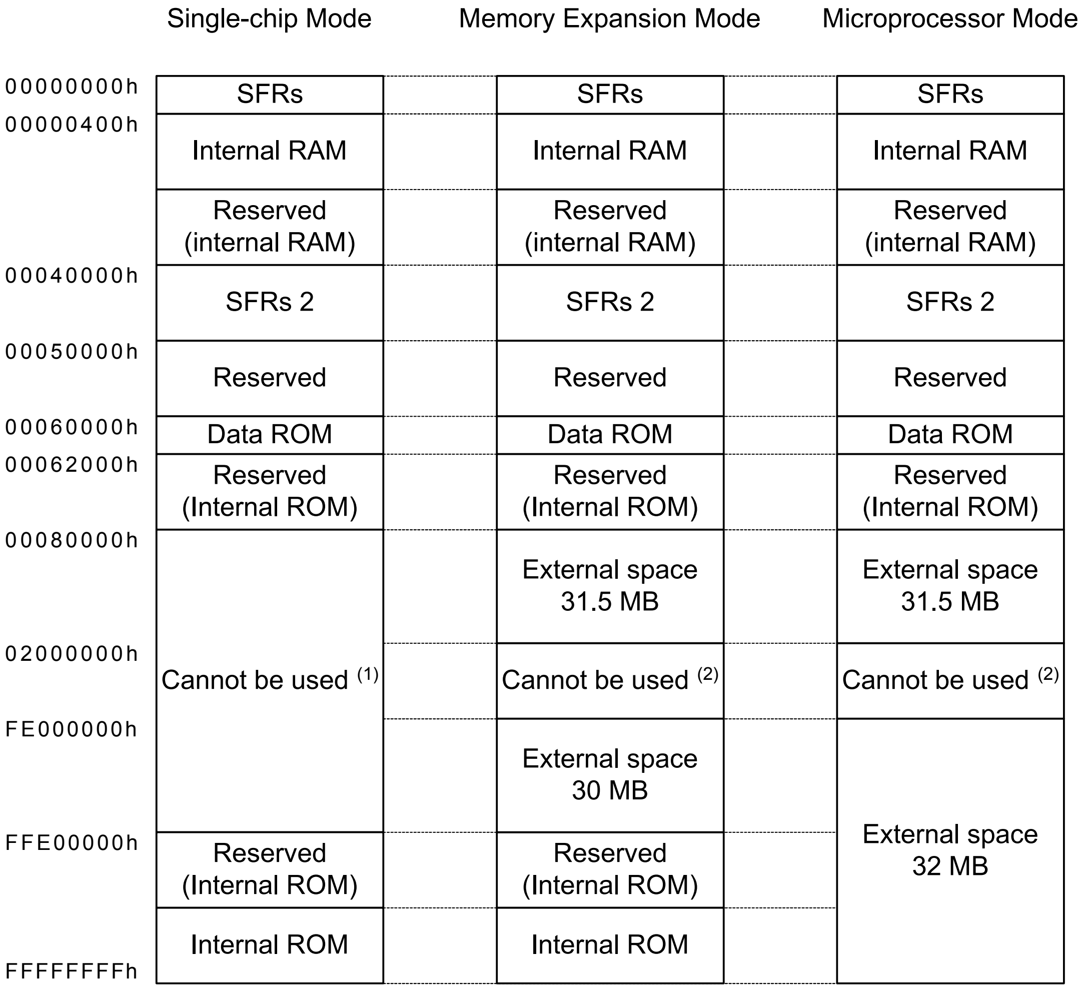
Symbol	Address	Reset Value
PM0	40044h	1000 0000b (CNVSS pin is held low) 0000 0011b (CNVSS pin is held high)

Bit Symbol	Bit Name	Function	RW
PM00	Processor Mode Bit [2, 3]	b1 b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Do not use this combination 1 1 : Microprocessor mode	RW
PM01			RW
PM02	R/W Mode Select Bit	0: $\overline{RD}$ / $\overline{WR}$ / $\overline{BC0}$ / $\overline{BC1}$ / $\overline{BC2}$ / $\overline{BC3}$ 1: $\overline{RD}$ / $\overline{WR0}$ / $\overline{WR1}$ / $\overline{WR2}$ / $\overline{WR3}$	RW
PM03	Software Reset Bit	The MCU is reset when this bit is set to 1. The bit is read as 0	RW
— (b6-b4)	Reserved	Should be written with 0	RW
PM07	BCLK Output Function Select Bit [4]	0: Output BCLK [5] 1: Do not output BCLK. Select a function for port P5_3 using bits CM01 and CM00 in the CM0 register	RW

Notes:

1. Rewrite this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. The processor mode is not changed even when the PM03 bit is set to 1 (software reset).
3. Rewrite bits PM01 and PM00 with 01b or 11b after other bit(s) is/are rewritten. They should not be rewritten simultaneously.
4. In single-chip mode, the BCLK is not output even when the PM07 bit is set to 0.
   To stop clock output in memory expansion mode or microprocessor mode, set the PM07 bit to 1 and bits CM01 and CM00 in the CM0 register to 00b (I/O port P5_3). I/O port P5_3 outputs a low signal in this case.
5. Set bits CM01 and CM00 to 00b when the PM07 bit is 0.

Figure 7.1 PM0 Register



	Single-chip Mode	Memory Expansion Mode	Microprocessor Mode
00000000h	SFRs	SFRs	SFRs
00000400h	Internal RAM	Internal RAM	Internal RAM
	Reserved (internal RAM)	Reserved (internal RAM)	Reserved (internal RAM)
00040000h	SFRs 2	SFRs 2	SFRs 2
00050000h	Reserved	Reserved	Reserved
00060000h	Data ROM	Data ROM	Data ROM
00062000h	Reserved (Internal ROM)	Reserved (Internal ROM)	Reserved (Internal ROM)
00080000h	Cannot be used (1)	External space 31.5 MB	External space 31.5 MB
02000000h		Cannot be used (2)	Cannot be used (2)
FE000000h		External space 30 MB	External space 32 MB
FFE00000h	Reserved (Internal ROM)	Reserved (Internal ROM)	
	Internal ROM	Internal ROM	
FFFFFFFFh			

Notes:

1. This space cannot be externally expanded in single-chip mode.
2. This space cannot be used in any processor mode.

Figure 7.2 Memory Map of Each Processor Mode

8. Clock Generator

8.1 Clock Generator Types

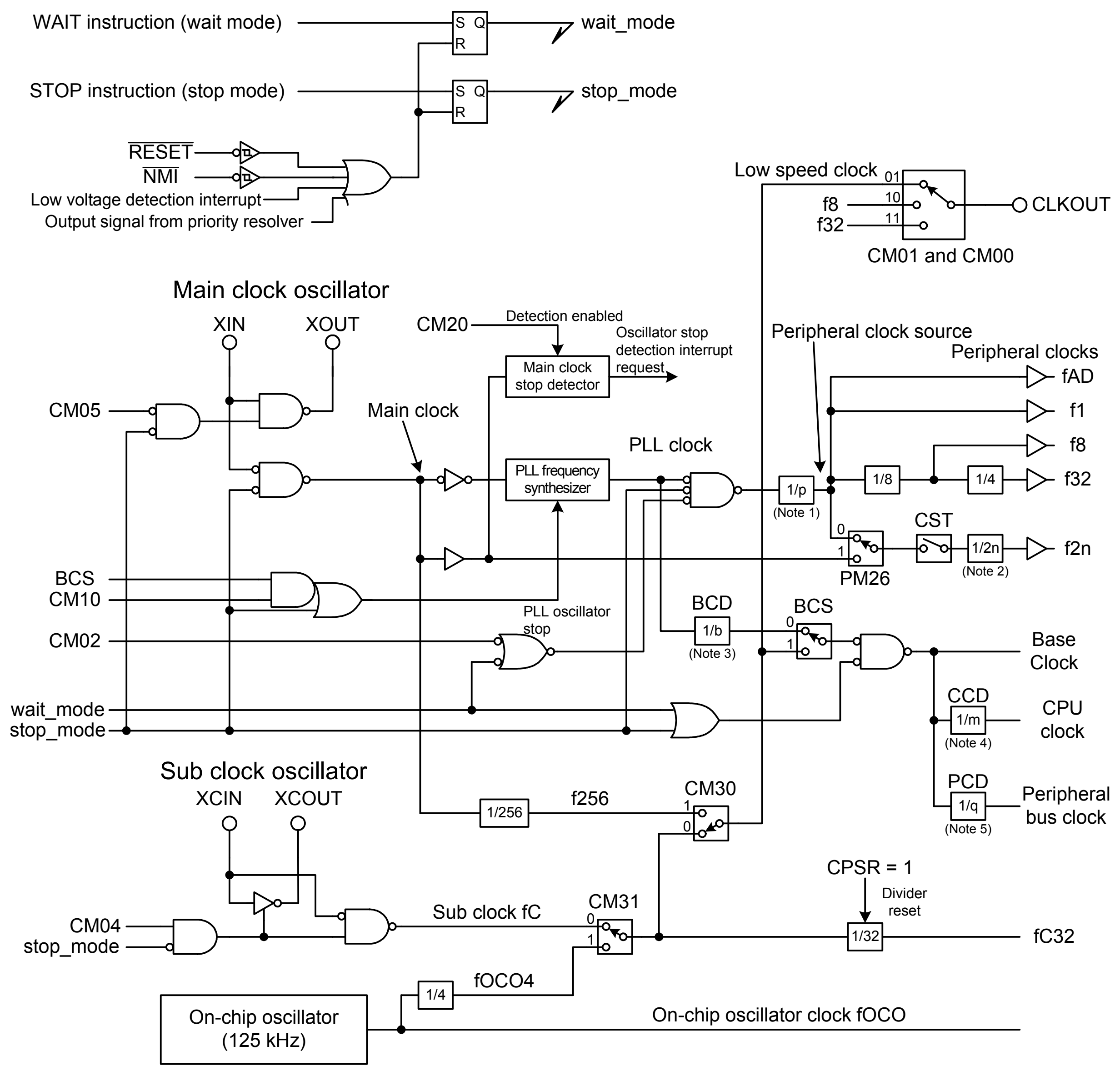
The clock generator consists of four circuits:

- Main clock oscillator
- Sub clock oscillator
- PLL frequency synthesizer
- On-chip oscillator (OCO)

Table 8.1 lists the specifications of the clock generator. Figure 8.1 shows a block diagram of the clock generator, and Figures 8.2 to 8.10 show registers associated with clock control.

Table 8.1 Clock Generator Specifications

Item	Main Clock Oscillator	Sub Clock Oscillator	PLL Frequency Synthesizer	On-chip Oscillator
Used as	• PLL reference clock source • Peripheral clock source	• CPU clock source • Clock source for timers A and B	• CPU clock source • Peripheral clock source	• CPU clock source • Clock source for timers A and B
Clock frequency	4 to 16 MHz	32.768 kHz	$f_{SO(PLL)}$ or $f_{(PLL)}$	Approx. 125 kHz
Connectable oscillators or additional circuits	Ceramic resonator Crystal oscillator	Crystal oscillator	—	—
Pins for oscillators or additional circuits	XIN, XOUT	XCIN, XCOUT	—	—
Oscillator stop/ restart function	Available	Available	Available	Available
Oscillator state after a reset	Running	Stopped	Running	Stopped
Note	Externally generated clock can be input	Externally generated clock can be input	When the main clock oscillator stops running, the PLL frequency synthesizer oscillates at its own frequency of $f_{SO(PLL)}$	The on-chip oscillator starts running by setting the CSPM bit in the OFS area to 0 after a reset



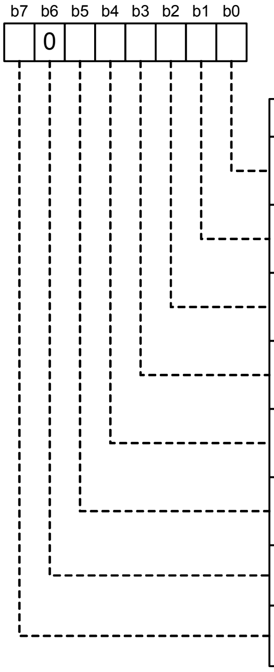
CM00 to CM02, CM04, and CM05: Bits in the CM0 register
CM10: Bit in the CM1 register
CM20: Bit in the CM2 register
CM30 and CM31: Bits in the CM3 register
PM26: Bit in the PM2 register
CST: Bit in the TCSPR register
CPSR: Bit in the CPSRF register
BCS: Bit in the CCR register

Notes:

1. The value of p can be selected by setting bits PM36 and PM35 in the PM3 register (p = 2, 4, 6, 8).
2. The value of n can be selected by setting bits CNT3 to CNT0 in the TCSPR register (n = 0 to 15). When n is 0, the clock is not divided.
3. The value of b can be selected by setting bits BCD1 and BCD0 in the CCR register (b = 2, 3, 4, 6).
4. The value of m can be selected by setting bits CCD1 and CCD0 in the CCR register (m = 1 to 4).
5. The value of q can be selected by setting bits PCD1 and PCD0 in the CCR register (q = 2 to 4).

Figure 8.1 Clock Generation Circuitry

Clock Control Register [1]



Symbol
CCR

Address
0004h

Reset Value
0001 1000b

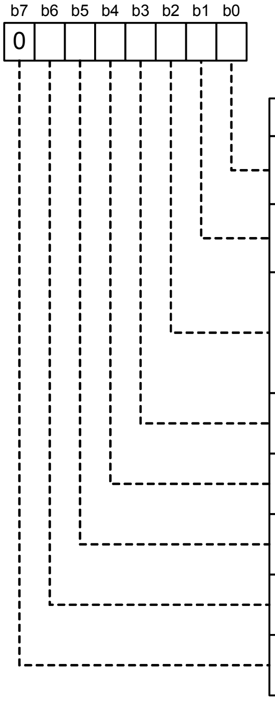
Bit Symbol	Bit Name	Function	RW
BCD0	Base Clock Divide Ratio Select Bit [2]	b1 b0 0 0 : Divide-by-6 0 1 : Divide-by-4 1 0 : Divide-by-3 1 1 : Divide-by-2	RW
BCD1			RW
CCD0	CPU Clock Divide Ratio Select Bit [3]	b3 b2 0 0 : Divide-by-4 0 1 : Divide-by-3 1 0 : Divide-by-2 1 1 : No division	RW
CCD1			RW
PCD0	Peripheral Bus Clock Divide Ratio Select Bit [2, 3, 4]	b5 b4 0 0 : Do not use this combination 0 1 : Divide-by-2 1 0 : Divide-by-3 1 1 : Divide-by-4	RW
PCD1			RW
— (b6)	Reserved	Should be written with 0	RW
BCS	Base Clock Source Select Bit	0: PLL clock 1: fC, fOCO4, or f256 [5, 6]	RW

Notes:

1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The divide ratios of the base clock and peripheral bus clock should not be changed simultaneously. Doing so may cause the peripheral bus clock frequency to go over the maximum operating frequency.
3. The divide ratio of the CPU clock should be equal to or lower than that of peripheral bus clock.
4. Set this bit only once after a reset and do not change the setting afterwards. Rewrite the PBC register before rewriting this bit.
5. To set this bit to 1, a 32-bit write access to addresses 0004h to 0007h should be performed.
6. To use these low speed clocks, select one of them by setting bits CM31 and CM30 in the CM3 register and then set the BCS bit to 1.

Figure 8.2 CCR Register

System Clock Control Register 0 (1)



Symbol	Address	Reset Value
CM0	40046h	0000 1000b

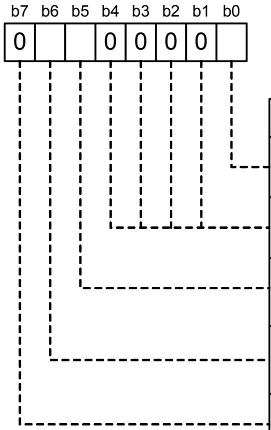
Bit Symbol	Bit Name	Function	RW
CM00	Clock Output Function Select Bit (2)	b1 b0 0 0 : I/O port P5_3 0 1 : Output a low speed clock 1 0 : Output f8 1 1 : Output f32	RW
CM01			RW
CM02	Peripheral Clock Source Stop Bit (3)	0: Peripheral clock source not stopped in wait mode 1: Peripheral clock source stopped in wait mode (4)	RW
CM03	XCIN-XCOUT Drive Strength Select Bit (5)	0: Low 1: High	RW
CM04	Port XC Switch Bit	0: I/O port 1: XCIN-XCOUT oscillator (6)	RW
CM05	Main Clock Oscillator (XIN-XOUT) Stop Bit (3, 7)	0: Main clock oscillator enabled 1: Main clock oscillator disabled	RW
CM06	Watchdog Timer Function Select Bit (8)	0: Watchdog timer interrupt 1: Reset (9)	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:

1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. When the PM07 bit in the PM0 register is 0 (output BCLK), bits CM01 and CM00 should be set to 00b. In memory expansion mode, when the PM07 bit is 1 (select a function for port P5_3 using bits CM01 and CM00 in the CM0 register) and bits CM01 and CM00 are set to 00b, the P5_3 pin is driven low (this pin does not function as port P5_3).
3. When the PM21 bit in the PM2 register is 1 (clock change disabled), bits CM02 bit and CM05 cannot be changed by a write access.
4. fC32 and f2n (whose clock source is the main clock) do not stop.
5. When entering stop mode, the CM03 bit becomes 1.
6. To set the CM04 bit to 1, set bits PD8_7 and PD8_6 in the PD8 register to 0 (input), and the PU25 bit in the PUR2 register to 0 (pull-up resistor disabled).
7. This bit stops the main clock when entering low power mode. It cannot detect whether or not the main clock oscillator stops. When this bit is set to 1, the clock applied to the XOUT pin becomes high. Since the on-chip feedback resistor remains connected, the XIN pin is connected to the XOUT pin via the feedback resistor.
8. Set this bit before activating the watchdog timer. When rewriting this bit while the watchdog timer is running, set it immediately after writing to the WDTS register.
9. Once this bit is set to 1, it cannot be set to 0 by a program.

Figure 8.3 CM0 Register

System Clock Control Register 1 (1)



Symbol
CM1

Address
40047h

Reset Value
0010 0000b

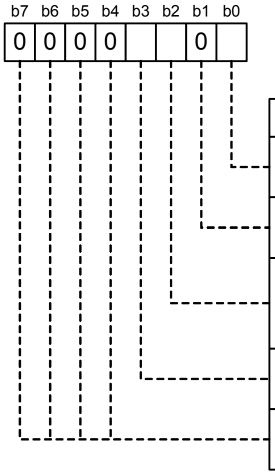
Bit Symbol	Bit Name	Function	RW
CM10	PLL Oscillator Stop Bit (2, 3)	0: PLL oscillator enabled 1: PLL oscillator disabled	RW
— (b4-b1)	Reserved	Should be written with 0	RW
CM15	XIN-XOUT Drive Strength Select Bit (4)	b6 b5 0 0 : Low 0 1 : High 1 0 : Super low (5) 1 1 : Do not use this combination	RW
CM16			RW
— (b7)	Reserved	Should be written with 0	RW

Notes:

1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. When the BCS bit in the CCR register is 0 (PLL clock selected as base clock source), the PLL frequency synthesizer does not stop oscillating even if the CM10 bit is set to 1.
3. When the PM21 bit in the PM2 register is 1 (clock change disabled), the CM10 bit cannot be changed by a write access.
4. These bits become 01b when the main clock is stopped. When setting to 00b or 10b, rewrite them after the main clock is fully stabilized.
5. The oscillator frequency should be 8 MHz or less to select super low mode.

Figure 8.4 CM1 Register

Oscillator Stop Detection Register [1]



Symbol CM2

Address 4004Dh

Reset Value 0000 0000b

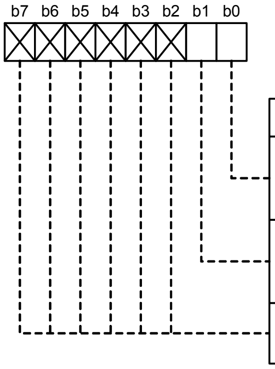
Bit Symbol	Bit Name	Function	RW
CM20	Oscillator Stop Detection Enable Bit [2, 3]	0: Disable oscillator stop detection 1: Enable oscillator stop detection	RW
— (b1)	Reserved	Should be written with 0	RW
CM22	Oscillator Stop Detection Flag [4]	0: Main clock oscillator has not been stopped 1: Main clock oscillator stop detected	RW
CM23	Main Clock Monitor Flag [5]	0: Main clock oscillator active 1: Main clock oscillator stopped	RO
— (b7-b4)	Reserved	Should be written with 0	RW

Notes:

1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. Set this bit to 0 when f256 is selected as the base clock source in low speed mode.
3. When the PM21 bit in the PM2 register is 1 (clock change disabled), the CM20 bit cannot be changed by a write access.
4. When a main clock oscillator stop is detected, this bit becomes 1. It can be set to 0 by a program, however not to 1. When it is set to 0 while the main clock oscillator is stopped, it does not become 1 until the next main clock oscillator stop is detected.
5. After an oscillator stop detection interrupt occurs, read this bit several times to determine the main clock state.

Figure 8.5 CM2 Register

Low Speed Mode Clock Control Register [1]



Symbol CM3

Address 4005Ah

Reset Value XXXX XX00b

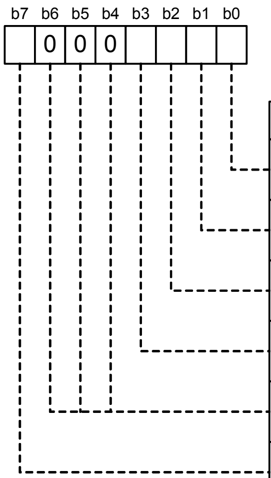
Bit Symbol	Bit Name	Function	RW
CM30	Low Speed Mode Base Clock Select Bit	b1 b0 0 0 : fC 0 1 : f256 (main clock divided by 256) 1 0 : fOCO4 (on-chip oscillator divided by 4) [2, 3] 1 1 : Do not use this combination	RW
CM31			RW
— (b7-b2)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Rewrite this register after setting the PRC27 bit in the PRCR2 register to 1 (write enabled) and while the BCS bit in the CCR register is 0 (PLL clock).
2. When the PM22 bit in the PM2 register is 1 (count source protected) and the PM23 bit or WDK4 bit in the WDK register is 1 (on-chip oscillator clock), the CM31 bit is not changed by a write access.
3. The on-chip oscillator clock starts when the CM31 bit is set to 1.

Figure 8.6 CM3 Register

Count Source Prescaler Register



Symbol TCSPR | Address 035Fh | Reset Value 0000 0000b

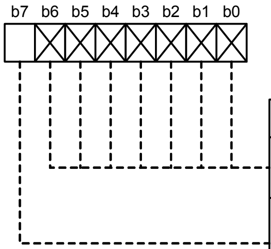
Bit Symbol	Bit Name	Function	RW
CNT0	Divide Ratio Select Bit [1]	f2n is either the main clock or peripheral clock source divided by 2n. If n = 0, the clock is not divided (n = setting value)	RW
CNT1			RW
CNT2			RW
CNT3			RW
— (b6-b4)	Reserved	Should be written with 0	RW
CST	Divider Operation Enable Bit	0: Stop divider operation 1: Start divider operation	RW

Note:

1. Set the CST bit to 0 before rewriting bits CNT3 to CNT0.

Figure 8.7 TCSPR Register

Clock Prescaler Reset Register

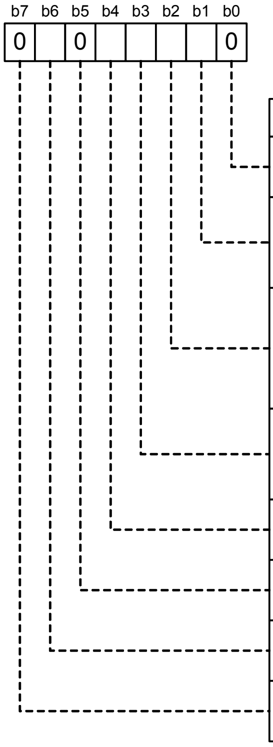


Symbol CPSRF | Address 0341h | Reset Value 0XXX XXXXb

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	No register bits; should be written with 0 and read as undefined value		—
CPSR	Clock Prescaler Reset Bit	When this bit is set to 1, the fC divided-by-32 divider is initialized. The bit is read as 0	RW

Figure 8.8 CPSRF Register

Processor Mode Register 2 [1]



Symbol
PM2

Address
40053h

Reset Value
0000 0000b

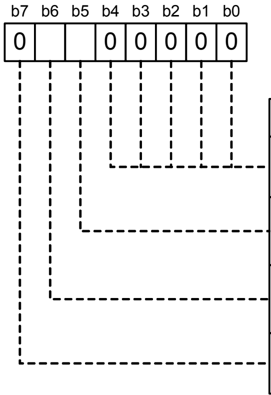
Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	RW
PM21	System Clock Protect Bit [2, 3]	0: Protect the clock by the PRCR register 1: Clock change disabled	RW
PM22	Watchdog Timer Count Source Protect Bit [2, 4]	The count source of watchdog timer is 0: Unprotected 1: Protected [5]	RW
PM23	Watchdog Timer Count Source Select Bit [2, 4, 6]	0: Peripheral bus clock or prescaler cascaded on-chip oscillator clock 1: On-chip oscillator clock	RW
PM24	NMI Enable Bit [2]	0: NMI disabled [7] 1: NMI enabled	RW
— (b5)	Reserved	Should be written with 0	RW
PM26	f2n Clock Source Select Bit [8]	0: Peripheral clock source 1: Main clock	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:

1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. Once this bit is set to 1, it cannot be set to 0 by a program.
3. When the PM21 bit is set to 1, the following bits cannot be changed by a write access:
   CM02 bit in the CM0 register (the peripheral clock source state in wait mode)
   CM05 bit in the CM0 register (main clock oscillator enabled/disabled)
   CM10 bit in the CM1 register (PLL oscillator enabled/disabled)
   CM20 bit in the CM2 register (oscillator stop detection enabled/disabled)
4. This bit should be set before the watchdog timer activates.
5. The count source protect mode is enabled only when the PM23 or WDK4 bit in the WDK register is 1 (on-chip oscillator clock selected). When this mode is enabled, the CM31 bit in the CM3 register is not changed by a write access.
6. When the PM23 bit is set to 1, the on-chip oscillator starts running.
7. When the PM24 bit is 0, the forced cutoff of the three-phase motor control timers is also disabled.
8. Stop all the peripherals that use f2n before rewriting this bit.

Figure 8.9 PM2 Register

Processor Mode Register 3 (1)



Symbol PM3

Address 40048h

Reset Value 0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Reserved	Should be written with 0	RW
PM35	Peripheral Clock Source Divide Ratio Select Bit (2)	b6 b5 0 0 : Divide-by-8 0 1 : Divide-by-6 1 0 : Divide-by-4 1 1 : Divide-by-2	RW
PM36			RW
— (b7)	Reserved	Should be written with 0	RW

Notes:

1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register. Stop all the peripherals that use fAD, f1, f8, f32, or f2n (when the clock source is the peripheral clock source) to rewrite this register.
2. Select a divide ratio so that the peripheral clock source frequency does not exceed the maximum value specified in the electrical characteristics

Figure 8.10 PM3 Register

The following sections illustrate clocks generated in clock generators.

8.1.1 Main Clock

The main clock is generated by the main clock oscillator. This clock can be a clock source for the PLL reference clock or peripheral clocks.

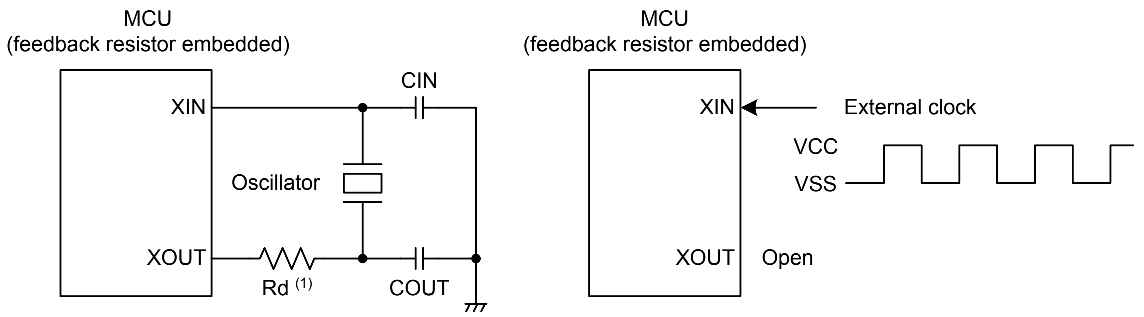
The main clock oscillator is configured with two pins, XIN and XOUT, connected by an oscillator or resonator. The circuit has an on-chip feedback resistor which is separated from the oscillator in stop mode to save power consumption. An external clock can be applied to the XIN pin in this circuit. Figure 8.11 shows an example of a main clock circuit connection.

Circuit constants vary depending on the oscillator. Circuit constants should be set as per the oscillator manufacturer’s recommendations.

After a reset, the main clock oscillator is still independently active and disconnected from the PLL frequency synthesizer. A PLL frequency synthesizer self-oscillating clock divided by 12 is provided to the CPU.

Setting the CM05 bit in the CM0 register to 1 (main clock oscillator disabled) enables power-saving. In this case, the clock applied to the XOUT pin becomes high. The XIN pin connected to the XOUT pin by an embedded feedback resistor is also driven high. Do not set the CM05 bit to 1 when an external clock is applied to the XIN pin.

All clocks, including the main clock, stop in stop mode. Refer to 8.7 “Power Control” for details.



Note:

1. Insert a damping resistor if required. Resistance values may vary according to oscillator setting. Values recommended by the manufacturer should be set. A feedback resistor should be placed between XIN and XOUT if the manufacturer recommends placing a resistor externally.

Figure 8.11 Main Clock Circuit Connection

8.1.2 Sub Clock (fC)

The sub clock is generated by the sub clock oscillator. This clock can be a clock source for the CPU clock and a count source for timers A and B. It can be output from the CLKOUT pin.

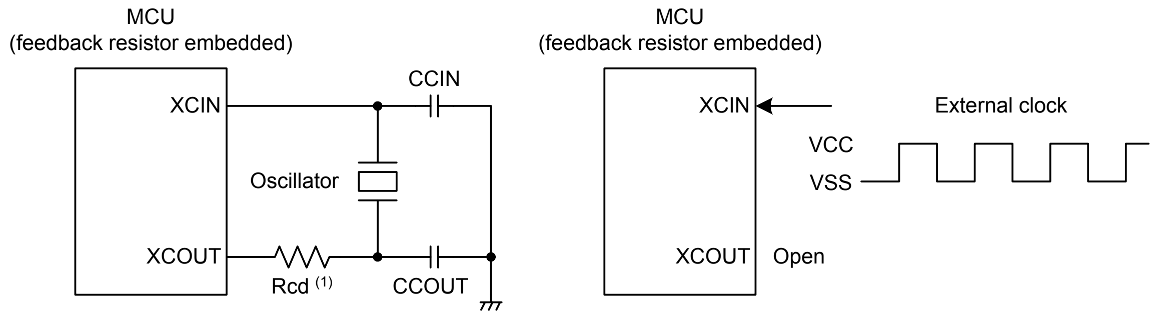
The sub clock oscillator is configured with pins XCIN and XCOUT connected by a crystal oscillator. The circuit has a on-chip feedback resistor which is separated from the oscillator in stop mode to save power consumption. An external clock can be applied to the XCIN pin. Figure 8.12 shows an example of a sub clock circuit connection. Circuit constants vary depending on the oscillator. Circuit constants should be set as per the oscillator manufacturer’s recommendations.

After a reset, the sub clock is stopped and the feedback resistor is separated from the oscillator. In order to start the sub clock oscillation, first set bits PD8_6 and PD8_7 in the PD8 register to 0 (input mode), and the PU25 bit in the PUR2 register to 0 (pull-up resistor disabled). Then, set the CM04 bit in the CM0 register to 1 (XCIN-XCOUT oscillator).

To input an external clock to the XCIN pin, set bits PD8_7 and PU25 to 0 and then the CM04 bit to 1. The clock applied to the XCIN pin becomes a clock source for the sub clock.

When the CM3 register is set to 00h (fC) and the BCS bit in the CCR register is set to 1 (fC, fOCO4, or f256) after the sub clock oscillation has stabilized, the sub clock becomes the base clock of the CPU clock and the peripheral bus clock.

All clocks, including the sub clock, stop in stop mode. Refer to 8.7 “Power Control” for details.



Note:

1. Insert a damping resistor if required. Resistance values vary according to oscillator setting. Values recommended by the manufacturer should be set. A feedback resistor should be placed between XCIN and XCOUT if the manufacturer recommends placing a resistor externally.

Figure 8.12 Sub Clock Circuit Connection

8.1.3 PLL Clock

The PLL clock is generated by the PLL frequency synthesizer based on the main clock. This clock can be a clock source for any clock including the CPU clock and the peripheral clock.

Figure 8.13 shows a block diagram of the PLL frequency synthesizer. Figures 8.14 and 8.15 show registers PLC0 and PLC1, respectively.

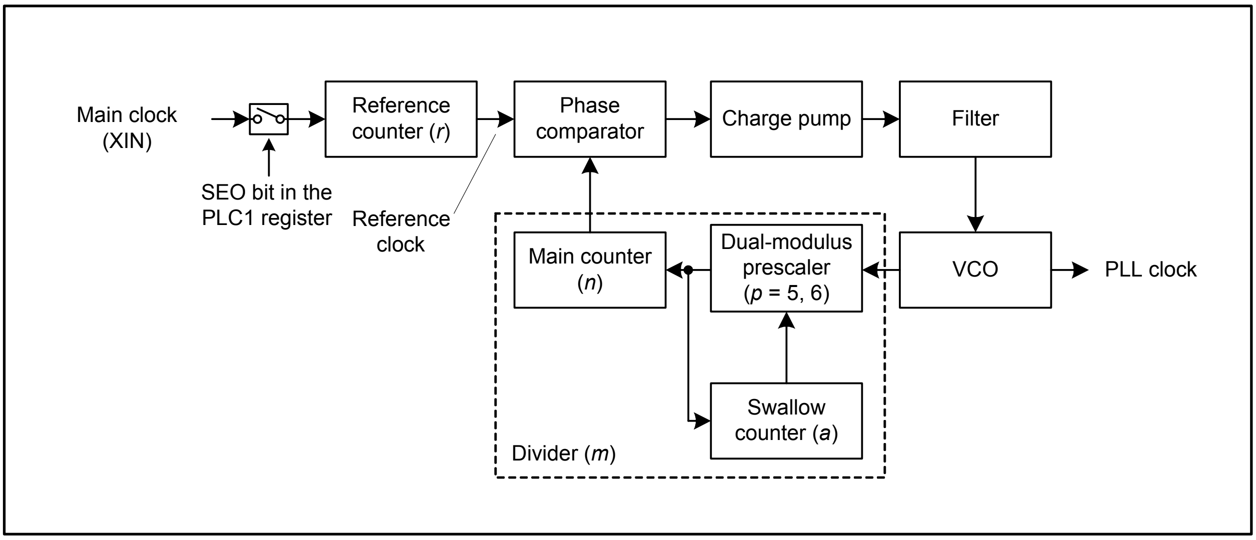


Figure 8.13 PLL Frequency Synthesizer Block Diagram

PLL Control Register 0 (1, 2)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
PLC0	40020h	0000 0001b

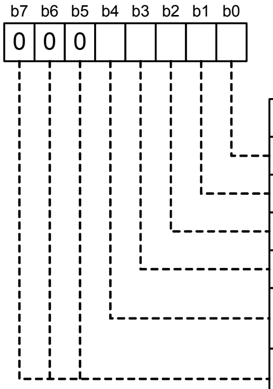
Bit Symbol	Bit Name	Function	RW
MCV0	Main Counter Divide Ratio Setting Bit	Set the bits to *n* - 1 (*n* = divide ratio of the main counter)	RW
MCV1			RW
MCV2			RW
MCV3			RW
MCV4			RW
SCV0	Swallow Counter Divide Ratio Setting Bit	The divide ratio of the dual-modulus prescaler is 6 (in *a* out of *n* times) or 5 (in other cases) (*a* = setting value)	RW
SCV1			RW
SCV2			RW

Notes:

1. Set the PRC2 bit in the PRCR register to 1 (write enabled) just before rewriting this register. No interrupt handling or DMA transfers should be inserted between these two instructions.
2. This register can be rewritten only once after a reset.

Figure 8.14 PLC0 Register

PLL Control Register 1 (1)



Symbol PLC1 | Address 40021h | Reset Value 0001 1111b

Bit Symbol	Bit Name	Function	RW
RCV0	Reference Counter Divide Ratio Setting Bit	Set the bits to *r* - 1 (*r* = divide ratio of the main clock)	RW
RCV1			RW
RCV2			RW
RCV3			RW
SEO	Self-oscillating Setting Bit	0: PLL lock-in 1: Self-oscillating	RW
— (b7-b5)	Reserved	Should be written with 0	RW

Note:

1. Set the PRC2 bit in the PRCR register to 1 (write enabled) just before rewriting this register. No interrupt handling or DMA transfers should be inserted between these two instructions.

Figure 8.15 PLC1 Register

In the PLL frequency synthesizer, the pulse-swallow operation is implemented. The divide ratio *m* is simply expressed by *n*×*p*. However, with the swallow counter, the divide ratio *p* is 6 in *a* out of *n*, or 5 in other cases, the actual *m* is therefore given by the formula below:

$$\begin{aligned} m &= n \times p \\ &= n \times \left(\frac{a}{n} \cdot 6 + \frac{n-a}{n} \cdot 5\right) \\ &= 5n + a \end{aligned}$$

The setting range of *a* is $0 \le a < 5$, $0 \le a \le n$.

As *r* is the divide ratio of the reference counter, the PLL clock has a *m/r* times the main clock (XIN) frequency.

$$\begin{aligned} \text{PLL clock frequency } f(PLL) &= \frac{m}{r} \cdot \text{main clock frequency} \\ &= \frac{5n+a}{r} \cdot \text{main clock frequency} \end{aligned}$$

After a reset, the reference counter is divided by 16, and the PLL frequency synthesizer is multiplied by 10. Since the main clock as a reference clock is disconnected, the PLL frequency synthesizer may self-oscillate at its own frequency of $f_{SO(PLL)}$.

Each register should be set to meet the following conditions:

-The reference clock, which is the main clock divided by *r*, should be between 2 to 4 MHz.

-The divide ratio *m* is $25 \le m \le 100$.

For the setting of registers PLC1 and PLC0, Table 8.2 should be applied. While the main clock oscillation is stable, a wait time of $t_{LOCK(PLL)}$ is necessary between rewriting registers PLC1 and PLC0, and the PLL clock becoming stable.

Table 8.2 PLC1 and PLC0 Register Settings (1)

Main Clock	*r*	Reference Clock	*n*	*a*	*m*	PLC1 Register Setting	PLC0 Register Setting	*m/r*	PLL Clock
4 MHz	2	2 MHz	9	3	48	01h	68h	24	96 MHz
6 MHz	2	3 MHz	6	2	32	01h	45h	16	96 MHz
8 MHz	3	2.6667 MHz	7	1	36	02h	26h	12	96 MHz
10 MHz	5	2 MHz	9	3	48	04h	68h	9.6	96 MHz
12 MHz	4	3 MHz	6	2	32	03h	45h	8	96 MHz
16 MHz	5	3.2 MHz	6	0	30	04h	05h	6	96 MHz
4 MHz	1	4 MHz	5	0	25	00h	04h	25	100 MHz
6 MHz	3	2 MHz	10	0	50	02h	09h	16.6667	100 MHz
8 MHz	2	4 MHz	5	0	25	01h	04h	12.5	100 MHz
10 MHz	3	3.3333 MHz	6	0	30	02h	05h	10	100 MHz
12 MHz	3	4 MHz	5	0	25	02h	04h	8.3333	100 MHz
16 MHz	4	4 MHz	5	0	25	03h	04h	6.25	100 MHz
4 MHz	1	4 MHz	6	0	30	00h	05h	30	120 MHz
6 MHz	2	3 MHz	8	0	40	01h	07h	20	120 MHz
8 MHz	2	4 MHz	6	0	30	01h	05h	15	120 MHz
10 MHz	3	3.3333 MHz	7	1	36	02h	26h	12	120 MHz
12 MHz	3	4 MHz	6	0	30	02h	05h	10	120 MHz
16 MHz	4	4 MHz	6	0	30	03h	05h	7.5	120 MHz
4 MHz	1	4 MHz	6	2	32	00h	45h	32	128 MHz
6 MHz	3	2 MHz	12	4	64	02h	8Bh	21.3333	128 MHz
8 MHz	2	4 MHz	6	2	32	01h	45h	16	128 MHz
10 MHz	5	2 MHz	12	4	64	04h	8Bh	12.8	128 MHz
12 MHz	3	4 MHz	6	2	32	02h	45h	10.6667	128 MHz
16 MHz	4	4 MHz	6	2	32	03h	45h	8	128 MHz

Note:

1. Registers PLC1 and PLC0 should be set according to the list above.

8.1.4 On-chip Oscillator Clock

The on-chip oscillator clock is generated by the on-chip oscillator (OCO). This clock can be a clock source for the CPU clock and a count source for timers A and B. This clock has a frequency of approximately 125 kHz. The on-chip oscillator clock divided by 4 can be used as the base clock for the CPU clock and peripheral bus clock.

When the WCSS bit in the OFS area is 1 and the WPCS bit is 0, the on-chip oscillator clock is stopped after a reset. It starts running when setting any of the following bits to 1; the CM31 bit in the CM3 register, the PM23 bit in the PM2 register, or the WDK4 bit in the WDK register. It is not necessary to wait for stabilization because the on-chip oscillator instantly starts oscillating.

8.2 Oscillator Stop Detection

This function detects the main clock is stopped when its oscillator stops running due to an external factor. When the CM20 bit in the CM2 register is 1 (enable oscillator stop detection), an oscillator stop detection interrupt request is generated as soon as the main clock stops. Simultaneously, the PLL frequency synthesizer starts to self-oscillate at its own frequency. If the PLL frequency synthesizer is the clock source for CPU clock and peripheral clock, these clocks continue running.

When an oscillator stop is detected, the following bits in the CM2 register become 1:

- The CM22 bit: main clock oscillator stop detected
- The CM23 bit: main clock oscillator stopped

8.2.1 How to Use Oscillator Stop Detection

The oscillator stop detection interrupt shares vectors with the watchdog timer interrupt and the low voltage detection interrupt. When using these interrupts simultaneously, read the CM22 bit with an interrupt handler to determine if an oscillator stop detection interrupt request has been generated.

When the main clock oscillator resumes running after an oscillator stop is detected, the PLL clock frequency may temporarily exceed the preset value before the PLL frequency synthesizer oscillation stabilizes. As soon as an oscillator stop is detected, the main clock oscillator should be stopped from resuming (set the CM05 bit in the CM0 register to 1) or the divide ratios of the base clock and peripheral clock source should be increased by a program. They can be set using bits BCD1 and BCD0 in the CCR register and bits PM36 and PM35 in the PM3 register.

In low speed mode, when the main clock oscillator stops running, an oscillator stop detection interrupt request is generated if the CM20 bit is set to 1 (enable oscillator stop detection). The CPU clock remains running with a low speed clock source. Note that if the base clock is f256, which is the main clock divided by 256, oscillator stop detection cannot be used.

The oscillator stop detection is provided to handle main clock stop caused by external factors. To stop the main clock oscillator by a program, i.e., to enter stop mode or to set the CM05 bit to 1 (main clock oscillator disabled), the CM20 bit in the CM2 register should be set to 0 (disable oscillator stop detection). To enter wait mode, this bit should be also set to 0.

The oscillator stop detection functions depending on the voltage of a capacitor which is being changed. In more concrete terms, this function detects that the oscillator is stopped when the main clock goes lower than approximately 500 kHz. Note that if the CM22 bit is set to 0 by a program in an interrupt handler while the frequency is around 500 kHz, a stack overflow may occur due to multiple interrupt requests.

8.3 Base Clock

The base clock is a reference clock for the CPU clock and peripheral bus clock. The base clock after a reset is the PLL clock divided by 6.

The base clock source is selected between the PLL clock and the low speed clocks which contain the sub clock (fC), on-chip oscillator clock divided by 4 (fOCO4), and main clock divided by 256 (f256).

If the PLL clock is selected, it is divided by 2, 3, 4, or 6 to become the base clock. If a low speed clock is selected, the clock itself can be the base clock.

The base clock source is set using the BCS bit in the CCR register and the divide ratio for the PLL clock is set using bits BCD1 and BCD0. Bits CM31 and CM30 in the CM3 register select a low speed clock.

8.4 CPU Clock and Peripheral Bus Clock

The CPU operating clock is referred to as the CPU clock. The CPU clock after a reset is the base clock divided by 2.

The CPU clock source is the base clock, and its divide ratio is selected by setting bits CCD1 and CCD0 in the CCR register. The base clock divided by 2 to 4 becomes the peripheral bus clock. Its divide ratio is selected by setting bits PCD1 and PCD0 in the CCR register. The peripheral bus clock also functions as a count source for the watchdog timer.

In memory expansion mode or microprocessor mode, the peripheral bus clock can be output as BCLK from the BCLK pin. This clock is used as a reference clock for external timing generation. Refer to 8.6 “Clock Output Function” for details.

To prevent the CPU clock, whose clock source is the PLL clock, from stopping when the CPU becomes out of control, set the following while the CM05 bit in the CM0 register is 0 (main clock oscillator enabled) and the BCS bit in the CCR register is 0 (PLL clock selected as base clock source):

(1) Set the PRC1 bit in the PRCR register to 1 (write enabled to the PM2 register).

(2) Set the PM21 bit in the PM2 register to 1 (clock change disabled).

8.5 Peripheral Clock

The peripheral clock is an operating clock or a count source for the peripherals excluding the watchdog timer. The source of this clock is generated by a clock, which has the same frequency as the PLL clock, divided by 2, 4, 6, or 8 according to the settings of bits PM36 and PM35 in the PM3 register. The peripheral clock is classified into three types of clock as follows:

(1) f1, f8, f32, f2n

f1, f8, and f32 are the peripheral clock sources divided by 1, 8, and 32, respectively. The clock source for f2n is selected between the peripheral clock source and the main clock by setting the PM26 bit in the PM2 register. The f2n divide ratio can be set using bits CNT3 to CNT0 in the TCSPR register (n = 1 to 15, not divided when n = 0).

f1, f8, f32, and f2n, whose clock source is the peripheral clock source, stop in low power mode or when the CM02 bit is set to 1 (peripheral clock source stopped in wait mode) to enter wait mode.

f1, f8, and f2n are used as a count source for timers A and B or an operating clock for the serial interface. f1 is used as an operating clock for the intelligent I/O as well.

f8 and f32 can be output from the CLKOUT pin. Refer to 8.6 “Clock Output Function” for details.

(2) fAD

fAD, which has the same frequency as peripheral clock source, is an operating clock for the A/D converter.

This clock stops in low power mode or when the CM02 bit is set to 1 (peripheral clock source stopped in wait mode) to enter wait mode.

(3) fC32

fC32, which is a sub clock divided by 32, or on-chip oscillator clock divided by 128, is used as the count source for timers A and B. This clock is available when the sub clock or on-chip oscillator clock is active.

8.6 Clock Output Function

Low speed clocks, f8, and f32 can be output from the CLKOUT pin.

In memory expansion mode or microprocessor mode, the BCLK, that is, the peripheral bus clock which is the base clock divided by 2 to 4 can also be output from the BCLK pin.

Tables 8.3 and 8.4 list the CLKOUT pin functions in single-chip mode and memory expansion mode or microprocessor mode, respectively.

Table 8.3 CLKOUT Pin Functions in Single-chip Mode

PM0 Register (1)	CM0 Register (2)		CLKOUT Pin Function
PM07	CM01	CM00	
0 or 1	0	0	I/O port P5_3
1	0	1	Output a low speed clock
1	1	0	Output f8
1	1	1	Output f32

Notes:

1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.

Table 8.4 CLKOUT Pin Functions in Memory Expansion Mode or Microprocessor Mode

PM0 Register (1)	CM0 Register (2)		CLKOUT Pin Function
PM07	CM01	CM00	
0	0 (3)	0 (3)	Output BCLK
1	0	0	Output low (not function as P5_3)
1	0	1	Output a low speed clock
1	1	0	Output f8
1	1	1	Output f32

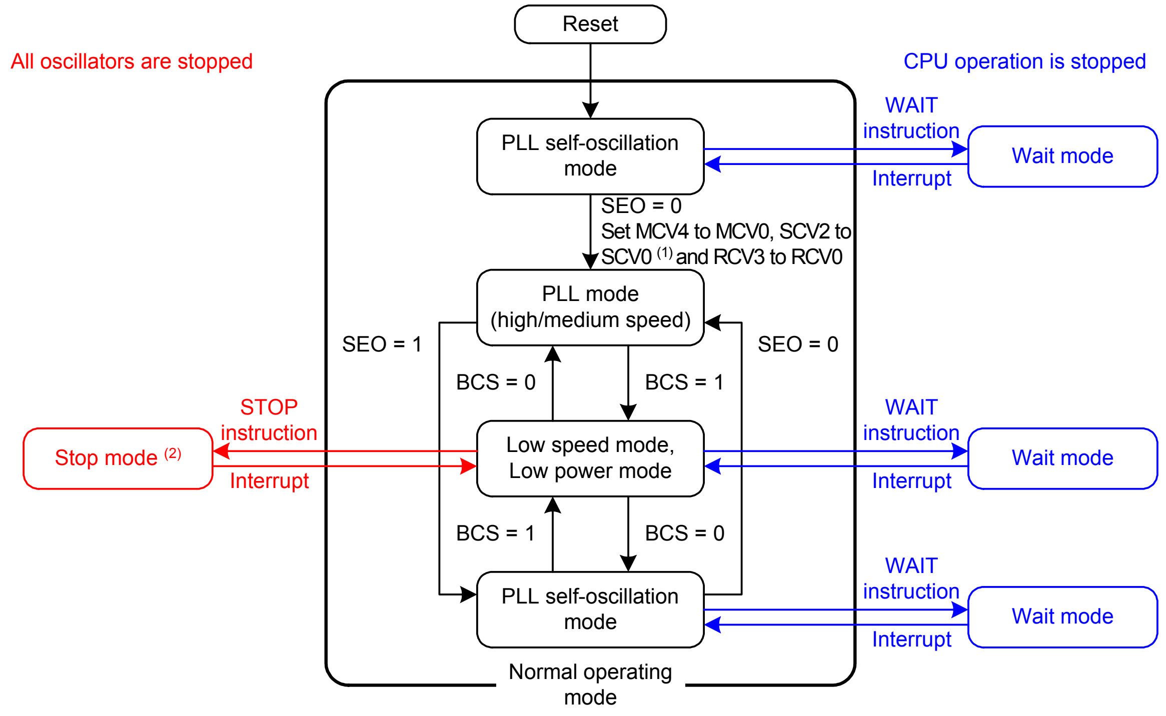
Notes:

1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
3. When the PM07 bit is set to 0 (output BCLK), set bits CM01 and CM00 to 00b (I/O port P5_3).

8.7 Power Control

Power control has three modes: wait mode, stop mode, and normal operating mode.

The name “normal operating mode” is used restrictively in this chapter, and it indicates all other modes except wait mode and stop mode. Figure 8.16 shows a block diagram of the state transition in normal operating mode, stop mode, and wait mode.



BCS: Bit in the CCR register
MCV4 to MCV0, SCV2 to SCV0: Bits in the PLC0 register
SEO, RCV3 to RCV0: Bits in the PLC1 register

Notes:

1. The PLC0 register can be set only once after a reset.
2. When the sub clock is selected as the base clock source, do not enter stop mode.

Figure 8.16 State Transition in Stop Mode and Wait Mode

8.7.1 Normal Operating Mode

Normal operating mode is classified into the five modes shown below.

In normal operating mode, the CPU clock and peripheral clock are provided to operate the CPU and peripherals. Power consumption is controlled by the CPU clock frequency. The higher the CPU clock frequency is, the more processing power increases. The lower the CPU clock frequency is, the less power consumption is required. Power consumption can be reduced by stopping oscillators that are not being used.

(1) PLL Mode (high speed mode)

In this mode, the PLL clock is selected as the base clock source, and the main clock is provided as the reference clock source for the PLL frequency synthesizer. High speed mode enables the CPU to operate at the maximum operating frequency. The PLL clock divided by 2 becomes the base clock. The base clock frequency should be identical to that of the CPU clock. fAD, f1, f8, f32, and f2n can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as the count source for timers A and B.

(2) PLL Mode (medium speed mode)

This mode indicates all modes in PLL mode except high speed mode. The PLL clock divided by 2, 3, 4, or 6 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock. fAD, f1, f8, f32, and f2n can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as the count source for timers A and B.

(3) Low Speed Mode

In this mode, a low speed clock is used as the base clock source. The low speed clock becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock. fAD, f1, f8, f32, and f2n can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as the count source for timers A and B.

(4) Low Power Mode

This is a state where the main clock oscillator and the PLL frequency synthesizer are stopped after switching to low speed mode. The sub clock or the on-chip oscillator clock divided by 4 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock. fC32, which is the only peripheral clock available, can be used as the count source for timers A and B. By setting the MRS bit in the VRCR register to 1 (main regulator stopped), this mode consumes even less power than the modes above.

(5) PLL Self-oscillation Mode

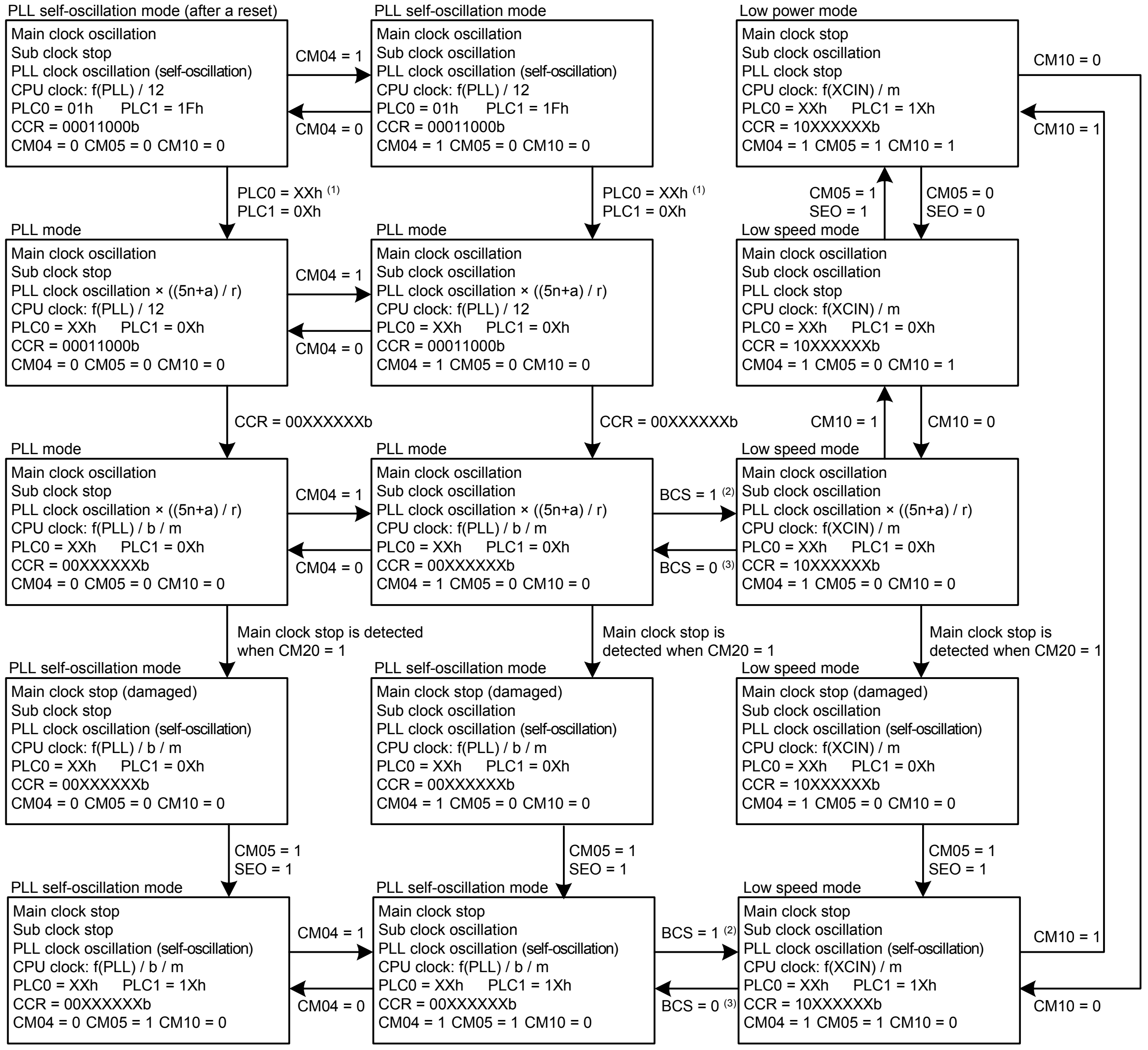
In this mode, the PLL clock is selected as the base clock source, and the main clock is not provided as the reference clock source for the PLL frequency synthesizer. The PLL frequency synthesizer self-oscillates at its own frequency. The PLL clock divided by 2, 3, 4, or 6 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock. fAD, f1, f8, f32, and f2n can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as the count source for timers A and B.

The state transition within normal operating mode can be very complicated; therefore only the block diagrams of typical state transitions are shown. Figures 8.17 to 8.19 show block diagrams of the respective state transitions: state when the sub clock is used, state when the main clock divided by 256 is used, and state when the on-chip oscillator clock is used. As for the state transitions other than the above, setting of each register and the usage notes below can be used as references.

- PLL can be switched from PLL oscillating to self-oscillating by setting the SEO bit in the PLC1 register to 1. Set the SEO bit to 1 (self-oscillating) before setting the CM05 bit in the CM0 register to 0 (main clock oscillator disabled) to stop the main clock.
- The divide ratio of the clock should be increased and the frequency should be decreased by using bits BCD1 to BCD0 in the CCR register or bits PM36 to PM35 in the PM3 register before setting the SEO bit to 0 (PLL oscillating) in order to switch back PLL self-oscillation mode to PLL mode. Set back the settings of bits BCD1 to BCD0 and bits PM36 to PM35 once PLL oscillation is stabilized after setting the SEO bit to 0.
- Before switching the CPU clock to another clock, that clock should be stabilized. In particular, the sub clock oscillator may require more time to stabilize [1]. Therefore, certain waiting time to switch should be taken by a program immediately after turning the MCU on or exiting stop mode.

Note:

1. Contact the oscillator manufacturer for details on oscillator stabilization time.



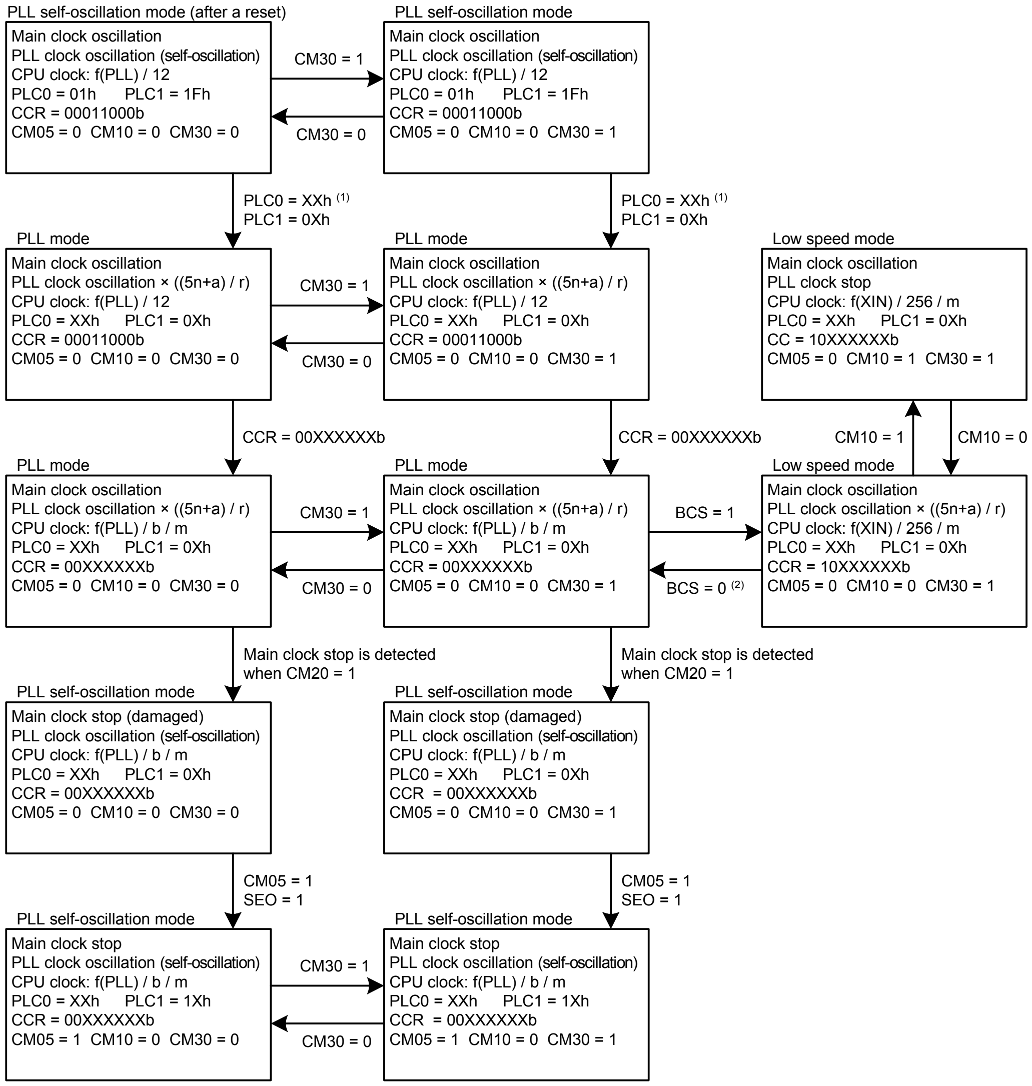
→ : Arrows indicate a one-way transition between modes. No transition should be made unless indicated.

BCS: Bit in the CCR register
CM04 and CM05: Bits in the CM0 register
CM10: Bit in the CM1 register
CM20: Bit in the CM2 register
SEO: Bit in the PLC1 register
PLC0 = XXh: The multiplication ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.
PLC1 = 0Xh: The divisor of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.
CCR = 00XXXXXXb: The divisor of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum values specified in the electrical characteristics.

Notes:

1. The PLC0 register can be set only once after reset is released.
2. This clock should be switched after the sub clock oscillation is fully stabilized.
3. This clock should be switched after the PLL clock oscillation is fully stabilized.

Figure 8.17 State Transition When Using the Sub Clock


PLL self-oscillation mode (after a reset)
Main clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / 12
PLC0 = 01h PLC1 = 1Fh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM30 = 0
CM30 = 1
CM30 = 0
PLL self-oscillation mode
Main clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / 12
PLC0 = 01h PLC1 = 1Fh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM30 = 1
PLC0 = XXh (1)
PLC1 = 0Xh
PLC0 = XXh (1)
PLC1 = 0Xh
PLL mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / 12
PLC0 = XXh PLC1 = 0Xh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM30 = 0
CM30 = 1
CM30 = 0
PLL mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / 12
PLC0 = XXh PLC1 = 0Xh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM30 = 1
Low speed mode
Main clock oscillation
PLL clock stop
CPU clock: f(XIN) / 256 / m
PLC0 = XXh PLC1 = 0Xh
CC = 10XXXXXXb
CM05 = 0 CM10 = 1 CM30 = 1
CCR = 00XXXXXXb
CCR = 00XXXXXXb
CM10 = 1
CM10 = 0
PLL mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 0
CM30 = 1
CM30 = 0
PLL mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 1
BCS = 1
BCS = 0 (2)
Low speed mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(XIN) / 256 / m
PLC0 = XXh PLC1 = 0Xh
CCR = 10XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 1
Main clock stop is detected
when CM20 = 1
Main clock stop is detected
when CM20 = 1
PLL self-oscillation mode
Main clock stop (damaged)
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 0
PLL self-oscillation mode
Main clock stop (damaged)
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 1
CM05 = 1
SEO = 1
CM05 = 1
SEO = 1
PLL self-oscillation mode
Main clock stop
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 1Xh
CCR = 00XXXXXXb
CM05 = 1 CM10 = 0 CM30 = 0
CM30 = 1
CM30 = 0
PLL self-oscillation mode
Main clock stop
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 1Xh
CCR = 00XXXXXXb
CM05 = 1 CM10 = 0 CM30 = 1

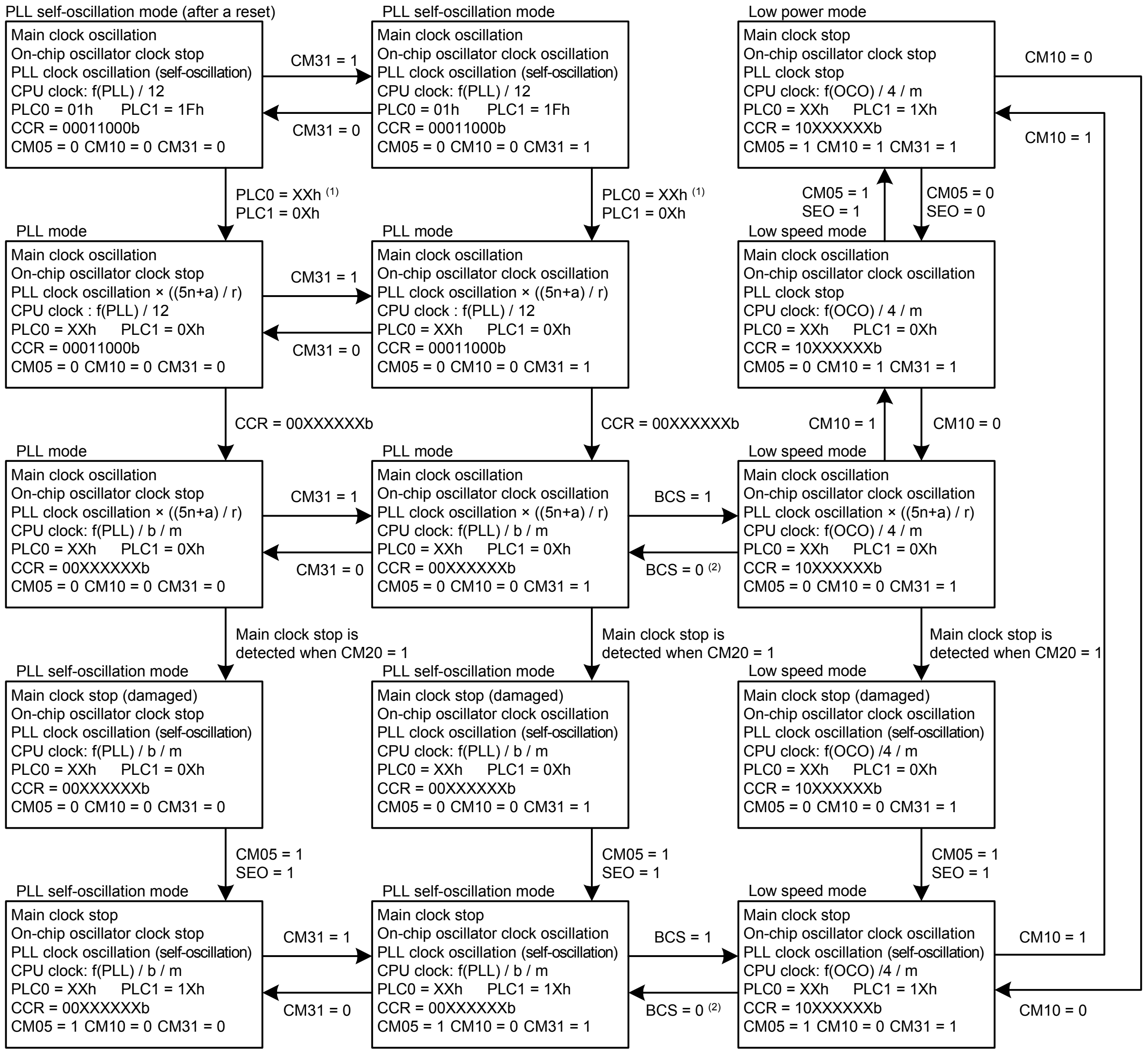
→ : Arrows indicate a one-way transition between modes. No transition should be made unless indicated.

BCS: Bit in the CCR register
CM05: Bit in the CM0 register
CM10: Bit in the CM1 register
CM20: Bit in the CM2 register
CM30: Bit in the CM3 register
SEO: Bit in the PLC1 register
PLC0 = XXh: The multiplication ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.
PLC1 = 0Xh: The divisor of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.
CCR = 00XXXXXXb: The divisor of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum values specified in the electrical characteristics.

Notes:

1. The PLC0 register can be set only once after reset is released.
2. This clock should be switched after the PLL clock oscillation is fully stabilized.

Figure 8.18 State Transition When Using the Main Clock Divided by 256


PLL self-oscillation mode (after a reset)
Main clock oscillation
On-chip oscillator clock stop
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / 12
PLC0 = 01h PLC1 = 1Fh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM31 = 0
CM31 = 1
CM31 = 0
PLL self-oscillation mode
Main clock oscillation
On-chip oscillator clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / 12
PLC0 = 01h PLC1 = 1Fh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM31 = 1
Low power mode
Main clock stop
On-chip oscillator clock stop
PLL clock stop
CPU clock: f(OCO) / 4 / m
PLC0 = XXh PLC1 = 1Xh
CCR = 10XXXXXXb
CM05 = 1 CM10 = 1 CM31 = 1
CM10 = 0
CM10 = 1
PLC0 = XXh (1)
PLC1 = 0Xh
PLC0 = XXh (1)
PLC1 = 0Xh
CM05 = 1
SEO = 1
CM05 = 0
SEO = 0
PLL mode
Main clock oscillation
On-chip oscillator clock stop
PLL clock oscillation × ((5n+a) / r)
CPU clock : f(PLL) / 12
PLC0 = XXh PLC1 = 0Xh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM31 = 0
CM31 = 1
CM31 = 0
PLL mode
Main clock oscillation
On-chip oscillator clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock : f(PLL) / 12
PLC0 = XXh PLC1 = 0Xh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM31 = 1
Low speed mode
Main clock oscillation
On-chip oscillator clock oscillation
PLL clock stop
CPU clock: f(OCO) / 4 / m
PLC0 = XXh PLC1 = 0Xh
CCR = 10XXXXXXb
CM05 = 0 CM10 = 1 CM31 = 1
CCR = 00XXXXXXb
CCR = 00XXXXXXb
CM10 = 1
CM10 = 0
PLL mode
Main clock oscillation
On-chip oscillator clock stop
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM31 = 0
CM31 = 1
CM31 = 0
PLL mode
Main clock oscillation
On-chip oscillator clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM31 = 1
BCS = 1
BCS = 0 (2)
Low speed mode
Main clock oscillation
On-chip oscillator clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(OCO) / 4 / m
PLC0 = XXh PLC1 = 0Xh
CCR = 10XXXXXXb
CM05 = 0 CM10 = 0 CM31 = 1
Main clock stop is
detected when CM20 = 1
Main clock stop is
detected when CM20 = 1
Main clock stop is
detected when CM20 = 1
PLL self-oscillation mode
Main clock stop (damaged)
On-chip oscillator clock stop
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM31 = 0
PLL self-oscillation mode
Main clock stop (damaged)
On-chip oscillator clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM31 = 1
Low speed mode
Main clock stop (damaged)
On-chip oscillator clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(OCO) /4 / m
PLC0 = XXh PLC1 = 0Xh
CCR = 10XXXXXXb
CM05 = 0 CM10 = 0 CM31 = 1
CM05 = 1
SEO = 1
CM05 = 1
SEO = 1
CM05 = 1
SEO = 1
PLL self-oscillation mode
Main clock stop
On-chip oscillator clock stop
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 1Xh
CCR = 00XXXXXXb
CM05 = 1 CM10 = 0 CM31 = 0
CM31 = 1
CM31 = 0
PLL self-oscillation mode
Main clock stop
On-chip oscillator clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 1Xh
CCR = 00XXXXXXb
CM05 = 1 CM10 = 0 CM31 = 1
BCS = 1
BCS = 0 (2)
Low speed mode
Main clock stop
On-chip oscillator clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(OCO) /4 / m
PLC0 = XXh PLC1 = 1Xh
CCR = 10XXXXXXb
CM05 = 1 CM10 = 0 CM31 = 1
CM10 = 1
CM10 = 0

→ : Arrows indicate a one-way transition between modes. No transition should be made unless indicated.

BCS: Bit in the CCR register
CM05: Bit in the CM0 register
CM10: Bit in the CM1 register
CM20: Bit in the CM2 register
CM31: Bit in the CM3 register
SEO: Bit in the PLC1 register
PLC0 = XXh: The multiplication ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.
PLC1 = 0Xh: The divisor of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.
CCR = 00XXXXXXb: The divisor of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum values specified in the electrical characteristics.

Notes:

1. The PLC0 register can be set only once after reset is released.
2. This clock should be switched after the PLL clock oscillation is fully stabilized.

Figure 8.19 State Transition When Using the On-chip Oscillator Clock

8.7.2 Wait Mode

The base clock stops in wait mode so that clocks generated by the base clock, the CPU clock and peripheral bus clock, stop running as well. Thus the CPU and watchdog timer, operated by these two clocks, also stop. However, the watchdog timer continues operating when the PM23 bit in the PM2 register is 1 (on-chip oscillator selected as count source for the watchdog timer) or when the WDK4 bit in the WDK register is 1 (on-chip oscillator selected as count source for the prescaler). Since the main clock, sub clock, PLL clock, and on-chip oscillator clock continue running, the peripherals using these clocks also continue operating.

8.7.2.1 Peripheral Clock Source Stop Function

When the CM02 bit in the CM0 register is 1 (peripheral clock source stopped in wait mode), power consumption is reduced since peripheral clocks f1, f8, f32, f2n (when the clock source is the peripheral clock source), and fAD stop running in wait mode. fC32 and f2n (when the clock source is the main clock) do not stop running.

8.7.2.2 Entering Wait Mode

To enter wait mode, the following procedures should be completed before the WAIT instruction is executed.

- Initial setting
  Set the wake-up interrupt priority level (bits RLVL2 to RLVL0 in registers RIPL1 and RIPL2) to 7. Then set each interrupt request level.
- Steps before entering wait mode
  (1) Set the I flag to 0.
  (2) Set the interrupt request level for each interrupt source (interrupt number from 1 to 127) to 0, if its interrupt request level is not 0.
  (3) Perform a dummy read of any of the interrupt control registers.
  (4) Set the processor interrupt priority level (IPL) in the flag register to 0.
  (5) Enable interrupts temporarily by executing the following instructions:
  FSET I
  NOP
  NOP
  FCLR I
  (6) Set the interrupt request level for the interrupt to exit wait mode.
  Do not rewrite the interrupt control register after this step.
  (7) Set the IPL in the flag register.
  (8) Set the interrupt priority level for resuming to the same level as the IPL.
  Interrupt request level for the interrupt to exit wait mode > IPL = Interrupt priority level for resuming
  (9) Set the CM20 bit in the CM2 register to 0 (disable oscillator stop detection) when the oscillator stop detection is used.
  (10)Enter either PLL self-oscillation mode, low speed mode, or low power mode.
  (11)Set the I flag to 1.
  (12)Execute the WAIT instruction.
- After exiting wait mode
  Set the wake-up interrupt priority level to 7 immediately after exiting wait mode.

8.7.2.3 Pin State in Wait Mode

Table 8.5 lists the pin state in wait mode.

Table 8.5 Pin State in Wait Mode

Pin		Memory Expansion Mode/ Microprocessor Mode	Single-chip Mode
Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{BC0}}$ to $\overline{\text{BC3}}$		The state immediately before entering wait mode is held	—
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$		High	—
$\overline{\text{HLDA}}$, BCLK		High	—
ALE		High	—
Ports		The state immediately before entering wait mode is held	
DA0, DA1		The state immediately before entering wait mode is held	
CLKOUT	When a low speed clock is selected	The clock is output	
	When f8 or f32 is selected	The clock is output when the CM02 bit in the CM0 register is 0 (no peripheral clock source stopped in wait mode). The state immediately before entering wait mode is held when the CM02 bit is 1 (peripheral clock source stopped in wait mode)	

8.7.2.4 Exiting Wait Mode

The MCU exits wait mode by a hardware reset, an NMI, or a peripheral interrupt assigned to software interrupt number from 0 to 63.

To exit wait mode using either a hardware reset or NMI, without using peripheral interrupts, set bits ILVL2 to ILVL0 for the peripheral interrupts to 000b (interrupt disabled) before executing the WAIT instruction.

The CM02 bit setting in the CM0 register affects the peripheral interrupts. When the CM02 bit is 0 (peripheral clock source not stopped in wait mode), peripheral interrupts for software interrupt numbers from 0 to 63 can be used to exit wait mode. When this bit is 1 (peripheral clock source stopped in wait mode), peripherals operated using clocks (f1, f8, f32, f2n whose clock source is the peripheral clock source, and fAD) generated by the peripheral clock source stop operating. Therefore, the peripheral interrupts cannot be used to exit wait mode. However, peripherals operated using clocks which are independent from the peripheral clock source (fC32, external clock, and f2n whose clock source is the main clock) do not stop operating. Thus, interrupts generated by these peripherals and assigned to software interrupt numbers from 0 to 63 can be used to exit wait mode.

The CPU clock used when exiting wait mode by a peripheral interrupt or an NMI is the same clock used when the WAIT instruction is executed.

Table 8.6 lists interrupts used to exit wait mode and usage conditions.

Table 8.6 Interrupts for Exiting Wait Mode and Usage Conditions

Interrupt	When the CM02 Bit is 0	When the CM02 Bit is 1
NMI	Available	Available
External interrupt [1]	Available	Available
Key input interrupt	Available	Available
Low voltage detection interrupt	Available	Available
Timer A interrupt Timer B interrupt	Available in any mode	Available in event counter mode, or when the count source is fC32 or f2n (when the main clock is selected as the clock source)
Serial interface interrupt [2]	Available when an internal or external clock is used	Available when the external clock or f2n (when the main clock is selected as the clock source) is used
A/D conversion interrupt	Available in single mode or single-sweep mode	Should not be used
Intelligent I/O interrupt	Available	Should not be used
I^2C-bus interface interrupt	Available	Should not be used
I^2C-bus line interrupt	Available	Available

Notes:

1. INT6 to INT8 are available in the intelligent I/O interrupt only.
2. UART7 to UART10 are excluded.

8.7.3 Stop Mode

In stop mode, all of the clocks, except for those that are protected, stop running. That is, the CPU and peripherals, operated by the CPU clock and peripheral clock, also stop. This mode saves the most power.

8.7.3.1 Entering Stop Mode

To enter stop mode, the following procedures should be done before the STOP instruction is executed.

- Initial setting
  Set the wake-up interrupt priority level (bits RLVL2 to RLVL0 in registers RIPL1 and RIPL2) to 7. Then set each interrupt request level.
- Steps before entering stop mode
  (1) Set the I flag to 0.
  (2) Set the interrupt request level for each interrupt source (interrupt number from 1 to 127) to 0, if the interrupt request level is not 0.
  (3) Perform a dummy read of any of the interrupt control registers.
  (4) Set the processor interrupt priority level (IPL) in the flag register to 0.
  (5) Enable interrupts temporarily by executing the following instructions:
  FSET I
  NOP
  NOP
  FCLR I
  (6) Set the interrupt request level for the interrupt to exit stop mode.
  Do not rewrite the interrupt control register after this step.
  (7) Set the IPL in the flag register.
  (8) Set the interrupt priority level for resuming to the same level as the IPL.
  Interrupt request level for the interrupt to exit stop mode > IPL = Interrupt priority level for resuming
  (9) Set the CM20 bit in the CM2 register to 0 (oscillator stop detection disabled) when the oscillator stop detection is used.
  (10)Change the base clock to either the main clock divided by 256 (f256) or the on-chip oscillator clock divided by 4 (fOCO4).
  (11)Set the I flag to 1.
  (12)Execute the STOP instruction.
- After exiting stop mode
  Set the wake-up interrupt priority level to 7 immediately after exiting stop mode.

8.7.3.2 Pin State in Stop Mode

Table 8.7 lists the pin state in stop mode.

Table 8.7 Pin State in Stop Mode

Pin		Memory Expansion Mode/ Microprocessor Mode	Single-chip Mode
Address bus, data bus, $\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS3}}$, $\overline{\mathrm{BC0}}$ to $\overline{\mathrm{BC3}}$		The state immediately before entering stop mode is held	—
$\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, $\overline{\mathrm{WR0}}$ to $\overline{\mathrm{WR3}}$		High	—
$\overline{\mathrm{HLDA}}$, BCLK		High	—
ALE		High	—
Ports		The state immediately before entering stop mode is held	
DA0, DA1		The state immediately before entering stop mode is held	
CLKOUT	When a low speed clock is selected	High	
	When f8 or f32 is selected	The state immediately before entering stop mode is held	
XIN		High-impedance	
XOUT		High	
XCIN, XCOUT		High-impedance	

8.7.3.3 Exiting Stop Mode

The MCU exits stop mode by a hardware reset, NMI, low voltage detection interrupt, or a peripheral interrupt assigned to software interrupt number from 0 to 63.

To exit stop mode using either a hardware reset or NMI, without using peripheral interrupts, set bits ILVL2 to ILVL0 for the peripheral interrupts to 000b (interrupt disabled) before executing the STOP instruction.

The CPU clock used when exiting stop mode by a peripheral interrupt or NMI is the same clock used when the STOP instruction is executed.

Table 8.8 lists interrupts used to exit stop mode and usage conditions.

Table 8.8 Interrupts for Exiting Stop Mode and Usage Conditions

Interrupt	Usage Condition
NMI	
Low voltage detection interrupt	
External interrupt	INT6 to INT8 are available when intelligent I/O interrupt is used
Key input interrupt	
Timer A interrupt Timer B interrupt	Available when a timer counts an external pulse with a frequency of 100 Hz or less in event counter mode
Serial interface interrupt (1)	Available when an external clock is used
I^2C-bus line interrupt	

Note:

1. UART7 to UART10 are excluded.

8.8 System Clock Protection

The system clock protection disables clock change when the PLL clock is selected as the base clock source. This prevents the CPU clock from stopping due to a runaway program.
When the PM21 bit in the PM2 register is set to 1 (clock change disabled), the following bits cannot be written to:

- Bits CM02 and CM05 in the CM0 register
- The CM10 bit in the CM1 register
- The CM20 bit in the CM2 register
- The PM27 bit in the PM2 register

To use the system clock protection, set the CM05 bit in the CM0 register to 0 (main clock oscillator enabled) and the BCS bit in the CCR register to 0 (PLL clock selected as base clock source) before the following procedure is done:

(1) Set the PRC1 bit in the PRCR register to 1 (write to the PM2 register enabled).
(2) Set the PM21 bit in the PM2 register to 1 (clock change disabled).
(3) Set the PRC1 bit in the PRCR register to 0 (write to the PM2 register disabled).

8.9 Notes on Clock Generator

8.9.1 Sub Clock

8.9.1.1 Oscillator Constant Matching

The constant matching of the sub clock oscillator should be evaluated in both cases when the drive strength is high and low.
Contact the oscillator manufacturer for details on the oscillation circuit constant matching.

8.9.2 Power Control

Do not switch the base clock source until the oscillation of the clock to be used has stabilized. However, this does not apply to the on-chip oscillator since it starts running immediately after the CM31 bit in the CM3 register is set to 1.
To switch the base clock source from the PLL clock to a low speed clock, use the MOV.L or OR.L instruction to set the BCS bit in the CCR register to 1.

- Program example in assembly language

```
OR.L      #80h, 0004h
```

- Program example in C language

```
asm("OR.L  #80h, 0004h");
```

8.9.2.1 Stop Mode

- To exit stop mode using a reset, apply a low signal to the $\overline{\text{RESET}}$ pin until the main clock oscillation stabilizes.

8.9.2.2 Suggestions for Power Saving

The following are suggestions to reduce power consumption when programming or designing systems.

- I/O pins:
  If inputs are floating, both transistors may be conducting. Set unassigned pins to input mode and connect each of them to VSS via a resistor, or set them to output mode and leave them open.
- A/D converter:
  When not performing the A/D conversion, set the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). To perform the A/D conversion, set the VCUT bit to 1 (VREF connected) and wait at least 1 μs before starting conversion.
- D/A converter:
  When not performing the D/A conversion, set the DAiE bit in the DACON register (i = 0, 1) to 0 (output disabled) and the DAi register to 00h.
- Peripheral clock stop:
  When entering wait mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop the peripheral clock source. However, this setting does not stop the fC32.

9. Bus

This MCU has an internal bus and an external bus. The internal bus contains a fast bus (CPU bus) and a slow bus (peripheral bus). Figure 9.1 shows a block diagram of the bus.

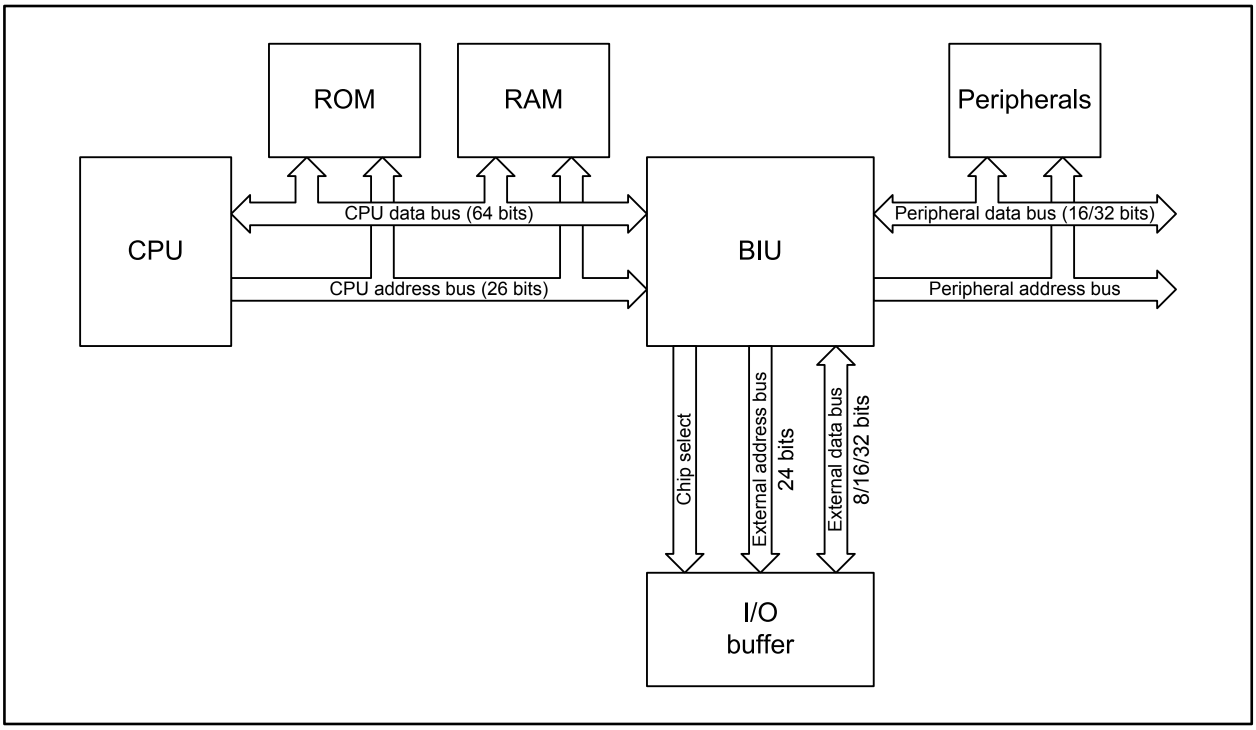


Figure 9.1 Bus Block Diagram

In memory expansion mode or microprocessor mode, some pins function as bus control pin to control the address bus and the data bus. The bus control pins are as follows: A0 to A23, D0 to D31, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{WR0}}/\overline{\text{WR}}$, $\overline{\text{BC0}}$, $\overline{\text{WR1}}/\overline{\text{BC1}}$, $\overline{\text{WR2}}/\overline{\text{BC2}}$, $\overline{\text{WR3}}/\overline{\text{BC3}}$, $\overline{\text{RD}}$, BCLK, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$, ALE, and $\overline{\text{RDY}}$.

9.1 Bus Settings

The bus settings are controlled by the two lowest bits of the reset vector, the PBC register, registers EBC0 to EBC3, and CSOP0 to CSOP2.

Table 9.1 lists bus settings and their sources.

Table 9.1 Bus Settings and Sources

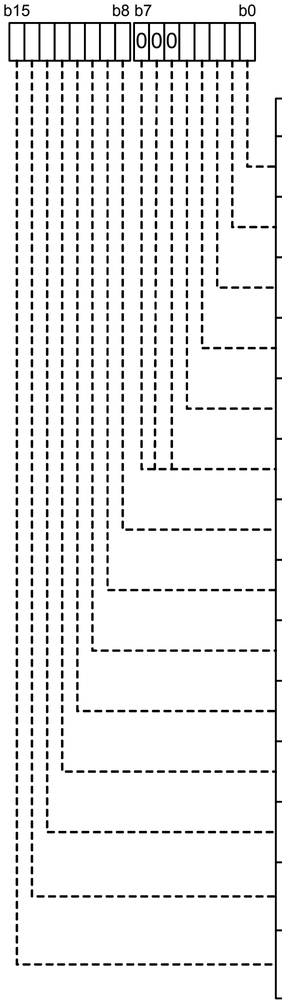
Bus Settings	Sources
Internal SFR bus timing	PBC register
External bus timing	Registers EBC0 to EBC3
External data bus width	PBC register, registers EBC0 to EBC3
External data bus width after reset	Two lowest bits of the reset vector
Separate bus/multiplexed bus selection	PBC register, registers EBC0 to EBC3
Pins outputting chip select signals	Registers CSOP0 to CSOP2

9.2 Peripheral Bus Timing Setting

The 16-/32-bit wide peripheral bus operates at a frequency up to 32 MHz (the theoretical value and the maximum frequency of each product group are as defined by f(BCLK) in 28. “Electrical Characteristics”). The timing adjustment and bus-width conversion with the faster, 64-bit wide CPU bus are controlled in the bus interface unit (BIU).

Figure 9.2 shows the PBC register which determines the peripheral bus timing.

Peripheral Bus Control Register (1, 2)



Symbol	Address	Reset Value
PBC	001Fh-001Eh	0504h

Bit Symbol	Bit Name	Function	RW
PRD0	Read Timing Setting Bit	Select from the three options below according to the peripheral bus clock setting (bits PCD1 and PCD0 in the CCR register). When bits PCD1 and PCD0 are set to: 1. 01b : 00100b 2. 10b : 01101b 3. 11b : 01111b	RW
PRD1			
PRD2			
PRD3			
PRD4			
— (b7-b5)	Reserved	Should be written with 0	RW
PWR0	Write Timing Setting Bit	Select from the three options below according to the peripheral bus clock setting (bits PCD1 and PCD0 in the CCR register). When bits PCD1 and PCD0 are set to: 1. 01b : 00101b 2. 10b : 01010b 3. 11b : 01111b	RW
PWR1			
PWR2			
PWR3			
PWR4			
EXMPX	External Bus Format Select Bit (3)	0: Separate bus in some spaces 1: Multiplexed bus in all spaces	RW
EXBW0	External Bus Maximum Width Setting Bit (4)	b15b14 0 0 : 8-bit width 0 1 : 16-bit width 1 0 : 32-bit width 1 1 : Do not use this combination	RW
EXBW1			

Notes:

1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. Set this register only once after a reset. Do not rewrite this register after setting the CCR register.
3. If this bit is set to 1 when the all MPX bits in registers EBC0 to EBC3 are set to 1, ports P0, P1, and P4_0 to P4_3 can be used as programmable I/O ports.
4. This bit should be the maximum bus width set in bits BW1 and BW0 in registers EBC0 to EBC3. The functions of ports P1, P12, and P13 vary with this bit setting.

Figure 9.2 PBC Register

9.3 External Bus Setting

The 8-/16-/32-bit wide external bus operates at a frequency up to 32 MHz (the theoretical value and the maximum frequency of each product group are as defined by f(BCLK) in 28. “Electrical Characteristics”). The timing adjustment and bus-width conversion with the faster 64-bit wide CPU bus are controlled in the bus interface unit (BIU).

9.3.1 External Address Space Setting

The internal address bus of the R32C/100 Series MCU consists of 26 address lines (A0 to A25). Since A25 is sign extended to A26 to A31, the MCU has 64 MB of accessible space addresses from 00000000h to 01FFFFFFh and from FE000000h to FFFFFFFFh.

Up to 24 address lines from A0 to A23 can be used for external output. Decoded A18 to A25 function as 4 chip select signals ($\overline{\text{CS3}}$ to $\overline{\text{CS0}}$). If a 16 MB space is assigned to each chip select signal, up to 63.5 MB can be used as external address space. When the processor mode is changed from single-chip mode to memory expansion mode, the address bus status is undefined until an external space is accessed.

Chip select signals $\overline{\text{CS3}}$ to $\overline{\text{CS0}}$ share pins with A20 to A23, respectively. Other combinations of signal and output port are also available as follows: signals $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ with ports P11_0 to P11_3, and signals $\overline{\text{CS1}}$ to $\overline{\text{CS3}}$ with ports P5_4, P5_6, and P5_7.

In microprocessor mode, the $\overline{\text{CS0}}$ signal is output from port P4_7 after a reset. The maximum space per chip select signal is 8 MB since A23 is not available. Signals $\overline{\text{CS1}}$ to $\overline{\text{CS3}}$ are output only when being set.

$\overline{\text{CSi}}$ (i = 0 to 3) is held low while accessing an external space i. It becomes high when accessing another external space. Figure 9.3 shows output examples of address bus and chip select signals.

Set registers CSOP0 to CSOP2 to select a chip select signal to be used and its output pin. Set registers CB01, CB12, and CB23 to set the address space for each chip select signal.

Figures 9.4 to 9.6 show registers CSOP0 to CSOP2. Figures 9.7, 9.8, and 9.9 show registers CB01, CB12, and CB23, respectively. Figures 9.10 and 9.11 show the chip select space.

A chip select signal should not be set for more than two output pins in registers CSOP0 to CSOP2.

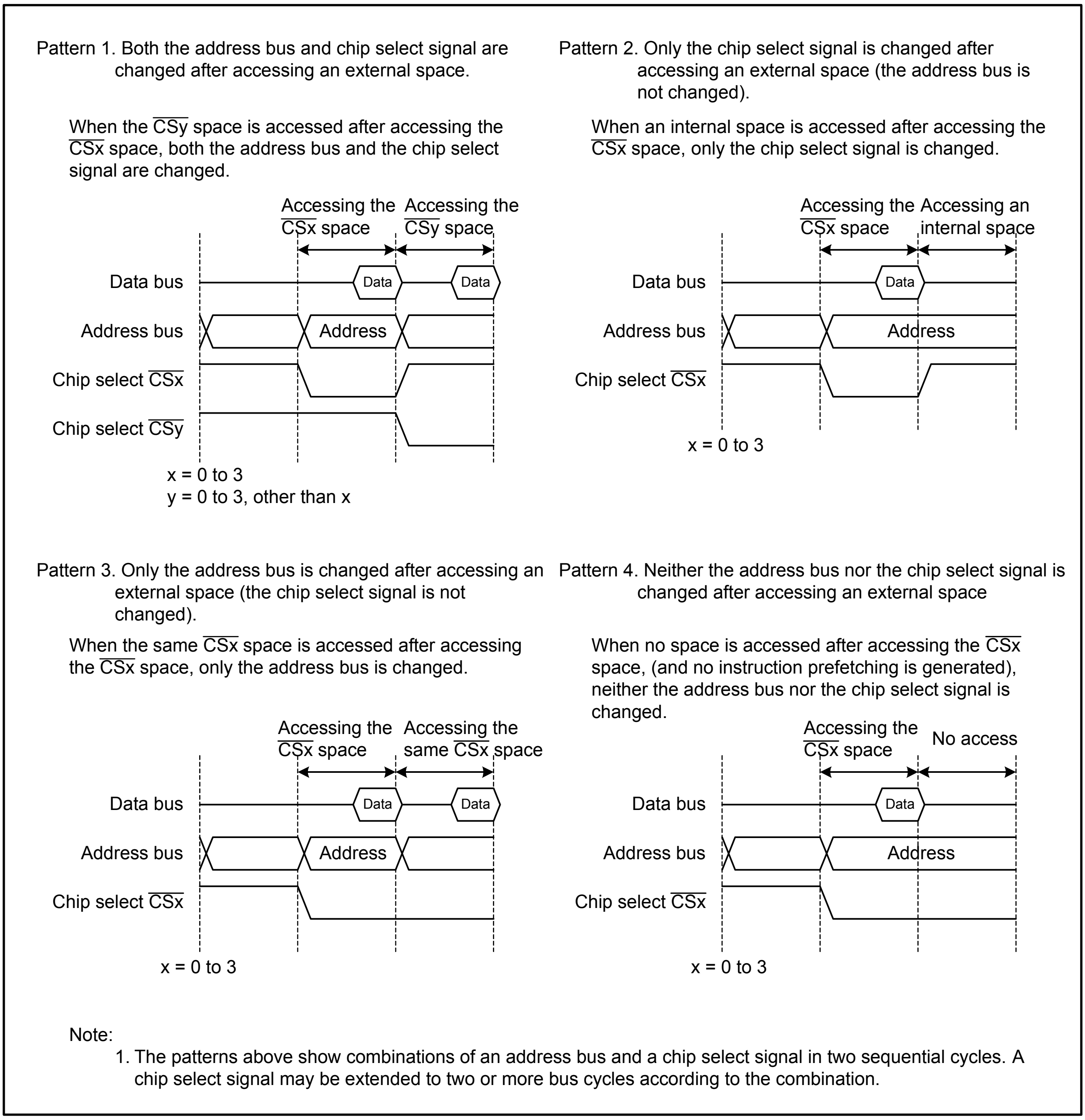
Registers CB01, CB12, and CB23 should be set to meet the conditions below:

- In memory expansion mode

$$\text{0080000h} \le (\text{CB23} \times 2^{18}) \le (\text{CB12} \times 2^{18}) \le (\text{CB01} \times 2^{18}) \le \text{3DC0000h}$$

- In microprocessor mode

$$\text{0080000h} \le (\text{CB23} \times 2^{18}) \le (\text{CB12} \times 2^{18}) \le (\text{CB01} \times 2^{18}) \le \text{3FC0000h}$$



Pattern 1. Both the address bus and chip select signal are changed after accessing an external space.

When the $\overline{\mathrm{CSy}}$ space is accessed after accessing the $\overline{\mathrm{CSx}}$ space, both the address bus and the chip select signal are changed.

Accessing the $\overline{\mathrm{CSx}}$ space
Accessing the $\overline{\mathrm{CSy}}$ space
Data bus
Data
Data
Address bus
Address
Chip select $\overline{\mathrm{CSx}}$
Chip select $\overline{\mathrm{CSy}}$

x = 0 to 3
y = 0 to 3, other than x

Pattern 2. Only the chip select signal is changed after accessing an external space (the address bus is not changed).

When an internal space is accessed after accessing the $\overline{\mathrm{CSx}}$ space, only the chip select signal is changed.

Accessing the $\overline{\mathrm{CSx}}$ space
Accessing an internal space
Data bus
Data
Address bus
Address
Chip select $\overline{\mathrm{CSx}}$

x = 0 to 3

Pattern 3. Only the address bus is changed after accessing an external space (the chip select signal is not changed).

When the same $\overline{\mathrm{CSx}}$ space is accessed after accessing the $\overline{\mathrm{CSx}}$ space, only the address bus is changed.

Accessing the $\overline{\mathrm{CSx}}$ space
Accessing the same $\overline{\mathrm{CSx}}$ space
Data bus
Data
Data
Address bus
Address
Chip select $\overline{\mathrm{CSx}}$

x = 0 to 3

Pattern 4. Neither the address bus nor the chip select signal is changed after accessing an external space

When no space is accessed after accessing the $\overline{\mathrm{CSx}}$ space, (and no instruction prefetching is generated), neither the address bus nor the chip select signal is changed.

Accessing the $\overline{\mathrm{CSx}}$ space
No access
Data bus
Data
Address bus
Address
Chip select $\overline{\mathrm{CSx}}$

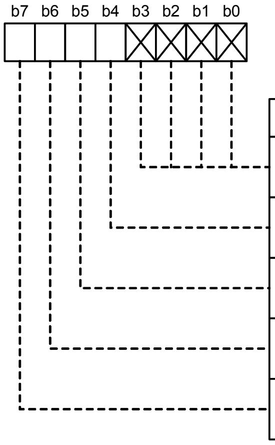
x = 0 to 3

Note:

1. The patterns above show combinations of an address bus and a chip select signal in two sequential cycles. A chip select signal may be extended to two or more bus cycles according to the combination.

Figure 9.3 Address Bus and Chip Select Signal Output Patterns (in separate bus format)

Chip Select Output Pin Setting Register 0 [(1)]



Symbol	Address	Reset Value
CSOP0	40054h	1000 XXXXb

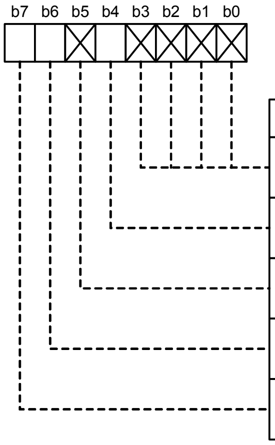
Bit Symbol	Bit Name	Function	RW
— (b3-b0)	No register bits; should be written with 0 and read as undefined value		—
P4_4B	P4_4 Bus Function Setting Bit	0: Output A20 from P4_4 1: Output $\overline{\text{CS3}}$ from P4_4	RW
P4_5B	P4_5 Bus Function Setting Bit	0: Output A21 from P4_5 1: Output $\overline{\text{CS2}}$ from P4_5	RW
P4_6B	P4_6 Bus Function Setting Bit	0: Output A22 from P4_6 1: Output $\overline{\text{CS1}}$ from P4_6	RW
P4_7B	P4_7 Bus Function Setting Bit	0: Output A23 from P4_7 [(2)] 1: Output $\overline{\text{CS0}}$ from P4_7	RW

Notes:

1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. The P4_7B bit should not be set to 0 when starting an operation in microprocessor mode.

Figure 9.4 CSOP0 Register

Chip Select Output Pin Setting Register 1 [(1)]



Symbol	Address	Reset Value
CSOP1	40055h	01X0 XXXXb

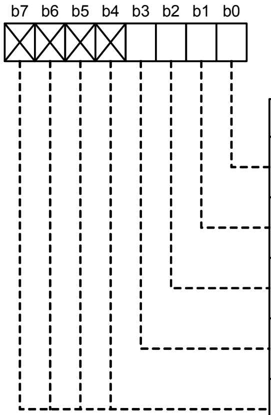
Bit Symbol	Bit Name	Function	RW
— (b3-b0)	No register bits; should be written with 0 and read as undefined value		—
P5_4B	P5_4 Bus Function Setting Bit	0: Output $\overline{\text{HLDA}}$ from P5_4 1: Output $\overline{\text{CS1}}$ from P5_4	RW
— (b5)	No register bit; should be written with 0 and read as undefined value		—
P5_6B	P5_6 Bus Function Setting Bit	0: Output ALE from P5_6 1: Output $\overline{\text{CS2}}$ from P5_6	RW
P5_7B	P5_7 Bus Function Setting Bit	0: $\overline{\text{RDY}}$ input pin 1: Output $\overline{\text{CS3}}$ from P5_7	RW

Notes:

1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.

Figure 9.5 CSOP1 Register

Chip Select Output Pin Setting Register 2 [1]



Symbol	Address	Reset Value
CSOP2	40056h	XXXX 0000b

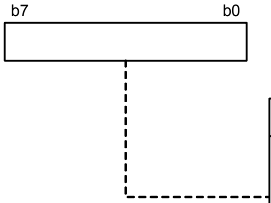
Bit Symbol	Bit Name	Function	RW
P11_0B	P11_0 Bus Function Setting Bit	0: Use P11_0 for a peripheral function 1: Output $\overline{\text{CS0}}$ from P11_0	RW
P11_1B	P11_1 Bus Function Setting Bit	0: Use P11_1 for a peripheral function 1: Output CS1 from P11_1	RW
P11_2B	P11_2 Bus Function Setting Bit	0: Use P11_2 for a peripheral function 1: Output CS2 from P11_2	RW
P11_3B	P11_3 Bus Function Setting Bit	0: Use P11_3 for a peripheral function 1: Output $\overline{\text{CS3}}$ or $\overline{\text{WR2}}$ from P11_3 [2]	RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. $\overline{\text{WR2}}$ is output when the PM02 bit in the PM0 register is 1 ($\overline{\text{RD}}$/$\overline{\text{WR0}}$/$\overline{\text{WR1}}$/$\overline{\text{WR2}}$/$\overline{\text{WR3}}$) and bits EXBW1 and EXBW0 in the PBC register are 10b (32-bit width as the maximum width of external bus); otherwise, $\overline{\text{CS3}}$ is output.

Figure 9.6 CSOP2 Register

Chip Selects 0 and 1 Boundary Setting Register [1]



Symbol	Address	Reset Value
CB01	001Ah	00h

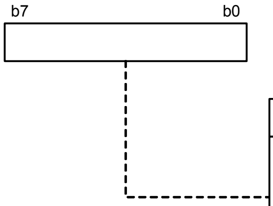
Function	Setting Range	RW
Set this register to the value from A25 to A18 of the start address in the $\overline{\text{CS0}}$ space. The immediately preceding address mentioned above and lower is designated for $\overline{\text{CS1}}$ space	02h to F8h [2] in memory expansion mode 02h to FFh [2] in microprocessor mode	RW

Notes:

1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The setting value should be equal to or greater than that of the CB12 register.

Figure 9.7 CB01 Register

Chip Selects 1 and 2 Boundary Setting Register [1]



Symbol	Address	Reset Value
CB12	0016h	00h

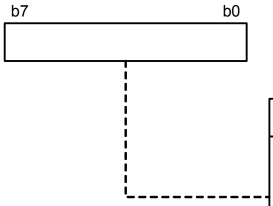
Function	Setting Range	RW
Set this register to the value from A25 to A18 of the start address in the $\overline{CS1}$ space. The immediately preceding address mentioned above and lower is designated for $\overline{CS2}$ space	02h to F8h [2] in memory expansion mode 02h to FFh [2] in microprocessor mode	RW

Notes:

1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The setting value should be equal to or greater than that of the CB23 register and should be equal to or less than that of the CB01 register.

Figure 9.8 CB12 Register

Chip Selects 2 and 3 Boundary Setting Register [1]



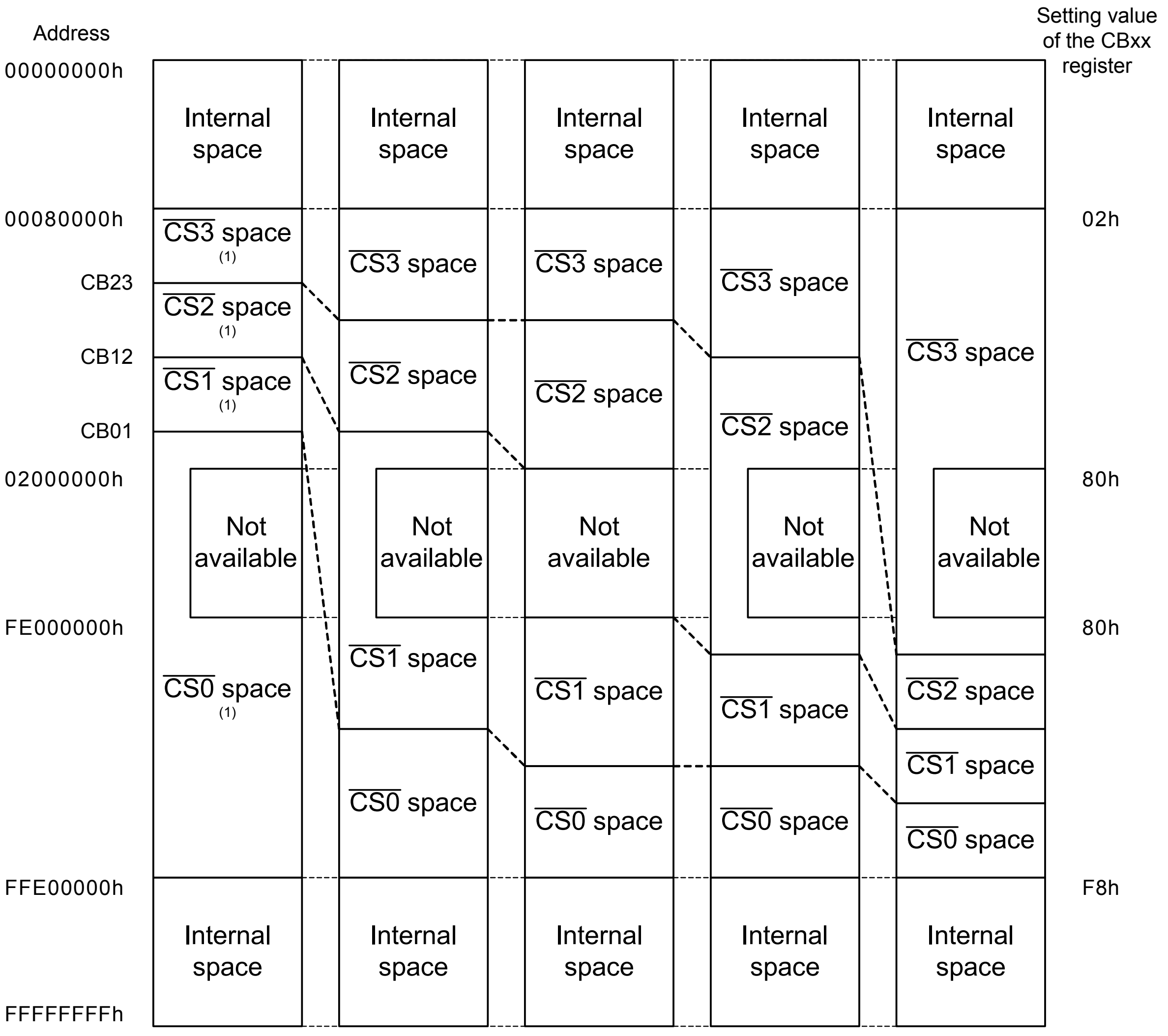
Symbol	Address	Reset Value
CB23	0012h	00h

Function	Setting Range	RW
Set this register to the value from A25 to A18 of the start address in the $\overline{CS2}$ space. The immediately preceding address mentioned above and lower is designated for $\overline{CS3}$ space	02h to F8h [2] in memory expansion mode 02h to FFh [2] in microprocessor mode	RW

Notes:

1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The setting value should be equal to or less than that of the CB12 register.

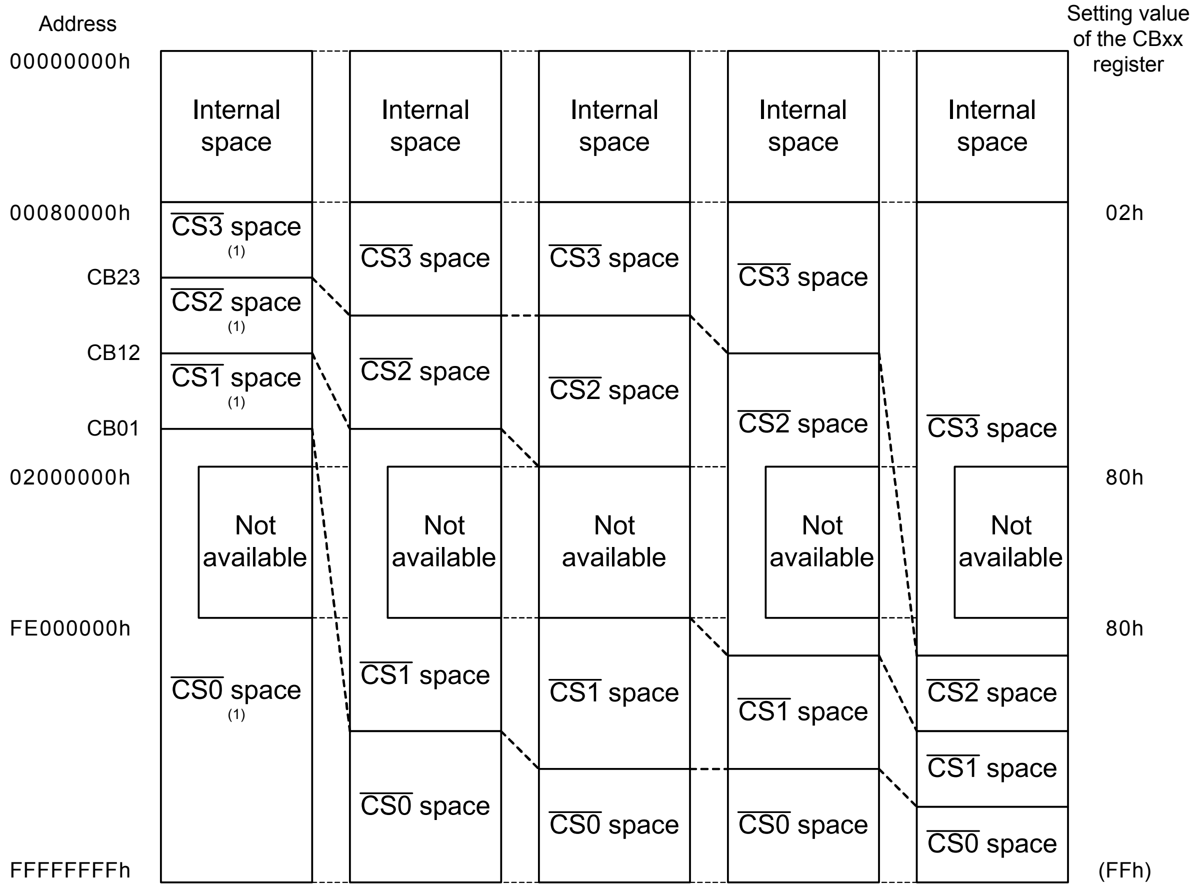
Figure 9.9 CB23 Register



Note:

1. Each $\overline{CS}$ space can be up to 16 MB when the $\overline{CS}$ signal is not output from port P4. If a space is oversized, the same data is shown every 16 MB. When the $\overline{CS}$ signal is output from port P4, the maximum valid size is reduced depending on the number of address lines reduced.

Figure 9.10 Chip Select Spaces in Memory Expansion Mode



Note:

1. Each $\overline{\text{CS}}$ space can be up to 8 MB when the $\overline{\text{CS}}$ signal (except for the $\overline{\text{CS0}}$ signal) is not output from port P4. If a space is oversized, the same data is shown every 8 MB. When the $\overline{\text{CS}}$ signal (except for the $\overline{\text{CS0}}$ signal) is output from port P4, the maximum valid size is reduced depending on the number of address lines reduced.

Figure 9.11 Chip Select Spaces in Microprocessor Mode

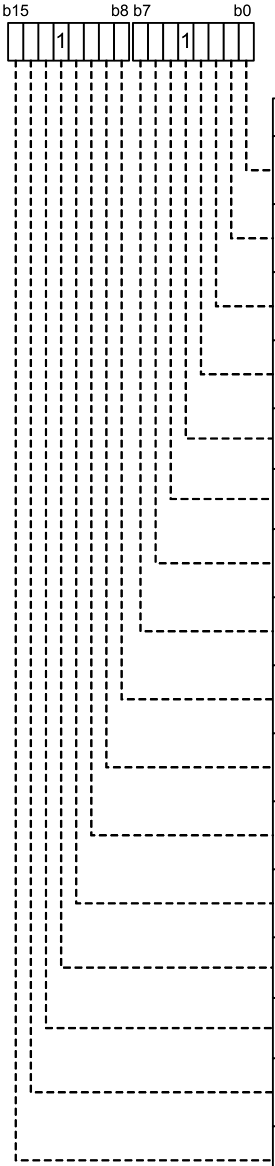
9.3.2 External Data Bus Width Setting

The external data bus width is selectable among 8 bits, 16 bits, and 32 bits. The bus width of each space is selected by setting bits BW1 and BW0 in registers EBC0 to EBC3. The maximum bus width for all spaces is selected by setting bits EXBW1 and EXBW0 in the PBC register. The bus width specified in bits EXBW1 and EXBW0 should be equal to or greater than the value specified in bits BW1 and BW0.

When an accessed space has a bus width less than that specified in bits EXBW1 and EXBW0, an undefined value is output from the unused data output pins.

Figure 9.12 shows registers EBC0 to EBC3.

External Bus Control Register i (i = 0 to 3) [1]



Symbol	Address	Reset Value
EBC0, EBC1	001Dh-001Ch, 0019h-0018h	0000h
EBC2, EBC3	0015h-0014h, 0011h-0010h	0000h

Bit Symbol	Bit Name	Function	RW
ESUR0	Address Setup Cycles Before Read Setting Bit [2]	b1 b0 0 0 : *sur* = 0 0 1 : *sur* = 1 1 0 : *sur* = 2 1 1 : *sur* = 3	RW
ESUR1			
EWR0	Read Pulse Width Setting Bit [2]	b3 b2 0 0 : *wr* = 1 0 1 : *wr* = 2 1 0 : *wr* = 3 1 1 : *wr* = 4	RW
EWR1			
— (b4)	Reserved	Should be written with 1	RW
RDY	$\overline{\text{RDY}}$ Monitor Bit	0: Ignore $\overline{\text{RDY}}$ 1: Use $\overline{\text{RDY}}$	RW
MPY0	Multiplied Cycle Setting Bit [2]	b7 b6 0 0 : *mpy* = 1 0 1 : *mpy* = 2 1 0 : *mpy* = 3 1 1 : *mpy* = 4	RW
MPY1			
ESUW0	Address Setup Cycles Before Write Setting Bit [2]	b9 b8 0 0 : *suw* = 0 0 1 : *suw* = 1 1 0 : *suw* = 2 1 1 : *suw* = 3	RW
ESUW1			
EWW0	Write Pulse Width Setting Bit [2]	b11b10 0 0 : *ww* = 1 0 1 : *ww* = 2 1 0 : *ww* = 3 1 1 : *ww* = 4	RW
EWW1			
— (b12)	Reserved	Should be written with 1	RW
MPX	External Bus Format Select Bit	0: Separate bus 1: Multiplexed bus	RW
BW0	External Bus Width Setting Bit [3]	b15b14 0 0 : 8-bit width 0 1 : 16-bit width 1 0 : 32-bit width 1 1 : Do not use this combination	RW
BW1			

Notes:

1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. Refer to 9.3.5 “External Bus Timing” for the relation between register settings and practical timing.
3. The maximum value set here should be applied to bits EXBW1 and EXBW0 in the PBC register.

Figure 9.12 Registers EBC0 to EBC3

9.3.3 Separate Bus/Multiplexed Bus Selection

The bus format is selectable between separate bus format and multiplexed bus format. The bus format for each space is selected by setting the MPX bit in registers EBC0 to EBC3. To select the multiplexed bus format for all spaces, the EXPMX bit in the PBC register should be set to 1 (multiplexed bus in all spaces). In this case, ports P0, P1, and P4_0 to P4_3 can be used as programmable I/O ports.

(1) Separate Bus

In this bus format, the data bus and address bus have their own I/O pins.

To select separate bus mode, the MPX bit in registers EBC0 to EBC3 should be set to 0. The data bus width is selectable among 8 bits, 16 bits, and 32 bits by setting bits BW1 and BW0 in registers EBC0 to EBC3.

When bits EXBW1 and EXBW0 in the PBC register are 00b (8-bit width), port P0 is the data bus, and ports P1, P12, and P13 are programmable I/O ports.

When bits EXBW1 and EXBW0 are 01b (16-bit width), ports P0 and P1 are data buses, and Ports P12 and P13 are programmable I/O ports. Note that port P1 (D8 to D15) becomes undefined if the MCU accesses an space where bits BW1 and BW0 are to 00b (8-bit width).

When bits EXBW1 and EXBW0 are 10b (32-bit width), ports P0, P1, P12, and P13 are data lines. Note that ports P1, P12, and P13 (D8 to D31) become undefined if the MCU accesses an space where bits BW1 and BW0 are 00b (8-bit width), and ports P12 and P13 (D16 to D31) become undefined if the MCU accesses an space where bits BW1 and BW0 are 01b (16-bit width).

(2) Multiplexed Bus

In this bus format, the data bus and address bus are time division multiplexed.

To select multiplexed bus mode, the MPX bit in registers EBC0 to EBC3 should be set to 1.

When bits BW1 and BW0 in registers EBC0 to EBC3 are 00b (8-bit width), D0 to D7 are multiplexed with A0 to A7. When bits BW1 and BW0 are 01b (16-bit width) or 10b (32-bit width), D0 to D15 are multiplexed with $\overline{\mathrm{BC0}}$, A1/$\overline{\mathrm{BC2}}$, and A2 to A15.

In microprocessor mode, an operation is started in separate bus format after a reset. Therefore the multiplexed bus format can only be used for $\overline{\mathrm{CS1}}$ to $\overline{\mathrm{CS3}}$ spaces and cannot be used for the $\overline{\mathrm{CS0}}$ space.

Table 9.2 lists pin functions for each processor mode and Table 9.3 lists pin functions for each bus format.

Table 9.2 Processor Mode and Pin Functions

Process or Mode	Single-chip Mode	Microprocessor Mode/Memory Expansion Mode						Memory Expansion Mode		
Bus format	—	Separate bus only (EXMPX = 0)			Separate bus and multiplexed bus (mixed) (EXMPX = 0)			Multiplexed bus only (EXMPX = 1)		
Data bus width	—	8 bits only	8/16 bits (mixed)	8/16/32 bits (mixed)	8 bits only	8/16 bits (mixed)	8/16/32 bits (mixed)	8 bits only	8/16 bits (mixed)	8/16/32 bits (mixed)
P0_0 to P0_7	I/O ports	D0 to D7						I/O ports		
P1_0 to P1_7	I/O ports	I/O ports	D8 to D15		I/O ports	D8 to D15		I/O ports		
P2_0	I/O port	A0	A0 or $\overline{\text{BC0}}$		A0 or A0/D0	A0, A0/D0, $\overline{\text{BC0}}$, or $\overline{\text{BC0}}$/D0		A0/D0	A0/D0 or $\overline{\text{BC0}}$/D0	
P2_1	I/O port	A1	A1 or $\overline{\text{BC2}}$		A1 or A1/D1		A1,A1/ D1,$\overline{\text{BC2}}$, or $\overline{\text{BC2}}$/D1	A1/D1		A1/D1 or $\overline{\text{BC2}}$/D1
P2_2 to P2_7	I/O ports	A2 to A7			A2 to A7 or A2/D2 to A7/D7			A2/D2 to A7/D7		
P3_0 to P3_7	I/O ports	A8 to A15			A8 to A15	A8 to A15 or A8/D8 to A15/D15		A8 to A15	A8/D8 to A15/D15	
P4_0 to P4_3	I/O ports	A16 to A19						I/O ports		
P4_4	I/O port	A20 or $\overline{\text{CS3}}$								
P4_5	I/O port	A21 or $\overline{\text{CS2}}$								
P4_6	I/O port	A22 or $\overline{\text{CS1}}$								
P4_7	I/O port	A23 or $\overline{\text{CS0}}$								
P5_0	I/O port	$\overline{\text{WR}}$ or $\overline{\text{WR0}}$								
P5_1	I/O port	Undefined[1]	$\overline{\text{BC1}}$ or $\overline{\text{WR1}}$		Undefined[1]	$\overline{\text{BC1}}$ or $\overline{\text{WR1}}$		Undefined[1]	$\overline{\text{BC1}}$ or $\overline{\text{WR1}}$	
P5_2	I/O port	$\overline{\text{RD}}$								
P5_3	I/O port	BCLK								
P5_4	I/O port	$\overline{\text{HLDA}}$ or $\overline{\text{CS1}}$								
P5_5	I/O port	$\overline{\text{HOLD}}$								
P5_6	I/O port	ALE or $\overline{\text{CS2}}$			Set to ALE					
P5_7	I/O port	$\overline{\text{RDY}}$ or $\overline{\text{CS3}}$								
P11_0 to P11_2	I/O ports	$\overline{\text{CS0}}$ to $\overline{\text{CS2}}$ or I/O ports								
P11_3	I/O port	$\overline{\text{CS3}}$ or I/O port		$\overline{\text{CS3}}$ or $\overline{\text{WR2}}$	$\overline{\text{CS3}}$ or I/O port		$\overline{\text{CS3}}$ to $\overline{\text{WR2}}$	$\overline{\text{CS3}}$ or I/O port		$\overline{\text{CS3}}$ or $\overline{\text{WR2}}$
P11_4	I/O port	I/O port		$\overline{\text{BC3}}$ or $\overline{\text{WR3}}$	I/O port		$\overline{\text{BC3}}$ to $\overline{\text{WR3}}$	I/O port		$\overline{\text{BC3}}$ or $\overline{\text{WR3}}$
P12_0 to P12_7	I/O ports	I/O ports		D16 to D23	I/O ports		D16 to D23	I/O ports		D16 to D23
P13_0 to P13_7	I/O ports	I/O ports		D24 to D31	I/O ports		D24 to D31	I/O ports		D24 to D31

Note:

1. An undefined value is output.

Table 9.3 Bus Format and Pin Functions (in Microprocessor Mode/Memory Expansion Mode)

Bus Format	Separate Bus			Multiplexed Bus		
MPX bit	0			1		
Bus width	8 bits	16 bits	32 bits	8 bits	16 bits	32 bits
Bits BW1 to BW0	00b	01b	10b	00b	01b	10b
P0_0 to P0_7	D0 to D7			I/O ports		
P1_0 to P1_7	I/O ports	D8 to D15		I/O ports		
P2_0	A0	$\overline{BC0}$		A0/D0	$\overline{BC0}$/D0	
P2_1	A1		$\overline{BC2}$	A1/D1		$\overline{BC2}$/D1
P2_2 to P2_7	A2 to A7			A2/D2 to A7/D7		
P3_0 to P3_7	A8 to A15			A8/D8 to A15/D15		
P4_0 to P4_3	A16 to A19			A16 to A19 or I/O ports		
P4_4	A20 or $\overline{CS3}$					
P4_5	A21 or $\overline{CS2}$					
P4_6	A22 or $\overline{CS1}$					
P4_7	A23 or $\overline{CS0}$ ($\overline{CS0}$ fixed in microprocessor mode)					
P5_0	$\overline{WR}$ or $\overline{WR0}$					
P5_1	Undefined (1)	$\overline{BC1}$ or $\overline{WR1}$		Undefined (1)	$\overline{BC1}$ or $\overline{WR1}$	
P5_2	$\overline{RD}$					
P5_3	BCLK					
P5_4	$\overline{HLDA}$ or $\overline{CS1}$					
P5_5	$\overline{HOLD}$					
P5_6	ALE or $\overline{CS2}$			Set to ALE		
P5_7	$\overline{RDY}$ or $\overline{CS3}$					
P11_0 to P11_2	$\overline{CS0}$ to $\overline{CS2}$ or I/O ports					
P11_3	$\overline{CS3}$ or I/O port		$\overline{CS3}$ or $\overline{WR2}$	$\overline{CS3}$ or I/O port		$\overline{CS3}$ or $\overline{WR2}$
P11_4	I/O port		$\overline{BC3}$ or $\overline{WR3}$	I/O port		$\overline{BC3}$ or $\overline{WR3}$
P12_0 to P12_7	I/O ports		D16 to D23	I/O ports		D16 to D23
P13_0 to P13_7	I/O ports		D24 to D31	I/O ports		D24 to D31

Note:

1. An undefined value is output.

9.3.4 Read and Write Signals

When the data bus is 16 or 32 bits, set the PM02 bit in the PM0 register to select a combination of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, $\overline{\mathrm{BC0}}$, $\overline{\mathrm{BC1}}$, $\overline{\mathrm{BC2}}$, and $\overline{\mathrm{BC3}}$, or $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR0}}$, $\overline{\mathrm{WR1}}$, $\overline{\mathrm{WR2}}$, and $\overline{\mathrm{WR3}}$ as read or write signals.

When bits EXBW1 and EXBW0 in the PBC register are 00b (8-bit width), the PM02 bit should be set to 0 ($\overline{\mathrm{RD}}$/$\overline{\mathrm{WR}}$/$\overline{\mathrm{BC0}}$/$\overline{\mathrm{BC1}}$/$\overline{\mathrm{BC2}}$/$\overline{\mathrm{BC3}}$). When accessing an 8-bit space while bits EXBW1 and EXBW0 are 01b (16-bit width) or 10b (32-bit width), the combination of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, $\overline{\mathrm{BC0}}$, $\overline{\mathrm{BC1}}$, $\overline{\mathrm{BC2}}$, and $\overline{\mathrm{BC3}}$ is selected irrespective of the PM02 bit setting.

Tables 9.4 and 9.5 list the operation of each signal.

The read and write signals after a reset are in the following combination: $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, $\overline{\mathrm{BC0}}$, $\overline{\mathrm{BC1}}$, $\overline{\mathrm{BC2}}$, and $\overline{\mathrm{BC3}}$. To change to the combination of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR0}}$, $\overline{\mathrm{WR1}}$, $\overline{\mathrm{WR2}}$, and $\overline{\mathrm{WR3}}$, set the PM02 bit before writing data to external memory.

Table 9.4 $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR0}}$, $\overline{\mathrm{WR1}}$, $\overline{\mathrm{WR2}}$, and $\overline{\mathrm{WR3}}$ Signals

Data Bus Width	$\overline{\mathrm{RD}}$	$\overline{\mathrm{WR0}}$	$\overline{\mathrm{WR1}}$	$\overline{\mathrm{WR2}}$	$\overline{\mathrm{WR3}}$	External Data Bus Status
32 bits	L	H	H	H	H	Read 4-byte data
	H	L	H	H	H	Write 1-byte data to address 4n+0
	H	H	L	H	H	Write 1-byte data to address 4n+1
	H	H	H	L	H	Write 1-byte data to address 4n+2
	H	H	H	H	L	Write 1-byte data to address 4n+3
	H	L	L	H	H	Write 2-byte data to addresses 4n+0 to 4n+1
	H	H	L	L	H	Write 2-byte data to addresses 4n+1 to 4n+2
	H	H	H	L	L	Write 2-byte data to addresses 4n+2 to 4n+3
	H	L	L	L	H	Write 3-byte data to addresses 4n+0 to 4n+2
	H	H	L	L	L	Write 3-byte data to addresses 4n+1 to 4n+3
	H	L	L	L	L	Write 4-byte data to addresses 4n+0 to 4n+3
16 bits	L	H	H	H/L (A1)	—	Read 2-byte data
	H	L	H	H/L (A1)	—	Write 1-byte data to even address
	H	H	L	H/L (A1)	—	Write 1-byte data to odd address
	H	L	L	H/L (A1)	—	Write 2-byte data to both even and odd addresses
8 bits	L	H ($\overline{\mathrm{WR}}$)	—	H/L (A1)	—	Read 1-byte data
	H	L ($\overline{\mathrm{WR}}$)	—	H/L (A1)	—	Write 1-byte data

Table 9.5 $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BC0}}$, $\overline{\text{BC1}}$, $\overline{\text{BC2}}$, and $\overline{\text{BC3}}$ Signals

Data Bus Width	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{BC0}}$	$\overline{\text{BC1}}$	$\overline{\text{BC2}}$	$\overline{\text{BC3}}$	External Data Bus Status
32 bits	L	H	L	L	L	L	Read 4-byte data
	H	L	L	H	H	H	Write 1-byte data to address 4n+0
	H	L	H	L	H	H	Write 1-byte data to address 4n+1
	H	L	H	H	L	H	Write 1-byte data to address 4n+2
	H	L	H	H	H	L	Write 1-byte data to address 4n+3
	H	L	L	L	H	H	Write 2-byte data to addresses 4n+0 to 4n+1
	H	L	H	L	L	H	Write 2-byte data to addresses 4n+1 to 4n+2
	H	L	H	H	L	L	Write 2-byte data to addresses 4n+2 to 4n+3
	H	L	L	L	L	H	Write 3-byte data to addresses 4n+0 to 4n+2
	H	L	H	L	L	L	Write 3-byte data to addresses 4n+1 to 4n+3
	H	L	L	L	L	L	Write 4-byte data to addresses 4n+0 to 4n+3
16 bits	L	H	L	L	H/L (A1)	—	Read 2-byte data
	H	L	L	H	H/L (A1)	—	Write 1-byte data to even address
	H	L	H	L	H/L (A1)	—	Write 1-byte data to odd address
	H	L	L	L	H/L (A1)	—	Write 2-byte data to both even and odd addresses
8 bits	L	H	H/L (A0)	—	H/L (A1)	—	Read 1-byte data
	H	L	H/L (A0)	—	H/L (A1)	—	Write 1-byte data

9.3.5 External Bus Timing

The external bus timing is configured by setting registers EBC0 to EBC3. The reference clock is the base clock selected by setting bits BCD1 and BCD0 in the CCR register.

Table 9.6 lists the bit setting of MPY1, MPY0, ESUR1, and ESUR0 and the Tsu(A-R) (address setup cycles before read), Table 9.7 lists the bit setting of MPY1, MPY0, EWR1, and EWR0 and the Tw(R) (read pulse width), Table 9.8 lists the bit setting of MPY1, MPY0, ESUW1, and ESUW0 and the Tsu(A-W) (address setup cycles before write), Table 9.9 lists the bit setting of MPY1, MPY0, EWW1, and EWW0 and the Tw(W) (write pulse width).

Table 9.6 Tsu(A-R) and Bit Settings: MPY1, MPY0, ESUR1, and ESUR0 (unit: cycles)

ESUR1 and ESUR0 Bit Settings		Separate Bus				Multiplexed Bus			
		MPY1 and MPY0 bit settings				MPY1 and MPY0 bit settings			
		00b	01b	10b	11b	00b	01b	10b	11b
		$mpy = 1$	$mpy = 2$	$mpy = 3$	$mpy = 4$	$mpy = 1$	$mpy = 2$	$mpy = 3$	$mpy = 4$
00b	$sur = 0$	0.5	0.5	0.5	0.5	1	1	1	1
01b	$sur = 1$	1.5	2.5	3.5	4.5	2	3	4	5
10b	$sur = 2$	2.5	4.5	6.5	8.5	3	5	7	9
11b	$sur = 3$	3.5	6.5	9.5	12.5	4	7	10	13
Formula		Tsu(A-R) = $sur \times mpy + 0.5$				Tsu(A-R) = $sur \times mpy + 1$			

Table 9.7 Tw(R) and Bit Settings: MPY1, MPY0, EWR1, and EWR0 (unit: cycles)

EWR1 and EWR0 Bit Settings		Separate Bus				Multiplexed Bus			
		MPY1 and MPY0 bit setting				MPY1 and MPY0 bit setting			
		00b	01b	10b	11b	00b	01b	10b	11b
		$mpy = 1$	$mpy = 2$	$mpy = 3$	$mpy = 4$	$mpy = 1$	$mpy = 2$	$mpy = 3$	$mpy = 4$
00b	$wr = 1$	1.5	2.5	3.5	4.5	0.5 (1)	1.5	2.5	3.5
01b	$wr = 2$	2.5	4.5	6.5	8.5	1.5	3.5	5.5	7.5
10b	$wr = 3$	3.5	6.5	9.5	12.5	2.5	5.5	8.5	11.5
11b	$wr = 4$	4.5	8.5	12.5	16.5	3.5	7.5	11.5	15.5
Formula		Tw(R) = $wr \times mpy + 0.5$				Tw(R) = $wr \times mpy - 0.5$			

Note:

1. Do not set this value.

Table 9.8 **Tsu(A-W) and the Bit Settings: MPY1, MPY0, ESUW1, and ESUW0 (unit: cycles)**

ESUW1 and ESUW0 Bit Settings		MPY1 and MPY0 Bit Settings			
		00b	01b	10b	11b
		$mpy = 1$	$mpy = 2$	$mpy = 3$	$mpy = 4$
00b	$suw = 0$	1	1	1	1
01b	$suw = 1$	2	3	4	5
10b	$suw = 2$	3	5	7	9
11b	$suw = 3$	4	7	10	13
Formula		Tsu(A-W) = $suw \times mpy + 1$			

Table 9.9 **Tw(W) and the Bit Settings: MPY1, MPY0, EWW1, and EWW0 (unit: cycles)**

EWW1 and EWW0 Bit Settings		MPY1 and MPY0 Bit Settings			
		00b	01b	10b	11b
		$mpy = 1$	$mpy = 2$	$mpy = 3$	$mpy = 4$
00b	$ww = 1$	0.5 (1)	1.5	2.5	3.5
01b	$ww = 2$	1.5	3.5	5.5	7.5
10b	$ww = 3$	2.5	5.5	8.5	11.5
11b	$ww = 4$	3.5	7.5	11.5	15.5
Formula		Tw(W) = $ww \times mpy - 0.5$			

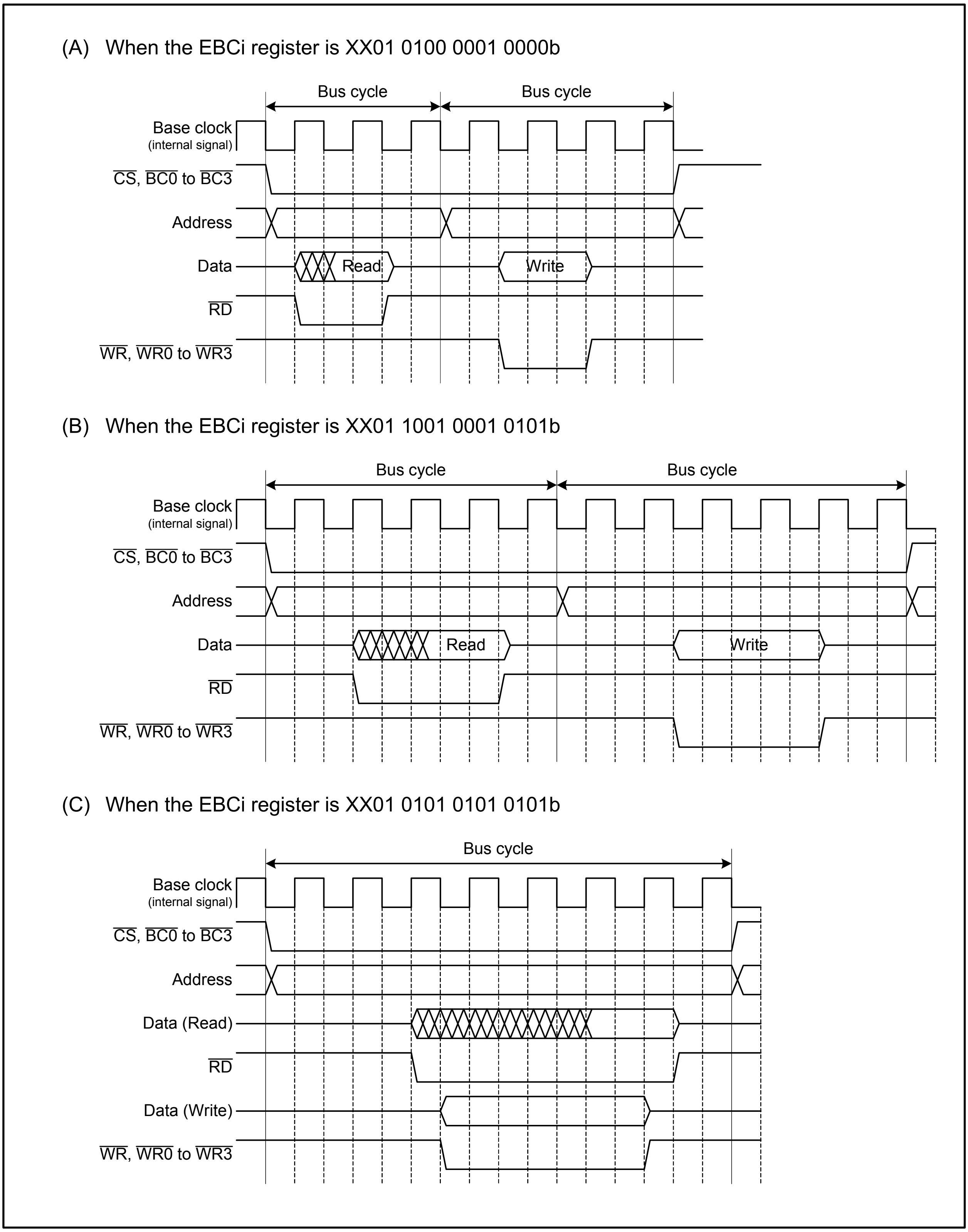
Note:

1. Do not set this value.

Figure 9.13 and 9.14 show examples of external bus timing in separate bus format (the MPX bit is set to 0) and in multiplexed bus format (the MPX bit is set to 1), respectively.

Note that the actual bus cycles are adjusted to be the integral multiple of peripheral bus clock as follows:

- Peripheral bus clock divided by 2: If the calculation result is odd, an idle cycle is inserted so that the bus cycles becomes even.
- Peripheral bus clock divided by 3: If the calculation result is not a multiple of three, (an) idle cycle(s) is/are inserted so that the bus cycles becomes a multiple of three.
- Peripheral bus clock divided by 4: If the calculation result is not a multiple of four, (an) idle cycle(s) is/are inserted so that the bus cycles becomes a multiple of four.

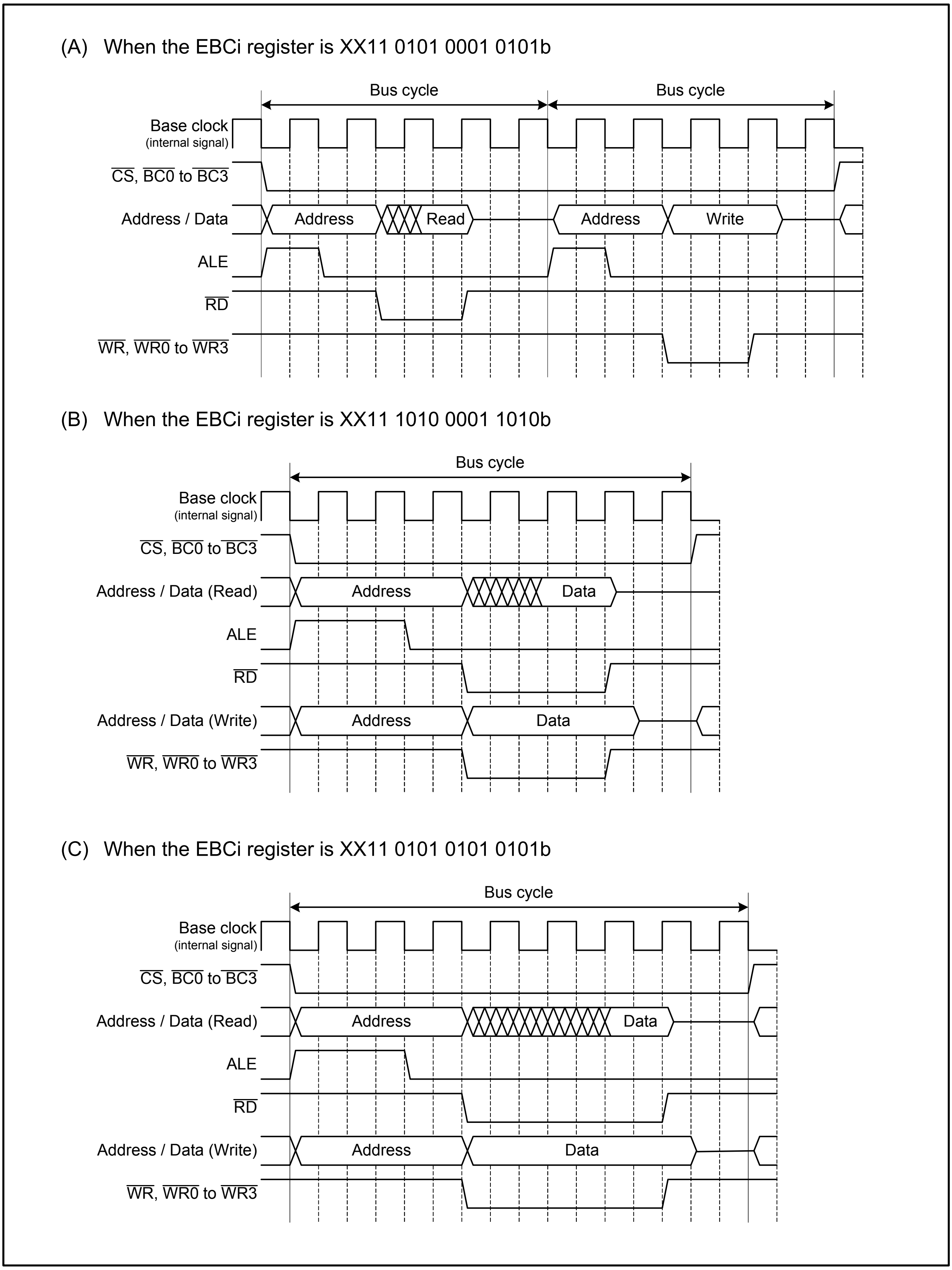


(A) When the EBCi register is XX01 0100 0001 0000b

(B) When the EBCi register is XX01 1001 0001 0101b

(C) When the EBCi register is XX01 0101 0101 0101b

Figure 9.13 External Bus Timing in Separate Bus Format (i = 0 to 3)



(A) When the EBCi register is XX11 0101 0001 0101b

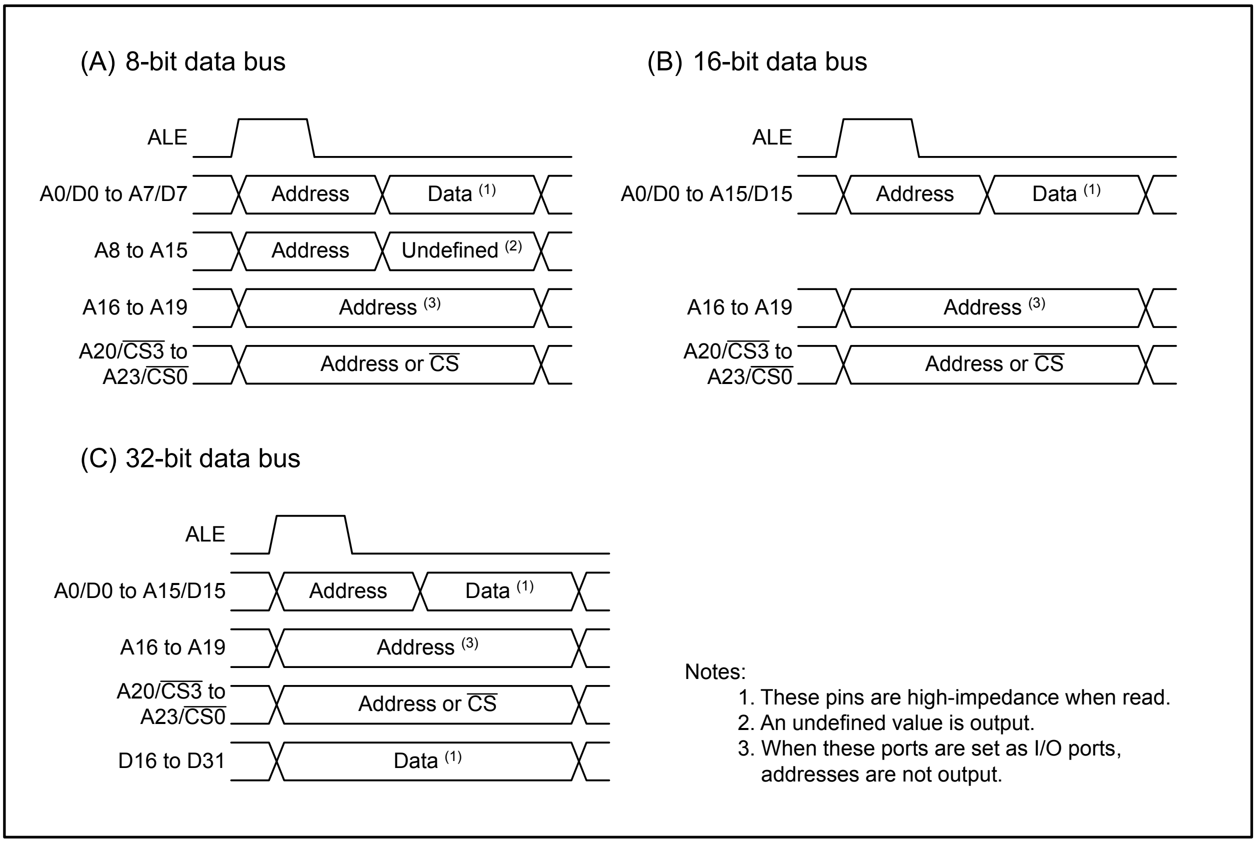
(B) When the EBCi register is XX11 1010 0001 1010b

(C) When the EBCi register is XX11 0101 0101 0101b

Figure 9.14 External Bus Timing in Multiplexed Bus Format (i = 0 to 3)

9.3.6 ALE Signal

The ALE signal latches an address of the multiplexed bus. The address should be latched on the falling edge of the ALE signal. This signal is output to internal space or external space.



Notes:

1. These pins are high-impedance when read.
2. An undefined value is output.
3. When these ports are set as I/O ports, addresses are not output.

Figure 9.15 ALE Signal and Address Bus/Data Bus

The ALE signal becomes high when a bus cycle is started and changes to low at 1/2 base clock before (when Tsu(A-R) is 1 and Tsu(A-W) is 1) or 1 base clock before (when Tsu(A-R) is greater than 1 and Tsu(A-W) is greater than 1) $\overline{\text{RD}}$ or $\overline{\text{WR}}$ becomes low.

9.3.7 $\overline{\text{RDY}}$ Signal

The $\overline{\text{RDY}}$ signal facilitates access to external devices requiring longer access time. It is used when accessing an external device with a lower access rate than the timing set in registers EBC0 to EBC3, or when accessing multiple devices with different access timing in a $\overline{\text{CS}}$ space.

When the RDY bit in registers EBC0 to EBC3 is set to 1 (use $\overline{\text{RDY}}$), the $\overline{\text{RDY}}$ pin is sampled on the every *mpy*th falling edge of the base clock. If the $\overline{\text{RDY}}$ pin is held low when sampled, wait states are inserted into the bus cycle. The sampling continues until the $\overline{\text{RDY}}$ pin is held high so that the bus cycle starts running again.

Since the base clock is not output to external pins, drive the $\overline{\text{RDY}}$ signal low when the $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$ signals are held in a low level, and drive the $\overline{\text{RDY}}$ signal high synchronizing the rise of the BCLK signal.

Figure 9.16 shows an example of $\overline{\text{RDY}}$ signal generator and Table 9.10 lists setting conditions of registers EBC0 to EBC3 to use this circuit. Figure 9.17 shows examples of bus cycle that is extended by the $\overline{\text{RDY}}$ signal.

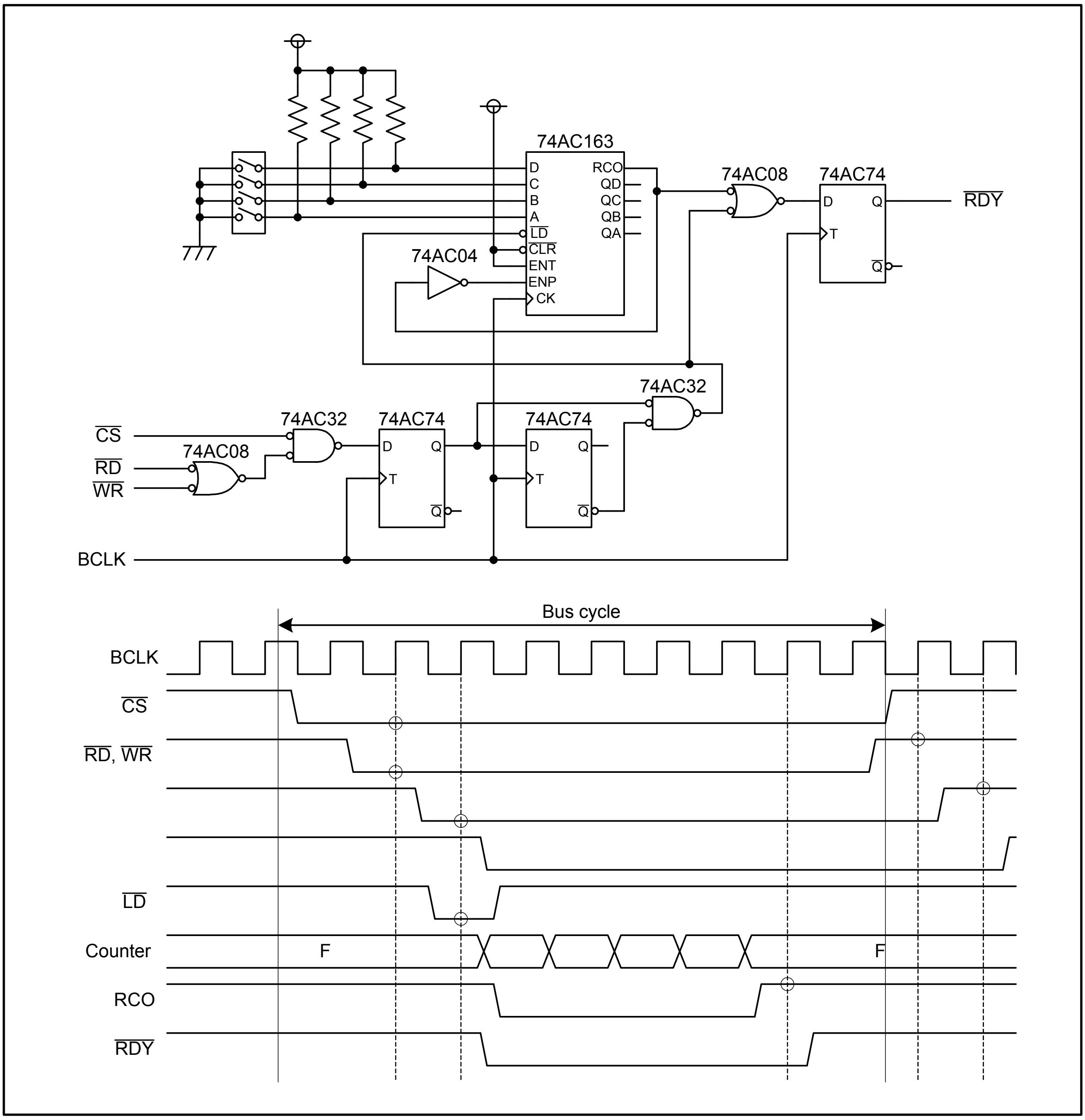
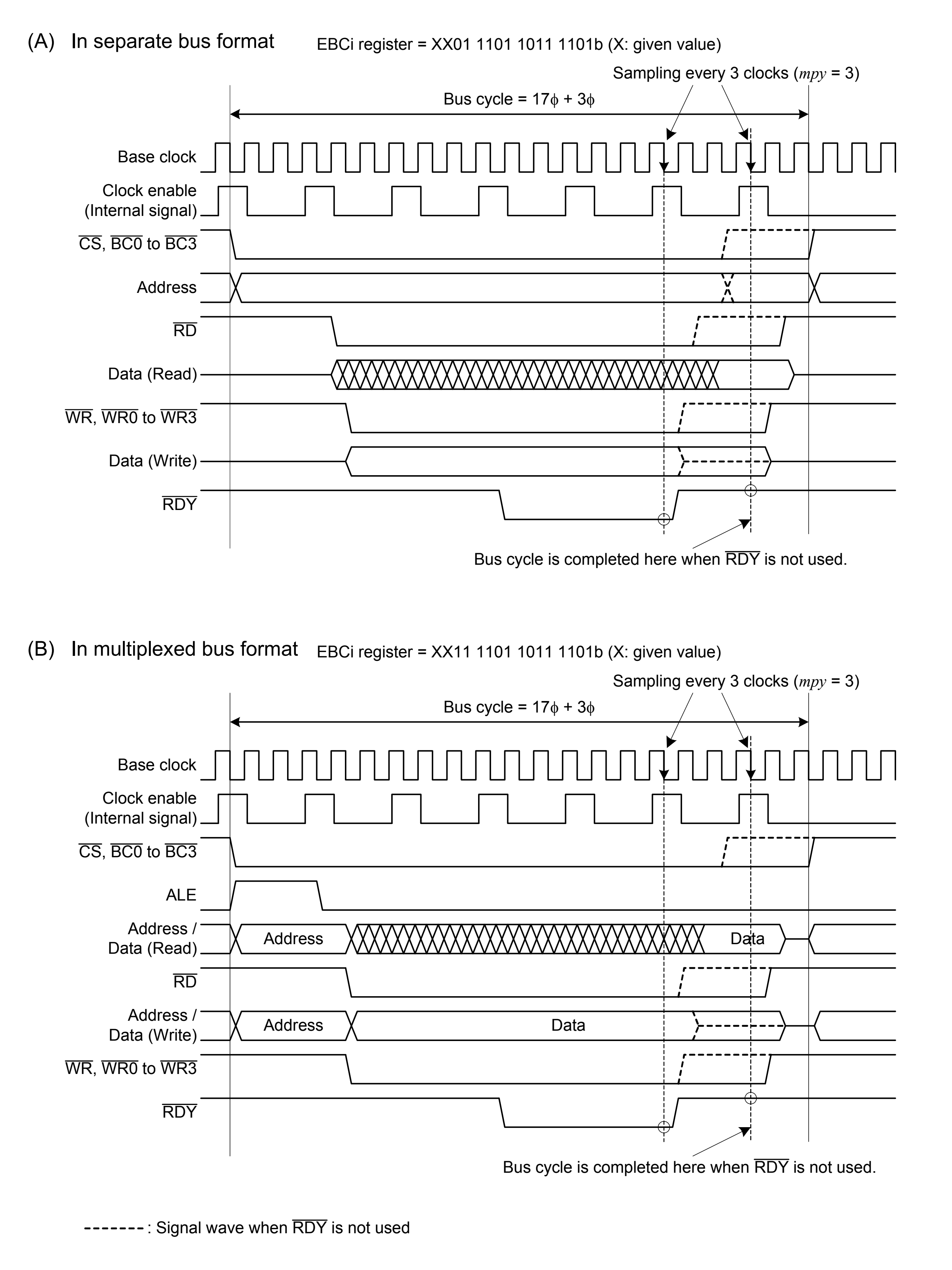


Figure 9.16 $\overline{\text{RDY}}$ Signal Generation Circuitry

Table 9.10 EBCi Register Setting Conditions when Using the Circuit in Figure 9.16 (i = 0 to 3)

Peripheral Bus Clock Frequency	Setting Condition	Setting Example
BCLK = 1/2 base clock	$mpy = 3$ In separate bus format $\overline{\text{RD}}$ pulse width ≥ 9.5 $\overline{\text{WR}}$ pulse width ≥ 11.5 $\overline{\text{RD}}$/$\overline{\text{WR}}$ high level width ≥ 2.5 In multiplexed bus format $\overline{\text{RD}}$ pulse width ≥ 11.5 $\overline{\text{WR}}$ pulse width ≥ 11.5	In separate bus format EBCi = XX01 1101 1011 1001b etc. In multiplexed bus format EBCi = XX11 1101 1011 1101b etc.
BCLK = 1/3 base clock	$mpy = 3$ In separate bus format $\overline{\text{RD}}$ pulse width ≥ 12.5 $\overline{\text{WR}}$ pulse width ≥ 11.5 $\overline{\text{RD}}$/$\overline{\text{WR}}$ high level width ≥ 3.5 In multiplexed bus format $\overline{\text{RD}}$ pulse width ≥ 11.5 $\overline{\text{WR}}$ pulse width ≥ 11.5	In separate bus format EBCi = XX01 1101 1011 1101b etc. In multiplexed bus format EBCi = XX11 1101 1011 1101b etc.
BCLK = 1/4 base clock	$mpy = 4$ In separate bus format $\overline{\text{RD}}$ pulse width ≥ 20.5 $\overline{\text{WR}}$ pulse width ≥ 19.5 $\overline{\text{RD}}$/$\overline{\text{WR}}$ high level width ≥ 4.5 In multiplexed bus format $\overline{\text{RD}}$ pulse width ≥ 19.5 $\overline{\text{WR}}$ pulse width ≥ 19.5	In separate bus format Not available In multiplexed bus format Not available

X: Given value



(A) In separate bus format

EBCi register = XX01 1101 1011 1101b (X: given value)

(B) In multiplexed bus format

EBCi register = XX11 1101 1011 1101b (X: given value)

Bus cycle is completed here when $\overline{\text{RDY}}$ is not used.

-------: Signal wave when $\overline{\text{RDY}}$ is not used

Figure 9.17 An Example of Bus Cycle Extended by $\overline{\text{RDY}}$ Signal (f(BCLK) = 1/2 f(Base)) (i = 0 to 3)

9.3.8 $\overline{\text{HOLD}}$ Signal

The $\overline{\text{HOLD}}$ signal is used when an external bus master requests the external bus from the CPU. When the external bus master drives the $\overline{\text{HOLD}}$ pin low, the CPU outputs a low signal from the $\overline{\text{HLDA}}$ pin after the ongoing bus access is completed. Then the CPU grants the external bus to the external bus master. While the $\overline{\text{HOLD}}$ pin is held low, the CPU does not start the next bus cycle.

To hand over the external bus to the CPU, the external bus master should verify the $\overline{\text{HLDA}}$ pin is held low, and then drive the $\overline{\text{HOLD}}$ pin high.

Table 9.11 lists the MCU state in a hold state.

The bus is used in the following priority order: External bus master, DMAC, and CPU.

Table 9.11 MCU State in Hold State

Item	State
Oscillation	On
Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{BC0}}$ to $\overline{\text{BC3}}$	High-impedance
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$	High-impedance
Programmable I/O port	The state when $\overline{\text{HOLD}}$ was received is held
$\overline{\text{HLDA}}$ pin	Low is output
Internal peripheral circuit	On (excluding the watchdog timer)
ALE pin	Low is output

9.3.9 BCLK Output

The BCLK, which has the same frequency as peripheral bus clock, is a divided clock derived from the PLL clock. In memory expansion mode or microprocessor mode, BCLK is output from port P5_3 when the PM07 bit in the PM0 register is set to 0 (output BCLK) and bits CM01 and CM00 in the CM0 register are set to 00b (I/O port P5_3). In single-chip mode, BCLK cannot be output. Refer to 8. “Clock Generator” for details.

9.4 External Bus State when Accessing Internal Space

Table 9.12 lists the external bus state when accessing an internal space.

Table 9.12 External Bus State when Accessing Internal Space

Pin		Pin State when Accessing SFR	Pin State when Accessing Internal Memory
Address bus		Address is output	The address of an SFR or external space last accessed is held
Data bus	Read cycle	High-impedance	High-impedance
	Write cycle	Data is output	Undefined
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$		High is output	High is output
$\overline{\text{BC0}}$ to $\overline{\text{BC3}}$		$\overline{\text{BC0}}$ to $\overline{\text{BC3}}$ are output	The address of SFR or external space last accessed is held
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$		$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$ are output	High is output
ALE		The ALE signal is output	The ALE signal is output

9.5 Notes on Bus

9.5.1 Notes on Register Settings

9.5.1.1 Chip Select Boundary Select Registers

When not using memory expansion mode, do not change values after a reset for registers CB01, CB12, and CB23.
When using memory expansion mode, set all of these registers to a value within the specified range whether or not each chip select space is used.

10. Protection

This function protects important registers from being easily overwritten when a program goes out of control. Registers used to protect other registers from being rewritten are as follows: PRCR, PRCR2 to PRCR4, and PRR.

10.1 Protect Register (PRCR Register)

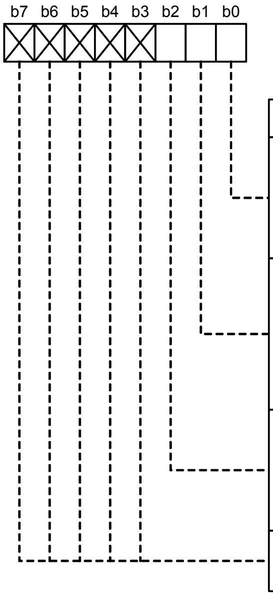
Figure 10.1 shows the PRCR register. Registers protected by bits in the PRCR register are listed in Table 10.1.

Table 10.1 Registers Protected by the PRCR Register

Bit	Protected Registers
PRC0	CM0, CM1, CM2, and PM3
PRC1	PM0, PM2, CSOP0, CSOP1, CSOP2, INVC0, INVC1, IOBC, and I2CMR
PRC2	PLC0, PLC1, PD9, and P9_iS (i = 0 to 7)

The PRC2 bit becomes 0 (write disabled) when a write operation is performed in any other address after this bit is set to 1 (write enabled). Set the PRC2 bit to 1 just before rewriting registers PD9, P9_iS, PLC0, and PLC1 (i = 0 to 7). No interrupt handling or DMA transfers should be inserted between these two instructions. Bits PRC1 and PRC0 do not become 0 even if a write operation is performed in any other address. These bits should be set to 0 by a program.

Protect Register



Symbol
PRCR

Address
4004Ah

Reset Value
XXXX X000b

Bit Symbol	Bit Name	Function	RW
PRC0	Protect Bit 0	Enable writing to registers CM0, CM1, CM2, and PM3 0: Write disabled 1: Write enabled	RW
PRC1	Protect Bit 1	Enable writing to registers PM0, PM2, CSOP0, CSOP1, CSOP2, INVC0, INVC1, IOBC, and I2CMR 0: Write disabled 1: Write enabled	RW
PRC2	Protect Bit 2 [1]	Enable writing to registers PLC0, PLC1, PD9, and P9_iS (i = 0 to 7) 0: Write disabled 1: Write enabled	RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Note:

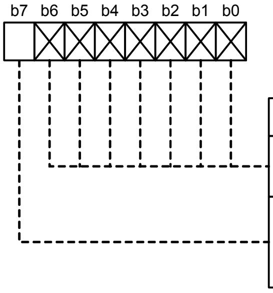
1. The PRC2 bit becomes 0 when a write operation is performed in any other address after this bit is set to 1.

Figure 10.1 PRCR Register

10.2 Protect Register 2 (PRCR2 Register)

Figure 10.2 shows the PRCR2 register which protects the CM3 register only.

Protect Register 2



Symbol PRCR2 | Address 4405Fh | Reset Value 0XXX XXXXb

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	No register bits; should be written with 0 and read as undefined value		—
PRC27	CM3 Protect Bit	Enable writing to the CM3 register 0: Write disabled 1: Write enabled	RW

Figure 10.2 PRCR2 Register

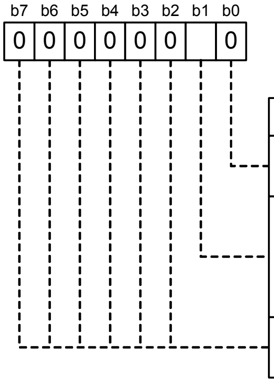
10.3 Protect Register 3 (PRCR3 Register)

Figure 10.3 shows the PRCR3 register. Registers protected by the bits in the PRCR3 register are listed in Table 10.2.

Table 10.2 Registers Protected by the PRCR3 Register

Bit	Protected Registers
PRC31	VRCR, LVDC, and DVCR

Protect Register 3



Symbol PRCR3 | Address 4004Ch | Reset Value 0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	RW
PRC31	Protect Bit 31	Enable writing to registers VRCR, LVDC, and DVCR 0: Write disabled 1: Write enabled	RW
— (b7-b2)	Reserved	Should be written with 0	RW

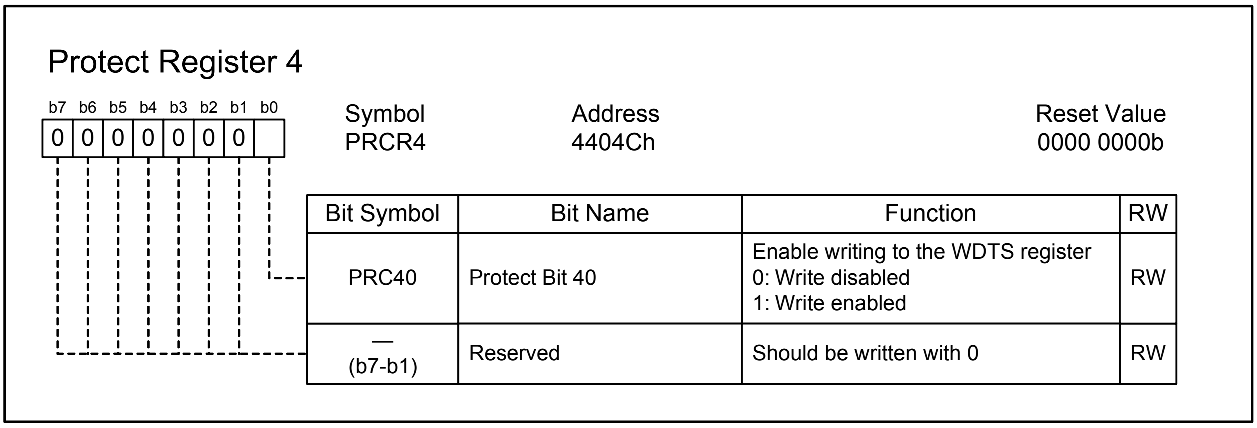
Figure 10.3 PRCR3 Register

10.4 Protect Register 4 (PRCR4 Register)

Figure 10.4 shows the PRCR4 register. Registers protected by the bits in the PRCR4 register are listed in Table 10.3.

Table 10.3 Registers Protected by the PRCR4 Register

Bit	Protected Registers
PRC40	WDTS



Protect Register 4

Symbol	Address	Reset Value
PRCR4	4404Ch	0000 0000b

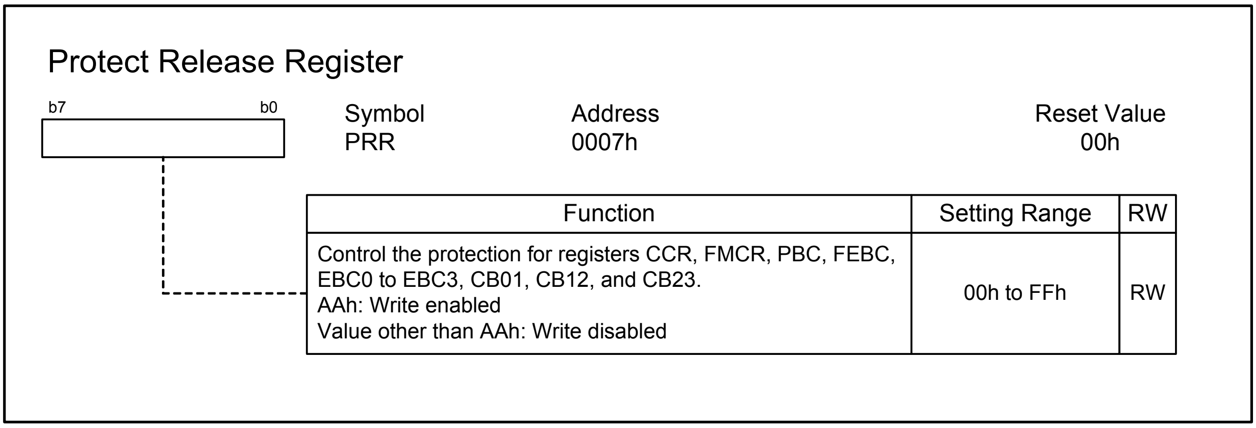
Bit Symbol	Bit Name	Function	RW
PRC40	Protect Bit 40	Enable writing to the WDTS register 0: Write disabled 1: Write enabled	RW
— (b7-b1)	Reserved	Should be written with 0	RW

Figure 10.4 PRCR4 Register

10.5 Protect Release Register (PRR Register)

Figure 10.5 shows the PRR register. Registers protected by the PRR register are as follows: CCR, FMCR, PBC, FEBC, EBC0 to EBC3, CB01, CB12, and CB23.

To write to the registers above, the PRR register should be set to AAh (write enabled). Otherwise, the PRR register should be set to any value other than AAh to protect the above registers from unexpected write accesses.



Protect Release Register

Symbol PRR

Address 0007h

Reset Value 00h

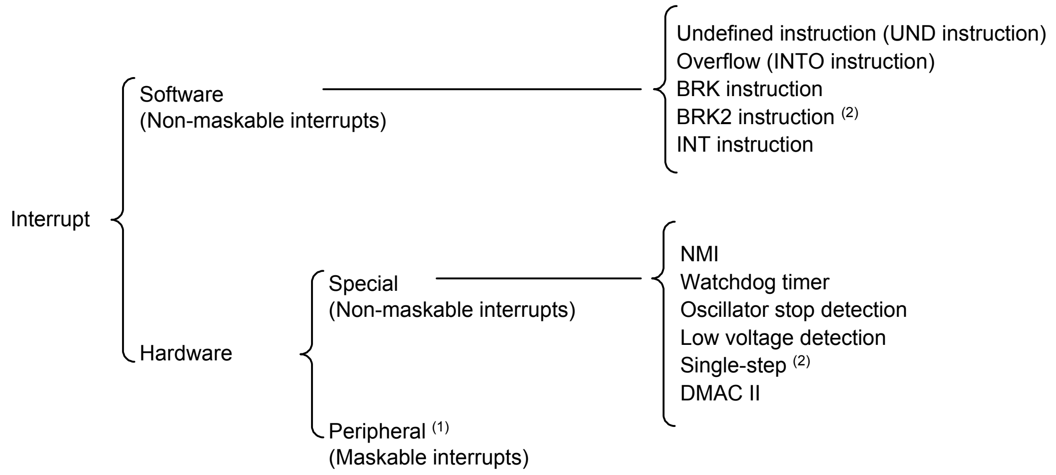
Function	Setting Range	RW
Control the protection for registers CCR, FMCR, PBC, FEBC, EBC0 to EBC3, CB01, CB12, and CB23. AAh: Write enabled Value other than AAh: Write disabled	00h to FFh	RW

Figure 10.5 PRR Register

11. Interrupts

11.1 Interrupt Types

Figure 11.1 shows the types of interrupts.



Notes:

1. The peripheral interrupts are generated by the corresponding peripherals in the MCU.
2. This interrupt is used exclusively as a development support tool. Users are not allowed to use this interrupt.

Figure 11.1 Interrupts

Interrupts are also classified into maskable/non-maskable.

(1) Maskable Interrupts

Maskable interrupts can be disabled by the interrupt enable flag (I flag).
The priority can be configured by assigning an interrupt request level.

(2) Non-maskable Interrupts

Maskable interrupts cannot be disabled by the interrupt enable flag (I flag).
The interrupt priority cannot be configured.

11.2 Software Interrupts

Software interrupts are non-maskable. A software interrupt occurs by executing an instruction.
There are five types of software interrupts shown below.

(1) Undefined Instruction Interrupt

This interrupt occurs when the UND instruction is executed.

(2) Overflow Interrupt

This interrupt occurs when the INTO instruction is executed while the O flag is 1. The following instructions may change the O flag to 1, depending on the operation result:
ABS, ADC, ADCF, ADD, ADDF, ADSF, CMP, CMPF, CNVIF, DIV, DIVF, DIVU, DIVX, EDIV, EDIVU, EDIVX, MUL, MULF, MULU, MULX, NEG, RMPA, ROUND, SBB, SCMPU, SHA, SUB, SUBF, SUNTIL, and SWHILE

(3) BRK Instruction Interrupt

This interrupt occurs when the BRK instruction is executed.

(4) BRK2 Instruction Interrupt

This interrupt occurs when the BRK2 instruction is executed.
This interrupt is only meant for use as a development support tool and users are not allowed to use it.

(5) INT Instruction Interrupt

This interrupt occurs when the INT instruction is executed with a selected software interrupt number from 0 to 255. Software interrupt numbers 0 to 127 are designated for peripheral interrupts. That is, the INT instruction with a software interrupt number from 0 to 127 has the same interrupt handler as that for peripheral interrupts.
The stack pointer (SP) used for this interrupt differs depending on the software interrupt numbers. For software interrupt numbers 0 to 127, when an interrupt request is accepted, the U flag is saved and set to 0 to select the interrupt stack pointer (ISP) during the interrupt sequence. The saved data of the U flag is restored upon returning from the interrupt handler. For software interrupt numbers 128 to 255, the stack pointer does not change during the interrupt sequence.

11.3 Hardware Interrupts

There are two kinds of hardware interrupts: special interrupts and peripheral interrupts.
In peripheral interrupts, only one interrupt with the highest priority can be specified as a fast interrupt.

11.3.1 Special Interrupts

Special interrupts are non-maskable. There are five special interrupts shown below.

(1) NMI (Non Maskable Interrupt)

This interrupt occurs when an input signal at the $\overline{\text{NMI}}$ pin switches from high to low. Refer to 11.11 “NMI” for details.

(2) Watchdog Timer Interrupt

The watchdog timer generates this interrupt. Refer to 12. “Watchdog Timer” for details.

(3) Oscillator Stop Detection Interrupt

This interrupt occurs when the MCU detects a main clock oscillator stop. Refer to 8.2 “Oscillator Stop Detection” for details.

(4) Low Voltage Detection Interrupt

This interrupt occurs when a low voltage input to VCC is detected by the voltage detector. Refer to 6.2 “Low Voltage Detector” for details.

(5) Single-step Interrupt

This interrupt is only meant for use as a development support tool and users are not allowed to use it.

11.3.2 Peripheral Interrupts

Peripheral interrupts occur when an interrupt request from a peripheral in the MCU is accepted. They share the interrupt vector with software interrupt numbers 0 to 127 for the INT instruction. Peripheral interrupts are maskable.
Refer to Tables 11.2 to 11.5 for details on the interrupt sources. Refer to the relevant descriptions for details on each function.

11.4 Fast Interrupt

A fast interrupt enables the CPU to accelerate interrupt response. In peripheral interrupts, only one interrupt with the highest priority can be specified as the fast interrupt.

Use the following procedure to enable a fast interrupt:

(1) Set the both FSIT bit in registers RIPL1 and RIPL2 to 1 (interrupt request level 7 available for fast interrupt).

(2) Set the both DMAII bit in registers RIPL1 and RIPL2 to 0 (interrupt request level 7 available for interrupts).

(3) Set the start address of the fast interrupt handler to the VCT register.

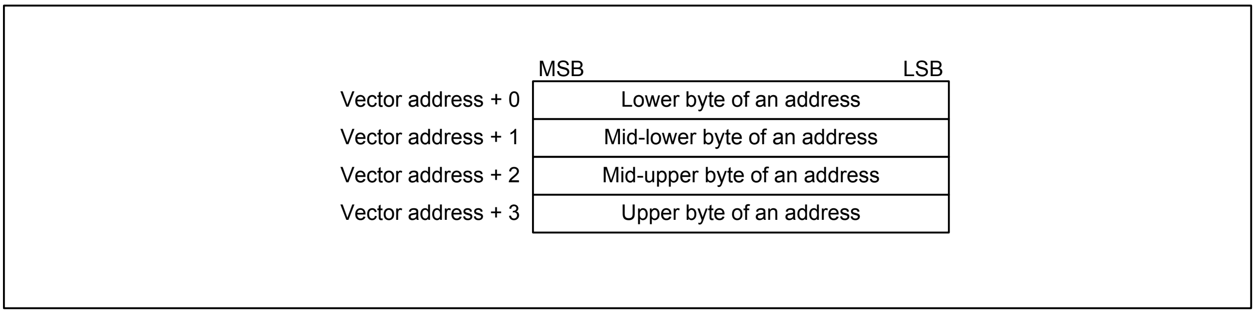
Under the conditions above, bits ILVL2 to ILVL0 in the interrupt control register should be set to 111b (level 7) to enable the fast interrupt. No other interrupts should be set to interrupt request level 7.

When the fast interrupt is accepted, the flag register (FLG) and program counter (PC) are saved to the save flag register (SVF) and save PC register (SVP), respectively. The program is executed from the address indicated by the VCT register.

To return from the fast interrupt handler, the FREIT instruction should be executed. The values saved into registers SVF and SVP are restored to the FLG register and PC, respectively.

11.5 Interrupt Vectors

Each interrupt vector has a 4-byte memory space, in which the start address of the associated interrupt handler is stored. When an interrupt request is accepted, a jump to the address set in the interrupt vector takes place. Figure 11.2 shows an interrupt vector.



	MSB LSB
Vector address + 0	Lower byte of an address
Vector address + 1	Mid-lower byte of an address
Vector address + 2	Mid-upper byte of an address
Vector address + 3	Upper byte of an address

Figure 11.2 Interrupt Vector

11.5.1 Fixed Vector Table

The fixed vector table is allocated in addresses FFFFFFDCh to FFFFFFFFh. Table 11.1 lists the fixed vector table.

Table 11.1 Fixed Vector Table

Interrupt Source	Vector Addresses (Address (L) to Address (H))	Remarks	Reference
Undefined instruction	FFFFFFDCh to FFFFFFDFh	Interrupt by the UND instruction	R32C/100 Series Software Manual
Overflow	FFFFFFE0h to FFFFFFE3h	Interrupt by the INTO instruction	
BRK instruction	FFFFFFE4h to FFFFFFE7h	If address FFFFFFE7h is FFh, a jump to the interrupt vector of software interrupt number 0 in the relocatable vector table takes place	
—	FFFFFFE8h to FFFFFFEBh	Reserved	
—	FFFFFFECh to FFFFFFEFh	Reserved	
Watchdog timer Oscillator stop detection Low voltage detection	FFFFFFF0h to FFFFFFF3h	These addresses are shared by the watchdog timer interrupt, oscillator stop detection interrupt, and low voltage detection interrupt	12. “Watchdog Timer” 8. “Clock Generator” 6.2 “Low Voltage Detector”
—	FFFFFFF4h to FFFFFFF7h	Reserved	
NMI	FFFFFFF8h to FFFFFFFBh	External interrupt by the $\overline{\text{NMI}}$ pin	
Reset	FFFFFFFCh to FFFFFFFFh		5. “Resets”

11.5.2 Relocatable Vector Table

The relocatable vector table occupies a 1024-byte memory space from the start address set in the INTB register. Tables 11.2 to 11.5 list the relocatable vector table entries.
An address in a multiple of 4 should be set in the INTB register for a faster interrupt sequence.

Table 11.2 Relocatable Vector Table (1/4)

Interrupt Source	Vector Table Relative Addresses (Address (L) to Address (H)) [1]	Software Interrupt Number	Reference
BRK instruction [2]	+0 to +3 (0000h to 0003h)	0	R32C/100 Series Software Manual
Reserved	+4 to +7 (0004h to 0007h)	1	
UART5 transmission, NACK [3]	+8 to +11 (0008h to 000Bh)	2	18. “Serial Interface”
UART5 reception, ACK [3]	+12 to +15 (000Ch to 000Fh)	3	
UART6 transmission, NACK [3]	+16 to +19 (0010h to 0013h)	4	
UART6 reception, ACK [3]	+20 to +23 (0014h to 0017h)	5	
Bus collision detection, start condition detection, or stop condition detection (UART5 or UART6) [3, 4]	+24 to +27 (0018h to 001Bh)	6	
Reserved	+28 to +31 (001Ch to 001Fh)	7	
DMA0 transfer complete	+32 to +35 (0020h to 0023h)	8	13. “DMAC”
DMA1 transfer complete	+36 to +39 (0024h to 0027h)	9	
DMA2 transfer complete	+40 to +43 (0028h to 002Bh)	10	
DMA3 transfer complete	+44 to +47 (002Ch to 002Fh)	11	
Timer A0	+48 to +51 (0030h to 0033h)	12	16.1 “Timer A”
Timer A1	+52 to +55 (0034h to 0037h)	13	
Timer A2	+56 to +59 (0038h to 003Bh)	14	
Timer A3	+60 to +63 (003Ch to 003Fh)	15	
Timer A4	+64 to +67 (0040h to 0043h)	16	
UART0 transmission, NACK [3]	+68 to +71 (0044h to 0047h)	17	18. “Serial Interface”
UART0 reception, ACK [3]	+72 to +75 (0048h to 004Bh)	18	
UART1 transmission, NACK [3]	+76 to +79 (004Ch to 004Fh)	19	
UART1 reception, ACK [3]	+80 to +83 (0050h to 0053h)	20	
Timer B0	+84 to +87 (0054h to 0057h)	21	16.2 “Timer B”
Timer B1	+88 to +91 (0058h to 005Bh)	22	
Timer B2	+92 to +95 (005Ch to 005Fh)	23	
Timer B3	+96 to +99 (0060h to 0063h)	24	
Timer B4	+100 to +103 (0064h to 0067h)	25	
INT5	+104 to +107 (0068h to 006Bh)	26	11.10 “External Interrupt”
INT4	+108 to +111 (006Ch to 006Fh)	27	
INT3	+112 to +115 (0070h to 0073h)	28	
INT2	+116 to +119 (0074h to 0077h)	29	
INT1	+120 to +123 (0078h to 007Bh)	30	
INT0	+124 to +127 (007Ch to 007Fh)	31	
Timer B5	+128 to +131 (0080h to 0083h)	32	16.2 “Timer B”

Notes:

1. Each entry is relative to the base address in the INTB register.
2. Interrupts from this source cannot be disabled by the I flag.
3. In I^{2}C mode, interrupts are generated by NACK, ACK, or detection of a start condition/stop condition.
4. The IFSR16 bit in the IFSR1 register selects either the interrupt source in UART5 or UART6.

Table 11.3 Relocatable Vector Table (2/4)

Interrupt Source	Vector Table Relative Addresses (Address (L) to Address (H)) (1)	Software Interrupt Number	Reference
UART2 transmission, NACK (2)/I²C-bus interface (3)	+132 to +135 (0084h to 0087h)	33	18. "Serial Interface"/24. "Multi-master I²C-bus Interface"
UART2 reception, ACK (2)/I²C-bus line (3)	+136 to +139 (0088h to 008Bh)	34	
UART3 transmission, NACK (2)	+140 to +143 (008Ch to 008Fh)	35	
UART3 reception, ACK (2)	+144 to +147 (0090h to 0093h)	36	
UART4 transmission, NACK (2)	+148 to +151 (0094h to 0097h)	37	
UART4 reception, ACK (2)	+152 to +155 (0098h to 009Bh)	38	
Bus collision detection, start condition detection or stop condition detection (UART2) (2)	+156 to +159 (009Ch to 009Fh)	39	
Bus collision detection, start condition detection or stop condition detection (UART3 or UART0) (2, 4)	+160 to +163 (00A0h to 00A3h)	40	
Bus collision detection, start condition detection or stop condition detection (UART4 or UART1) (2, 4)	+164 to +167 (00A4h to 00A7h)	41	
A/D0	+168 to +171 (00A8h to 00ABh)	42	19. "A/D Converter"
Key input	+172 to +175 (00ACh to 00AFh)	43	11.12 "Key Input Interrupt"
Intelligent I/O interrupt 0	+176 to +179 (00B0h to 00B3h)	44	11.13 "Intelligent I/O Interrupt", 23. "Intelligent I/O"
Intelligent I/O interrupt 1	+180 to +183 (00B4h to 00B7h)	45	
Intelligent I/O interrupt 2	+184 to +187 (00B8h to 00BBh)	46	
Intelligent I/O interrupt 3	+188 to +191 (00BCh to 00BFh)	47	
Intelligent I/O interrupt 4	+192 to +195 (00C0h to 00C3h)	48	
Intelligent I/O interrupt 5	+196 to +199 (00C4h to 00C7h)	49	
Intelligent I/O interrupt 6	+200 to +203 (00C8h to 00CBh)	50	
Intelligent I/O interrupt 7	+204 to +207 (00CCh to 00CFh)	51	
Intelligent I/O interrupt 8	+208 to +211 (00D0h to 00D3h)	52	
Intelligent I/O interrupt 9	+212 to +215 (00D4h to 00D7h)	53	
Intelligent I/O interrupt 10	+216 to +219 (00D8h to 00DBh)	54	
Intelligent I/O interrupt 11	+220 to +223 (00DCh to 00DFh)	55	
Reserved	+224 to +227 (00E0h to 00E3h)	56	
Reserved	+228 to +231 (00E4h to 00E7h)	57	
Reserved	+232 to +235 (00E8h to 00EBh)	58	
Reserved	+236 to +239 (00ECh to 00EFh)	59	
Reserved	+240 to +243 (00F0h to 00F3h)	60	
Reserved	+244 to +247 (00F4h to 00F7h)	61	
Reserved	+248 to +251 (00F8h to 00FBh)	62	
Reserved	+252 to +255 (00FCh to 00FFh)	63	

Notes:

1. Each entry is relative to the base address in the INTB register.
2. In I²C mode, interrupts are generated by NACK, ACK, or detection of a start condition/stop condition.
3. Select an interrupt source either of UART2 or I²C-bus interface by setting the I2CEN bit in the I2CMR register.
4. The IFSR06 bit in the IFSR0 register selects either the interrupt source in UART0 or UART3. The IFSR07 bit selects either the interrupt source in UART1 or that in UART4.

Table 11.4 Relocatable Vector Table (3/4) (1)

Interrupt Source	Vector Table Relative Addresses (Address (L) to Address (H)) (2)	Software Interrupt Number	Reference
Reserved	+256 to +259 (0100h to 0103h)	64	
Reserved	+260 to +263 (0104h to 0107h)	65	
Reserved	+264 to +267 (0108h to 010Bh)	66	
Reserved	+268 to +271 (010Ch to 010Fh)	67	
Reserved	+272 to +275 (0110h to 0113h)	68	
Reserved	+276 to +279 (0114h to 0117h)	69	
Reserved	+280 to +283 (0118h to 011Bh)	70	
Reserved	+284 to +287 (011Ch to 011Fh)	71	
Reserved	+288 to +291 (0120h to 0123h)	72	
Reserved	+292 to +295 (0124h to 0127h)	73	
Reserved	+296 to +299 (0128h to 012Bh)	74	
Reserved	+300 to +303 (012Ch to 012Fh)	75	
Reserved	+304 to +307 (0130h to 0133h)	76	
Reserved	+308 to +311 (0134h to 0137h)	77	
Reserved	+312 to +315 (0138h to 013Bh)	78	
Reserved	+316 to +319 (013Ch to 013Fh)	79	
Reserved	+320 to +323 (0140h to 0143h)	80	
Reserved	+324 to +327 (0144h to 0147h)	81	
Reserved	+328 to +331 (0148h to 014Bh)	82	
Reserved	+332 to +335 (014Ch to 014Fh)	83	
Reserved	+336 to +339 (0150h to 0153h)	84	
Reserved	+340 to +343 (0154h to 0157h)	85	
Reserved	+344 to +347 (0158h to 015Bh)	86	
Reserved	+348 to +351 (015Ch to 015Fh)	87	
Reserved	+352 to +355 (0160h to 0163h)	88	
Reserved	+356 to +359 (0164h to 0167h)	89	
Reserved	+360 to +363 (0168h to 016Bh)	90	
Reserved	+364 to +367 (016Ch to 016Fh)	91	
Reserved	+368 to +371 (0170h to 0173h)	92	
INT8	+372 to +375 (0174h to 0177h)	93	11.10 “External Interrupt”
INT7	+376 to +379 (0178h to 017Bh)	94	
INT6	+380 to +383 (017Ch to 017Fh)	95	

Notes:

1. Entries in this table cannot be used to exit wait mode or stop mode.
2. Each entry is relative to the base address in the INTB register.

Table 11.5 Relocatable Vector Table (4/4) [1]

Interrupt Source	Vector Table Relative Addresses (Address (L) to Address (H)) [2]	Software Interrupt Number	Reference
Reserved	+384 to +387 (0180h to 0183h)	96	
Reserved	+388 to +391 (0184h to 0187h)	97	
Reserved	+392 to +395 (0188h to 018Bh)	98	
Reserved	+396 to +399 (018Ch to 018Fh)	99	
Reserved	+400 to +403 (0190h to 0193h)	100	
Reserved	+404 to +407 (0194h to 0197h)	101	
Reserved	+408 to +411 (0198h to 019Bh)	102	
Reserved	+412 to +415 (019Ch to 019Fh)	103	
Reserved	+416 to +419 (01A0h to 01A3h)	104	
Reserved	+420 to +423 (01A4h to 01A7h)	105	
Reserved	+424 to +427 (01A8h to 01ABh)	106	
Reserved	+428 to +431 (01ACh to 01AFh)	107	
Reserved	+432 to +435 (01B0h to 01B3h)	108	
Reserved	+436 to +439 (01B4h to 01B7h)	109	
Reserved	+440 to +443 (01B8h to 01BBh)	110	
Reserved	+444 to +447 (01BCh to 01BFh)	111	
Reserved	+448 to +451 (01C0h to 01C3h)	112	
Reserved	+452 to +455 (01C4h to 01C7h)	113	
Reserved	+456 to +459 (01C8h to 01CBh)	114	
Reserved	+460 to +463 (01CCh to 01CFh)	115	
Reserved	+464 to +467 (01D0h to 01D3h)	116	
Reserved	+468 to +471 (01D4h to 01D7h)	117	
Reserved	+472 to +475 (01D8h to 01DBh)	118	
Reserved	+476 to +479 (01DCh to 01DFh)	119	
UART9 transmission	+480 to +483 (01E0h to 01E3h)	120	18. “Serial Interface”
UART9 reception	+484 to +487 (01E4h to 01E7h)	121	
UART10 transmission	+488 to +491 (01E8h to 01EBh)	122	
UART10 reception	+492 to +495 (01ECh to 01EFh)	123	
UART7 transmission	+496 to +499 (01F0h to 01F3h)	124	
UART7 reception	+500 to +503 (01F4h to 01F7h)	125	
UART8 transmission	+504 to +507 (01F8h to 01FBh)	126	
UART8 reception	+508 to +511 (01FCh to 01FFh)	127	
INT instruction [3]	+0 to +3 (0000h to 0003h) to +1020 to +1023 (03FCh to 03FFh)	0 to 255	11.2 “Software Interrupts”

Notes:

1. Entries in this table cannot be used to exit wait mode or stop mode.
2. Each entry is relative to the base address in the INTB register.
3. Interrupts from this source cannot be disabled by the I flag.

11.6 Interrupt Request Acceptance

Software interrupts and special interrupts are accepted whenever their interrupt request is generated. Peripheral interrupts, however, are only accepted if the conditions below are met:

- I flag is 1
- IR bit is 1
- Bits ILVL2 to ILVL0 > IPL

The I flag, IPL, IR bit, and bits ILVL2 to ILVL0 do not affect each other. The I flag and IPL are in the FLG register. The IR bit and bits ILVL2 to ILVL0 are in the interrupt control register.

The following section describes these flag and bits.

11.6.1 I Flag and IPL

The I flag (interrupt enable flag) enables or disables maskable interrupts. When the I flag is set to 1 (enabled), all maskable interrupts are enabled; when it is set to 0 (disabled), they are disabled. The I flag becomes 0 after a reset.

The IPL (processor interrupt priority level) consists of 3 bits and indicates eight interrupt priority levels from 0 to 7. An interrupt becomes acceptable when its interrupt request level is higher than the specified IPL (bits ILVL2 to ILVL0 > IPL).

Table 11.6 lists interrupt request levels classified by the IPL.

Table 11.6 Acceptable Interrupt Request Levels and IPL

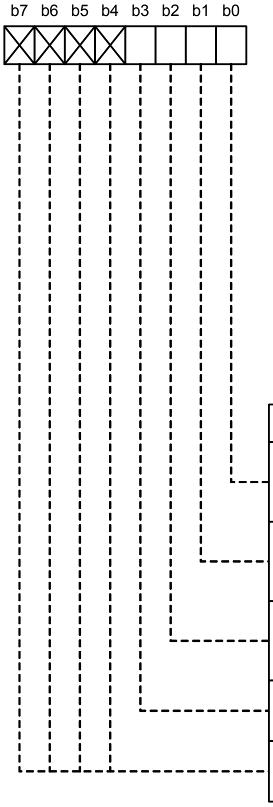
IPL			Acceptable Interrupt Request Levels
IPL2	IPL1	IPL0	
1	1	1	All maskable interrupts are disabled
1	1	0	Level 7 only
1	0	1	Level 6 and above
1	0	0	Level 5 and above
0	1	1	Level 4 and above
0	1	0	Level 3 and above
0	0	1	Level 2 and above
0	0	0	Level 1 and above

11.6.2 Interrupt Control Registers

Each peripheral interrupt is controlled by an interrupt control register.

11.4 show the interrupt control registers.

Interrupt Control Register



Symbol	Address	Reset Value
TA0IC to TA4IC	006Ch, 008Ch, 006Eh, 008Eh, 0070h	XXXX X000b
TB0IC to TB5IC	0094h, 0076h, 0096h, 0078h, 0098h, 0061h	XXXX X000b
S0TIC to S4TIC	0090h, 0092h, 0081h [(1)], 0083h, 0085h	XXXX X000b
S5TIC to S10TIC	0062h, 0064h, 00DDh, 00DFh, 00D9h, 00DBh	XXXX X000b
S0RIC to S4RIC	0072h, 0074h, 0063h [(2)], 0065h, 0067h	XXXX X000b
S5RIC to S10RIC	0082h, 0084h, 00FDh, 00FFh, 00F9h, 00FBh	XXXX X000b
BCN0IC to BCN3IC	0069h, 0089h, 0087h, 0069h [(3)]	XXXX X000b
BCN4IC to BCN6IC	0089h [(4)], 0066h, 0066h [(5)]	XXXX X000b
DM0IC to DM3IC	0068h, 0088h, 006Ah, 008Ah	XXXX X000b
AD0IC	006Bh	XXXX X000b
KUPIC	008Bh	XXXX X000b
IIO0IC to IIO5IC	006Dh, 008Dh, 006Fh, 008Fh, 0071h, 0091h	XXXX X000b
IIO6IC to IIO11IC	0073h, 0093h, 0075h, 0095h, 0077h, 0097h	XXXX X000b
I2CIC, I2CLIC	0081h [(1)], 0063h [(2)]	XXXX X000b

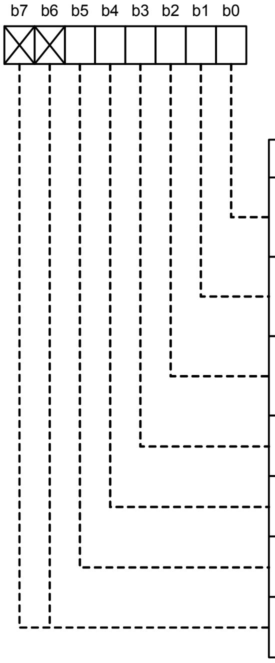
Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt Request Level Select Bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt Request Flag	0: No interrupt requested 1: Interrupt requested [(6)]	RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. The S2TIC register shares an address with the I2CIC register.
2. The S2RIC register shares an address with the I2CLIC register.
3. The BCN0IC register shares an address with the BCN3IC register.
4. The BCN1IC register shares an address with the BCN4IC register.
5. The BCN5IC register shares an address with the BCN6IC register.
6. This bit can only be set to 0 (do not set it to 1).

Figure 11.3 Interrupt Control Register (1/2)

Interrupt Control Register



Symbol	Address	Reset Value
INT0IC to INT2IC	009Eh, 007Eh, 009Ch	XX00 X000b
INT3IC to INT5IC [1]	007Ch, 009Ah, 007Ah	XX00 X000b
INT6IC to INT8IC	00FEh, 00DEh, 00FCh	XX00 X000b

Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt Request Level Select Bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt Request Flag	0: No interrupt requested 1: Interrupt requested [2]	RW
POL	Polarity Select Bit	0: Select the falling edge or a low 1: Select the rising edge or a high [3]	RW
LVS	Level/Edge Sensitive Select Bit	0: Edge sensitive 1: Level sensitive [4]	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. When the 16- or 32-bit data bus is used in microprocessor mode or memory expansion mode, pins $\overline{\text{INT3}}$ to $\overline{\text{INT5}}$ function as data bus. In this case, set bits ILVL2 to ILVL0 in registers INT3IC to INT5IC to 000b.
2. This bit can only be set to 0 (do not set it to 1).
3. Set this bit to 0 (the falling edge) to set the corresponding bit in registers IFSR0 and IFSR1 to 1 (both edges).
4. Set the corresponding bit in registers IFSR0 and IFSR1 to 0 (one edge) to select the level sensitive.

Figure 11.4 Interrupt Control Register (2/2)

Bits ILVL2 to ILVL0

The interrupt request level is selected by setting bits ILVL2 to ILVL0. The higher the level is, the higher interrupt priority is.

When an interrupt request is generated, its request level is compared to the IPL. The interrupt is accepted only when the interrupt request level is higher than the IPL. When bits ILVL2 to ILVL0 are set to 000b, the interrupt is disabled.

IR bit

The IR bit becomes 1 (interrupt requested) when an interrupt request is generated; this bit setting is retained until the interrupt request is accepted. When the request is accepted and a jump to the corresponding interrupt vector takes place, the IR bit becomes 0 (no interrupt requested).

The IR bit can be set to 0 by a program. This bit should not be set to 1.

When rewriting the interrupt control register, no corresponding interrupt request should be generated. If there is a possibility that an interrupt request may be generated, disable the interrupt request before rewriting the register.
When enabling an interrupt immediately after changing the interrupt control register, insert NOPs between two instructions or perform a dummy read of the interrupt control register so that the interrupt enable flag (I flag) cannot become 1 (interrupt enabled) before writing to the interrupt control register is completed.
If an interrupt request is generated for the register being rewritten, the IR bit may not become 1 depending on the instruction being used. If it matters, use one of the following instructions to rewrite the register:

- AND
- OR
- BCLR
- BSET

If the AND or BCLR instruction is used to set the IR bit to 0, the IR bit may not become 0 as these instructions cause the interrupt request to be retained during the rewrite. To prevent this from happening, rewrite the register using the MOV instruction. To set just the IR bit to 0, first temporarily store the read value to memory or a CPU internal register, then execute either the AND or BCLR instruction in the stored area. After that, write the value back to the register using the MOV instruction.

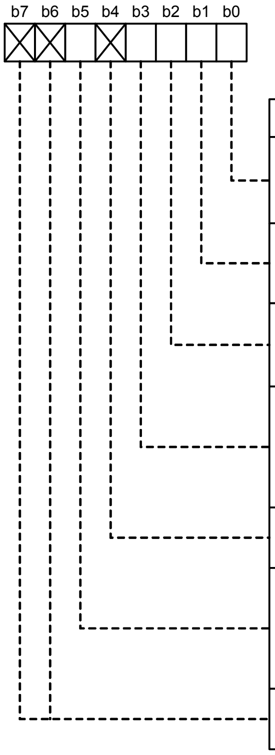
11.6.3 Wake-up IPL Setting Register

Set the wake-up IPL setting registers (registers RIPL1 and RIPL2) when using an interrupt to exit wait or stop mode, or using the fast interrupt.

Refer to 8.7.2 “Wait Mode”, 8.7.3 “Stop Mode”, or 11.4 “Fast Interrupt” for details.

Figure 11.5 shows registers RIPL1 and RIPL2.

Wake-up IPL Setting Register i (i = 1, 2) (1)



Symbol	Address	Reset Value
RIPL1, RIPL2	4407Fh, 4407Dh	XX0X 0000b

Bit Symbol	Bit Name	Function	RW
RLVL0	Interrupt Priority Level for Wake-up Select Bit (2)	b2 b1 b0 0 0 0 : Level 0 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
RLVL1			RW
RLVL2			RW
FSIT	Fast Interrupt Select Bit (3)	0: Use interrupt request level 7 for normal interrupt 1: Use interrupt request level 7 for fast interrupt (4)	RW
— (b4)	No register bit; should be written with 0 and read as undefined value		—
DMAII	DMA II Select Bit (5)	0: Use interrupt request level 7 for interrupt 1: Use interrupt request level 7 for DMA II transfer (4)	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Registers RIPL1 and RIPL2 should be set with the same values.
2. The MCU exits wait mode or stop mode when the request level of the requested interrupt is higher than the level selected using bits RLVL2 to RLVL0. Set these bits to the same value as the IPL in the FLG register.
3. When the FSIT bit is 1, an interrupt with interrupt request level 7 becomes the fast interrupt. In this case, set the interrupt request level to level 7 with only one interrupt.
4. Set either the FSIT or DMAII bit to 1. The fast interrupt and DMAC II cannot be used simultaneously.
5. Set bits ILVL2 to ILVL0 in the interrupt control register after the DMAII bit is set. DMA II transfer is not affected by the I flag or IPL.

Figure 11.5 Registers RIPL1 and RIPL2

11.6.4 Interrupt Sequence

An interrupt sequence is performed from when an interrupt request has been accepted until the interrupt handler starts.

When an interrupt request is generated while an instruction is being executed, the requested interrupt is evaluated in the priority resolver after the current instruction is completed, and the interrupt sequence starts from the next cycle.

However, for instructions RMPA, SCMPU, SIN, SMOVB, SMOVF, SMOVU, SOUT, SSTR, SUNTIL, and SWHILE, when an interrupt request is generated while an instruction is being executed, the current instruction is suspended, and the interrupt sequence starts.

The interrupt sequence is as follows:

(1) The CPU acknowledges the interrupt request to obtain the interrupt information (the interrupt number, and the interrupt request level) from the interrupt controller. Then the corresponding IR bit becomes 0 (no interrupt requested).

(2) The FLG register value before the interrupt sequence is stored to a temporary register in the CPU. The temporary register is inaccessible to users.

(3) The following bits in the FLG register become 0:
- The I flag (interrupt enable flag): interrupt disabled
- The D flag (debug flag): single-step interrupt disabled
- The U flag (stack pointer select flag): ISP selected

(4) The temporary register value in the CPU is saved to the stack, or to the SVF register in case of the fast interrupt.

(5) The PC value is saved to the stack, or to the SVP register in case of the fast interrupt.

(6) The interrupt request level for the accepted interrupt is set in the IPL (processor interrupt priority level).

(7) The corresponding interrupt vector is read from the interrupt vector table.

(8) This interrupt vector is stored into the PC.

After the interrupt sequence is completed, an instruction is executed from the start address of the interrupt handler.

11.6.5 Interrupt Response Time

The interrupt response time, as shown in Figure 11.6, consists of two non-overlapping time segments: (a) the period from when an interrupt request is generated until the instruction being executed is completed; and (b) the period required for the interrupt sequence.

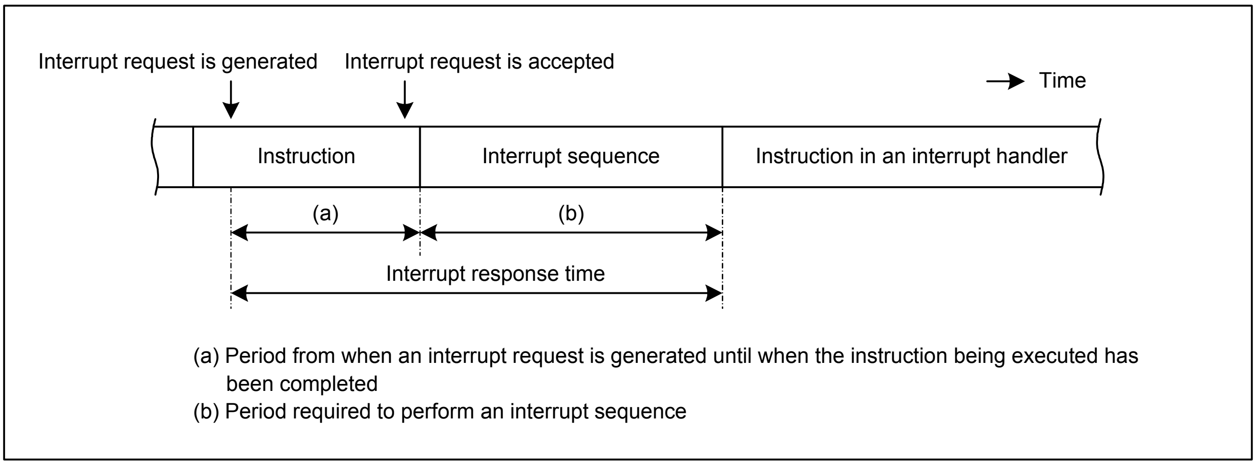

Interrupt request is generated
Interrupt request is accepted
Time
Instruction
Interrupt sequence
Instruction in an interrupt handler
(a)
(b)
Interrupt response time
(a) Period from when an interrupt request is generated until when the instruction being executed has been completed
(b) Period required to perform an interrupt sequence

Figure 11.6 Interrupt Response Time

Period (a) varies depending on the instruction being executed. Instructions, such as LDCTX and STCTX in which registers are sequentially saved into or restored from the stack, require the longest time. For example, the STCTX instruction requires at least 30 cycles for 10 registers to be saved. It requires more time if the WAIT instruction is in the stack.
Period (b) is listed in Table 11.7.

Table 11.7 Interrupt Sequence Execution Time [1]

Interrupt	Execution Time in Terms of CPU Clock
Peripherals	13 + α cycles [2]
INT instruction	11 cycles
NMI	10 cycles
Watchdog timer Oscillator stop detection Low voltage detection	11 cycles
Undefined instruction	12 cycles
Overflow	12 cycles
BRK instruction (relocatable vector table)	16 cycles
BRK instruction (fixed vector table)	19 cycles
BRK2 instruction	19 cycles
Fast interrupt	11 cycles

Notes:

1. These are the values when the interrupt vectors are aligned to the addresses in multiples of 4 in the internal ROM. However, the condition does not apply to the fast interrupt.
2. α is the number of waits to access SFRs minus 2.

11.6.6 IPL after Accepting an Interrupt Request

When a peripheral interrupt request is accepted, the interrupt request level is set in the IPL (processor interrupt priority level).

Software interrupts and special interrupts have no interrupt request level. When these interrupt requests are accepted, the value listed in Table 11.8 is set in the IPL as the interrupt request level.

Table 11.8 Interrupts without Interrupt Request Level and IPL

Interrupt Sources without Interrupt Request Level	IPL Value to be Set
NMI, watchdog timer, oscillator stop detection, low voltage detection	7
Reset	0
Software	Unchanged

11.6.7 Register Saving

In the interrupt sequence, the FLG register and PC values are saved to the stack, in that order. Figure 11.7 shows the stack status before and after an interrupt request is accepted.

In the fast interrupt sequence, the FLG register and PC values are saved to registers SVF and SVP, respectively.

If there are any other registers to be saved to the stack, save them at the beginning of the interrupt handler. A single PUSHM instruction saves all registers except the frame base register (FB) and stack pointer (SP).

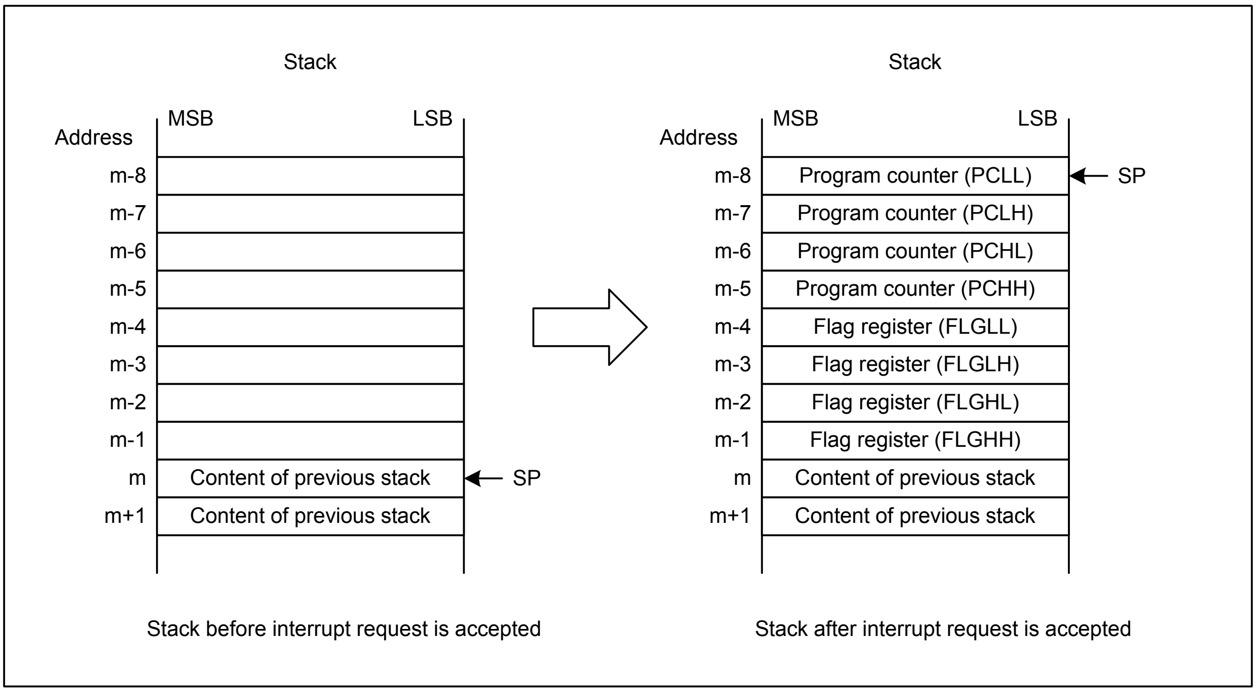


Figure 11.7 Stack Before and After an Interrupt Request is Accepted

11.7 Register Restoring from Interrupt Handler

When the REIT instruction is executed at the end of the interrupt handler, the FLG register and PC values, which are saved in the stack, are restored, and the program resumes the operation that was interrupted. In the fast interrupt, execute the FREIT instruction to restore them from the save registers, instead.

To restore the register values which are saved by software in the interrupt handler, use an instruction such as POPM before the REIT or FREIT instruction.

If the register bank is switched in the interrupt handler, the bank is automatically switched back to the original register bank by the REIT or FREIT instruction.

11.8 Interrupt Priority

If two or more interrupt requests are detected at an interrupt request sampling point, the interrupt request with higher priority is accepted.

For maskable interrupts (peripheral interrupts), the interrupt request level select bits (bits ILVL2 to ILVL0) select a request level. If two or more interrupt requests have the same request level, the interrupt with higher priority, predetermined by hardware, is accepted.

The priorities of the reset and special interrupts, such as the watchdog timer interrupt, are determined by the hardware. Note that the reset has the highest priority. The following is the priority order determined by the hardware:

Watchdog timer
Reset > Oscillator stop detection > NMI > Peripherals
Low voltage detection

Software interrupts are not governed by priority. A jump to the interrupt handler takes place whenever the relevant instruction is executed.

11.9 Priority Resolver

The priority resolver selects an interrupt that has the highest priority among requested interrupts detected at the same sampling point.

Figure 11.8 shows the priority resolver.

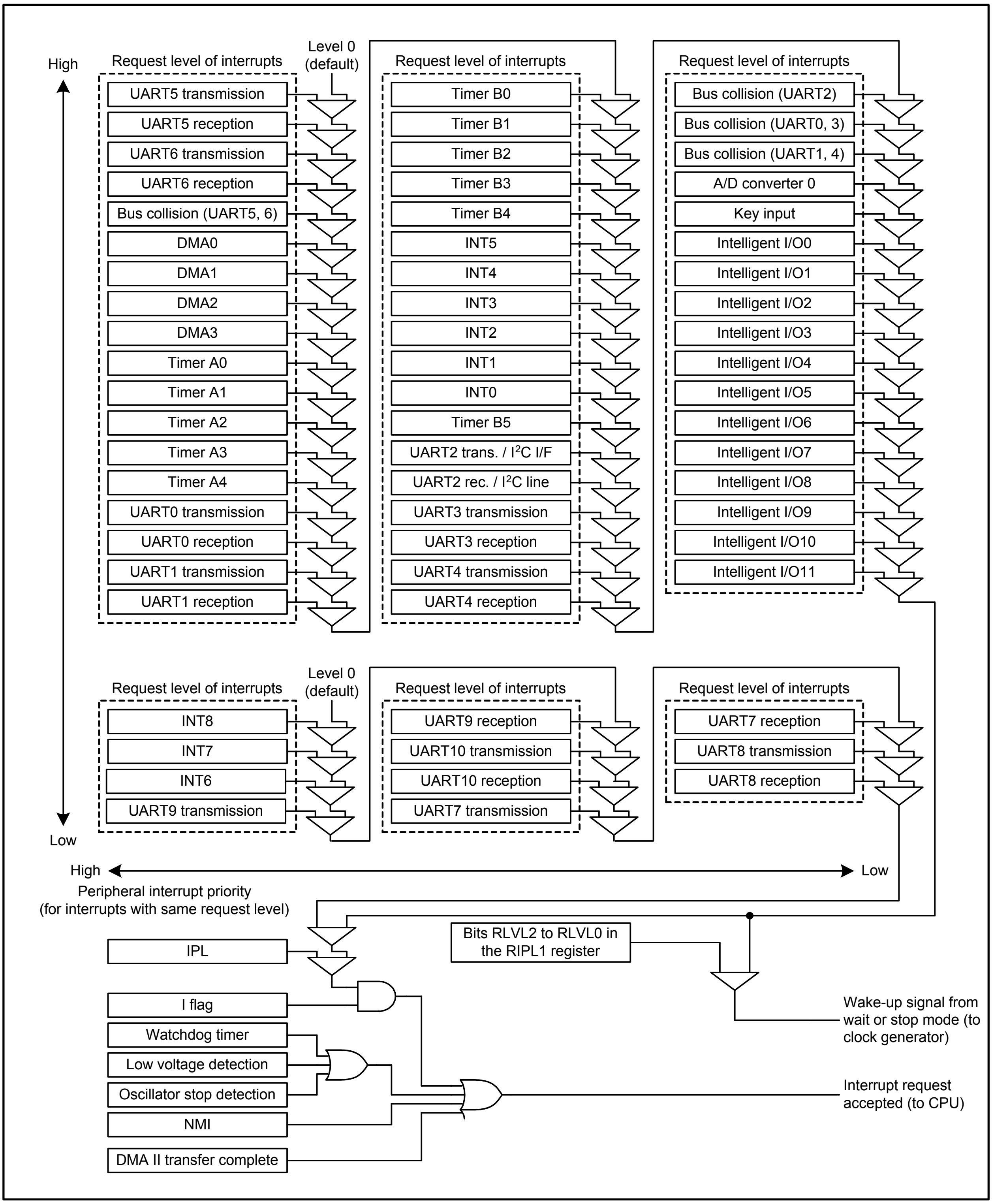


Figure 11.8 Priority Resolver

11.10 External Interrupt

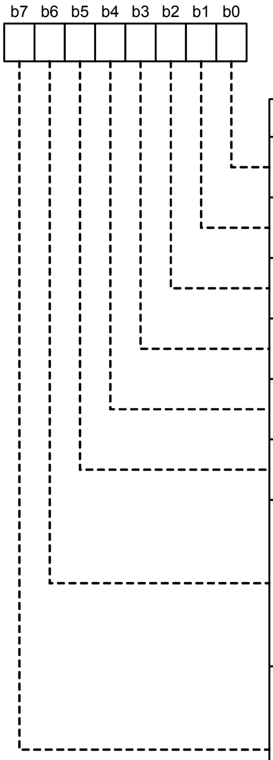
An external interrupt occurs by an external input applied to the $\overline{\text{INTi}}$ pin (i = 0 to 8). Set the LVS bit in the INTiIC register to select whether an interrupt is triggered by the effective edge(s) (edge sensitive), or by the effective level (level sensitive) of the input signal. The polarity of the input signal is selected by setting the POL bit in the same register.

When using edge-triggered interrupts, setting the IFSR0j bit in the IFSR0 register to 1 (both edges) causes interrupt requests to be generated on both rising and falling edges of the external input applied to the $\overline{\text{INTj}}$ pin (j = 0 to 5). This also applies to setting the IFSR1n bit (n = m - 6) in the IFSR1 register to 1 (both edges) for the $\overline{\text{INTm}}$ pin (m = 6 to 8). Set the POL bit in the corresponding register to 0 (falling edge) to set the IFSR0j bit or the IFSR1n bit to 1.

When using level-triggered interrupts, set the IFSR0j or IFSR1n bit to 0 (one edge). When an effective level, which is selected by the POL bit, is detected on the $\overline{\text{INTi}}$ pin, the IR bit in the INTiIC register becomes 1. The IR bit does not become 0 even if the signal level at the $\overline{\text{INTi}}$ pin changes. This bit is set to 0 when the INTi interrupt is accepted or it is set to 0 by a program.

Figures 11.9 and 11.10 show registers IFSR0 and IFSR1, respectively.

External Interrupt Request Source Select Register 0



Symbol	Address	Reset Value
IFSR0	4406Fh	0000 0000b

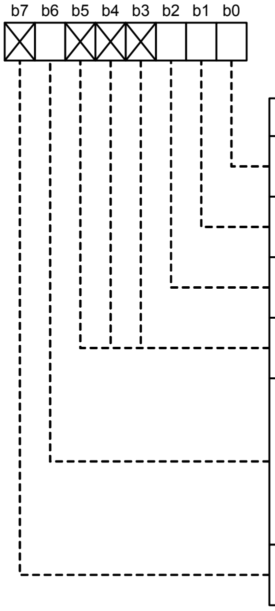
Bit Symbol	Bit Name	Function	RW
IFSR00	$\overline{\text{INT0}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
IFSR01	$\overline{\text{INT1}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
IFSR02	$\overline{\text{INT2}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
IFSR03	$\overline{\text{INT3}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
IFSR04	$\overline{\text{INT4}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
IFSR05	$\overline{\text{INT5}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
IFSR06	UART0/UART3 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART3 1: Bus collision, start condition detection, stop condition detection in UART0	RW
IFSR07	UART1/UART4 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART4 1: Bus collision, start condition detection, stop condition detection in UART1	RW

Note:

1. Set this bit to 0 to select the level sensitive input as trigger. To set this bit to 1, set the POL bit in the corresponding INTiIC register to 0 (falling edge) (i = 0 to 5).

Figure 11.9 IFSR0 Register

External Interrupt Request Source Select Register 1



Symbol	Address	Reset Value
IFSR1	4406Dh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
IFSR10	$\overline{\text{INT6}}$ Pin Polarity Select Bit [1]	0: One edge 1: Both edges	RW
IFSR11	$\overline{\text{INT7}}$ Pin Polarity Select Bit [1]	0: One edge 1: Both edges	RW
IFSR12	$\overline{\text{INT8}}$ Pin Polarity Select Bit [1]	0: One edge 1: Both edges	RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
IFSR16	UART5/UART6 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART5 1: Bus collision, start condition detection, stop condition detection in UART6	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Note:

1. Set this bit to 0 to select the level sensitive input as trigger. To set this bit to 1, set the POL bit in the corresponding INTiIC register (i = 6 to 8) to 0 (falling edge).

Figure 11.10 IFSR1 Register

11.11 NMI

The NMI (non maskable interrupt) occurs when an input signal at the $\overline{\text{NMI}}$ pin switches from high to low. This non maskable interrupt is disabled after a reset. To enable this interrupt, set the PM24 bit in the PM2 register to 1 after setting the interrupt stack pointer (ISP) at the beginning of the program. The $\overline{\text{NMI}}$ pin shares a pin with port P8_5, which enables the P8_5 bit in the P8 register to indicate the input level at the $\overline{\text{NMI}}$ pin.

Note:

1. When not using the NMI, do not change the reset value of the PM24 bit in the PM2 register.

11.12 Key Input Interrupt

The key input interrupt is enabled by setting ports P10_4 to P10_7 as input ports.

The interrupt request is generated if any of the signals applied to ports P10_4 to P10_7 switch from high to low. This interrupt also functions as key wake-up to exit wait or stop mode. Figure 11.11 shows a block diagram of the key input interrupt. If any of the ports are held low, signals applied to other ports are not detected as interrupt request signals.

To use the key input interrupt, every register from P10_4S to P10_7S should be set to 00h (I/O port) and bits PD10_4 to PD10_7 should be set to 0 (input). This is the only setting available for the key input interrupt.

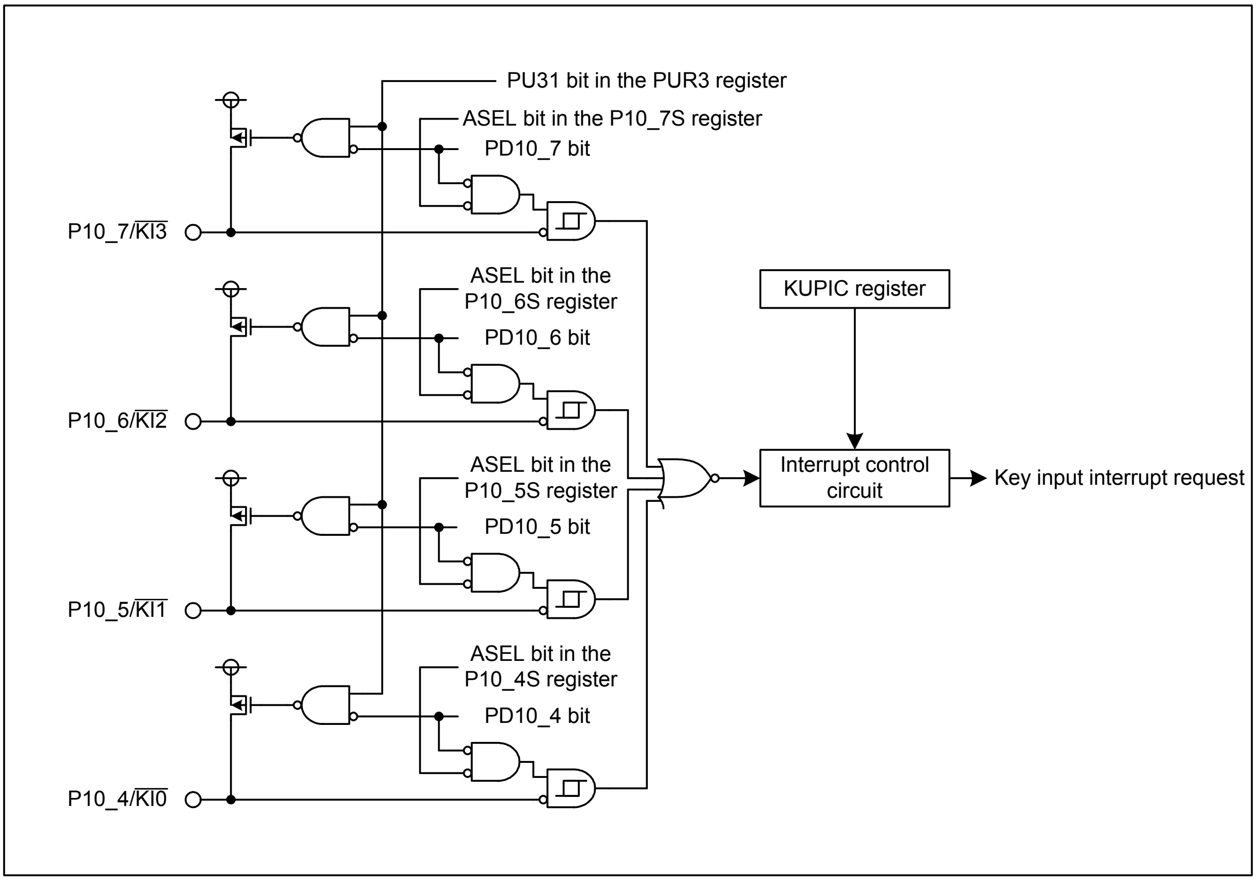

PU31 bit in the PUR3 register
ASEL bit in the P10_7S register
PD10_7 bit
P10_7/KI3
ASEL bit in the P10_6S register
PD10_6 bit
P10_6/KI2
KUPIC register
Interrupt control circuit
Key input interrupt request
ASEL bit in the P10_5S register
PD10_5 bit
P10_5/KI1
ASEL bit in the P10_4S register
PD10_4 bit
P10_4/KI0

Figure 11.11 Key Input Interrupt Block Diagram

11.13 Intelligent I/O Interrupt

The intelligent I/O interrupt is assigned to software interrupt numbers 44 to 55.

Figure 11.12 shows a block diagram of the intelligent I/O interrupt. Figures 11.13 and 11.14 show registers IIOiIR and IIOiIE, respectively (i = 0 to 11).

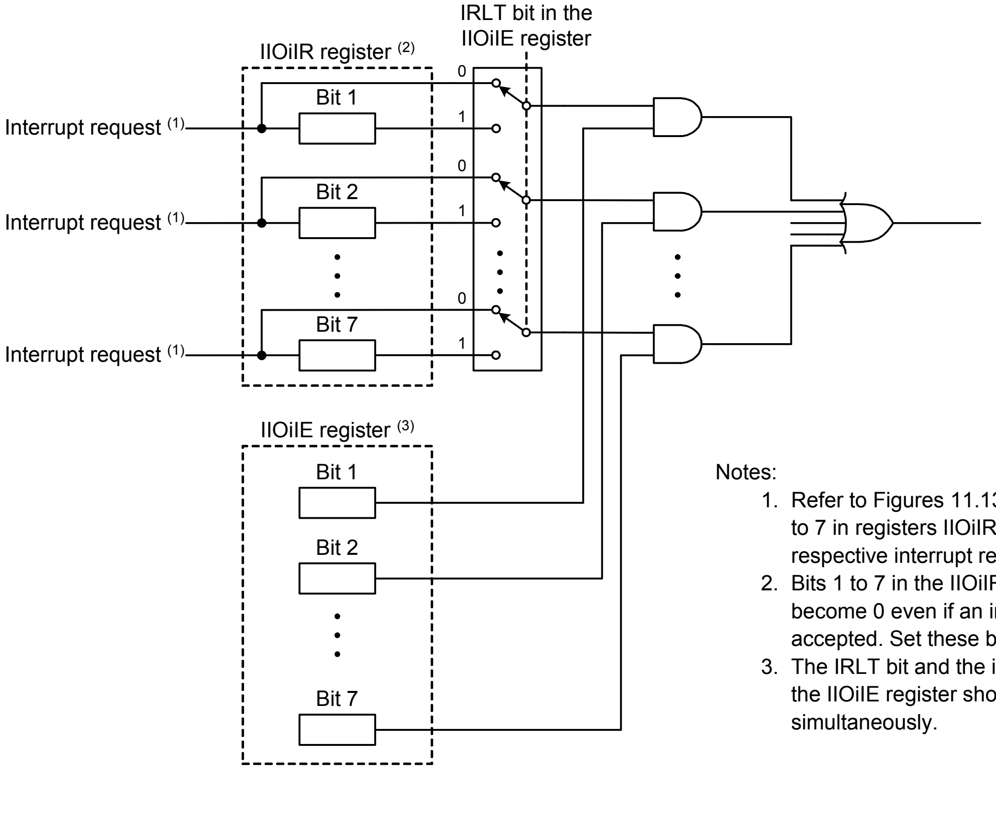
To use the intelligent I/O interrupt, set the IRLT bit in the IIOiIE register to 1 (interrupt requests used for interrupt).

The intelligent I/O interrupt has multiple request sources. When an interrupt request is generated with an intelligent I/O function, the corresponding bit in the IIOiIR register becomes 1 (interrupt requested). If the corresponding bit in the IIOiIE register is 1 (interrupt enabled), the IR bit in the corresponding IIOiIC register changes to 1 (interrupt requested).

After the IR bit setting changes from 0 to 1, it remains unchanged if a bit in the IIOiIR register becomes 1 by another interrupt request source and the corresponding bit in the IIOiIE register is 1.

Bits in the IIOiIR register do not become 0 even if an interrupt is accepted. They should be set to 0 by either the AND or BCLR instruction. Note that every generated interrupt request is ignored until these bits are set to 0.

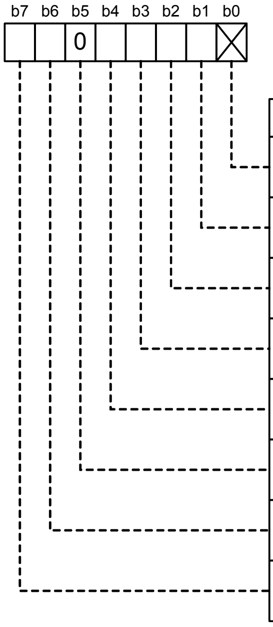
To use the intelligent I/O interrupt as a DMAC II trigger, set the IRLT bit in the IIOiIE register to 0 (interrupt requests used for DMA or DMA II) and the bit used for the interrupt source to 1 (interrupt enabled) in the IIOiIE register.



Notes:
1. Refer to Figures 11.13 and 11.14 for bits 1 to 7 in registers IIOiIR and IIOiIE and their respective interrupt request sources.
2. Bits 1 to 7 in the IIOiIR register do not become 0 even if an interrupt request is accepted. Set these bits to 0 by a program.
3. The IRLT bit and the interrupt enable bit in the IIOiIE register should not be rewritten simultaneously.

Figure 11.12 Intelligent I/O Interrupt Block Diagram (i = 0 to 11)

Intelligent I/O Interrupt Request Register i (i = 0 to 11)



Symbol
IIO0IR to IIO11IR

Address
Refer to the table below

Reset Value
??0? ???1b [1]

Bit Symbol	Bit Name	Function	RW
— (b0)	No register bit; this bit is read as 1		—
(Note 2)	0: No interrupt requested 1: Interrupt requested [3]		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested [3]		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested [3]		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested [3]		RW
— (b5)	Reserved	Should be written with 0	RW
(Note 2)	0: No interrupt requested 1: Interrupt requested [3]		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested [3]		RW

Notes:

1. When the register has any function-assigned bit, the reset value is X (undefined); otherwise, the reset value is 0.
2. Refer to the table below for bit symbols.
3. When this bit is function-assigned, it can only be set to 0. It should not be set to 1. To set it to 0, either the AND or BCLR instruction should be used; when the bit is not function-assigned (reserved), it should be set to 0.

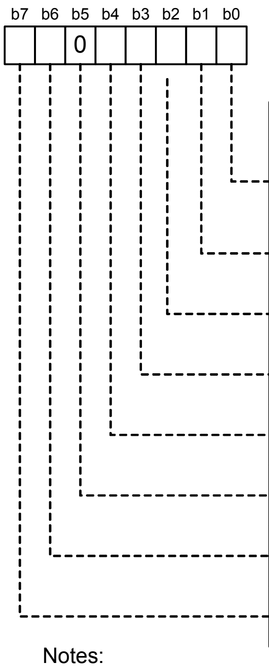
Bit Symbols for the Intelligent I/O Interrupt Request Register

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IIO0IR	00A0h	—	—	—	—	—	TM13R/PO13R	TM02R/PO02R	—
IIO1IR	00A1h	—	—	—	—	—	TM14R/PO14R	TM00R/PO00R	—
IIO2IR	00A2h	—	—	—	—	—	TM12R/PO12R	—	—
IIO3IR	00A3h	—	—	—	—	PO27R	TM10R/PO10R	TM03R/PO03R	—
IIO4IR	00A4h	—	—	—	BT1R	—	TM17R/PO17R	TM04R/PO04R	—
IIO5IR	00A5h	—	—	—	SIO2RR	—	PO21R	TM05R/PO05R	—
IIO6IR	00A6h	—	—	—	SIO2TR	—	PO20R	TM06R/PO06R	—
IIO7IR	00A7h	IE0R	—	—	BT0R	—	PO22R	TM07R/PO07R	—
IIO8IR	00A8h	IE1R	IE2R	—	BT2R	—	PO23R	TM11R/PO11R	—
IIO9IR	00A9h	—	INT6R	—	—	—	PO24R	TM15R/PO15R	—
IIO10IR	00AAh	—	INT7R	—	—	—	PO25R	TM16R/PO16R	—
IIO11IR	00ABh	—	INT8R	—	—	—	PO26R	TM01R/PO01R	—

BTxR: Intelligent I/O group x base timer interrupt request (x = 0 to 2)
TMxyR: Intelligent I/O group x time measurement channel y interrupt request (x = 0, 1; y = 0 to 7)
POxyR: Intelligent I/O group x waveform generation channel y interrupt request (x = 0 to 2; y = 0 to 7)
IEzR: Intelligent I/O group 2 IEBus interrupt request (z = 0 to 2)
SIO2RR: Intelligent I/O group 2 receive interrupt request
SIO2TR: Intelligent I/O group 2 transmit interrupt request
INTmR: INTm interrupt request (m = 6 to 8)

Figure 11.13 Registers IIO0IR to IIO11IR

Intelligent I/O Interrupt Enable Register i (i = 0 to 11)



Symbol: IIO0IE to IIO11IE
Address: Refer to the table below.
Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
IRLT	Interrupt Request Select Bit (2)	0: Use interrupt requests for DMA or DMA II 1: Use interrupt requests for interrupt	RW
(Note 1)	0: Disable the interrupt of bit 1 in the IIOiIR register 1: Enable the interrupt of bit 1 in the IIOiIR register		RW
(Note 1)	0: Disable the interrupt of bit 2 in the IIOiIR register 1: Enable the interrupt of bit 2 in the IIOiIR register		RW
(Note 1)	0: Disable the interrupt of bit 3 in the IIOiIR register 1: Enable the interrupt of bit 3 in the IIOiIR register		RW
(Note 1)	0: Disable the interrupt of bit 4 in the IIOiIR register 1: Enable the interrupt of bit 4 in the IIOiIR register		RW
— (b5)	Reserved	Should be written with 0	RW
(Note 1)	0: Disable the interrupt of bit 6 in the IIOiIR register 1: Enable the interrupt of bit 6 in the IIOiIR register		RW
(Note 1)	0: Disable the interrupt of bit 7 in the IIOiIR register 1: Enable the interrupt of bit 7 in the IIOiIR register		RW

Notes:

1. Refer to the table below for bit symbols.
2. To use interrupt requests for interrupt, the IRLT bit should be set to 1, then bits 1 to 4, 6, and 7 should be set to 1.

Bit Symbols for the Intelligent I/O Interrupt Enable Register

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IIO0IE	00B0h	—	—	—	—	—	TM13E/PO13E	TM02E/PO02E	IRLT
IIO1IE	00B1h	—	—	—	—	—	TM14E/PO14E	TM00E/PO00E	IRLT
IIO2IE	00B2h	—	—	—	—	—	TM12E/PO12E	—	IRLT
IIO3IE	00B3h	—	—	—	—	PO27E	TM10E/PO10E	TM03E/PO03E	IRLT
IIO4IE	00B4h	—	—	—	BT1E	—	TM17E/PO17E	TM04E/PO04E	IRLT
IIO5IE	00B5h	—	—	—	SIO2RE	—	PO21E	TM05E/PO05E	IRLT
IIO6IE	00B6h	—	—	—	SIO2TE	—	PO20E	TM06E/PO06E	IRLT
IIO7IE	00B7h	IE0E	—	—	BT0E	—	PO22E	TM07E/PO07E	IRLT
IIO8IE	00B8h	IE1E	IE2E	—	BT2E	—	PO23E	TM11E/PO11E	IRLT
IIO9IE	00B9h	—	INT6E	—	—	—	PO24E	TM15E/PO15E	IRLT
IIO10IE	00BAh	—	INT7E	—	—	—	PO25E	TM16E/PO16E	IRLT
IIO11IE	00BBh	—	INT8E	—	—	—	PO26E	TM01E/PO01E	IRLT

BTxE: Intelligent I/O group x base timer interrupt enabled (x = 0 to 2)
TMxyE: Intelligent I/O group x time measurement channel y interrupt enabled (x = 0, 1; y = 0 to 7)
POxyE: Intelligent I/O group x waveform generation channel y interrupt enabled (x = 0 to 2; y = 0 to 7)
IEzE: Intelligent I/O group 2 IEBus interrupt enabled (z = 0 to 2)
SIO2RE: Intelligent I/O group 2 receive interrupt enabled
SIO2TE: Intelligent I/O group 2 transmit interrupt enabled
INTmE: INTm interrupt enabled (m = 6 to 8)

Figure 11.14 Registers IIO0IE to IIO11IE

11.14 Notes on Interrupts

11.14.1 ISP Setting

The interrupt stack pointer (ISP) is initialized to 00000000h after a reset. Set a value to the ISP before an interrupt is accepted, otherwise the program may go out of control. A multiple of 4 should be set to the ISP, which enables faster interrupt sequence due to less memory access.
When using NMI, in particular, since this interrupt cannot be disabled, set the PM24 bit in the PM2 register to 1 (NMI enabled) after setting the ISP at the beginning of the program.

11.14.2 NMI

- NMI cannot be disabled once the PM24 bit in the PM2 register is set to 1 (NMI enabled). This bit setting should be done only when using NMI.

- When the PM24 bit in the PM2 register is 1 (NMI enabled), the P8_5 bit in the P8 register is enabled just for monitoring the $\overline{\text{NMI}}$ pin state. It is not enabled as a general port.

11.14.3 External Interrupts

- The input signal to the $\overline{\text{INTi}}$ pin requires the pulse width specified in the electrical characteristics (i = 0 to 8). If the pulse width is narrower than the specification, an external interrupt may not be accepted.

- When the effective level or edge of the $\overline{\text{INTi}}$ pin (i = 0 to 8) is changed by the following bits: bits POL, LVS in the INTiIC register, the IFSR0i bit (i = 0 to 5) in the IFSR0 register, and the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register, the corresponding IR bit may become 1 (interrupt requested). When setting the above mentioned bits, preset bits ILVL2 to ILVL0 in the INTiIC register to 000b (interrupt disabled). After setting the above mentioned bits, set the corresponding IR bit to 0 (no interrupt requested), then rewrite bits ILVL2 to ILVL0.

- The interrupt input signals to pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ are also connected to bits INT6R to INT8R in registers IIO9IR to IIO11IR. Therefore, these input signals, when assigned to the intelligent I/O, can be used as a source for exiting wait mode or stop mode. Note that these signals are enabled only on the falling edge and not affected by the following bit settings: bits POL and LVS in the INTiIC register (i = 0 to 8), IFSR0i bit (i = 0 to 5) in the IFSR0 register, and the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register.

12. Watchdog Timer

The watchdog timer is used to detect program runaway. The 15-bit watchdog counter decrements with the cycle which is the peripheral bus clock frequency or on-chip oscillator clock frequency divided by the prescaler.

Select either an interrupt request or a reset with the CM06 bit in the CM0 register for when the watchdog timer underflows. Once the CM06 bit is set to 1 (reset), it cannot be changed to 0 (watchdog timer interrupt) by a program. It can be set to 0 only by a reset. When the CM06 bit is 0, after the watchdog timer underflows, it can reload value to resume counting or stop the operation by setting the WDK5 bit in the WDK register.

The watchdog timer has two prescalers. One is the on-chip oscillator clock divided by 1, 2, 4 or 8; the other is the peripheral bus clock divided by 16 or 128. To select the divide ratio for the former, set bits WDK3 and WDK2 in the WDK register. To select the divide ratio for the latter, set bits WDK4 and WDC7 in the WDC register.

The count source for the watchdog timer is set by the combination of the 23 bit in the PM2 register and the WDK4 bit in the WDK register. When the peripheral bus clock is selected as the count source, the watchdog timer is stopped in wait mode, stop mode, or when the $\overline{\text{HOLD}}$ signal is driven low. It resumes counting from the value held when exiting the mode or state. When the on-chip oscillator clock is selected, and the count source protect mode is enabled, the watchdog timer does not stop. The count source protect mode is selected by setting the PM22 bit in the PM2 register.

The general formula to calculate a watchdog timer period is:

$$\text{Watchdog timer period} = \frac{\text{Prescaler divisor (16 or 128)} \times 32768}{\text{Peripheral bus clock frequency}}$$

or

$$\text{Watchdog timer period} = \frac{\text{Prescaler divider factor ((1, 2, 4, or 8)} \times \text{(1, 16, or 128))} \times 2048}{\text{On-chip oscillator clock frequency}}$$

For example, when the peripheral bus clock is selected as the count source and it is 1/2 of 64 MHz CPU clock and the prescaler has a divide-by-16 operation, the watchdog timer period is approximately 16.4 ms. When the on-chip oscillator clock whose frequency is 125 kHz is selected and it is divided by 8×128, the watchdog timer period is approximately 16.8 s. Depending on the timing of when a value is written to the WDTS register, a marginal error of one prescaler output cycle (maximum) may occur in the watchdog timer period.

The watchdog timer is initialized when a write operation to the WDTS register is performed or when a watchdog timer interrupt request is generated. The prescaler is initialized only when the MCU is reset.

After a reset, the watchdog timer starts counting automatically if the OFS area of the flash memory are preset. When the WDTON bit in the OFS area is 1, both the watchdog timer and the prescaler are stopped. They start counting when a write operation to the WDTS register is performed. When the WDTON bit is 0, both the watchdog timer and the prescaler automatically start counting after a reset.

Figure 12.1 shows a block diagram of the watchdog timer. Figures 12.2 to 12.5 show registers associated with the watchdog timer.

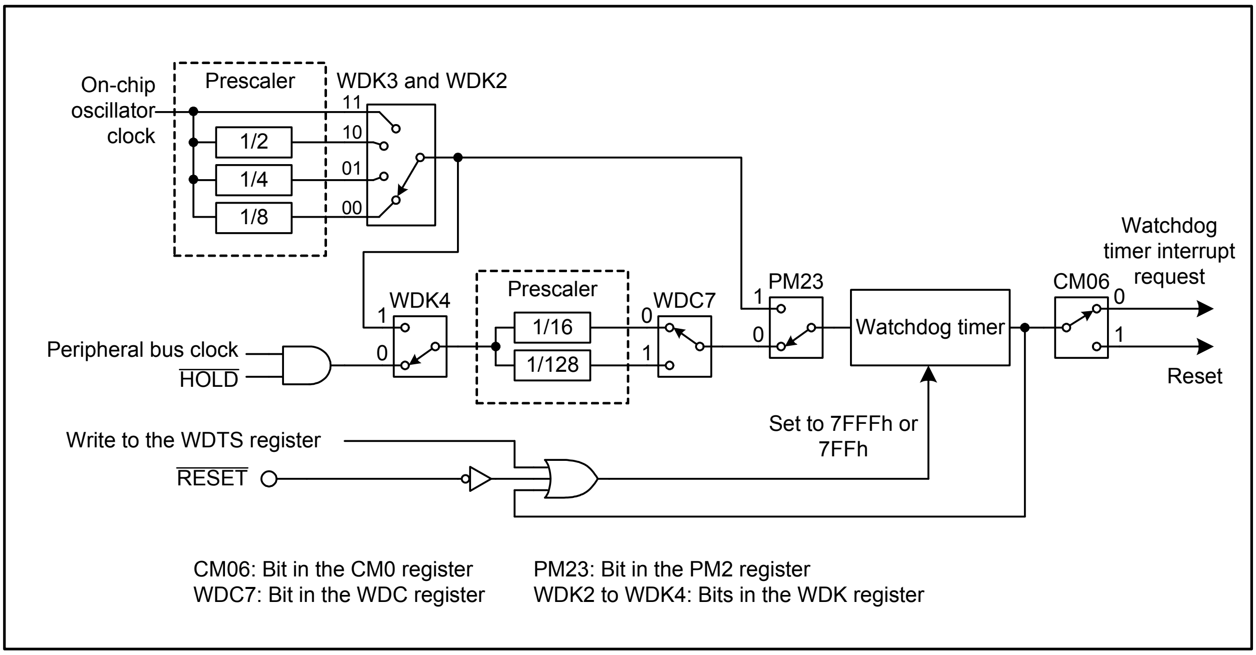

On-chip oscillator clock
Prescaler
1/2
1/4
1/8
WDK3 and WDK2
11
10
01
00
Peripheral bus clock
HOLD
WDK4
1
0
Prescaler
1/16
1/128
WDC7
0
1
PM23
1
0
Watchdog timer
CM06
0
1
Watchdog timer interrupt request
Reset
Set to 7FFFh or 7FFh
Write to the WDTS register
RESET
CM06: Bit in the CM0 register
WDC7: Bit in the WDC register
PM23: Bit in the PM2 register
WDK2 to WDK4: Bits in the WDK register

Figure 12.1 Watchdog Timer Block Diagram

Watchdog Timer Control Register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	Reset Value
WDC	4404Fh	000X XXXXb

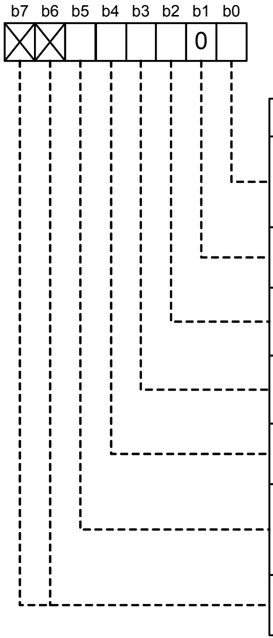
Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Upper 5 bits of the watchdog timer, when the peripheral bus clock is used: b14 to b10 when the on-chip oscillator is used: b10 to b6 (1)		RO
— (b6-b5)	Reserved	Should be written with 0	RW
WDC7	Prescaler Select Bit (2)	0: Divide-by-16 1: Divide-by-128	RW

Notes:

1. When the on-chip oscillator clock is used as the count source, the read value may be undefined due to a change in the count value while being read.
2. Set this bit before activating the watchdog timer.

Figure 12.2 WDC Register

Watchdog Timer Clock Control Register [1]



Symbol	Address	Reset Value
WDK	4404Dh	0000 000?b [2]

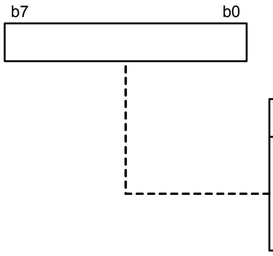
Bit Symbol	Bit Name	Function	RW
WDK0	Reset Source Detect Flag	The most recent reset is caused by 0: Non-watchdog timer 1: Watchdog timer	RO
— (b1)	Reserved	Should be written with 0	RW
WDK2	On-chip Oscillator Prescaler Select Bit [3]	b3 b2 0 0 : Divide-by-8 (approx. 15.6 kHz) 0 1 : Divide-by-4 (approx. 31.3 kHz) 1 0 : Divide-by-2 (approx. 62.5 kHz) 1 1 : No division (approx. 125 kHz)	RW
WDK3			RW
WDK4	Prescaler Count Source Select Bit [3]	0: Peripheral bus clock 1: On-chip oscillator clock [4]	RW
WDK5	Count Stop Function Select Bit [5]	The watchdog timer is 0: Not suspended 1: Suspended when it underflows	RW
— (b7-b6)	No register bits; should be written with 0 and read as 0		—

Notes:

1. This register cannot be rewritten when the WDTON bit in the OFS area is 0 (the watchdog timer starts counting automatically after a reset). When the WDTON bit is 1 (the watchdog timer stops after a reset), this register can be rewritten only once after a reset. When the count source protect mode is enabled, this register should be set at the very beginning of the program.
2. The WDK0 bit becomes 1 when the most recent reset is caused by the watchdog timer. Otherwise, it becomes 0.
3. Set these bits before activating the watchdog timer.
4. The on-chip oscillator starts running by setting the WDK4 bit to 1.
5. This bit setting is enabled when the CM06 bit in the CM0 register is 0 (watchdog timer interrupt).

Figure 12.3 WDK Register

Watchdog Timer Start Register [1]



Symbol	Address	Reset Value
WDTS	4404Eh	Undefined

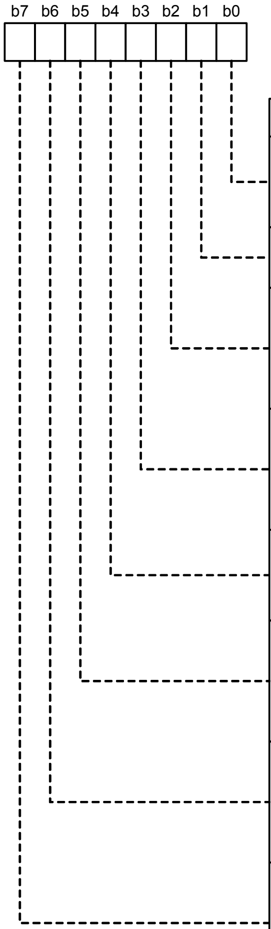
Function	RW
The watchdog timer is initialized by a write access. Then it starts counting downward. Regardless of the value written to, either 7FFFh (when the peripheral bus clock is used) or 7FFh (when the on-chip oscillator clock is used) is set as the default value by writing this register	WO

Note:

1. Set the PRC40 bit in the PRCR4 register to 1 (write enabled) before rewriting this register.

Figure 12.4 WDTS Register

Optional Function Select Area (1)



Symbol
OFS

Address
FFFFFFEFh

Factory Preset Value
FFh

Bit Symbol	Bit Name	Function	RW
WDTON	Watchdog Timer Start Timing Select Bit (2)	After a reset, the watchdog timer 0: Starts counting automatically 1: Stops	(1)
WPSC2	Watchdog Timer Prescaler Select Bit (3)	0: Divide-by-16 (WDC7 = 0) 1: Divide-by-128 (WDC7 = 1)	(1)
WPSC0	Watchdog Timer On-chip Oscillator Prescaler Select Bit (3)	b3 b2 0 0 : Divide-by-8 (WDK3 and WDK2 = 00b) 0 1 : Divide-by-4 (WDK3 and WDK2 = 01b) 1 0 : Divide-by-2 (WDK3 and WDK2 = 10b) 1 1 : No division (WDK3 and WDK2 = 11b)	(1)
WPSC1			(1)
WPCS	Watchdog Timer Prescaler Count Source Select Bit (2, 3)	0: Peripheral bus clock (WDK4 = 0) 1: On-chip oscillator clock (WDK4 = 1)	(1)
WSTP	Watchdog Timer Count Stop Function Select Bit (3)	The watchdog timer is 0: Not stopped (WDK5 = 0) 1: Suspended when it underflows (WDK5 = 1)	(1)
WCSS	Watchdog Timer Count Source Select Bit (2, 3)	0: On-chip oscillator clock (PM23 = 1) 1: Peripheral bus clock or prescaler cascaded on-chip oscillator clock (PM23 = 0)	(1)
CSPM	Post-reset Count Source Protect Mode Select Bit (2, 3)	After a reset, count source protect mode is 0: Enabled (PM22 = 1) 1: Disabled (PM22 = 0)	(1)

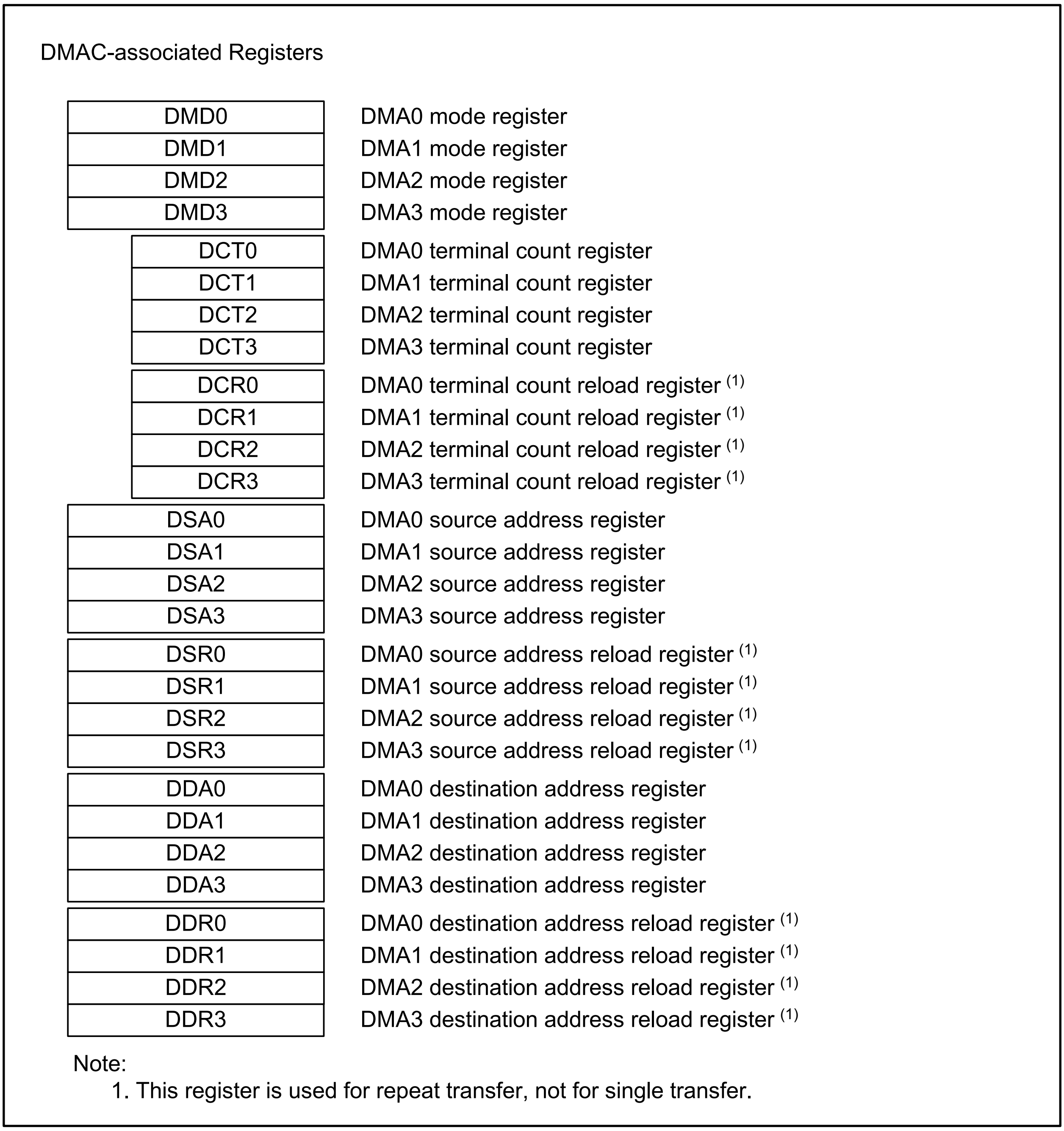
Notes:

1. Rewrite this area by the flash programmer. It cannot be rewritten by a program.
2. To enable the count source protect mode, set bits WDTON and CSPM to 0. Set also either the WPCS bit to 1 or WCSS bit to 0. Once the watchdog timer starts, the PM22 bit in the PM2 register and the WDK register cannot be rewritten.
3. These bit settings are disabled when the WDTON bit is 1. The values set to these bits are reflected to registers WDK and PM2 when the WDTON bit is 0.

Figure 12.5 OFS Area

13. DMAC

Direct memory access (DMA) is a system that can control data transfer without using a CPU instruction.
The R32C/100 Series' four channel DMA controller (DMAC) transmits 8-bit (byte), 16-bit (word), or 32-bit (long word) data in cycle-steal mode from a source address to a destination address each time a transfer request is generated.
The DMAC, which shares a data bus with the CPU, has a higher bus access priority than the CPU. This allows the DMAC to perform fast data transfer when a transfer request is generated.
Figure 13.1 shows a map of the CPU-internal registers associated with DMAC. Table 13.1 lists DMAC specifications. Figures 13.2 to 13.10 show registers associated with DMAC. Since the registers shown in Figure 13.1 are allocated in the CPU, the LDC or STC instruction should be used to write to the registers.



DMAC-associated Registers

DMD0	DMA0 mode register
DMD1	DMA1 mode register
DMD2	DMA2 mode register
DMD3	DMA3 mode register
DCT0	DMA0 terminal count register
DCT1	DMA1 terminal count register
DCT2	DMA2 terminal count register
DCT3	DMA3 terminal count register
DCR0	DMA0 terminal count reload register (1)
DCR1	DMA1 terminal count reload register (1)
DCR2	DMA2 terminal count reload register (1)
DCR3	DMA3 terminal count reload register (1)
DSA0	DMA0 source address register
DSA1	DMA1 source address register
DSA2	DMA2 source address register
DSA3	DMA3 source address register
DSR0	DMA0 source address reload register (1)
DSR1	DMA1 source address reload register (1)
DSR2	DMA2 source address reload register (1)
DSR3	DMA3 source address reload register (1)
DDA0	DMA0 destination address register
DDA1	DMA1 destination address register
DDA2	DMA2 destination address register
DDA3	DMA3 destination address register
DDR0	DMA0 destination address reload register (1)
DDR1	DMA1 destination address reload register (1)
DDR2	DMA2 destination address reload register (1)
DDR3	DMA3 destination address reload register (1)

Note:
1. This register is used for repeat transfer, not for single transfer.

Figure 13.1 CPU-internal Registers for DMAC

Table 13.1 DMAC Specifications (i = 0 to 3)

Item		Specification
Channels		4
Bus request mode		Cycle-steal mode
Transfer memory spaces		From a given address in a 64-Mbyte space (00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given address in the same space
Maximum transfer bytes		64-Mbytes (when 32-bit data is transferred), 32-Mbytes (when 16-bit data is transferred), 16-Mbytes (when 8-bit data is transferred)
DMA request sources [1]		Falling edge or both edges of signals applied to pins INT0 to INT3 or pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ Interrupt requests from timers A0 to A4 Interrupt requests from timers B0 to B5 Transmit/receive interrupt requests from UART0 to UART10 A/D conversion interrupt requests Intelligent I/O interrupt requests Multi-master I^{2}C-bus interrupt requests Software trigger
Channel priority		DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority)
Transfer sizes		8 bits, 16 bits, or 32 bits
Addressing modes		Incrementing addressing or non-incrementing addressing
Transfer modes	Single transfer	Transfer is completed when the DCTi register becomes 00000000h
	Repeat transfer	When the DCTi register becomes 00000000h, the value of the DCRi register is reloaded into the DCTi register to continue the DMA transfer
DMA transfer complete interrupt request generation timing		When the DCTi register changes from 00000001h to 00000000h
DMA transfer start-up	Single transfer	When a DMA transfer request is generated after the DCTi register is set to a value other than 00000000h and bits MDi1 and MDi0 in the DMDi register are set to 01b (single transfer)
	Repeat transfer	When a DMA transfer request is generated after the DCTi register is set to a value other than 00000000h and bits MDi1 and MDi0 are set to 11b (repeat transfer)
DMA transfer stop	Single transfer	When bits MDi1 and MDi0 are set to 00b (DMA transfer disabled)
	Repeat transfer	When bits MDi1 and MDi0 are set to 00b (DMA transfer disabled)
Reload timing to DCTi, DSAi, or DDAi register		When the DCTi register changes from 00000001h to 00000000h in repeat transfer mode
Minimum DMA transfer cycles		3

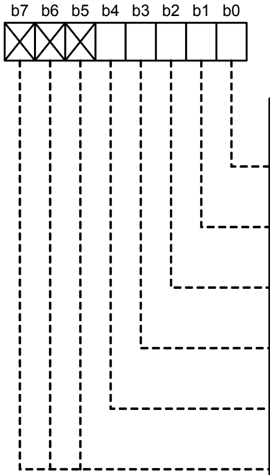
Note:

1. DMA transfer does not affect any interrupts.

The DMA transfer request is available by two different sources: software and hardware. More concretely, they are a write access to the DSR bit in the DMiSL2 register and an interrupt request output from a function specified in bits DSEL4 to DSEL0 in the DMiSL register, and in bits DSEL24 to DSEL20 in the DMiSL2 register (i = 0 to 3). Unlike interrupt requests, the DMA transfer request is not affected by the I flag or the interrupt control register. Therefore this request can be accepted even when interrupts are disabled. Since the DMA transfer does not affect any interrupts, either, the IR bit in the interrupt control register is not changed by the DMA transfer.

DMAi Request Source Select Register (i = 0 to 3)

Symbol	Address	Reset Value
DM0SL to DM3SL	44078h, 44079h, 4407Ah, 4407Bh	XXX0 0000b



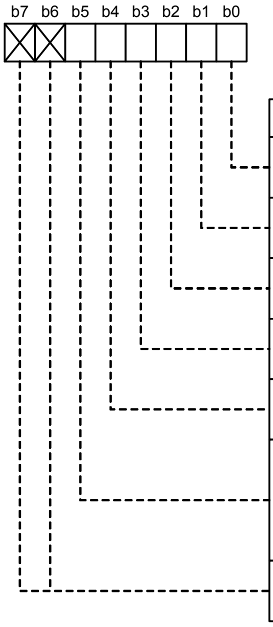
Bit Symbol	Bit Name	Function	RW
DSEL0	DMA Request Source Select Bit [1]	Refer to Table 13.2 "DMiSL Register Functions (i = 0 to 3)"	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
DSEL4			RW
— (b7-b5)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. Change the bit settings of bits DSEL4 to DSEL0 while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are 00b (DMA transfer disabled).

Figure 13.2 Registers DM0SL to DM3SL

DMAi Request Source Select Register 2 (i = 0 to 3)



Symbol	Address	Reset Value
DM0SL2 to DM3SL2	44070h, 44071h, 44072h, 44073h	XX00 0000b

Bit Symbol	Bit Name	Function	RW
DSEL20	DMA Request Source Select Bit [1]	Refer to Table 13.3 “DMiSL2 Register Functions (i = 0 to 3)”	RW
DSEL21			RW
DSEL22			RW
DSEL23			RW
DSEL24			RW
DSR	Software DMA Transfer Request Bit	When a software trigger is selected, a DMA transfer request is generated by setting this bit to 1 (the bit is read as 0)	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. Change the bit settings of bits DSEL24 to DSEL20 while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are 00b (DMA transfer disabled).

Figure 13.3 Registers DM0SL2 to DM3SL2

Table 13.2 DMiSL Register Functions (i = 0 to 3)

Setting Value	DMA Request Source			
b4 b3 b2 b1 b0	DMA0	DMA1	DMA2	DMA3
0 0 0 0 0	Select from DMiSL2 register			
0 0 0 0 1	Falling edge of $\overline{\text{INT0}}$ [1]	Falling edge of $\overline{\text{INT1}}$ [1]	Falling edge of $\overline{\text{INT2}}$ [1]	Falling edge of $\overline{\text{INT3}}$ [1, 2]
0 0 0 1 0	Both edges of $\overline{\text{INT0}}$ [1]	Both edges of $\overline{\text{INT1}}$ [1]	Both edges of $\overline{\text{INT2}}$ [1]	Both edges of $\overline{\text{INT3}}$ [1, 2]
0 0 0 1 1	Timer A0 interrupt request			
0 0 1 0 0	Timer A1 interrupt request			
0 0 1 0 1	Timer A2 interrupt request			
0 0 1 1 0	Timer A3 interrupt request			
0 0 1 1 1	Timer A4 interrupt request			
0 1 0 0 0	Timer B0 interrupt request			
0 1 0 0 1	Timer B1 interrupt request			
0 1 0 1 0	Timer B2 interrupt request			
0 1 0 1 1	Timer B3 interrupt request			
0 1 1 0 0	Timer B4 interrupt request			
0 1 1 0 1	Timer B5 interrupt request			
0 1 1 1 0	UART0 transmit interrupt request			
0 1 1 1 1	UART0 receive interrupt request or ACK interrupt request [3]			
1 0 0 0 0	UART1 transmit interrupt request			
1 0 0 0 1	UART1 receive interrupt request or ACK interrupt request [3]			
1 0 0 1 0	UART2 transmit interrupt request or I^{2}C-bus interface interrupt request [4]			
1 0 0 1 1	UART2 receive interrupt request, ACK interrupt request [3], or I^{2}C-bus line interrupt request [4]			
1 0 1 0 0	UART3 transmit interrupt request		UART5 transmit interrupt request	
1 0 1 0 1	UART3 receive interrupt request or ACK interrupt request [3]		UART5 receive interrupt request or ACK interrupt request [3]	
1 0 1 1 0	UART4 transmit interrupt request		UART6 transmit interrupt request	
1 0 1 1 1	UART4 receive interrupt request or ACK interrupt request [3]		UART6 receive interrupt request or ACK interrupt request [3]	
1 1 0 0 0	A/D0 interrupt request			
1 1 0 0 1	Intelligent I/O interrupt 0 request	Intelligent I/O interrupt 7 request	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request
1 1 0 1 0	Intelligent I/O interrupt 1 request	Intelligent I/O interrupt 8 request	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request
1 1 0 1 1	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request
1 1 1 0 0	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request	Intelligent I/O interrupt 5 request	Intelligent I/O interrupt 0 request
1 1 1 0 1	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request	Intelligent I/O interrupt 6 request	Intelligent I/O interrupt 1 request
1 1 1 1 0	Intelligent I/O interrupt 5 request	Intelligent I/O interrupt 0 request	Intelligent I/O interrupt 7 request	Intelligent I/O interrupt 2 request
1 1 1 1 1	Intelligent I/O interrupt 6 request	Intelligent I/O interrupt 1 request	Intelligent I/O interrupt 8 request	Intelligent I/O interrupt 3 request

Notes:

1. The falling edge and both edges of signals applied to the $\overline{\text{INTi}}$ pin become the DMA request sources (i = 0 to 3). These request sources are not affected by external interrupts (the IFSR0 register and bits POL and LVS in the INTiIC register), and vice versa.
2. When the $\overline{\text{INT3}}$ pin is used as data bus in memory expansion mode or microprocessor mode, it cannot be used as a signal input of the DMA3 request source.
3. Registers UiSMR and UiSMR2 are used to switch between the UARTi receive interrupt and ACK interrupt (i = 0 to 6).
4. Set the I2CEN bit in the I2CMR register to select an interrupt source from either UART2 or I^{2}C-bus.

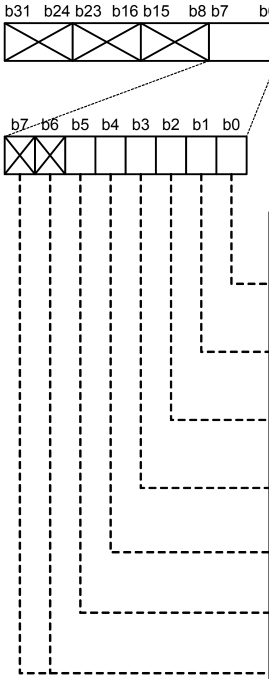
Table 13.3 DMiSL2 Register Functions (i = 0 to 3)

Setting Value	DMA Request Source			
b4 b3 b2 b1 b0	DMA0	DMA1	DMA2	DMA3
0 0 0 0 0	Software trigger			
0 0 0 0 1	Falling edge of $\overline{\text{INT6}}$ (1)	Falling edge of $\overline{\text{INT7}}$ (1)	Falling edge of $\overline{\text{INT8}}$ (1)	Reserved
0 0 0 1 0	Both edges of $\overline{\text{INT6}}$ (1)	Both edges of $\overline{\text{INT7}}$ (1)	Both edges of $\overline{\text{INT8}}$ (1)	Reserved
0 0 0 1 1	Reserved			
0 0 1 0 0	Reserved			
0 0 1 0 1	Reserved			
0 0 1 1 0	Reserved			
0 0 1 1 1	Reserved			
0 1 0 0 0	Reserved			
0 1 0 0 1	Reserved			
0 1 0 1 0	Reserved			
0 1 0 1 1	Reserved			
0 1 1 0 0	Reserved			
0 1 1 0 1	Reserved			
0 1 1 1 0	Reserved			
0 1 1 1 1	Reserved			
1 0 0 0 0	Reserved			
1 0 0 0 1	Reserved			
1 0 0 1 0	Reserved			
1 0 0 1 1	Reserved			
1 0 1 0 0	Reserved			
1 0 1 0 1	Reserved			
1 0 1 1 0	Reserved			
1 0 1 1 1	Reserved			
1 1 0 0 0	UART7 transmit interrupt request			
1 1 0 0 1	UART7 receive interrupt request			
1 1 0 1 0	UART8 transmit interrupt request			
1 1 0 1 1	UART8 receive interrupt request			
1 1 1 0 0	UART9 transmit interrupt request			
1 1 1 0 1	UART9 receive interrupt request			
1 1 1 1 0	UART10 transmit interrupt request			
1 1 1 1 1	UART10 receive interrupt request			

Note:

1. The falling edge and both edges of signals applied to the $\overline{\text{INTi}}$ pin become the DMA request sources (i = 6 to 8). These request sources are not affected by external interrupts (the IFSR1 register and bits POL and LVS in the INTiIC register), and vice versa.

DMAi Mode Register (i = 0 to 3) (1)



Symbol
DMD0 to DMD3

Address
(CPU internal register)

Reset Value
XXXX XXXX XXXX XXXX XXXX XXXX XX00 0000b

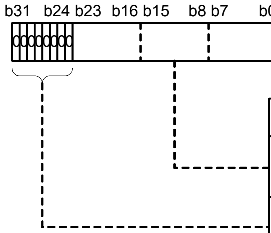
Bit Symbol	Bit Name	Function	RW
MDi0	Transfer Mode Select Bit (2)	b1 b0 0 0 : DMA transfer disabled 0 1 : Single transfer 1 0 : Do not use this combination 1 1 : Repeat transfer	RW
MDi1			RW
BWi0	Transfer Size Select Bit (3)	b3 b2 0 0 : 8 bits 0 1 : 16 bits 1 0 : 32 bits 1 1 : Do not use this combination	RW
BWi1			RW
USAi	Source Addressing Mode Select Bit (3)	0: Non-incrementing addressing 1: Incrementing addressing	RW
UDAi	Destination Addressing Mode Select Bit (3)	0: Non-incrementing addressing 1: Incrementing addressing	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—
— (b31-b8)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Use the LDC instruction to write to this register.
2. Set these bits after all other DMAC-associated registers are set.
3. Set bits MDi1 and MDi0 to 00b before rewriting these bits.

Figure 13.4 Registers DMD0 to DMD3

DMAi Terminal Count Register (i = 0 to 3) (1)



Symbol
DCT0 to DCT3

Address
(CPU internal register)

Reset Value
XXXX XXXXh

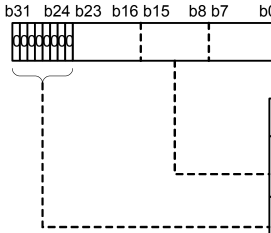
Function	Setting Range	RW
Set the number of transfers to be performed	000000h to FFFFFFh (2)	RW
Reserved	Should be set to 00h	RW

Notes:

1. Use the LDC instruction to write to this register. Set this register while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are 00b (DMA transfer disabled).
2. When these bits are set to 000000h, new DMA transfer requests cannot be accepted.

Figure 13.5 Registers DCT0 to DCT3

DMAi Terminal Count Reload Register (i = 0 to 3) (1)



Symbol	Address	Reset Value
DCR0 to DCR3	(CPU internal register)	XXXX XXXXh

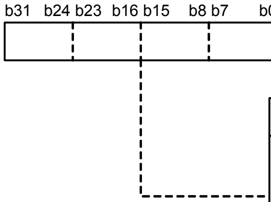
Function	Setting Range	RW
Set the number of transfers to be performed	000000h to FFFFFFh	RW
Reserved	Should be set to 00h	RW

Note:

1. Use the LDC instruction to write to this register. Set this register while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are 00b (DMA transfer disabled).

Figure 13.6 Registers DCR0 to DCR3

DMAi Source Address Register (i = 0 to 3) (1)



Symbol	Address	Reset Value
DSA0 to DSA3	(CPU internal register)	XXXX XXXXh

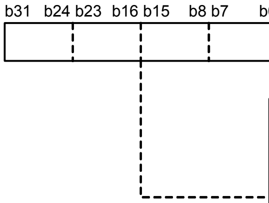
Function	Setting Range	RW
Set a source address	00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space)	RW

Note:

1. Use the LDC instruction to write to this register. Set this register while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are 00b (DMA transfer disabled).

Figure 13.7 Registers DSA0 to DSA3

DMAi Source Address Reload Register (i = 0 to 3) (1)



Symbol
DSR0 to DSR3

Address
(CPU internal register)

Reset Value
XXXX XXXXh

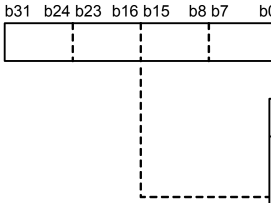
Function	Setting Range	RW
Set a source address	00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space)	RW

Note:

1. Use the LDC instruction to write to this register. Set this register while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are 00b (DMA transfer disabled).

Figure 13.8 Registers DSR0 to DSR3

DMAi Destination Address Register (i = 0 to 3) (1)



Symbol
DDA0 to DDA3

Address
(CPU internal register)

Reset Value
XXXX XXXXh

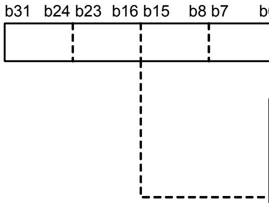
Function	Setting Range	RW
Set a destination address	00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space)	RW

Note:

1. Use the LDC instruction to write to this register. Set this register while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are 00b (DMA transfer disabled).

Figure 13.9 Registers DDA0 to DDA3

DMAi Destination Address Reload Register (i = 0 to 3) (1)



Symbol
DDR0 to DDR3

Address
(CPU internal register)

Reset Value
XXXX XXXXh

Function	Setting Range	RW
Set a destination address	00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space)	RW

Note:

1. Use the LDC instruction to write to this register. Set this register while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are 00b (DMA transfer disabled).

Figure 13.10 Registers DDR0 to DDR3

13.1 Transfer Cycle

The transfer cycle is composed of bus cycles to read data from (source read) or to write data to (destination write) memory or an SFR.

The read and write bus cycles vary with the setting of registers DSAi and DDAi, the width and timing of the data bus connected to the relevant device (i = 0 to 3).

13.1.1 Effect of Transfer Address and Data Bus Width

Table 13.4 lists the incremental bus cycles caused by transfer address alignment or data bus width.

Table 13.4 Incremental Bus Cycles Caused by Transfer Address and Data Bus Width

Transfer Data Unit	Data Bus Width	Transfer Address	Bus Cycles to be Incremented	Bus Cycles Generated
8-bit transfer	8 to 64 bits	n	0	[n]
16-bit transfer	8 bits	n	+1	[n] - [n + 1]
	16 bits	2n	0	[2n]
		2n + 1	+1	[2n + 1] - [2n + 2]
	32 bits	4n	0	[4n]
		4n + 1	0	[4n + 1]
		4n + 2	0	[4n + 2]
		4n + 3	+1	[4n + 3] - [4n + 4]
	64 bits	8n	0	[8n]
		8n + 1	0	[8n + 1]
		8n + 2	0	[8n + 2]
		8n + 3	0	[8n + 3]
		8n + 4	0	[8n + 4]
		8n + 5	0	[8n + 5]
		8n + 6	0	[8n + 6]
		8n + 7	+1	[8n + 7] - [8n + 8]
32-bit transfer	8 bits	n	+3	[n] - [n + 1] - [n + 2] - [n + 3]
	16 bits	4n	+1	[4n] - [4n + 2]
		4n + 1	+2	[4n + 1] - [4n + 2] - [4n + 4]
		4n + 2	+1	[4n + 2] - [4n + 4]
		4n + 3	+2	[4n + 3] - [4n + 4] - [4n + 6]
	32 bits	4n	0	[4n]
		4n + 1	+1	[4n + 1] - [4n + 4]
		4n + 2	+1	[4n + 2] - [4n + 4]
		4n + 3	+1	[4n + 3] - [4n + 4]
	64 bits	8n	0	[8n]
		8n + 1	0	[8n + 1]
		8n + 2	0	[8n + 2]
		8n + 3	0	[8n + 3]
		8n + 4	0	[8n + 4]
		8n + 5	+1	[8n + 5] - [8n + 8]
		8n + 6	+1	[8n + 6] - [8n + 8]
		8n + 7	+1	[8n + 7] - [8n + 8]

13.1.2 Effect of Bus Timing

In the R32C/100 Series, a separate bus is connected to each device. The bus width and bus timing vary with each device. Table 13.5 lists the bus width and access cycles for each device.

Table 13.5 Bus Width and Bus Cycles

Device	Addresses (1)	Bus Width	Access Cycles (2)	Reference Clock
Flash memory	FFE00000h to FFFFFFFFh	64-bit	2 or 3 (3)	CPU clock
Data flash	00060000h to 00061FFFh	64-bit	5	CPU clock
RAM	00000400h to 0003FFFFh	64-bit	1 or 2 (4)	CPU clock
SFR space	00000000h to 0000001Fh	16-bit	3 (5)	Peripheral bus clock
	00000020h to 000003FFh	16-bit	2 (5)	Peripheral bus clock
SFR2 space	00040000h to 00041FFFh	16-bit	2 (5)	Peripheral bus clock
	00042000h to 00043FFFh	32-bit	2 (5)	Peripheral bus clock
	00044000h to 000440DFh	16-bit	2 (5, 6)	Peripheral bus clock
	000440E0h to 000443FFh	16-bit	3 (5, 6)	Peripheral bus clock
	00044400h to 00045FFFh	16-bit	2 (5, 6)	Peripheral bus clock
	00046000h to 000467FFh	32-bit	3 (5, 6)	Peripheral bus clock
	00046800h to 00047FFFh	32-bit	2 (5, 6)	Peripheral bus clock
	00048000h to 0004FFFFh	64-bit	2	CPU clock
External bus	00080000h to 01FFFFFFh FE000000h to FFDFFFFFh	8-/16-/32-bit	Specified by the EBCn register (n = 0 to 3) (5)	Peripheral bus clock

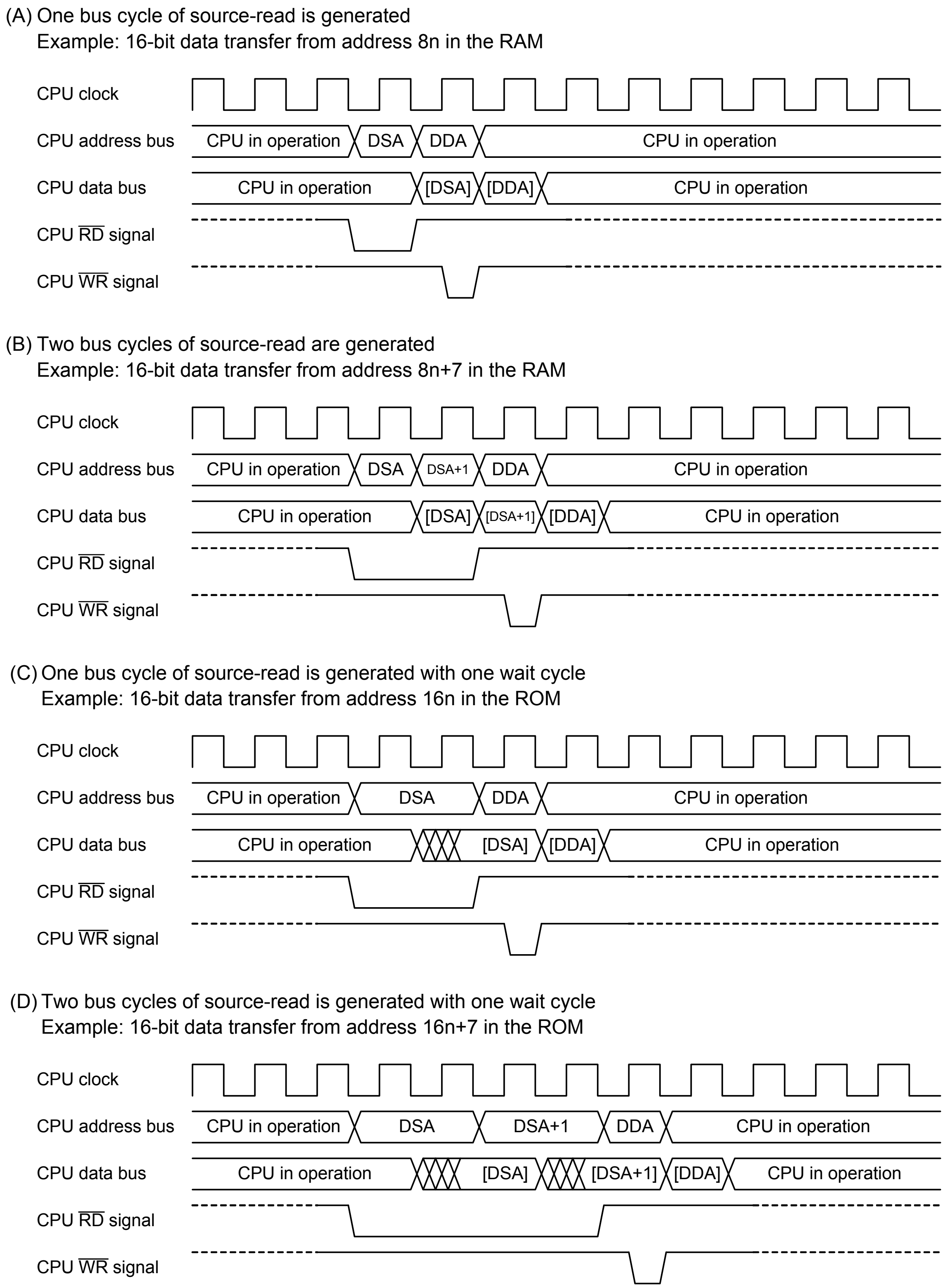
Notes:

1. Reserved spaces are included.
2. Access cycles are based on each bus clock.
3. An access to the same page as the previous time requires two cycles. Otherwise, three cycles are required.
4. If write cycles are generated sequentially, each write cycle except the initial one has two access cycles. A read cycle just after a write cycle has also two access cycles.
5. If SFRs are sequentially accessed, each access except the initial one has one additional base clock cycle.
6. Up to one access cycle may be added depending on the phase of peripheral bus clock.

Figure 13.11 shows an example of source-read bus cycles in a transfer cycle. In this figure, the number of source-read bus cycles is shown under different conditions, provided that the destination address is in an internal RAM with one bus cycle of destination-write. In a real operation, the transfer cycles change according to conditions for destination-write bus cycles as well as for source-read bus cycles. To calculate a transfer cycle, respective conditions should be applied to both destination-write bus cycle and source-read bus cycle. In (B) of Figure 13.11, for example, if two bus cycles are generated, bus cycles required for the destination-write is two as well as for the source-read.

13.1.3 Effect of $\overline{\text{RDY}}$ Signal

In memory expansion mode or microprocessor mode, the $\overline{\text{RDY}}$ signal affects a bus cycle in an external space. Refer to 9.3.7 “$\overline{\text{RDY}}$ Signal” for details.



(A) One bus cycle of source-read is generated
Example: 16-bit data transfer from address 8n in the RAM

(B) Two bus cycles of source-read are generated
Example: 16-bit data transfer from address 8n+7 in the RAM

(C) One bus cycle of source-read is generated with one wait cycle
Example: 16-bit data transfer from address 16n in the ROM

(D) Two bus cycles of source-read is generated with one wait cycle
Example: 16-bit data transfer from address 16n+7 in the ROM

Note:

1. The above assumes that destination-write completes in one bus cycle. In actual use, the number of destination-write bus cycles should be considered according to the conditions such as above.

Figure 13.11 Source-read Bus Cycles in a Transfer Cycle

13.2 DMA Transfer Cycle

The DMA transfer cycles are calculated as follows:

$$\textit{Number of transfer cycles} = \textit{Source-read bus cycles} \times j + \textit{Destination-write bus cycles} \times k + 1$$

where:
j = access cycles for read
k = access cycles for write (refer to Table 13.5)

Each bus cycle, source-read and destination-write, requires at least one cycle. In addition, more cycles may be required depending on the transfer address. Refer to Table 13.4 for details on the required bus cycles.
"+1" in the formula above means a cycle required to decrement the value of DCTi register (i = 0 to 3).

The following are calculation examples:
To transfer 32-bit data from address 400h in the RAM to address 800h in the RAM,

$$\begin{aligned}\textit{Number of the transfer cycles} &= 1 \times 1 + 1 \times 1 + 1\\ &= 3\end{aligned}$$

Thus, there are three cycles.

To transfer 16-bit data from the AD00 register at address 380h to registers P1 and P0 at addresses 3C1h and 3C0h, respectively, when the peripheral bus clock frequency is half the CPU clock,

$$\begin{aligned}\textit{Number of the transfer cycles} &= 1 \times 2 \times 2 + 1 \times 2 \times 2 + 1\\ &= 9\end{aligned}$$

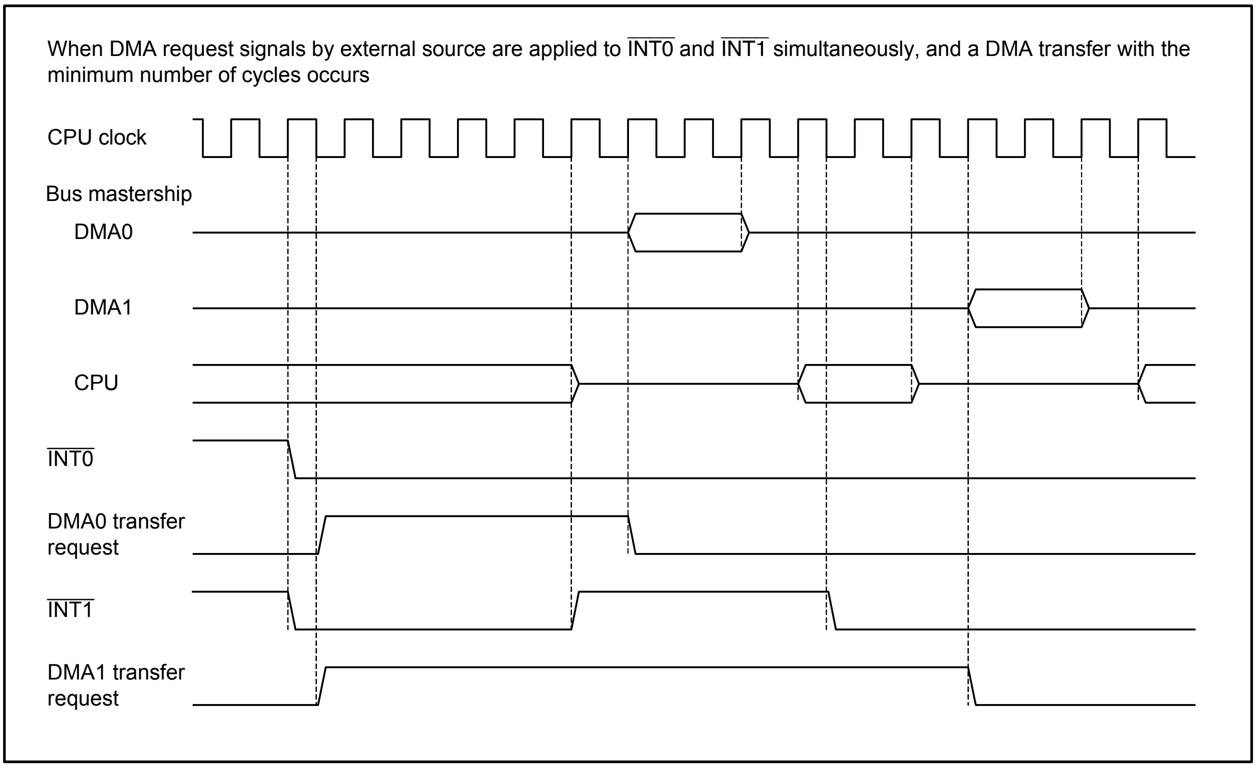
Thus, there are nine cycles.

13.3 Channel Priority and DMA Transfer Timing

When multiple DMA transfer requests are generated in the same sampling period, between the falling edge of the CPU clock and the next falling edge, these requests are simultaneously input into the DMAC. Channel priority in this case is: DMA0 > DMA1 > DMA2 > DMA3.

Figure 13.12 shows an example of the DMA transfer by external source, specifically when DMA0 and DMA1 requests are simultaneously generated. The DMA0, whose request priority is higher than that of DMA1, is received first to start the transfer and then hands over the bus to the CPU after completing one DMA0 transfer. Once the CPU completes one bus access, the DMA1 transfer starts. The CPU takes the bus back from the DMA1 after one DMA1 transfer is completed.

DMA transfer requests cannot be counted. Only a single transfer is performed even when an $\overline{\text{INTi}}$ interrupt occurs more than once before the bus is granted, as shown by DMA1 in Figure 13.12.



When DMA request signals by external source are applied to $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ simultaneously, and a DMA transfer with the minimum number of cycles occurs

Figure 13.12 DMA Transfer by External Source

13.4 Notes on DMAC

13.4.1 DMAC-associated Register Settings

- Set DMAC-associated registers while bits MDi1 and MDi0 in the DMDi register are 00b (DMA transfer disabled) (i = 0 to 3). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure. This procedure also applies when rewriting bits UDAi, USAi, and BWi1 and BWi0 in the DMDi register.

- When rewriting the DMAC-associated registers while DMA transfer is enabled, stop the peripherals that can be DMA triggers so that no DMA transfer request is generated, then set bits MDi1 and MDi0 in the DMDi register of the corresponding channel to 00b (DMA transfer disabled).

- Once a DMA transfer request is accepted, DMA transfer cannot be disabled even if setting bits MDi1 and MDi0 in the DMDi register to 00b (DMA transfer disabled). Do not change the settings of any DMAC-associated registers other than bits MDi1 and MDi0 until the DMA transfer is completed.

- After setting registers DMiSL and DMiSL2, wait at least six peripheral bus clocks to set bits MDi1 and MDi0 in the DMDi register to 01b (single transfer) or 11b (repeat transfer).

13.4.2 Reading DMAC-associated Registers

- Use the following read order to sequentially read registers DMiSL and DMiSL2:
  DM0SL, DM1SL, DM2SL, and DM3SL
  DM0SL2, DM1SL2, DM2SL2, and DM3SL2

14. DMAC II

DMAC II starts by an interrupt request from any peripheral and performs data transfer without a CPU instruction. Transfer sources are selectable from memory, immediate data, memory + memory, and immediate data + memory.

Table 14.1 lists specifications of DMAC II.

Table 14.1 DMAC II Specifications

Item	Specification
DMAC II request sources	Interrupt requests from the peripherals of which bits ILVL2 to ILVL0 in the corresponding interrupt control register are set to 111b (level 7)
Transfer types	• Data in memory is transferred to memory (memory-to-memory transfer) • Immediate data is transferred to memory (immediate data transfer) • Data in memory + data in memory are transferred to memory (calculation result transfer) • Immediate data + data in memory are transferred to memory (calculation result transfer)
Transfer sizes	8 bits or 16 bits
Transfer memory spaces	From a given address in a 64-Mbyte space (00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given address in the same space (1)
Addressing modes	Individually selectable for each source address and destination address from the following two modes: • Non-incrementing addressing: Address is held constant throughout a data transfer/DMA II transaction • Incrementing addressing: Address increments by 1 (when 8-bit data is transferred) or 2 (when 16-bit data is transferred) after each data transfer
Transfer modes	• Single transfer: Only one data transfer is performed by one transfer request • Burst transfer: Data transfers are continuously performed for the number of times set in the transfer counter by one transfer request • Multiple transfer: Multiple memory-to-memory transfers are performed from different source addresses to different destination addresses by one transfer request
Chain transfer	Data transfer is sequentially performed by switching among multiple DMAC II indexes (transfer information)
DMA II transfer complete interrupt request	An interrupt request is generated when the transfer counter reaches 0000h

Note:

1. When the transfer size is 16 bits and the destination address is FFFFFFFFh, data is transferred to FFFFFFFFh and 00000000h. This also applies when the source address is FFFFFFFFh.

14.1 DMAC II Settings

To use DMAC II, set the following:

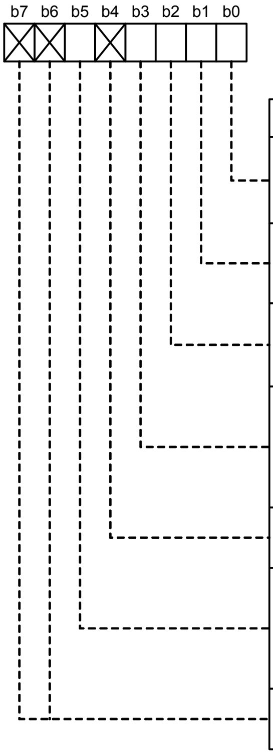
- Registers RIPL1 and RIPL2
- DMAC II index
- Interrupt control registers of the peripherals that trigger DMAC II
- Relocatable vectors of the peripherals that trigger DMAC II
- The IRLT bit in the IIOiIE register when using the intelligent I/O interrupt (i = 0 to 11). Refer to 11. “Interrupts” for details on the IIOiIE register.

14.1.1 Registers RIPL1 and RIPL2

When the DMAII bit in registers RIPL1 and RIPL2 is set to 1 (DMA II transfer selected) and the FSIT bit is set to 0 (normal interrupt selected), DMAC II starts by an interrupt request from any peripheral whose bits ILVL2 to ILVL0 in the corresponding interrupt control register are set to 111b (level 7).

Figure 14.1 shows registers RIPL1 and RIPL2.

Wake-up IPL Setting Register i (i = 1, 2) (1)



Symbol	Address	Reset Value
RIPL1, RIPL2	4407Fh, 4407Dh	XX0X 0000b

Bit Symbol	Bit Name	Function	RW
RLVL0	Interrupt Priority Level for Wake-up Select Bit (2)	b2 b1 b0 0 0 0 : Level 0 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
RLVL1			RW
RLVL2			RW
FSIT	Fast Interrupt Select Bit (3)	0: Use interrupt request level 7 for normal interrupt 1: Use interrupt request level 7 for fast interrupt (4)	RW
— (b4)	No register bit; should be written with 0 and read as undefined value		—
DMAII	DMA II Select Bit (5)	0: Use interrupt request level 7 for interrupt 1: Use interrupt request level 7 for DMA II transfer (4)	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Registers RIPL1 and RIPL2 should be set with the same values.
2. The MCU exits wait mode or stop mode when the request level of the requested interrupt is higher than the level selected using bits RLVL2 to RLVL0. Set these bits to the same value as the IPL in the FLG register.
3. When the FSIT bit is 1, an interrupt with interrupt request level 7 becomes the fast interrupt. In this case, set the interrupt request level to level 7 with only one interrupt.
4. Set either the FSIT or DMAII bit to 1. The fast interrupt and DMAC II cannot be used simultaneously.
5. Set bits ILVL2 to ILVL0 in the interrupt control register after the DMAII bit is set. DMA II transfer is not affected by the I flag or IPL.

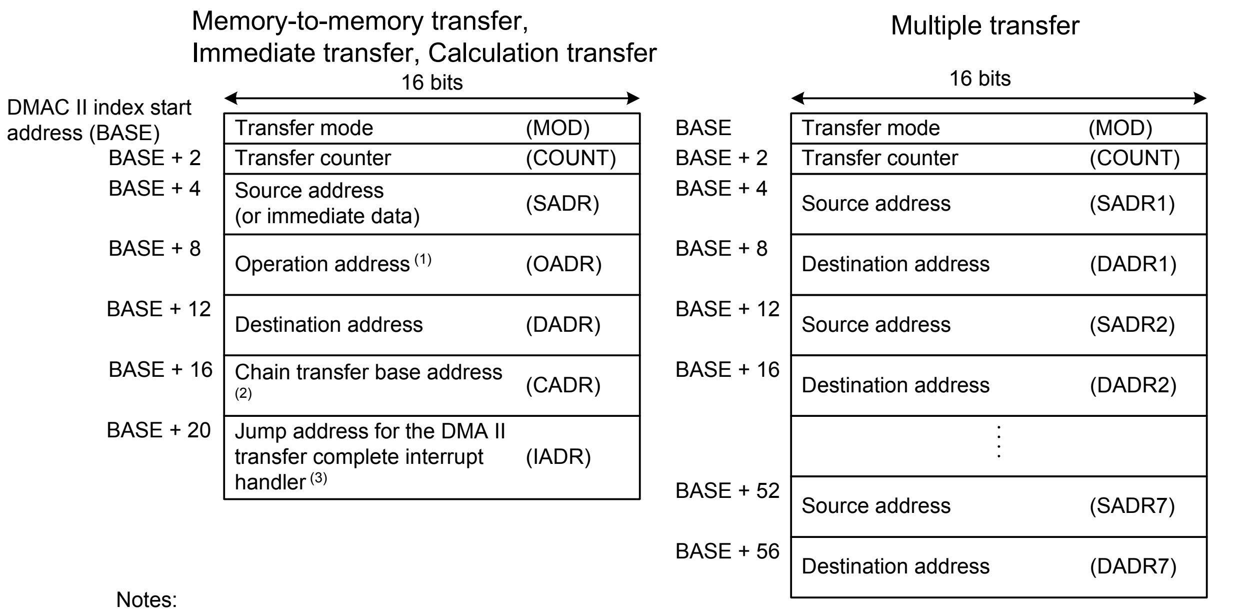
Figure 14.1 Registers RIPL1 and RIPL2

14.1.2 DMAC II Index

The DMAC II index is a data table of 12 to 60 bytes. It stores parameters for transfer mode, transfer counter, source address (or immediate data), operation address as an address to be calculated, destination address, chain transfer base address, and jump address for the DMA II transfer complete interrupt handler.

This DMAC II index should be allocated on the RAM.

Figure 14.2 shows a configuration of the DMAC II index and Table 14.2 lists a configuration example of the DMAC II index.



Memory-to-memory transfer, Immediate transfer, Calculation transfer

16 bits

DMAC II index start address (BASE)	Transfer mode	(MOD)
BASE + 2	Transfer counter	(COUNT)
BASE + 4	Source address (or immediate data)	(SADR)
BASE + 8	Operation address [1]	(OADR)
BASE + 12	Destination address	(DADR)
BASE + 16	Chain transfer base address [2]	(CADR)
BASE + 20	Jump address for the DMA II transfer complete interrupt handler [3]	(IADR)

Multiple transfer

16 bits

BASE	Transfer mode	(MOD)
BASE + 2	Transfer counter	(COUNT)
BASE + 4	Source address	(SADR1)
BASE + 8	Destination address	(DADR1)
BASE + 12	Source address	(SADR2)
BASE + 16	Destination address	(DADR2)
	⋮	
BASE + 52	Source address	(SADR7)
BASE + 56	Destination address	(DADR7)

Notes:

1. This data is required only for the calculation transfer.
2. This data is required only for the chain transfer.
3. This data is required only for the DMA II transfer complete interrupt.

The DMAC II index should be allocated on the RAM. Required data should be set front-aligned. For example, when the calculation transfer is not used, the destination address should be set to BASE + 8 (refer to the “DMAC II Index Configuration” on the next page).
Start address of the DMAC II index should be set in the interrupt vector space for the peripheral interrupt triggering DMAC II.

Figure 14.2 DMAC II Index

The following are the details on the DMAC II index. These parameters should be aligned in the order listed in Table 14.2 according to the transfer mode to be used.

- Transfer mode (MOD)
  Set a transfer mode in 2 bytes. Refer to Figure 14.3 for details on the setting of MOD.
- Transfer counter (COUNT)
  Set a number of transfers in 2 bytes.
- Source address (SADR)
  Set a source address or immediate data in 4 bytes. Note that the two upper bytes of immediate data are ignored.
- Operation address (OADR)
  Set an address in a to-be calculated memory in 4 bytes. This data setting is required only for the calculation transfer.
- Destination address (DADR)
  Set a destination address in 4 bytes.
- Chain transfer base address (CADR)
  Set the start address of the DMAC II index for the next transfer (BASE) in 4 bytes. This data setting is required only for the chain transfer.
- Jump address for the DMA II transfer complete interrupt handler (IADR)
  Set the start address for the DMA II transfer complete interrupt handler in 4 bytes. This data setting is required only for the DMA II transfer complete interrupt.

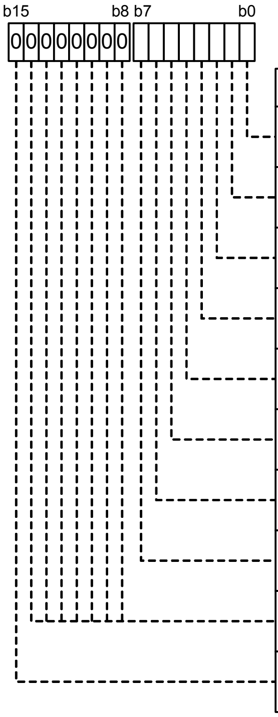
The symbols above are hereinafter used in place of their respective parameters.

Table 14.2 DMAC II Index Configuration

Transfer Data	Memory-to-memory Transfer/ Immediate Data Transfer				Calculation Transfer				Multiple Transfer
Chain transfer	Not used	Used	Not used	Used	Not used	Used	Not used	Used	Not available
DMA II transfer complete interrupt	Not used	Not used	Used	Used	Not used	Not used	Used	Used	Not available
DMAC II index	MOD COUNT SADR DADR 12 bytes	MOD COUNT SADR DADR CADR 16 bytes	MOD COUNT SADR DADR IADR 16 bytes	MOD COUNT SADR DADR CADR IADR 20 bytes	MOD COUNT SADR OADR DADR 16 bytes	MOD COUNT SADR OADR DADR CADR 20 bytes	MOD COUNT SADR OADR DADR IADR 20 bytes	MOD COUNT SADR OADR DADR CADR IADR 24 bytes	MOD COUNT SADR1 DADR1 SADRi DADRi i = 1 to 7 max. 60 bytes (when i = 7)

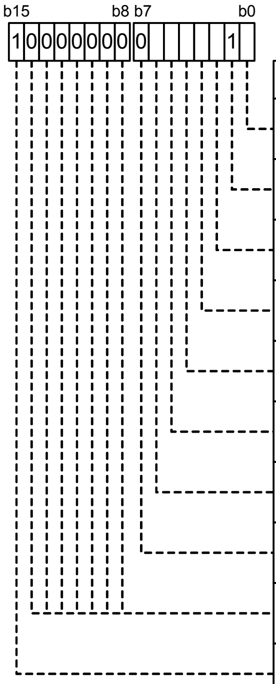
Transfer Mode (MOD) (1)

When multiple transfer is not selected (MULT = 0)



Bit Symbol	Bit Name	Function	RW
SIZE	Transfer Size Select Bit	0: 8 bits 1: 16 bits	RW
IMM	Transfer Source Select Bit	0: Immediate data 1: Memory	RW
UPDS	Source Addressing Select Bit	0: Non-incrementing addressing 1: Incrementing addressing	RW
UPDD	Destination Addressing Select Bit	0: Non-incrementing addressing 1: Incrementing addressing	RW
OPER	Calculation Result Transfer Select Bit	0: Not used 1: Used	RW
BRST	Burst Transfer Select Bit	0: Single transfer 1: Burst transfer	RW
INTE	DMA II Transfer Complete Interrupt Select Bit	0: Not used 1: Used	RW
CHAIN	Chain Transfer Select Bit	0: Not used 1: Used	RW
— (b14-b8)	Reserved	Should be written with 0	RW
MULT	Multiple Transfer Select Bit	0: Not used	RW

When multiple transfer is selected (MULT = 1)



Bit Symbol	Bit Name	Function	RW
SIZE	Transfer Size Select Bit	0: 8 bits 1: 16 bits	RW
IMM	Reserved	Should be written with 1	RW
UPDS	Source Addressing Select Bit	0: Non-incrementing addressing 1: Incrementing addressing	RW
UPDD	Destination Addressing Select Bit	0: Non-incrementing addressing 1: Incrementing addressing	RW
CNT0	Number of Transfers Setting Bit	b6 b5 b4 0 0 0 : Do not use this combination 0 0 1 : Once 0 1 0 : Twice : : 1 1 1 : Seven times	RW
CNT1			RW
CNT2			RW
CHAIN	Reserved	Should be written with 0	RW
— (b14-b8)	Reserved	Should be written with 0	RW
MULT	Multiple Transfer Select Bit	1: Used	RW

Note:
1. The MOD should be allocated on the RAM.

Figure 14.3 MOD

14.1.3 Interrupt Control Register of the Peripherals

Set bits ILVL2 to ILVL0 in the interrupt control register for the peripheral interrupt triggering DMAC II to 111b (level 7).

14.1.4 Relocatable Vector Table of the Peripherals

Set the start address of the DMAC II index in the interrupt vector space for the peripheral interrupt triggering DMAC II.
To use the chain transfer, allocate the relocatable vector table on the RAM.

14.1.5 IRLT Bit in the IIOiIE Register (i = 0 to 11)

To use the intelligent I/O interrupt as a trigger for DMAC II, set the IRLT bit in the corresponding IIOiIE register to 0 (interrupt request for DMA or DMA II used).

14.2 DMAC II Operation

Set the DMAII bit in registers RIPL1 and RIPL2 to 1 (interrupt request level 7 used for DMA II transfer) to perform a DMA II transfer. DMAC II starts by an interrupt request from any peripheral whose bits ILVL2 to ILVL0 in the corresponding interrupt control register are set to 111b (level 7). These peripheral interrupt requests are available only for DMA II transfer and cannot be used for the CPU.
When an interrupt request is generated with interrupt request level 7, DMAC II starts irrespective of the state of the I flag or IPL.
When a peripheral interrupt request triggering DMAC II and a higher-priority request such as the watchdog timer interrupt, low voltage detection interrupt, oscillator stop detection interrupt, or NMI are simultaneously generated, the higher-priority interrupt is accepted prior to the DMA II transfer, and the DMA II transfer starts after the higher-priority interrupt sequence.

14.3 Transfer Types

DMAC II transfers three types of 8-bit or 16-bit data as follows:

- Memory-to-memory transfer: Data is transferred from a given memory location in a 64-Mbyte space (addresses 00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given memory location in the same space.
- Immediate data transfer: Immediate data is transferred to a given memory location in a 64-Mbyte space.
- Calculation transfer: Two data are added together and the result is transferred to a given memory location in a 64-Kbyte space.

When 16-bit data is transferred to DADR at FFFFFFFFh, it is transferred to 00000000h as well as FFFFFFFFh. The same transfer is performed when SADR is FFFFFFFFh.

14.3.1 Memory-to-memory Transfer

Data transfer between any two memory locations can be:

- A transfer from a fixed address to another fixed address
- A transfer from a fixed address to an address range in memory
- A transfer from an address range in memory to a fixed address
- A transfer from an address range in memory to another address range in memory

When increment addressing mode is selected, SADR and DADR increment by 1 in an 8-bit transfer and by 2 in a 16-bit transfer after a data transfer for the next transfer. When SADR or DADR exceeds FFFFFFFFh by the incrementation, it returns to 00000000h. Likewise, when SADR or DADR exceeds 01FFFFFFh, it becomes 02000000h, but an actual transfer is performed for FE000000h.

14.3.2 Immediate Data Transfer

DMAC II transfers immediate data to a given memory location. Either incrementing or non-incrementing addressing mode can be selected for the destination address. Store the immediate data to be transferred into SADR. To transfer 8-bit immediate data, set the data to the lower 1 byte of SADR. The upper 3 bytes are ignored. To transfer 16-bit immediate data, set the data to the lower 2 bytes. The upper 2 bytes are ignored.

14.3.3 Calculation Result Transfer

After two memory data or immediate data and memory data are added together, DMAC II transfers the calculated result to a given memory location. Set an address to be calculated or immediate data to SADR and set the other address to be calculated to OADR. Either incrementing or non-incrementing addressing mode can be selected for source and destination addresses when performing data in memory + data in memory calculation transfer. If the source addressing is in incrementing mode, the operation addressing is also in incrementing mode. When performing immediate data + data in memory calculation transfer, the addressing mode is selectable only for the destination address.

14.4 Transfer Modes

DMAC II provides three types of basic transfer mode: single transfer, burst transfer, and multiple transfer. COUNT determines the number of transfers to be performed. Transfers are not performed when COUNT is 0000h.

14.4.1 Single Transfer

Set the BRST bit in the MOD to 0.

A single data transfer is performed by one transfer request.

When incrementing addressing mode is selected for the source and/or destination address, the address or addresses increment after a data transfer for the next transfer.

COUNT is decremented each time a data transfer is performed. When COUNT reaches 0000h, the DMA II transfer complete interrupt request is generated if the INTE bit in the MOD is 1 (DMA II transfer complete interrupt used).

14.4.2 Burst Transfer

Set the BRST bit in the MOD to 1.

DMAC II continuously transfers data for the number of times determined by COUNT with one transfer request. COUNT decrements each time a data transfer is performed. When COUNT reaches 0000h, the burst transfer is completed. The DMA II transfer complete interrupt request is generated if the INTE bit is 1 (DMA II transfer complete interrupt used).

No interrupts are accepted during a burst transfer.

14.4.3 Multiple Transfer

Set the MULT bit in the MOD to 1.

Multiple memory-to-memory transfers are performed from different source addresses to different destination addresses using one transfer request.

Set bits CNT2 to CNT0 in the MOD to select the number of transfers to be performed from 001b (once) to 111b (seven times). Do not set these bits to 000b.

Allocate the required number of SDARs and DADRs alternately following MOD and COUNT.

When the multiple transfer is selected, the following transfer functions are not available: calculation result transfer, burst transfer, chain transfer, and DMA II transfer complete interrupt.

14.5 Chain Transfer

The chain transfer is available when the CHAIN bit in the MOD is 1.

The chain transfer is performed as follows:

(1) When a transfer request is generated, a data transfer is performed according to the DMAC II index specified by the corresponding interrupt vector. Either a single transfer (the BRST bit in the MOD is 0) or burst transfer (the BRST bit is 1) is performed according to the BRST bit setting.

(2) When COUNT reaches 0000h, the value in the interrupt vector in (1) above is overwritten with the value in CADR. Simultaneously, the DMA II transfer complete interrupt request is generated when the INTE bit in the MOD is 1.

(3) When the next DMA II transfer request is generated, the data transfer is performed according to the DMAC II index specified by the peripheral interrupt vector in (2) above.

Figure 14.4 shows the relocatable vector and DMAC II index in a chain transfer.

To use the chain transfer, the relocatable vector table should be allocated on the RAM.

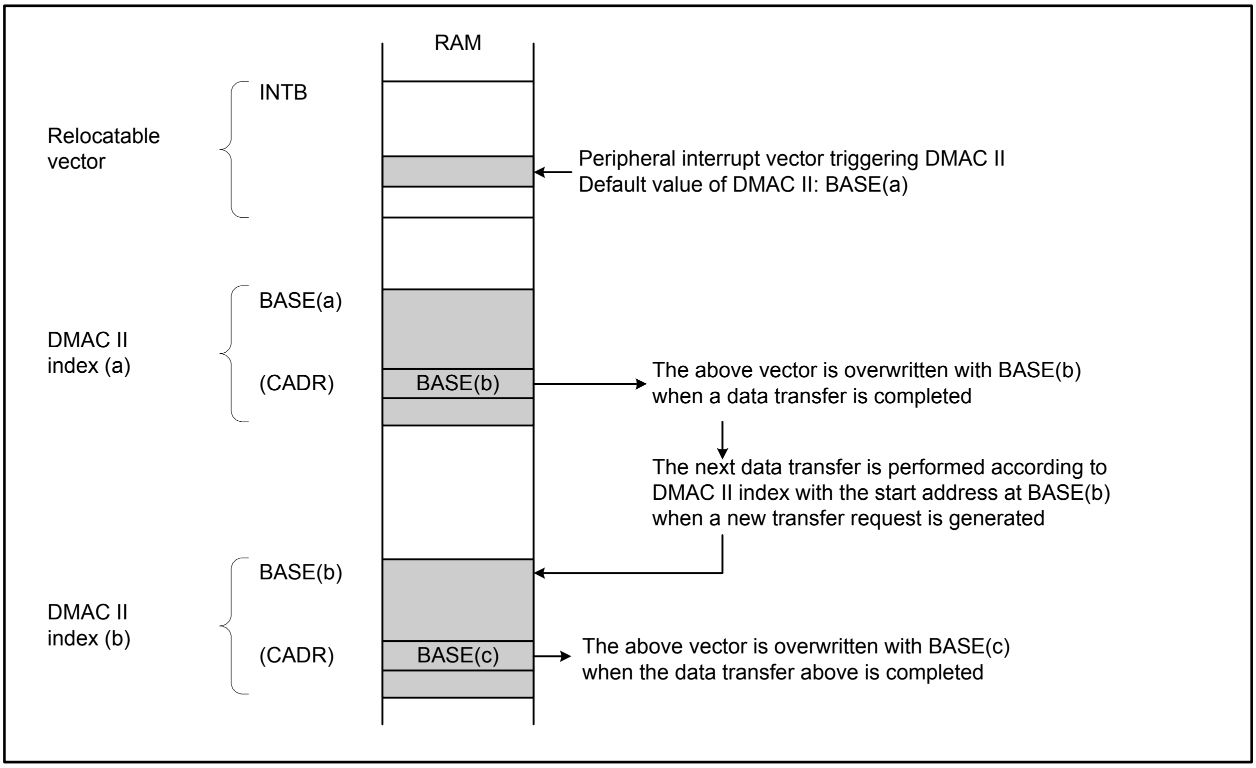

RAM
Relocatable vector
INTB
Peripheral interrupt vector triggering DMAC II
Default value of DMAC II: BASE(a)
DMAC II index (a)
BASE(a)
(CADR)
BASE(b)
The above vector is overwritten with BASE(b) when a data transfer is completed
The next data transfer is performed according to DMAC II index with the start address at BASE(b) when a new transfer request is generated
DMAC II index (b)
BASE(b)
(CADR)
BASE(c)
The above vector is overwritten with BASE(c) when the data transfer above is completed

Figure 14.4 Relocatable Vector and DMAC II Index in a Chain Transfer

14.6 DMA II Transfer Complete Interrupt

The DMA II transfer complete interrupt is available when the INTE bit in the MOD is 1.

Set the start address of the DMA II transfer complete interrupt handler to IADR. The interrupt request is generated when COUNT reaches 0000h.

The initial instruction of the interrupt handler is executed in the eighth cycle after a DMA II transfer is completed.

14.7 Execution Time

The DMAC II execution cycle is calculated by the following equations:

Mode other than multiple transfer: $t = 6 + (26 + a + b + c + d) \times m + (4 + e) \times n$ cycles
When using multiple transfer: $t = 21 + (11 + b + c) \times k$ cycles

a: When IMM is 0 (transfer source is immediate data), a is 0;
When IMM is 1 (transfer source is memory), a is -1
b: When UPDS is 1 (source addressing is incrementing), b is 0;
When UPDS is 0 (source addressing is non-incrementing), b is 1
c: When UPDD is 1 (destination addressing is incrementing), c is 0;
When UPDD is 0 (destination addressing is non-incrementing), c is 1
d: When OPER is 0 (calculation transfer is not selected), d is 0;
When OPER is 1 (calculation transfer is selected) and UPDS is 0 (source addressing is immediate data or non-incrementing), d is 7;
When OPER is 1 (calculation transfer is selected) and UPDS is 1 (source addressing is incrementing), d is 8
e: When CHAIN is 0 (chain transfer is not selected), e is 0;
When CHAIN is 1 (chain transfer is selected), e is 4
m: When BRST is 0 (single transfer), m is 1;
When BRST is 1 (burst transfer), m is COUNT
n: When COUNT is 0001h, n is 0; if COUNT is 0002h or more, n is 1
k: The number of transfers set using bits CNT2 to CNT0

The equations above are estimations. The number of cycles may vary depending on CPU state, bus wait state, and DMAC II index allocation.

The figure below applies under the following conditions:
memory-to-memory transfer; incrementing source address; non-incrementing destination address; number of transfers = 2; single transfer mode; using a transfer complete interrupt (transfer counter = 2); no chain transfer

($a = -1, b = 0, c = 1, d = 0, e = 0, m = 1$)

First DMA II transfer $t = 6 + (26 - 1 + 0 + 1 + 0) \times 1 + (4 + 0) \times 1 = 36$ cycles
Second DMA II transfer $t = 6 + (26 - 1 + 0 + 1 + 0) \times 1 + (4 + 0) \times 0 = 32$ cycles

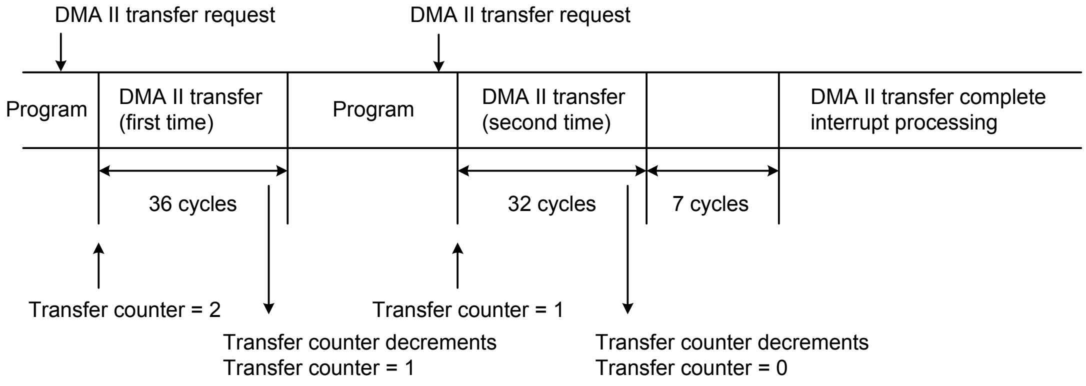
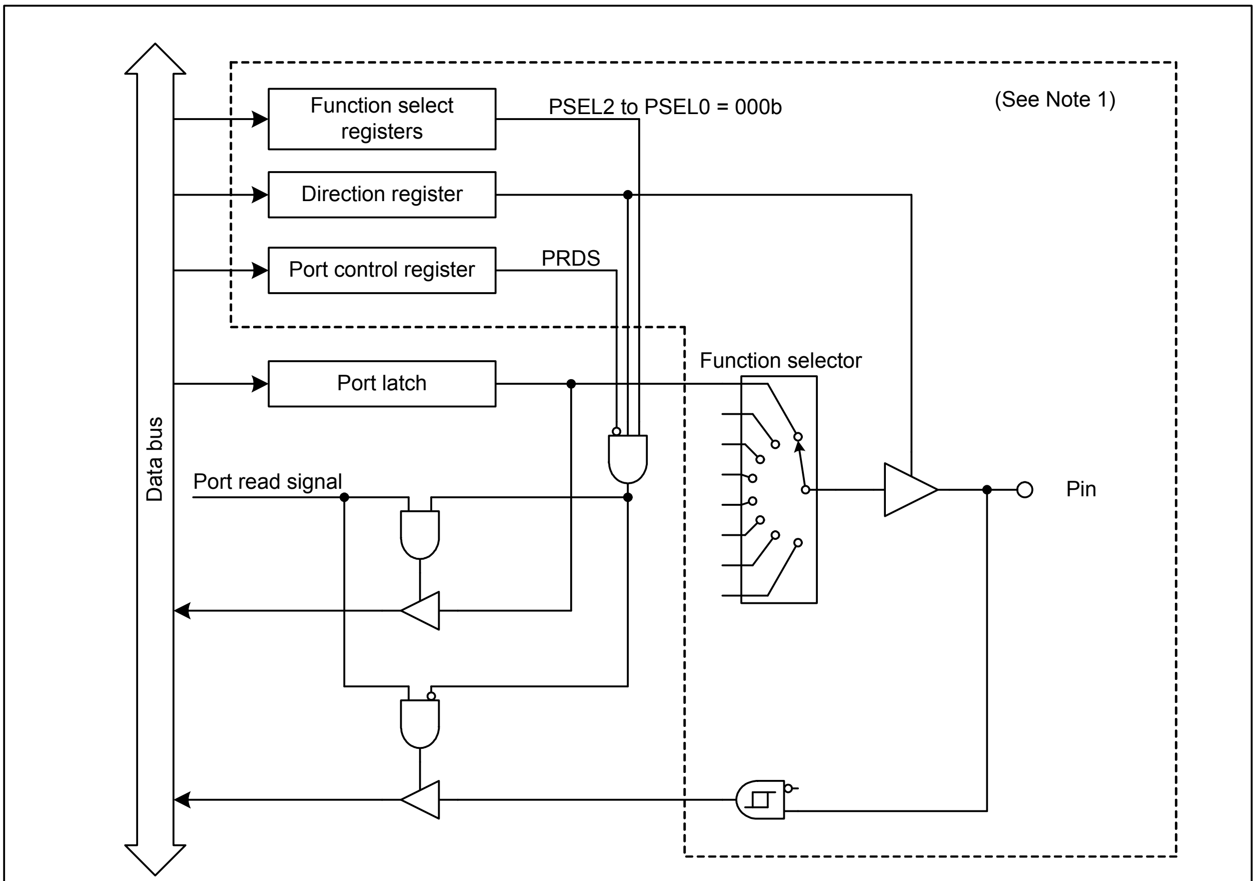


Figure 14.5 Transfer Cycles

15. Programmable I/O Ports

The programmable I/O ports in each pin package are designated as follows:
144-pin package: 124 ports from P0 to P15 (excluding P8_5, and P14_0 to P14_2)
176-pin package: 156 ports from P0 to P19 (excluding P8_5, and P14_0 to P14_2)
Each port status, input or output, can be selected using the direction register except P8_5 and P14_1 which are input only. The P8_5 bit in the P8 register indicates an $\overline{\text{NMI}}$ input level since the P8_5 shares a pin with the $\overline{\text{NMI}}$.
Figure 15.1 shows a configuration of programmable I/O ports, and Figures 15.2 and Figure 15.3 show a configuration of each input-only port.



Note:
1. Refer to 26. “I/O Pins” for details on the area enclosed with the dotted line above.

Figure 15.1 Programmable I/O Port Configuration

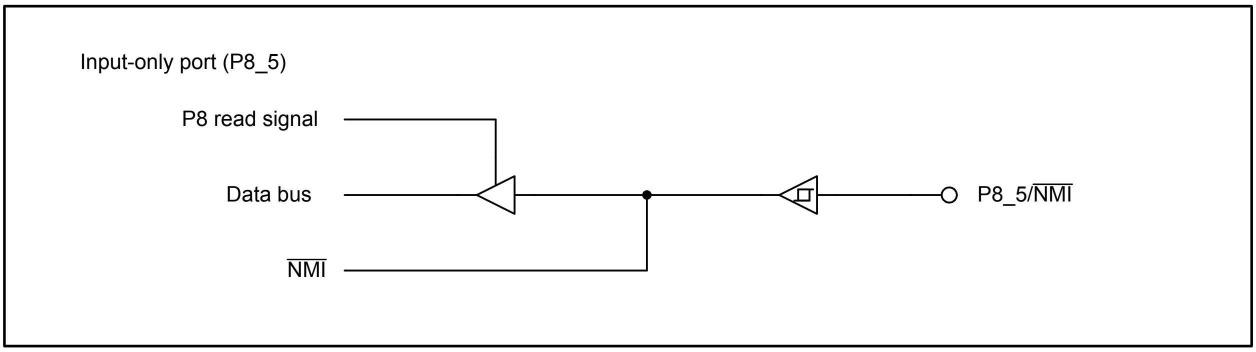


Figure 15.2 Input-only Port Configuration (1/2)

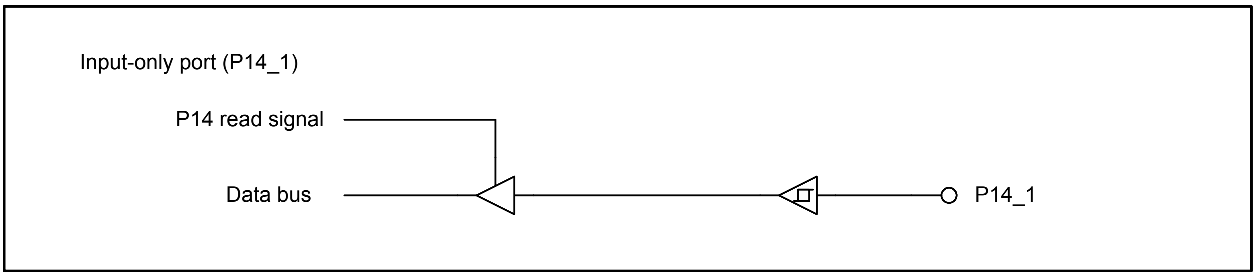


Figure 15.3 Input-only Port Configuration (2/2)

15.1 Port Pi Register (Pi register, i = 0 to 19)

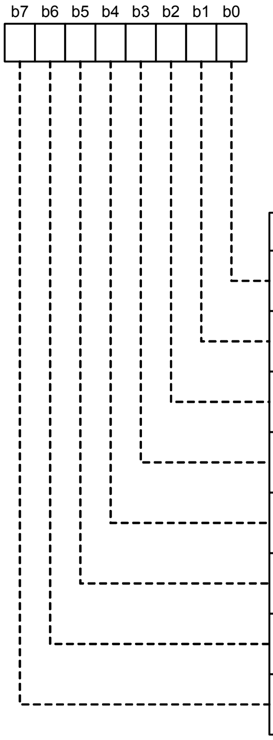
A write/read operation to the Pi register is required to communicate with external devices. This register consists of a port latch to hold output data and a circuit to read pin states. Bits in the Pi register correspond to respective ports.

When a programmable I/O port is selected in the output function select registers, the value in the port latch is read for output and the pin state is read for input.

In memory expansion mode and microprocessor mode, this register cannot control pins being assigned bus control signals (A0 to A23, D0 to D31, $\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS3}}$, $\overline{\mathrm{WR}}/\overline{\mathrm{WR0}}$, $\overline{\mathrm{BC0}}$, $\overline{\mathrm{BC1}}/\overline{\mathrm{WR1}}$, $\overline{\mathrm{BC2}}/\overline{\mathrm{WR2}}$, $\overline{\mathrm{BC3}}/\overline{\mathrm{WR3}}$, $\overline{\mathrm{RD}}$, CLKOUT/BCLK, $\overline{\mathrm{HLDA}}$, $\overline{\mathrm{HOLD}}$, ALE, and $\overline{\mathrm{RDY}}$).

Figure 15.4 shows the Pi register.

Port Pi Register (i = 0 to 19) [1, 2]



Symbol	Address	Reset Value
P0 to P3	03C0h, 03C1h, 03C4h, 03C5h	Undefined
P4 to P7	03C8h, 03C9h, 03CCh, 03CDh	Undefined
P8 [3], P9, P10	03D0h, 03D1h, 03D4h	Undefined
P11 to P13	03D5h, 03D8h, 03D9h	Undefined
P14 [3, 4], P15	03DCh, 03DDh	Undefined
P16 to P19 [2]	03E0h, 03E1h, 03E4h, 03E5h	Undefined

Bit Symbol	Bit Name	Function	RW
Pi_0	Port Pi_0 Bit [4]	When the direction bit is 0 (input) A value is written to the corresponding bit. It is not output due to input mode selected. The read value is the corresponding pin state as follows: 0: Low 1: High	RW
Pi_1	Port Pi_1 Bit [3]		RW
Pi_2	Port Pi_2 Bit [4]		RW
Pi_3	Port Pi_3 Bit		RW
Pi_4	Port Pi_4 Bit	When the direction bit is 1 (output) A value is written to the corresponding bit as follows: 0: Output low 1: Output high The read value has the same output level as that written to the corresponding bit or pin state	RW
Pi_5	Port Pi_5 Bit [3]		RW
Pi_6	Port Pi_6 Bit		RW
Pi_7	Port Pi_7 Bit		RW

Notes:

1. In memory expansion mode and microprocessor mode, this register cannot control pins being assigned bus control signals (A0 to A23, D0 to D31, $\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS3}}$, $\overline{\mathrm{WR}}/\overline{\mathrm{WR0}}$, $\overline{\mathrm{BC0}}$, $\overline{\mathrm{BC1}}/\overline{\mathrm{WR1}}$, $\overline{\mathrm{BC2}}/\overline{\mathrm{WR2}}$, $\overline{\mathrm{BC3}}/\overline{\mathrm{WR3}}$, $\overline{\mathrm{RD}}$, CLKOUT/BCLK, $\overline{\mathrm{HLDA}}$, $\overline{\mathrm{HOLD}}$, ALE, and $\overline{\mathrm{RDY}}$).
2. Registers P16 to P18 are available in the 176-pin package only.
3. The P8_5 bit in the P8 register and the P14_1 bit in the P14 register are read only.
4. Bits P14_0 and P14_2 in the P14 register are reserved. These bits should be written with 0 and read as undefined values.

Figure 15.4 Registers P0 to P19

15.2 Self Test Function

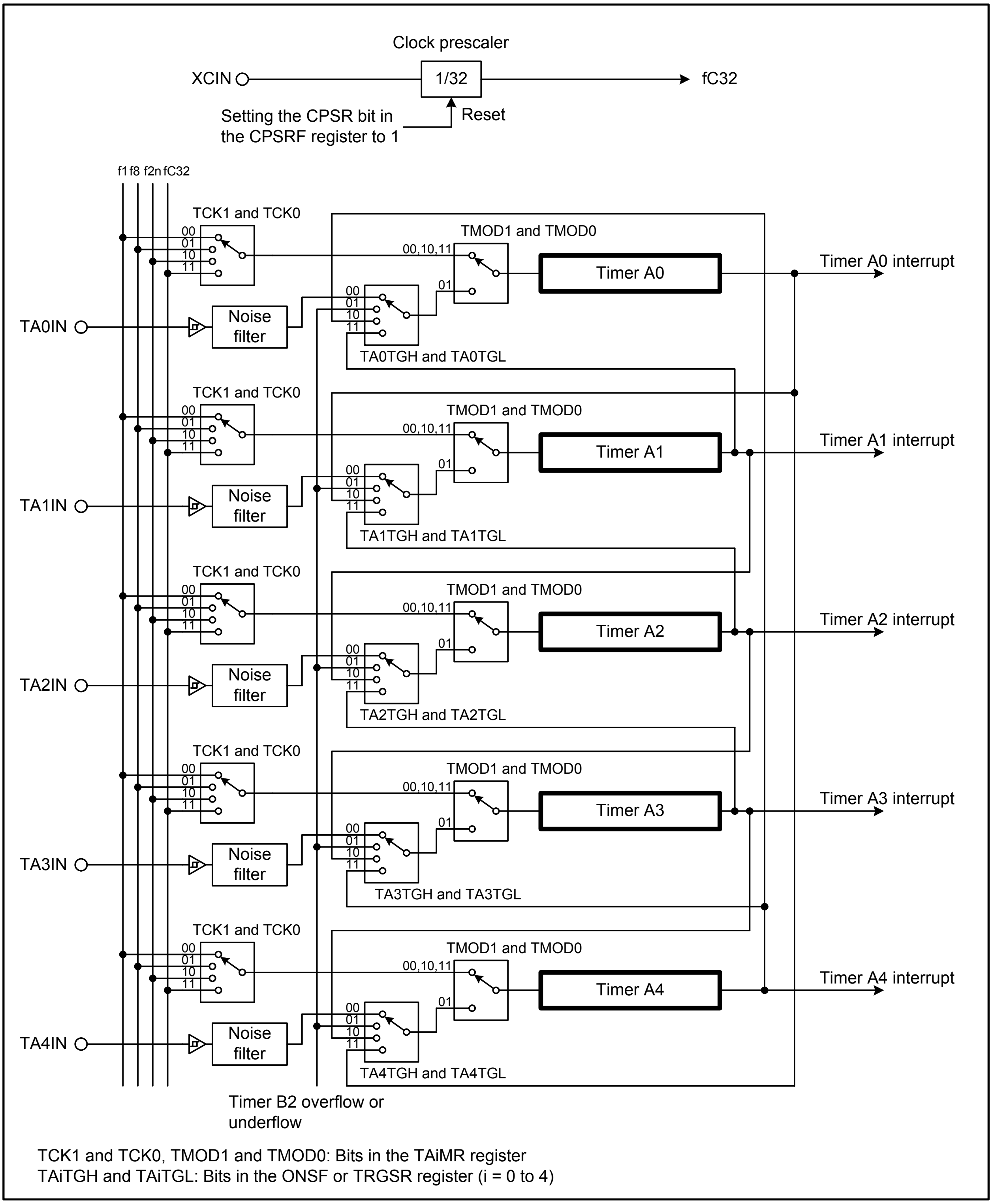
An output pin state can be verified by an output test of ports.
When the PRDS bit in the PCR register is set to 1 (pin state is read) while the PDi_j bit in the PDi register is 1 (output), the state of the Pi_j pin is read. When the read value is identical to the value written in the Pi_j bit in the Pi register, the setting value is output to the corresponding pin without any error. Otherwise, the pin may have a problem.
For example, when the read value is 0 despite the setting value is 1, this pin must be shorted out to a pin held low like VSS pin. Reversely, when the read value is 1 although the setting value is 0, the pin must be shorted out to a pin held high like VCC pin.
When the both setting values 0 and 1 are identical to the respective read values, at least, the pin is not shorted out to the VSS pin or VCC pin.

To rewrite a single bit in the Pi register, set the PRDS bit in the PCR register to 0 (value in the port latch is read). Otherwise, other bits may be also rewritten.

16. Timers

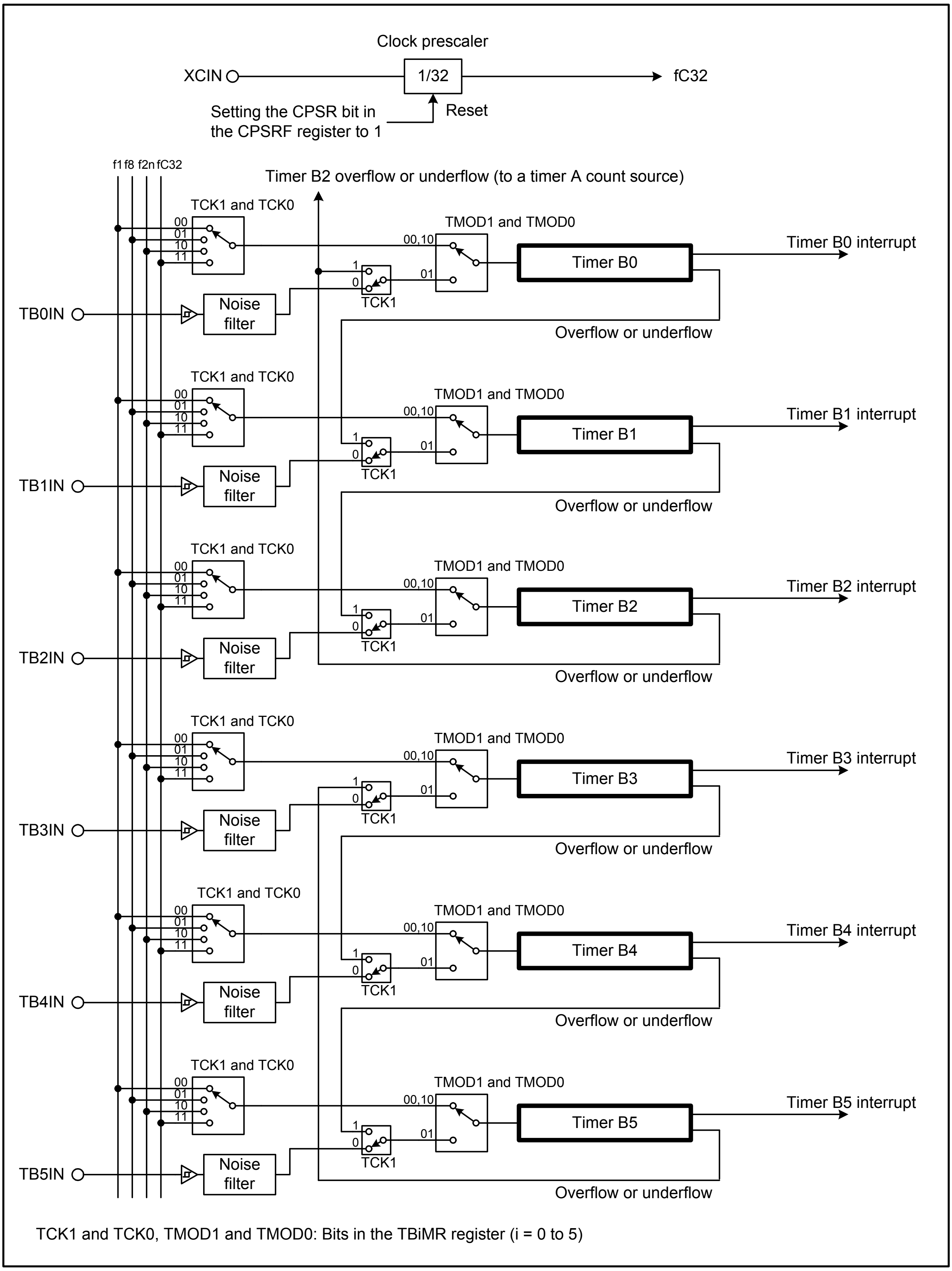
This MCU has eleven 16-bit timers which are divided into two groups according to their functions: five timer As and six timer Bs. Each timer functions individually. The count source of each timer provides the clock for timer operations such as counting and reloading.

Figure 16.1 and Figure 16.2 show the configuration of timers A and B, respectively.



TCK1 and TCK0, TMOD1 and TMOD0: Bits in the TAiMR register
TAiTGH and TAiTGL: Bits in the ONSF or TRGSR register (i = 0 to 4)

Figure 16.1 Timer A Configuration



TCK1 and TCK0, TMOD1 and TMOD0: Bits in the TBiMR register (i = 0 to 5)

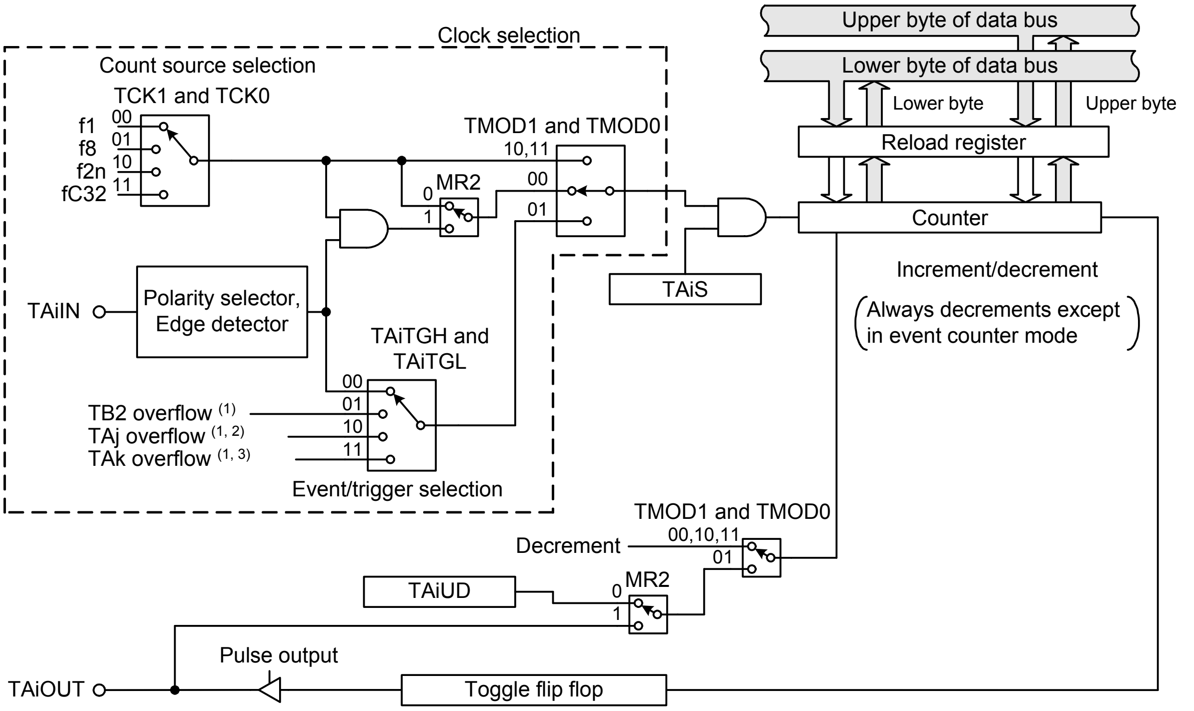
Figure 16.2 Timer B Configuration

16.1 Timer A

Figure 16.3 shows a block diagram of timer A and Figure 16.4 to Figure 16.10 show registers associated with timer A.

Timer A supports the four modes shown below. Timers A0 to A4 in any mode other than the event counter mode have the same function. Select a mode by setting bits TMOD1 and TMOD0 in the TAiMR register (i = 0 to 4).

- Timer mode: The timer counts an internal count source
- Event counter mode: The timer counts an external pulse or overflow and underflow of other timers
- One-shot timer mode: The timer outputs one valid pulse before the counter reaches 0000h
- Pulse-width modulation mode: The timer successively outputs pulses of a given width



TAi	TAj	TAk
Timer A0	Timer A4	Timer A1
Timer A1	Timer A0	Timer A2
Timer A2	Timer A1	Timer A3
Timer A3	Timer A2	Timer A4
Timer A4	Timer A3	Timer A0

i = 0 to 4

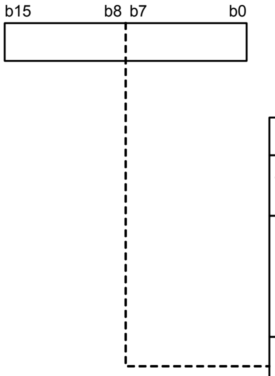
Notes:

1. The timer overflows or underflows.
2. j = i - 1, or j = 4 if i = 0 (refer to the list on the right)
3. k = i + 1, or k = 0 if i = 4 (refer to the list on the right)

TCK1 and TCK0, TMOD1 and TMOD0, MR2: Bits in the TAiMR register
TAiTGH and TAiTGL: Bits in the ONSF register (i = 0) or in the TRGSR register (i = 1 to 4)
TAiS: Bits in the TABSR register
TAiUD: Bits in the UDF register

Figure 16.3 Timer A Block Diagram

Timer Ai Register (i = 0 to 4) (1)



Symbol	Address	Reset Value
TA0 to TA2	0347h-0346h, 0349h-0348h, 034Bh-034Ah	Undefined
TA3, TA4	034Dh-034Ch, 034Fh-034Eh	Undefined

Mode	Function	Setting Range	RW
Timer Mode	Divides the count source by n+1 (n = setting value)	0000h to FFFFh	RW
Event Counter Mode	Divides the count source by FFFFh -n+1 (when incrementing) or by n+1 (when decrementing) (n = setting value) (2)	0000h to FFFFh	RW
One-shot Timer Mode	Divides the count source by n, then stops (n = setting value) (3)	0000h to FFFFh (4)	WO
Pulse-width Modulation Mode (16-bit PWM)	PWM period: $(2^{16}-1)$ / fj High level width of PWM pulse: n / fj (fj = count source frequency, n = setting value of the TAi register) (5)	0000h to FFFEh (4)	WO
Pulse-width Modulation Mode (8-bit PWM)	PWM period: $(2^{8}-1) \times (m+1)$ / fj High level width of PWM pulse: (m+1)n / fj (fj = count source frequency, n = setting value of the upper byte in the TAi register, m = setting value of the lower byte in the TAi register) (5)	00h to FEh (upper byte) 00h to FFh (lower byte) (4)	WO

fj: f1, f8, f2n, fC32

Notes:

1. A 16-bit read/write access to this register should be performed.
2. The timer counts an external input pulse or overflow and underflow of other timers.
3. When the TAi register is set to 0000h, the timer counter does not start, and the TAi interrupt request is not generated.
4. Use the MOV instruction to set the TAi register.
5. When the TAi register is set to 0000h, the pulse-width modulator does not operate, the TAiOUT pin is held low, and the TAi interrupt request is not generated. The same restrictions apply in 8-bit pulse-width modulator mode if the upper byte in the TAi register is set to 00h.

Figure 16.4 Registers TA0 to TA4

Timer Ai Mode Register (i = 0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
					0		

Symbol
TA0MR to TA4MR

Address
0356h, 0357h, 0358h, 0359h, 035Ah

Reset Value
0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse-width modulation mode	RW
TMOD1			RW
— (b2)	Reserved	Should be written with 0	RW
MR1	—	Function varies according to the operating mode	RW
MR2			RW
MR3			RW
TCK0	Count Source Select Bit	Function varies according to the operating mode	RW
TCK1			RW

Figure 16.5 Registers TA0MR to TA4MR

Count Start Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol
TABSR

Address
0340h

Reset Value
0000 0000b

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 Count Start Bit	0: Stop counter 1: Start counter	RW
TA1S	Timer A1 Count Start Bit	0: Stop counter 1: Start counter	RW
TA2S	Timer A2 Count Start Bit	0: Stop counter 1: Start counter	RW
TA3S	Timer A3 Count Start Bit	0: Stop counter 1: Start counter	RW
TA4S	Timer A4 Count Start Bit	0: Stop counter 1: Start counter	RW
TB0S	Timer B0 Count Start Bit	0: Stop counter 1: Start counter	RW
TB1S	Timer B1 Count Start Bit	0: Stop counter 1: Start counter	RW
TB2S	Timer B2 Count Start Bit	0: Stop counter 1: Start counter	RW

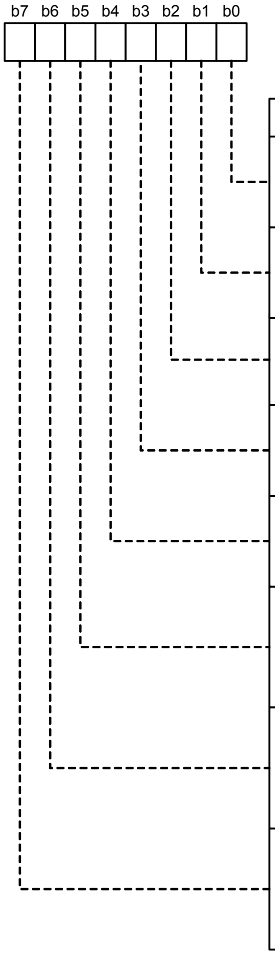
Figure 16.6 TABSR Register

Increment/Decrement Select Register [1]

Symbol UDF

Address 0344h

Reset Value 0000 0000b



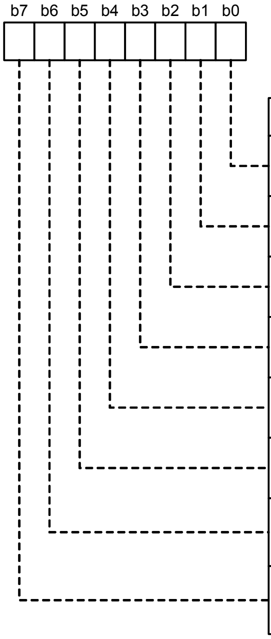
Bit Symbol	Bit Name	Function	RW
TA0UD	Timer A0 Increment/Decrement Select Bit	0: Count decremented 1: Count incremented [2]	RW
TA1UD	Timer A1 Increment/Decrement Select Bit	0: Count decremented 1: Count incremented [2]	RW
TA2UD	Timer A2 Increment/Decrement Select Bit	0: Count decremented 1: Count incremented [2]	RW
TA3UD	Timer A3 Increment/Decrement Select Bit	0: Count decremented 1: Count incremented [2]	RW
TA4UD	Timer A4 Increment/Decrement Select Bit	0: Count decremented 1: Count incremented [2]	RW
TA2P	Timer A2 Two-phase Pulse Signal Processing Select Bit	0: Two-phase pulse signal processing disabled 1: Two-phase pulse signal processing enabled [3]	WO
TA3P	Timer A3 Two-phase Pulse Signal Processing Select Bit	0: Two-phase pulse signal processing disabled 1: Two-phase pulse signal processing enabled [3]	WO
TA4P	Timer A4 Two-phase Pulse Signal Processing Select Bit	0: Two-phase pulse signal processing disabled 1: Two-phase pulse signal processing enabled [3]	WO

Notes:

1. Use the MOV instruction to set this register.
2. This bit is enabled in event counter mode and when the MR2 bit in the TAiMR register is set to 0 (the UDF register setting is the source of increment/decrement switching) (i = 0 to 4).
3. Set this bit to 0 when not using two-pulse signal processing.

Figure 16.7 UDF Register

One-shot Start Register



Symbol	Address	Reset Value
ONSF	0342h	0000 0000b

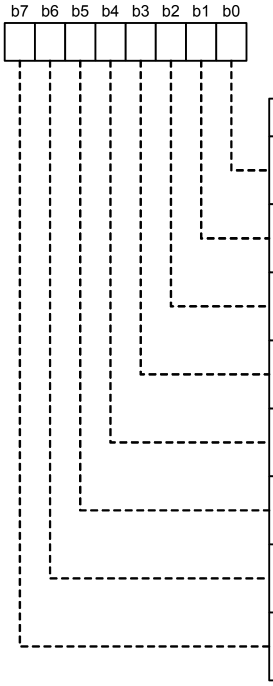
Bit Symbol	Bit Name	Function	RW
TA0OS	Timer A0 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TA1OS	Timer A1 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TA2OS	Timer A2 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TA3OS	Timer A3 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TA4OS	Timer A4 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TAZIE	Z-phase Input Enable Bit	0: Z-phase input disabled 1: Z-phase input enabled	RW
TA0TGL	Timer A0 Event/Trigger Select Bit	b7 b6 0 0 : Select the input to the TA0IN pin 0 1 : Select the overflow of TB2 (2) 1 0 : Select the overflow of TA4 (2) 1 1 : Select the overflow of TA1 (2)	RW
TA0TGH			RW

Notes:

1. This bit is read as 0.
2. The timer overflows or underflows.

Figure 16.8 ONSF Register

Trigger Select Register



Symbol	Address	Reset Value
TRGSR	0343h	0000 0000b

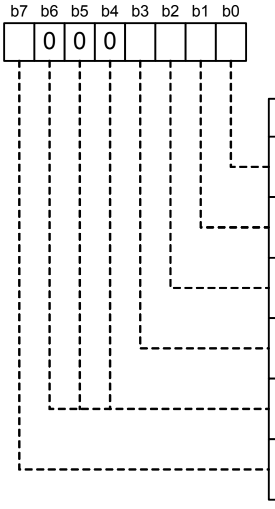
Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 Event/Trigger Select Bit	b1 b0 0 0 : Select the input to the TA1IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA0 (1) 1 1 : Select the overflow of TA2 (1)	RW
TA1TGH			RW
TA2TGL	Timer A2 Event/Trigger Select Bit	b2 b3 0 0 : Select the input to the TA2IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA1 (1) 1 1 : Select the overflow of TA3 (1)	RW
TA2TGH			RW
TA3TGL	Timer A3 Event/Trigger Select Bit	b4 b5 0 0 : Select the input to the TA3IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA2 (1) 1 1 : Select the overflow of TA4 (1)	RW
TA3TGH			RW
TA4TGL	Timer A4 Event/Trigger Select Bit	b6 b7 0 0 : Select the input to the TA4IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA3 (1) 1 1 : Select the overflow of TA0 (1)	RW
TA4TGH			RW

Note:

1. The timer overflows or underflows.

Figure 16.9 TRGSR Register

Count Source Prescaler Register



Symbol
TCSPR

Address
035Fh

Reset Value
0000 0000b

Bit Symbol	Bit Name	Function	RW
CNT0	Divide Ratio Select Bit [1]	f2n is either the main clock or peripheral clock source divided by 2n. If n = 0, the clock is not divided (n = setting value)	RW
CNT1			RW
CNT2			RW
CNT3			RW
— (b6-b4)	Reserved	Should be written with 0	RW
CST	Divider Operation Enable Bit	0: Stop divider operation 1: Start divider operation	RW

Note:

1. Set the CST bit to 0 before rewriting bits CNT3 to CNT0.

Figure 16.10 TCSPR Register

16.1.1 Timer Mode

In timer mode, the timer counts an internally generated count source. Table 16.1 lists the specifications of timer mode. Figure 16.11 shows registers TA0MR to TA4MR in this mode.

Table 16.1 Timer Mode Specifications (i = 0 to 4)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Decrement • When the timer counter underflows, the reload register value is reloaded into the counter to continue counting
Divide ratio	$\frac{1}{n+1}$ n: TAi register setting value, 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR register is 1 (start counter)
Count stop condition	The TAiS bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter underflows
TAiIN pin function	Functions as a programmable I/O port or a gate input
TAiOUT pin function	Functions as a programmable I/O port or a pulse output
Read from timer	The TAi register indicates the counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both the reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (it is transferred to the counter at the next reload timing)
Other functions	• Gate function Input signal to the TAiIN pin can control the count start/stop • Pulse output function The polarity of the TAiOUT pin is inverted each time the timer counter underflows. A low is output while the TAiS bit holds 0 (stop counter)

Timer Ai Mode Register (i = 0 to 4) (timer mode)

b7	b6	b5	b4	b3	b2	b1	b0
		0			0	0	0

Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
— (b2)	Reserved	Should be written with 0	RW
MR1	Gate Function Select Bit	b4 b3 0 X : No gate function (1) (TAiIN pin functions as programmable I/O port) 1 0 : Count only while the TAiIN pin is held low 1 1 : Count only while the TAiIN pin is held high	RW
MR2			RW
MR3	Should be written with 0 in timer mode		RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Note:

1. X can be set to either 0 or 1.

Figure 16.11 Registers TA0MR to TA4MR in Timer Mode

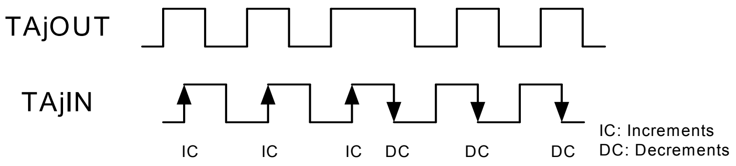
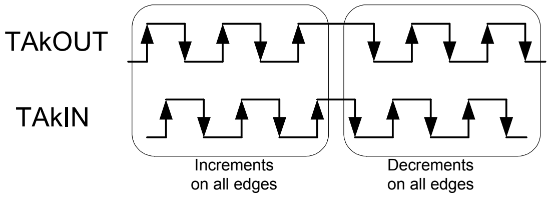
16.1.2 Event Counter Mode

In event counter mode, the timer counts an external signal or an overflow and underflow of other timers. Timers A2, A3, and A4 can count two-phase external signals. Table 16.2 lists the specifications in event count mode and Table 16.3 also lists the specifications when the timers use two-phase pulse signal processing. Figure 16.12 shows registers TA0MR to TA4MR in this mode.

Table 16.2 Event Counter Mode Specifications (without two-phase pulse signal processing) (i = 0 to 4)

Item	Specification
Count sources	• External signal applied to the TAiIN pin (valid edge is selectable by a program) • One of the following: the overflow and/or underflow signal of timer B2, the overflow and/or underflow signal of timer Aj (j = i - 1, or j = 4 if i = 0), or the overflow and/or underflow signal of timer Ak (k = i + 1, or k = 0 if i = 4)
Count operations	• Increment/decrement can be switched by an external signal or program • When the timer counter underflows or overflows, the reload register value is reloaded into the counter to continue counting. In a free-running count operation, the timer counter continues counting without reloading
Divide ratio	• $\frac{1}{FFFFh - n + 1}$ when incrementing • $\frac{1}{n+1}$ when decrementing n: TAi register setting value, 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR register is 1 (start counter)
Count stop condition	The TAiS bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter overflows or underflows
TAiIN pin function	Functions as a programmable I/O port or a count source input
TAiOUT pin function	Functions as a programmable I/O port, a pulse output, or an input for switching between increment/decrement
Read from timer	The TAi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both the reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (it is transferred to the counter at the next reload timing)
Other functions	• Free-running count function The reload register value is not reloaded even if the timer counter overflows or underflows • Pulse output function The polarity of the TAiOUT pin is inverted whenever the timer counter overflows or underflows. A low is output while the TAiS bit holds 0 (stop counter)

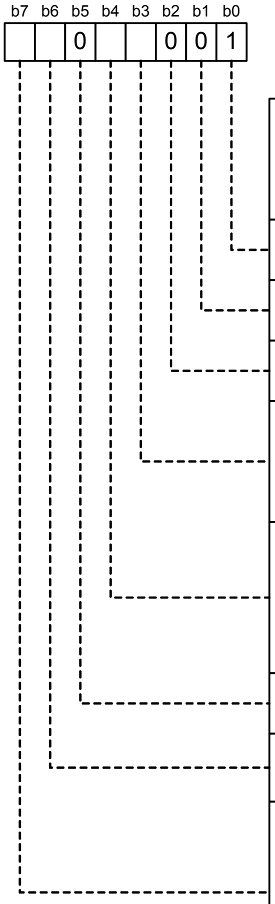
Table 16.3 Event Counter Mode Specifications (with two-phase pulse signal processing on timers A2 to A4) (i = 2 to 4)

Item	Specification
Count sources	Two-phase pulse signal applied to pins TAiIN and TAiOUT
Count operations	• Increment/decrement can be switched by a two-phase pulse signal • When the timer counter underflows or overflows, the reload register value is reloaded into the counter to continue counting. In a free-running count operation, the timer counter continues counting without reloading
Divide ratio	• $\frac{1}{FFFFh - n + 1}$ when incrementing • $\frac{1}{n + 1}$ when decrementing n: TAi register setting value, 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR register is 1 (start counter)
Count stop condition	The TAiS bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter overflows or underflows
TAiIN pin function	A two-phase pulse input
TAiOUT pin function	A two-phase pulse input
Read from timer	The TAi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both the reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (it is transferred to the counter at the next reload timing)
Other functions [(1)]	• Normal processing operation (timers A2 and A3) While the input signal applied to the TAjOUT pin is held high, the timer increments on the rising edge of the TAjIN pin and decrements on the falling edge (j = 2 or 3)  TAjOUT TAjIN IC IC IC DC DC DC IC: Increments DC: Decrements • Quadrupled processing operation (timers A3 and A4) When the input signal applied to the TAkOUT pin is held high on the rising edge of the TAkIN pin, the timer increments on both the rising and falling edges of pins TAkOUT and TAkIN (k = 3 or 4). When the signal is held high on the falling edge of the TAkIN pin, the timer decrements on both the rising and falling edges of pins TAkOUT and TAkIN  TAkOUT TAkIN Increments on all edges Decrements on all edges • Counter reset by Z-phase input (timer A3) The counter value is set to 0 by Z-phase input

Note:

1. Only timer A3 is available for any of the other functions. Timer A2 is exclusively for normal processing operations and timer A4 is for the quadrupled processing operation.

Timer Ai Mode Register (i = 0 to 4) (event counter mode)



Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function (without two-phase pulse signal processing)	Function (with two-phase pulse signal processing)	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 1 : Event counter mode (1)		RW
TMOD1				RW
— (b2)	Reserved	Should be written with 0		RW
MR1	Count Polarity Select Bit (2)	0: Count falling edges 1: Count rising edges	Should be written with 0	RW
MR2	Increment/Decrement Switching Source Select Bit	0: UDF register setting 1: Input signal to the TAiOUT pin (3)	Should be written with 1	RW
MR3	Should be written with 0 in event counter mode			RW
TCK0	Count Operation Type Select Bit	0: Reloading 1: Free-running		RW
TCK1	Two-phase Pulse Processing Operation Select Bit (4, 5)	Should be written with 0	0: Normal processing operation 1: Quadrupled processing operation	RW

Notes:

1. Set bits TAiTGH and TAiTGL in the ONSF or TRGSR register to select the count source in event counter mode.
2. This bit setting is enabled only when an external signal is counted.
3. The timer decrements when the input signal to the TAiOUT pin is held low, and increments when the signal is held high.
4. The TCK1 bit is enabled only in the TA3MR register.
5. For two-phase pulse signal processing, set the TAjP bit in the UDF register to 1 (two-phase pulse signal processing enabled) and bits TAjTGH and TAjTGL to 00b (input to the TAjIN pin) (j = 2 to 4).

Figure 16.12 Registers TA0MR to TA4MR in Event Counter Mode

16.1.2.1 Counter Reset by Two-phase Pulse Signal Processing

A Z-phase input signal resets the timer counter when a two-phase pulse signal is being processed.

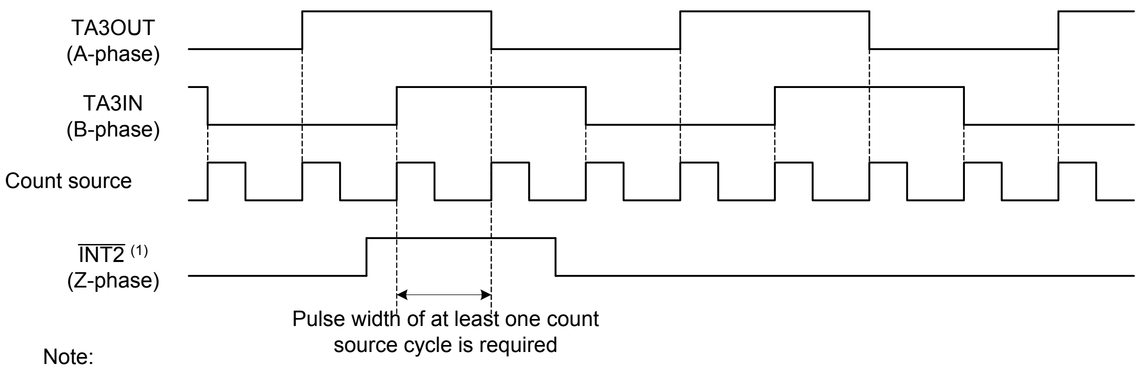
This function can be used under the following conditions: timer A3 event counter mode, two-phase pulse signal processing, free-running count operation, and quadrupled processing. The Z-phase signal is applied to the $\overline{\text{INT2}}$ pin.

When the TAZIE bit in the ONSF register is set to 1 (Z-phase input enabled), the timer counter can be reset by Z-phase input. To reset the counter, set the TA3 register to 0000h beforehand.

A Z-phase signal applied to the $\overline{\text{INT2}}$ pin is detected on an edge. The edge polarity is selected using the POL bit in the INT2IC register. The Z-phase signal should be input in order to have a pulse width of at least one count source cycle for timer A3. Figure 16.13 shows the two-phase pulse (phases A and B) and the Z-phase.

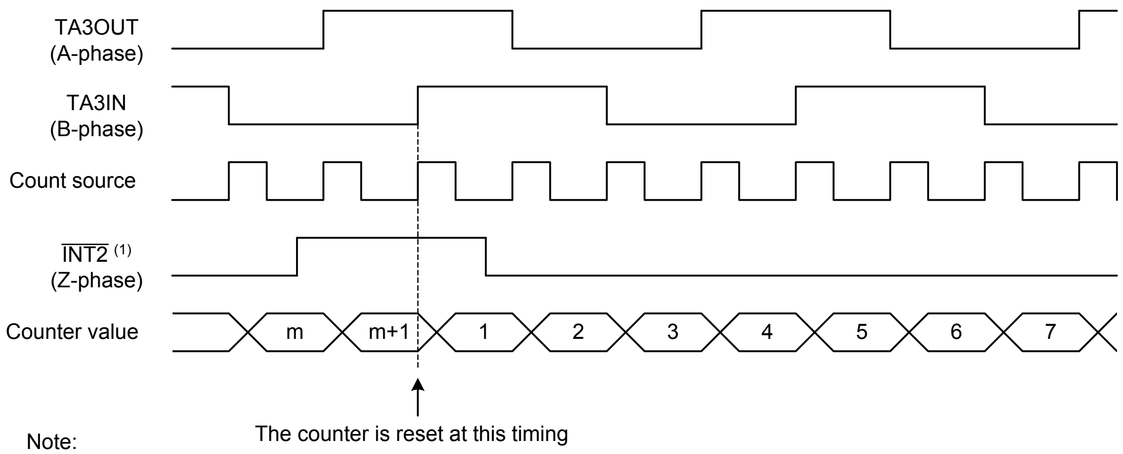
The timer counter is reset at the initial count source input after Z-phase input is detected. Figure 16.14 shows the counter reset timing.

When timer A3 overflows or underflows during a reset by the Z-phase input, two timer A3 interrupt requests are successively generated. To avoid this, the timer A3 interrupt request should not be used when using this function.


TA3OUT
(A-phase)
TA3IN
(B-phase)
Count source
$\overline{\text{INT2}}$ (1)
(Z-phase)
Pulse width of at least one count source cycle is required

Note:
1. This example assumes when the rising edge is selected for the $\overline{\text{INT}}$ signal.

Figure 16.13 Two-phase Pulse (phases A and B) and Z-phase



Note:
1. This example assumes when the rising edge is selected for the $\overline{\text{INT}}$ signal.

Figure 16.14 Counter Reset Timing

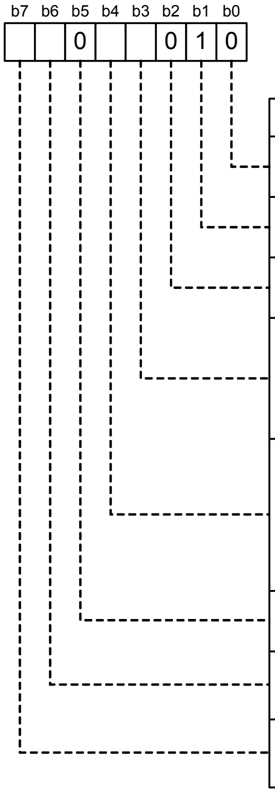
16.1.3 One-shot Timer Mode

In one-shot timer mode, the timer operates only once for each trigger. Table 16.4 lists specifications of one-shot timer mode. Once a trigger occurs, the timer starts and operates for a given period. Figure 16.15 shows registers TA0MR to TA4MR in this mode.

Table 16.4 One-shot Timer Mode Specifications (i = 0 to 4)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Decrement • When the timer counter reaches 0000h, it stops running after the reload register value is reloaded • When a trigger occurs while counting, the reload register value is reloaded into the counter to continue counting
Divide ratio	$\frac{1}{n}$ n: TAi register setting value, 0000h to FFFFh (Note that the timer counter does not run if n = 0000h)
Count start conditions	The TAiS bit in the TABSR register is 1 (start counter) and any of following triggers occurs: • An external trigger applied to the TAiIN pin • One of the following: the overflow and/or underflow signal of timer B2, the overflow and/or underflow signal of timer Aj (j = i - 1, or j = 4 if i = 0), or the overflow and/or underflow signal of timer Ak (k = i + 1, or k = 0 if i = 4) • The TAiOS bit in the ONSF register is 1 (start the timer)
Count stop conditions	• The timer counter reaches 0000h and the reload register value is reloaded • The TAiS bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter reaches 0000h
TAiIN pin function	A programmable I/O port or a trigger input
TAiOUT pin function	A programmable I/O port or a pulse output
Read from timer	The TAi register indicates an undefined value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both the reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (it is transferred to the counter at the next reload timing)
Other function	• Pulse output function A low is output while the timer counter is stopped and a high is output while the timer counter is running

Timer Ai Mode Register (i = 0 to 4) (one-shot timer mode)



Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 1 0 : One-shot timer mode	RW
TMOD1			RW
— (b2)	Reserved	Should be written with 0	RW
MR1	External Trigger Select Bit [1]	0: Falling edge of input signal to the TAiIN pin 1: Rising edge of input signal to the TAiIN pin	RW
MR2	Trigger Select Bit	0: TAiOS bit in the ONSF register is enabled 1: Selected using bits TAiTGH and TAiTGL in the ONSF or TRGSR register	RW
MR3	Should be written with 0 in one-shot timer mode		RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Note:

1. The MR1 bit setting is enabled only when bits TAiTGH and TAiTGL in the TRGSR register are set to 00b (input to the TAiIN pin). This bit can be set to either 0 or 1 when bits TAiTGH and TAiTGL are set to 01b (overflow or underflow of TB2), 10b (overflow or underflow of TAi), or 11b (overflow or underflow of TAi).

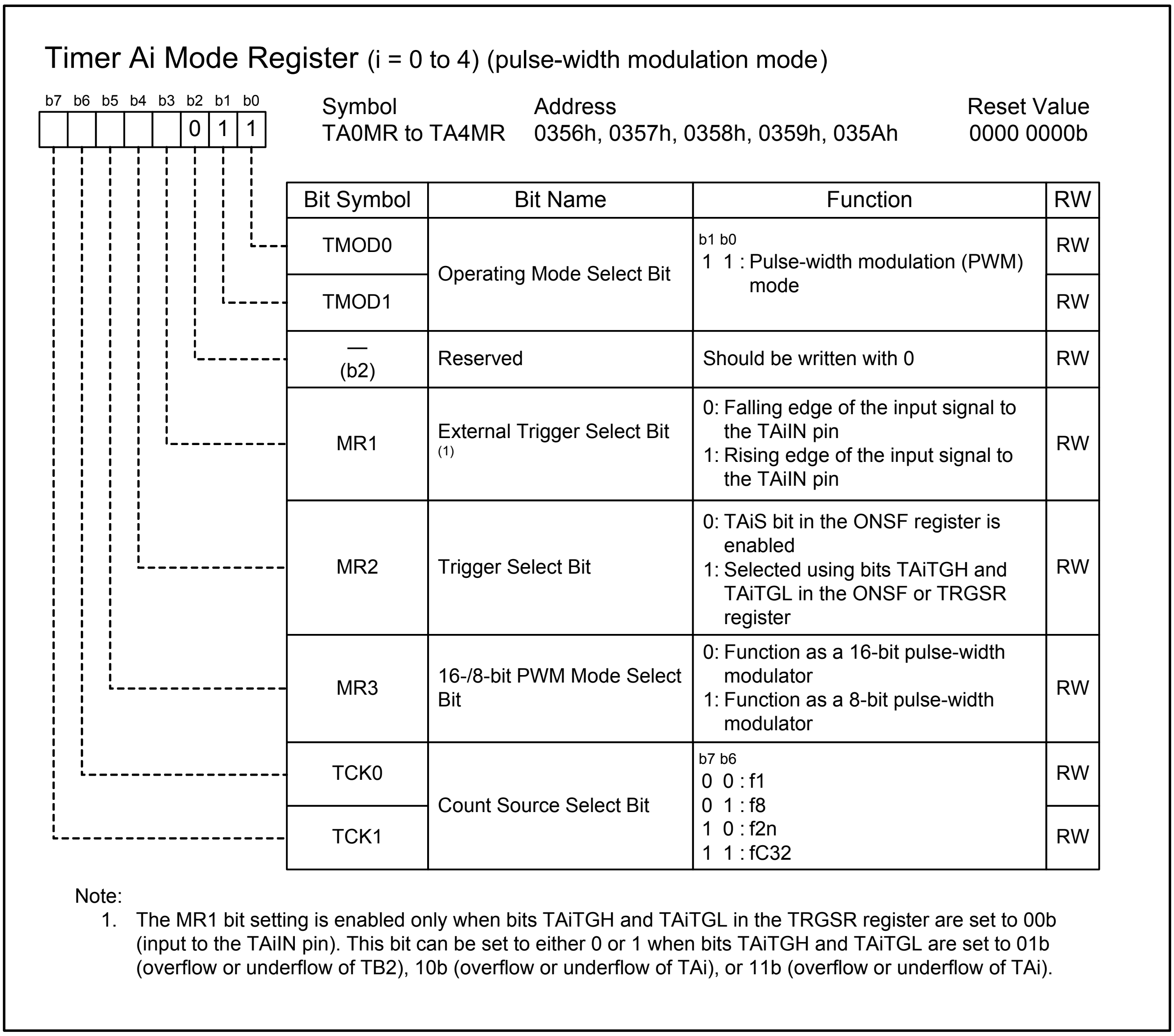
Figure 16.15 Registers TA0MR to TA4MR in One-shot Timer Mode

16.1.4 Pulse-width Modulation Mode

In pulse-width modulation mode, the timer outputs pulses of given width successively. Table 16.5 lists specifications of pulse-width modulation mode. The timer counter functions as either a 16-bit or 8-bit pulse-width modulator. Figure 16.16 shows registers TA0MR to TA4MR in this mode. Figures 16.17 and 16.18 show operation examples of 16-bit and 8-bit pulse-width modulators.

Table 16.5 Pulse-width Modulation Mode Specifications (i = 0 to 4)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Decrement (the timer counter functions as an 8-bit or a 16-bit pulse-width modulator) • The reload register value is reloaded on the rising edge of a PWM pulse to continue counting • The timer is not affected by a trigger that occurs while the counter is running
16-bit PWM	• High level width: $\frac{n}{fj}$ n: TAi register setting value, 0000h to FFFEh fj: Count source frequency • Period: fixed to $\frac{2^{16}-1}{fj}$
8-bit PWM	• High level width: $\frac{n \times (m+1)}{fj}$ • Period: $\frac{(2^8-1) \times (m+1)}{fj}$ n: Upper byte of the TAi register setting value, 00h to FEh m: Lower byte of the TAi register setting value, 00h to FFh
Count start conditions	• The TAiS bit in the TABSR register is 1 (start counter) • The TAiS bit is 1 and an external trigger is applied to the TAiIN pin • The TAiS bit is 1 and any of following triggers occurs: the overflow and/or underflow signal of timer B2, the overflow and/or underflow signal of timer Aj (j = i - 1, or j = 4 if i = 0), or the overflow and/ or underflow signal of timer Ak (k = i + 1, or k = 0 if i = 4)
Count stop condition	The TAiS bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	On the falling edge of the PWM pulse
TAiIN pin function	A programmable I/O port or trigger input
TAiOUT pin function	A pulse output
Read from timer	The TAi register indicates an undefined value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both the reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (it is transferred to the counter at the next reload timing)



Timer Ai Mode Register (i = 0 to 4) (pulse-width modulation mode)

Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

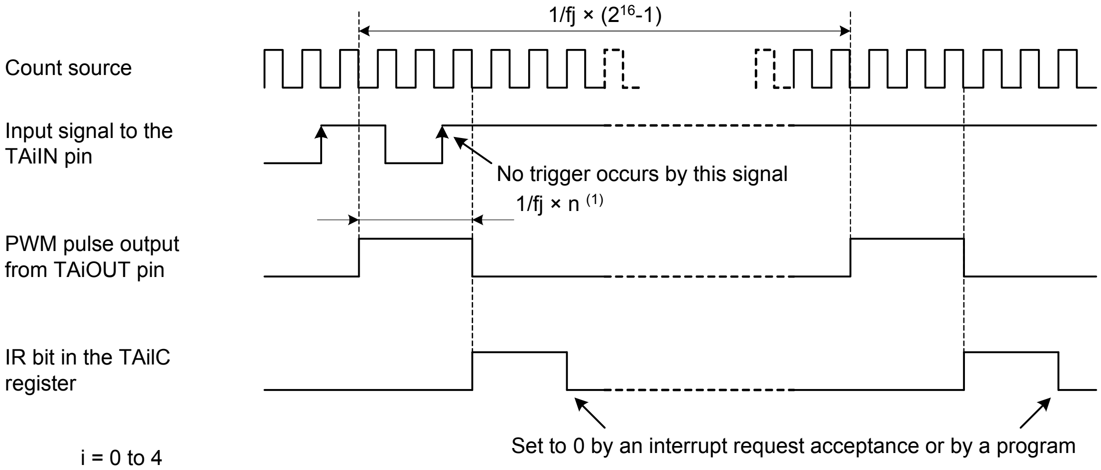
Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 1 1 : Pulse-width modulation (PWM) mode	RW
TMOD1			RW
— (b2)	Reserved	Should be written with 0	RW
MR1	External Trigger Select Bit [1]	0: Falling edge of the input signal to the TAiIN pin 1: Rising edge of the input signal to the TAiIN pin	RW
MR2	Trigger Select Bit	0: TAiS bit in the ONSF register is enabled 1: Selected using bits TAiTGH and TAiTGL in the ONSF or TRGSR register	RW
MR3	16-/8-bit PWM Mode Select Bit	0: Function as a 16-bit pulse-width modulator 1: Function as a 8-bit pulse-width modulator	RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Note:

1. The MR1 bit setting is enabled only when bits TAiTGH and TAiTGL in the TRGSR register are set to 00b (input to the TAiIN pin). This bit can be set to either 0 or 1 when bits TAiTGH and TAiTGL are set to 01b (overflow or underflow of TB2), 10b (overflow or underflow of TAi), or 11b (overflow or underflow of TAi).

Figure 16.16 Registers TA0MR to TA4MR in Pulse-width Modulation Mode

When the reload register is 0003h and an external trigger (the rising edge of an input signal applied to the TAiIN pin) is selected.


1/fj × (2^16-1)
Count source
Input signal to the TAiIN pin
No trigger occurs by this signal
1/fj × n (1)
PWM pulse output from TAiOUT pin
IR bit in the TAiC register
Set to 0 by an interrupt request acceptance or by a program

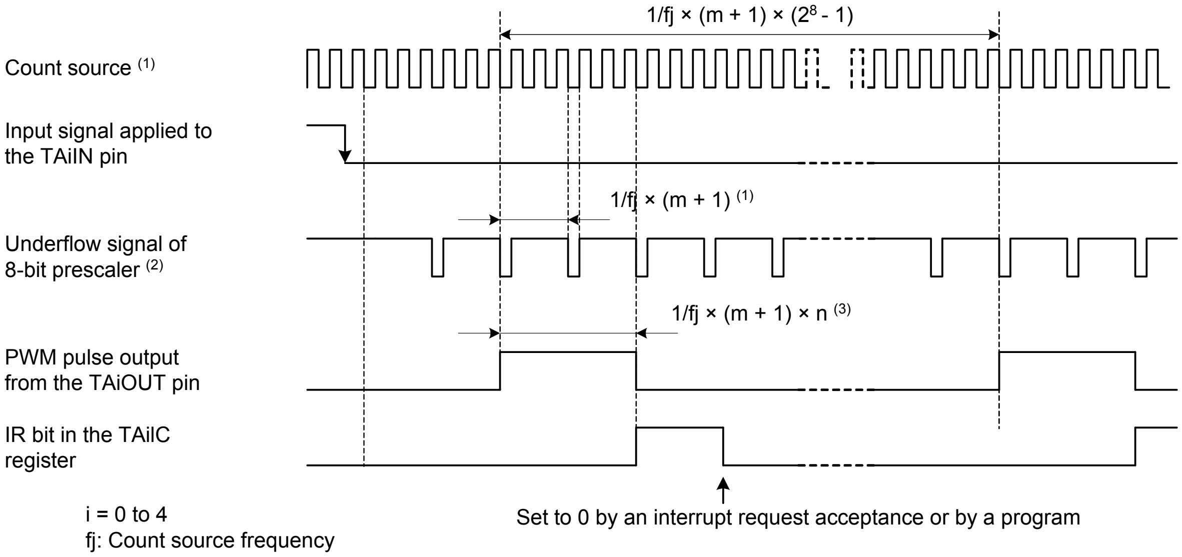
i = 0 to 4
fj: Count source frequency
(f1, f8, f2n, fC32)

Note:

1. n is a value from 0000h to FFFEh.

Figure 16.17 16-bit Pulse-width Modulator Operation

When the upper byte of the reload register is 02h, the lower byte is 02h and an external trigger (the falling edge of an input signal applied to the TAiIN pin) is selected.


1/fj × (m + 1) × (2^8 - 1)
Count source (1)
Input signal applied to the TAiIN pin
1/fj × (m + 1) (1)
Underflow signal of 8-bit prescaler (2)
1/fj × (m + 1) × n (3)
PWM pulse output from the TAiOUT pin
IR bit in the TAiC register
Set to 0 by an interrupt request acceptance or by a program

i = 0 to 4
fj: Count source frequency
(f1, f8, f2n, fC32)

Notes:

1. The 8-bit prescaler counts a count source.
2. The 8-bit pulse-width modulator counts underflow signals of the 8-bit prescaler.
3. m is a value from 00h to FFh, and n is a value from 00h to FEh.

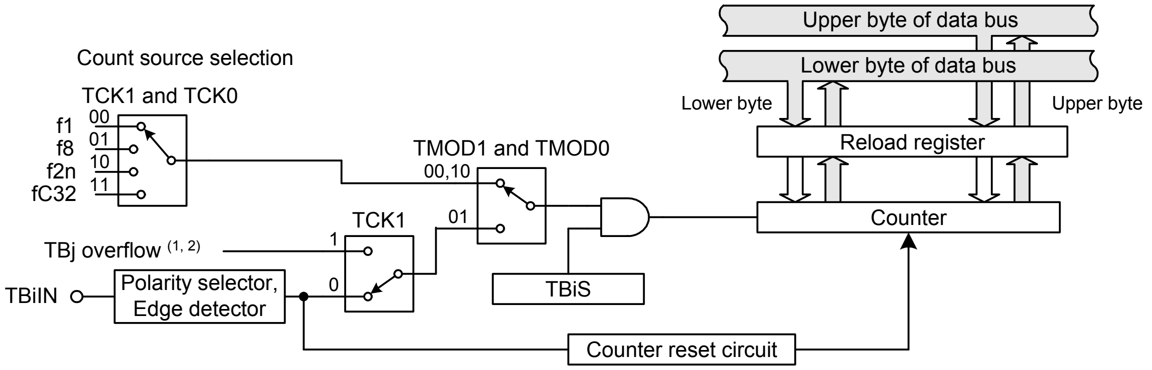
Figure 16.18 8-bit Pulse-width Modulator Operation

16.2 Timer B

Figure 16.19 shows a block diagram of timer B, and Figure 16.20 to Figure 16.23 show registers associated with timer B.

Timer B supports the three modes shown below. Select a mode by setting bits TMOD1 and TMOD0 in the TBiMR register (i = 0 to 5).

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts an external pulse or an overflow and underflow of other timers.
- Pulse period/pulse-width measure mode: The timer measures the pulse period or pulse width of an external signal.



i = 0 to 5

Notes:

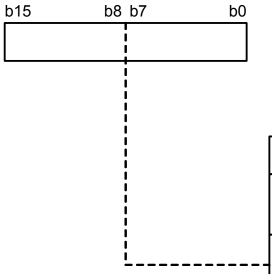
1. The timer overflows or underflows.
2. j = i - 1; j = 2 if i = 0; or j = 5 if i = 3 (refer to the list on the right)

TCK1 and TCK0, TMOD1 and TMOD0: Bits in the TBiMR register
TBiS: Bits in the TABSR or TBSR register

TBi	TBj
Timer B0	Timer B2
Timer B1	Timer B0
Timer B2	Timer B1
Timer B3	Timer B5
Timer B4	Timer B3
Timer B5	Timer B4

Figure 16.19 Timer B Block Diagram

Timer Bi Register (i = 0 to 5) (1)



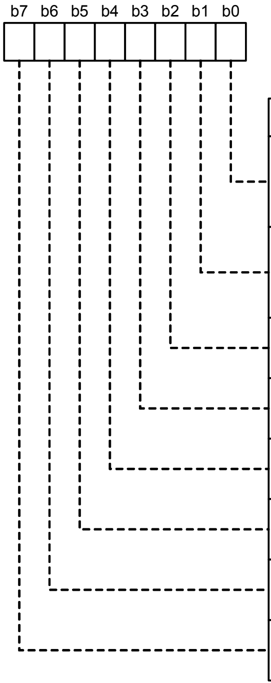
Symbol	Address	Reset Value
TB0 to TB2	0351h-0350h, 0353h-0352h, 0355h-0354h	Undefined
TB3 to TB5	0311h-0310h, 0313h-0312h, 0315h-0314h	Undefined

Mode	Function	Setting Range	RW
Timer Mode	Divides the count source by n+1 (n = setting value)	0000h to FFFFh	RW
Event Counter Mode	Divides the count source by n+1 (n = setting value) (2)	0000h to FFFFh	RW
Pulse Period/Pulse-width Measure Mode	Increments the counter between one valid edge and another of TBiIN input pulse	—	RO

Notes:
1. A 16-bit read/write access to this register should be performed.
2. The TBi register counts an external input pulse, or an overflow and underflow of other timers.

Figure 16.20 Registers TB0 to TB5

Timer Bi Mode Register (i = 0 to 5)



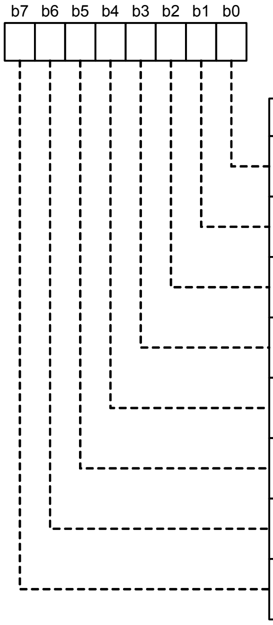
Symbol	Address	Reset Value
TB0MR to TB2MR	035Bh, 035Ch, 035Dh	00XX 0000b
TB3MR to TB5MR	031Bh, 031Ch, 031Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period measure mode, pulse-width measure mode 1 1 : Do not use this combination	RW
TMOD1			RW
MR0	—	Function varies according to the operating mode (1, 2)	RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count Source Select Bit	Function varies according to the operating mode	RW
TCK1			RW

Notes:
1. The MR2 bit is available for registers TB0MR and TB3MR only.
2. The MR2 bit in registers TB1MR, TB2MR, TB4MR and TB5MR are unavailable on this MCU. This bit should be written with 0 and read as undefined value.

Figure 16.21 Registers TB0MR to TB5MR

Count Start Register



Symbol
TABSR

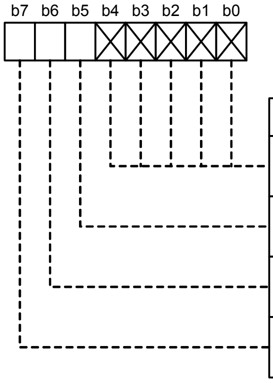
Address
0340h

Reset Value
0000 0000b

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 Count Start Bit	0: Stop counter 1: Start counter	RW
TA1S	Timer A1 Count Start Bit	0: Stop counter 1: Start counter	RW
TA2S	Timer A2 Count Start Bit	0: Stop counter 1: Start counter	RW
TA3S	Timer A3 Count Start Bit	0: Stop counter 1: Start counter	RW
TA4S	Timer A4 Count Start Bit	0: Stop counter 1: Start counter	RW
TB0S	Timer B0 Count Start Bit	0: Stop counter 1: Start counter	RW
TB1S	Timer B1 Count Start Bit	0: Stop counter 1: Start counter	RW
TB2S	Timer B2 Count Start Bit	0: Stop counter 1: Start counter	RW

Figure 16.22 TABSR Register

Count Start Register for Timers B3, B4, and B5



Symbol
TBSR

Address
0300h

Reset Value
000X XXXXb

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	No register bits; should be written with 0 and read as undefined value		—
TB3S	Timer B3 Count Start Bit	0: Stop counter 1: Start counter	RW
TB4S	Timer B4 Count Start Bit	0: Stop counter 1: Start counter	RW
TB5S	Timer B5 Count Start Bit	0: Stop counter 1: Start counter	RW

Figure 16.23 TBSR Register

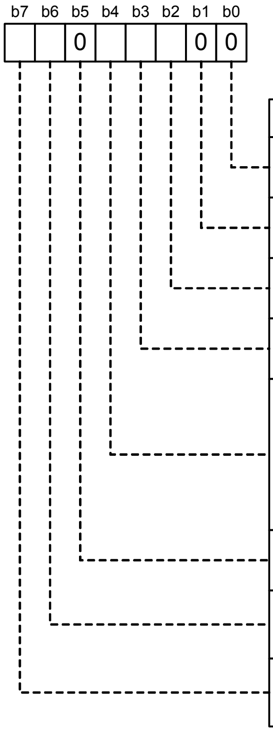
16.2.1 Timer Mode

In timer mode, the timer counts an internally generated count source. Table 16.6 lists specifications of timer mode. Figure 16.24 shows registers TB0MR to TB5MR in this mode.

Table 16.6 Timer Mode Specifications (i = 0 to 5)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Decrement • When the timer counter underflows, the reload register value is reloaded into the counter to continue counting
Divide ratio	$\frac{1}{n+1}$ n: TBi register setting value, 0000h to FFFFh
Count start condition	The TBiS bit in the TABSR or TBSR register is 1 (start counter)
Count stop condition	The TBiS bit in the TABSR or TBSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter underflows
TBiIN pin function	Functions as a programmable I/O port
Read from timer	The TBi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TBi register is written to both the reload register and the counter • While the timer counter is running, the value written to the TBi register is written to the reload register (it is transferred to the counter at the next reload timing)

Timer Bi Mode Register (i = 0 to 5) (timer mode)



Symbol	Address	Reset Value
TB0MR to TB2MR	035Bh, 035Ch, 035Dh	00XX 0000b
TB3MR to TB5MR	031Bh, 031Ch, 031Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Disabled in timer mode. Can be set to 0 or 1		RW
MR1			RW
MR2	In registers TB0MR and TB3MR: Reserved; should be written with 0		RW
	In registers TB1MR, TB2MR, TB4MR, and TB5MR: No register bit; should be written with 0 and read as undefined value		—
MR3	Disabled in timer mode. Should be written with 0 and read as undefined value		—
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Figure 16.24 Registers TB0MR to TB5MR in Timer Mode

16.2.2 Event Counter Mode

In event counter mode, the timer counts an external signal or the overflow or underflow of other timers. Table 16.7 lists specifications of event counter mode. Figure 16.25 shows the TBiMR register in this mode (i = 0 to 5).

Table 16.7 Event Counter Mode Specifications (i = 0 to 5)

Item	Specification
Count sources	• External signal applied to the TBiIN pin (valid edge is selectable among the falling edge, the rising edge, or both) • The overflow or underflow signal of TBj (j = i - 1; j = 2 if i = 0; or j = 5 if i = 3)
Count operations	• Decrement • When the timer counter underflows, the reload register value is reloaded into the counter to continue counting
Divide ratio	$\frac{1}{n+1}$ n: TBi register setting value, 0000h to FFFFh
Count start condition	The TBiS bit in the TABSR or TBSR register is 1 (start counter)
Count stop condition	The TBiS bit in the TABSR or TBSR register is 0 (stop counter)
Interrupt request generation timing	When the timer counter underflows
TBiIN pin function	Functions as a programmable I/O port or count source input
Read from timer	The TBi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TBi register is written to both the reload register and the counter • While the timer counter is running, the value written to the TBi register is written to the reload register (it is transferred to the counter at the next reload timing)

Timer Bi Mode Register (i = 0 to 5) (event counter mode)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol	Address	Reset Value
TB0MR to TB2MR	035Bh, 035Ch, 035Dh	00XX 0000b
TB3MR to TB5MR	031Bh, 031Ch, 031Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	Count Polarity Select Bit [1]	b3 b2 0 0 : Count falling edges 0 1 : Count rising edges 1 0 : Count both edges 1 1 : Do not use this combination	RW
MR1			RW
MR2	In registers TB0MR and TB3MR: Reserved; should be written with 0		RW
	In registers TB1MR, TB2MR, TB4MR, and TB5MR: No register bit; should be written with 0 and read as undefined value		—
MR3	Disabled in event counter mode. Should be written with 0 and read as undefined value		—
TCK0	Disabled in event counter mode. Can be set to 0 or 1		RW
TCK1	Event Clock Select Bit	0: Input signal to the TBiIN pin 1: Overflow or underflow of TBj [2]	RW

Notes:

1. These bit settings are enabled when the TCK1 bit is 0. When the TCK1 bit is 1, these bits can be set to either 0 or 1.
2. j = i - 1; j = 2 if i = 0; or j = 5 if i = 3.

Figure 16.25 Registers TB0MR to TB5MR in Event Counter Mode

16.2.3 Pulse Period/Pulse-width Measure Mode

In pulse period/pulse-width measure mode, the timer measures the pulse period or pulse width of an external signal. Table 16.8 lists specifications of the pulse period/pulse-width measure mode. Figure 16.26 shows registers TB0MR to TB5MR in this mode. Figures 16.27 and 16.28 show an operation example of pulse period measurement and pulse-width measurement, respectively.

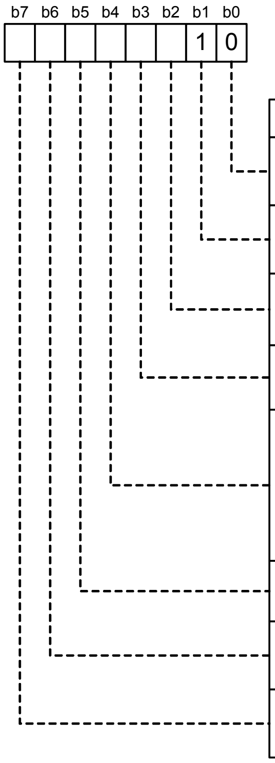
Table 16.8 Pulse Period/Pulse-width Measure Mode Specifications (i = 0 to 5)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Increment • The counter value is transferred to the reload register on the valid edge of a pulse to be measured, then it is set to 0000h to resume counting
Count start condition	The TBiS bit in the TABSR or TBSR register is 1 (start counter)
Count stop condition	The TBiS bit in the TABSR or TBSR register is 0 (stop counter)
Interrupt request generating timing	• On the valid edge of a pulse to be measured [1] • When the timer counter overflows (when the MR3 bit in the TBiMR register becomes 1 (overflow)) [2]
TBiIN pin function	A pulse input to be measured
Read from timer	The TBi register indicates a reload register value (measurement results) [3]
Write to timer	The value written to the TBi register is written to neither the reload register nor the counter

Notes:

1. No interrupt request is generated when the pulse to be measured is applied on the initial valid edge after the timer counter starts.
2. While the TBiS bit is 1 (start counter), after the MR3 bit becomes 1 (overflow) and at least one count source cycle has elapsed, a write operation to the TBiMR register sets the MR3 bit to 0 (no overflow).
3. The TBi register indicates an undefined value until the pulse to be measured is applied on the second valid edge after the timer counter starts.

Timer Bi Mode Register (i = 0 to 5) (pulse period/pulse-width measure mode)



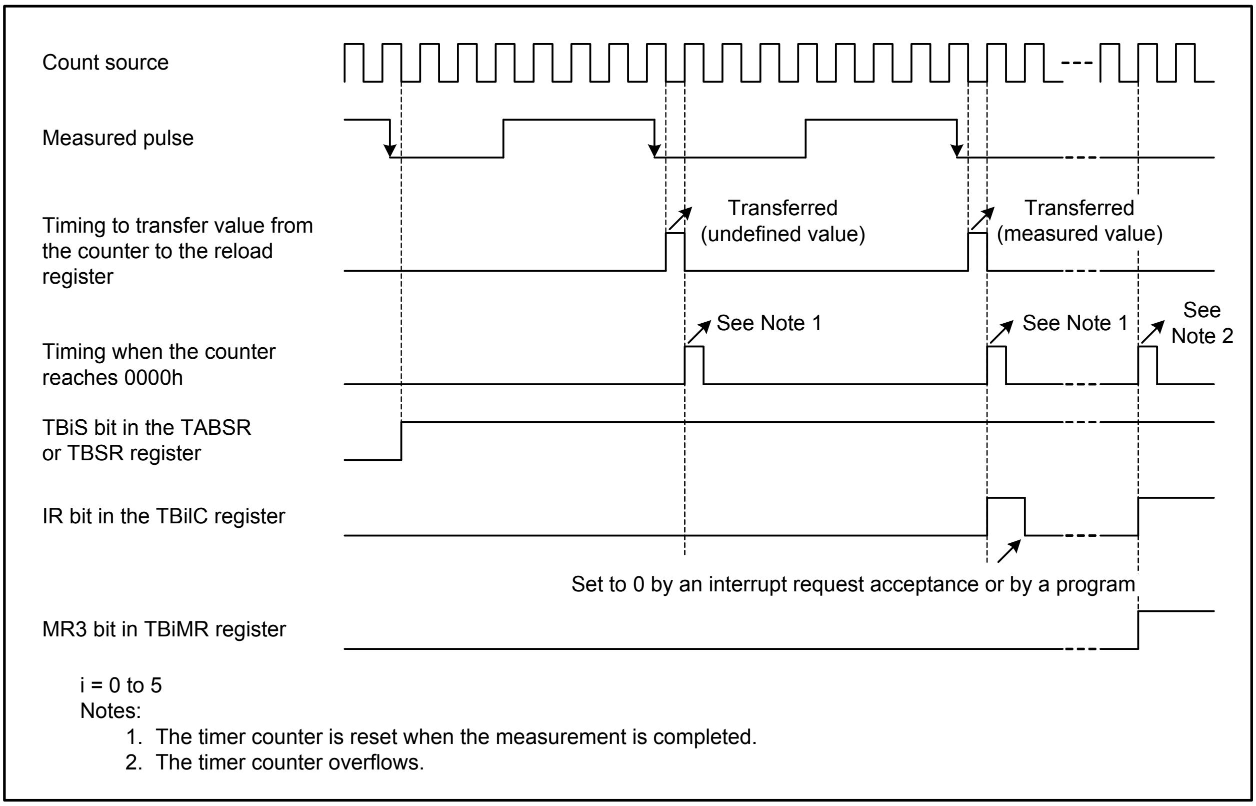
Symbol	Address	Reset Value
TB0MR to TB2MR	035Bh, 035Ch, 035Dh	00XX 0000b
TB3MR to TB5MR	031Bh, 031Ch, 031Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 1 0 : Pulse period/pulse-width measure mode	RW
TMOD1			RW
MR0	Measure Mode Select Bit [(1)]	b3 b2 0 0 : Pulse period measurement 1 0 1 : Pulse period measurement 2 1 0 : Pulse-width measurement 1 1 : Do not use this combination	RW
MR1			RW
MR2	In registers TB0MR and TB3MR: Reserved; should be written with 0		RW
	In registers TB1MR, TB2MR, TB4MR, and TB5MR: No register bit; should be written with 0 and read as undefined value		—
MR3	Timer Bi Overflow Flag [(2)]	0: No overflow 1: Overflow	RO
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Notes:

1. The measure modes selected by setting bits MR1 and MR0 are as follows:
   Pulse period measurement 1 (bits MR1 and MR0 = 00b):
   Measures between a falling edge and the next falling edge of a pulse
   Pulse period measurement 2 (bits MR1 and MR0 = 01b):
   Measures between a rising edge and the next rising edge of a pulse
   Pulse-width measurement (bits MR1 and MR0 = 10b):
   Measures between a falling edge and the next rising edge of a pulse and between the rising edge and the next falling edge of the pulse
2. The MR3 bit is undefined when the timer is reset.
   While the TBiS bit in the TABSR or TBSR register is 1 (start counter), after the MR3 bit becomes 1 and at least one count source cycle has elapsed, a write operation to the TBiMR register sets the MR3 bit to 0. The MR3 bit cannot be set to 1 by a program.

Figure 16.26 Registers TB0MR to TB5MR in Pulse Period/Pulse-width Measure Mode

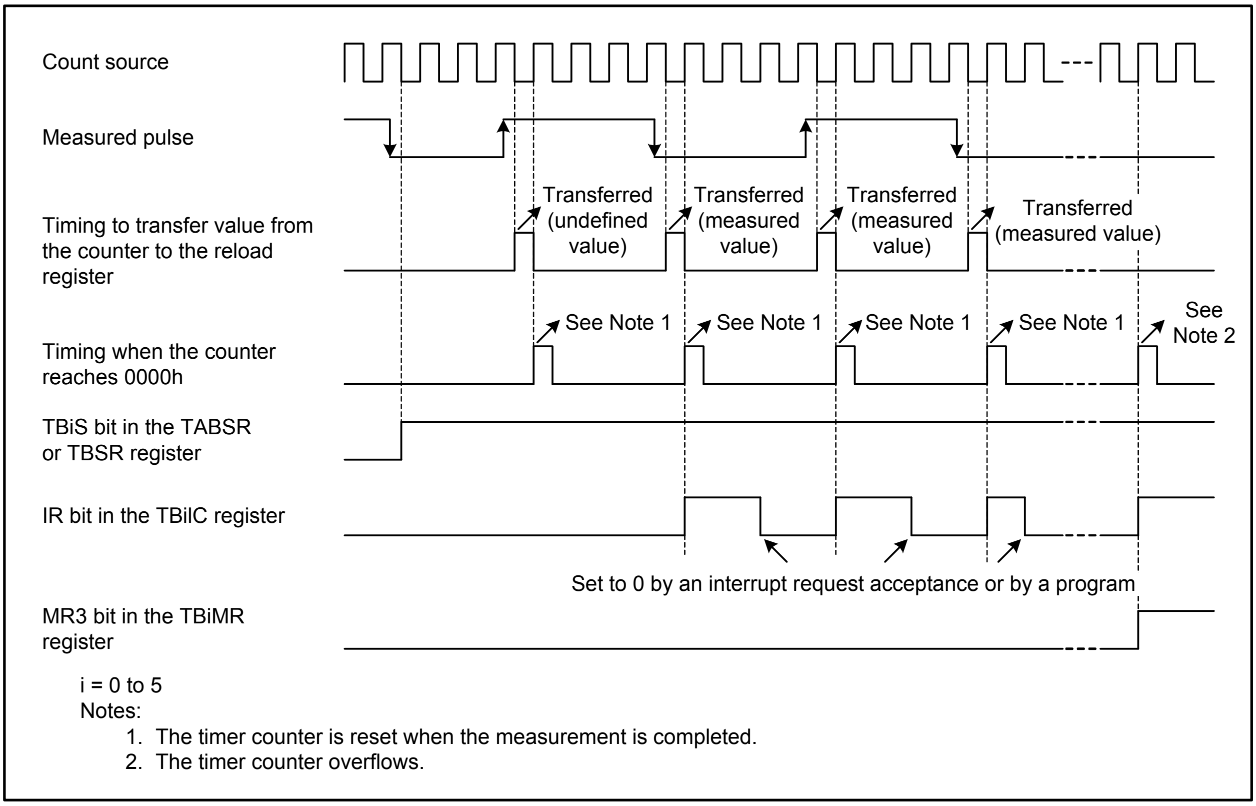

Count source
Measured pulse
Timing to transfer value from the counter to the reload register
Transferred (undefined value)
Transferred (measured value)
Timing when the counter reaches 0000h
See Note 1
See Note 1
See Note 2
TBiS bit in the TABSR or TBSR register
IR bit in the TBiIC register
Set to 0 by an interrupt request acceptance or by a program
MR3 bit in TBiMR register

i = 0 to 5

Notes:

1. The timer counter is reset when the measurement is completed.
2. The timer counter overflows.

Figure 16.27 Operation Example in Pulse Period Measurement


Count source
Measured pulse
Timing to transfer value from the counter to the reload register
Transferred (undefined value)
Transferred (measured value)
Transferred (measured value)
Transferred (measured value)
Timing when the counter reaches 0000h
See Note 1
See Note 1
See Note 1
See Note 1
See Note 2
TBiS bit in the TABSR or TBSR register
IR bit in the TBiIC register
Set to 0 by an interrupt request acceptance or by a program
MR3 bit in the TBiMR register

i = 0 to 5

Notes:

1. The timer counter is reset when the measurement is completed.
2. The timer counter overflows.

Figure 16.28 Operation Example in Pulse-width Measurement

16.3 Notes on Timers

16.3.1 Timer A and Timer B

All timers are stopped after a reset. To restart timers, configure parameters such as operating mode, count source, and counter value, then set the TAiS bit or TBjS bit in the TABSR or TBSR register to 1 (count starts) (i = 0 to 4; j = 0 to 5).

The following registers and bits should be set while the TAiS bit or TBjS bit is 0 (count stops):

- Registers TAiMR and TBjMR
- UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- TRGSR register

16.3.2 Timer A

16.3.2.1 Timer Mode

- While the timer counter is running, the TAi register indicates a counter value at any given time. However, FFFFh is read while reloading is in progress. A set value is read if the TAi register is set while the timer counter is stopped.

16.3.2.2 Event Counter Mode

- While the timer counter is running, the TAi register indicates a counter value at any given time. However, FFFFh is read if the timer counter underflows or 0000h if overflows while reloading is in progress. A set value is read if the TAi register is set while the timer counter is stopped.

16.3.2.3 One-shot Timer Mode

- If the TAiS bit in the TABSR register is set to 0 (count stops) while the timer counter is running, the following operations are performed:
  - The timer counter stops and the setting value of the TAi register is reloaded.
  - A low signal is output at the TAiOUT pin.
  - The IR bit in the TAiIC register becomes 1 (interrupts requested) after one CPU clock cycle.

- The one-shot timer is operated by an internal count source. When the trigger is an input to the TAiIN pin, the signal is output with a maximum one count source clock delay after a trigger input to the TAiIN pin.

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done:
  - Select one-shot timer mode after a reset.
  - Switch operating modes from timer mode to one-shot timer mode.
  - Switch operating modes from event counter mode to one-shot timer mode.

- If a retrigger occurs while counting, the timer counter decrements by one, reloads the setting value of the TAi register, and then continues counting. To generate a retrigger while counting, wait at least one count source cycle after the last trigger is generated.

- When an external trigger input is selected to start counting in timer A one-shot mode, do not provide an external retrigger for 300 ns before the timer counter reaches 0000h. Otherwise, it may stop counting.

16.3.2.4 Pulse-width Modulation Mode

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done (i = 0 to 4):
  - Select pulse-width modulation mode after a reset.
  - Switch operating modes from timer mode to pulse-width modulation mode.
  - Switch operating modes from event counter mode to pulse-width modulation mode.

- If the TAiS bit in the TABSR register is set to 0 (count stops) while PWM pulse is output, the following operations are performed:
  - The timer counter stops.
  - The output level at the TAiOUT pin changes from high to low. The IR bit becomes 1.
  - When a low signal is output at the TAiOUT pin, it does not change. The IR bit does not change, either.

16.3.3 Timer B

16.3.3.1 Timer Mode and Event Counter Mode

- While the timer counter is running, the TBj register indicates a counter value at any given time (j = 0 to 5). However, FFFFh is read while reloading is in progress. When a value is set to the TBj register while the timer counter is stopped, if the TBj register is read before the count starts, the set value is read.

16.3.3.2 Pulse Period/Pulse-width Measure Mode

- While the TBjS bit in the TABSR or TBSR register is 1 (start counter), after the MR3 bit becomes 1 (overflow) and at least one count source cycle has elapsed, a write operation to the TBjMR register sets the MR3 bit to 0 (no overflow).

- Use the IR bit in the TBjIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt handler.

- The counter value is undefined when the timer counter starts. Therefore, the timer counter may overflow before a measured pulse is applied on the initial valid edge and cause a timer Bj interrupt request to be generated.

- When the measured pulse is applied on the initial valid edge after the timer counter starts, an undefined value is transferred to the reload register. At this time, a timer Bj interrupt request is not generated.

- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBjMR register after the timer counter starts. However, if the same value is rewritten to bits MR1 and MR0, the IR bit does not change.

- Pulse width is continuously measured in pulse-width measure mode. Whether the measurement result is high-level width or not is determined by a program.

- When an overflow occurs at the same time a pulse is applied on the valid edge, this pulse is not recognized since an interrupt request is generated only once. Do not let an overflow occur in pulse period measure mode.

- In pulse-width measure mode, determine whether an interrupt source is a pulse applied on the valid edge or an overflow by reading the port level in the timer Bj interrupt handler.

17. Three-phase Motor Control Timers

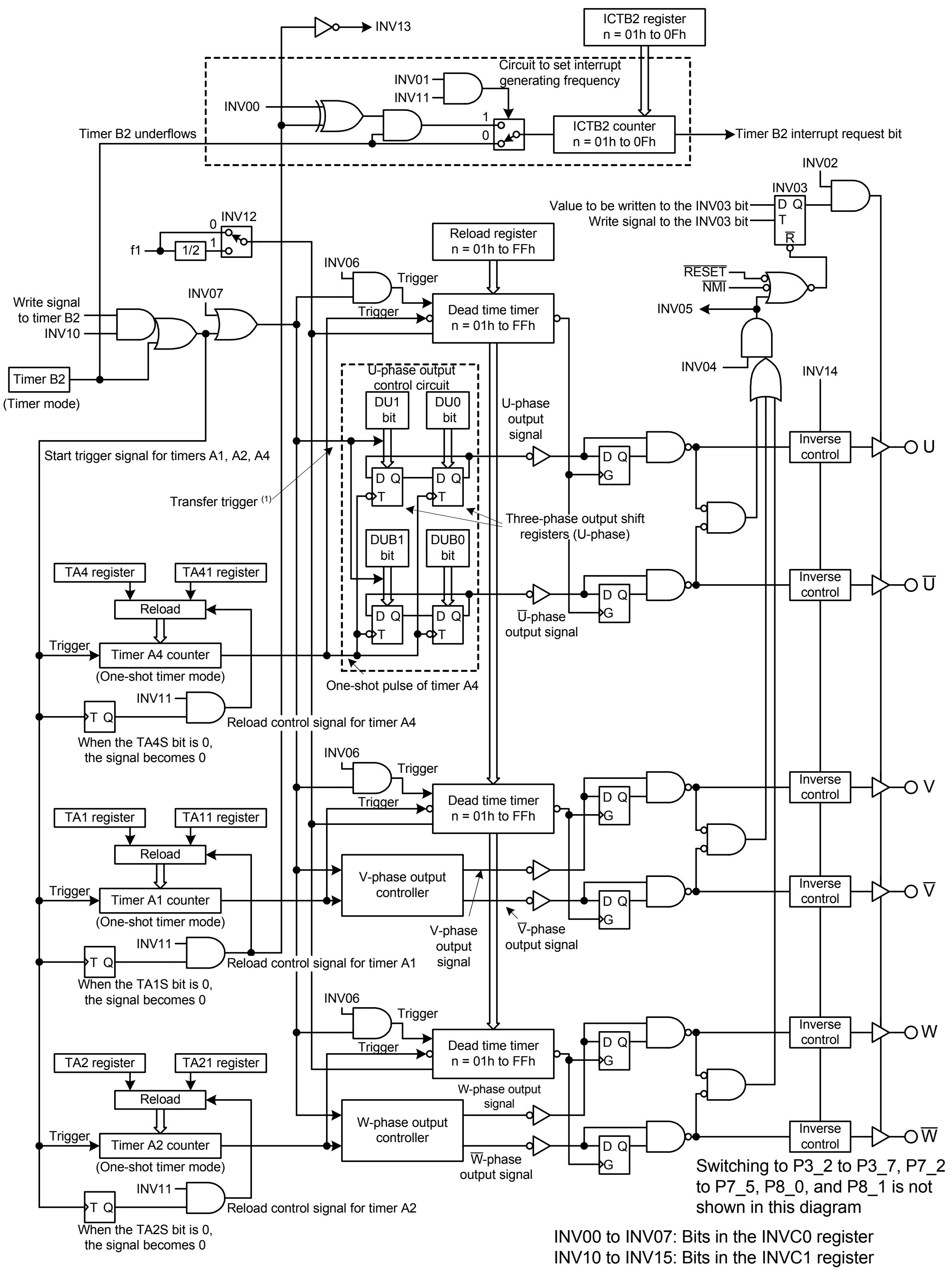
A three-phase motor driving waveform can be output using timers A1, A2, A4, and B2. The three-phase motor control timers are enabled by setting the INV02 bit in the INVC0 register to 1. Timer B2 is used for carrier wave control, and timers A1, A2, and A4 for three-phase PWM output (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$) control. Table 17.1 lists the specifications of the three-phase motor control timers and Figure 17.1 shows its block diagram. Figures 17.2 to 17.6 show registers associated with this function.

Table 17.1 Specifications for Three-phase Motor Control Timers

Item	Specification
Three-phase PWM waveform output pins	Six pins: U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$
Forced cutoff (1)	A low input to the $\overline{\text{NMI}}$ pin
Timers	Timers A4, A1, and A2 are used in one-shot timer mode: Timer A4 is used for U- and $\overline{U}$-phase waveform control Timer A1 is used for V- and $\overline{V}$-phase waveform control Timer A2 is used for W- and $\overline{W}$-phase waveform control Timer B2 is used in timer mode Carrier wave cycle control Dead time timer (three 8-bit timers share a reload register): Dead time control
Output waveforms	Triangular wave modulation and sawtooth wave modulation • Output of a high or a low waveform for one cycle • Separately settable levels of high side and low side
Carrier wave periods	Triangular wave modulation: count source × (m + 1) × 2 Sawtooth wave modulation: count source × (m + 1) m: TB2 register setting value from 0000h to FFFFh Count source: f1, f8, f2n, or fC32
Three-phase PWM output width	Triangular wave modulation: count source × n × 2 Sawtooth wave modulation: count source × n n: Setting value of registers TA4, TA1, and TA2 (registers TA4, TA41, TA1, TA11, TA2, and TA21 when the INV11 bit in the INVC1 register is 1) from 0001h to FFFFh Count source: f1, f8, f2n, or fC32
Dead time (width)	Count source × p or no dead time p: DTT register setting value from 01h to FFh Count source: f1 or f1 divided by 2
Active level	Selectable either active high or active low
Simultaneous conduction prevention	Function to detect simultaneous turn-on signal outputs, function to disable signal output when simultaneous turn-on signal outputs are detected
Interrupt frequency	Selectable from one through 15 time-carrier wave cycle-to-cycle basis for the timer B2 interrupt

Note:

1. Forced cutoff by a signal input to the $\overline{\text{NMI}}$ pin can be performed when the PM24 bit in the PM2 register is 1 (NMI enabled), the INV02 bit in the INVC0 register is 1 (three-phase motor control timers used), and the INV03 bit is 1 (three-phase motor control timer output enabled).



Switching to P3_2 to P3_7, P7_2 to P7_5, P8_0, and P8_1 is not shown in this diagram

INV00 to INV07: Bits in the INVC0 register
INV10 to INV15: Bits in the INVC1 register
DUi and DUBi: Bits in the IDBi register (i = 0, 1)
TA1S to TA4S: Bits in the TABSR register

Note:
1. When the INV06 bit is 0 (triangular wave modulation mode), the transfer trigger is generated only when the initial underflow of timer B2 occurs after setting registers IDB0 and IDB1.

Figure 17.1 Block Diagram for Three-phase Motor Control Timers

Three-phase PWM Control Register 0 [1]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
INVC0	0308h	0000 0000b

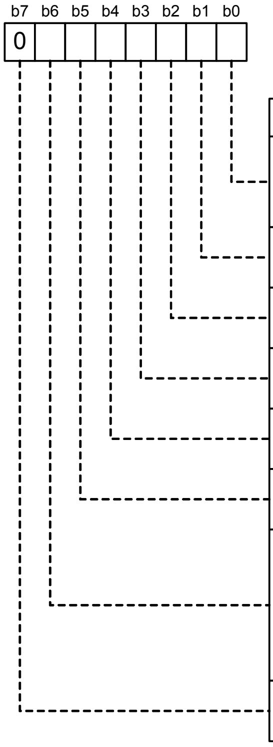
Bit Symbol	Bit Name	Function	RW
INV00	ICTB2 Count Condition Select Bit [2]	b1 b0 0 X : The underflow of timer B2 1 0 : The underflow of timer B2 when the reload control signal for timer A1 is 0 [3] 1 1 : The underflow of timer B2 when the reload control signal for timer A1 is 1 [3, 4]	RW
INV01			RW
INV02	Three-phase Motor Control Timers Select Bit	0: Do not use this function 1: Use this function [5, 6, 7]	RW
INV03	Three-phase Motor Control Timer Output Control Bit	0: Disables the three-phase motor control timer output [7] 1: Enables the three-phase motor control timer output [8]	RW
INV04	Simultaneous Conduction Prevention Bit	0: Ignores simultaneous turn-on signal output 1: Disables simultaneous turn-on signal output	RW
INV05	Simultaneous Conduction Detection Flag	0: Not detected 1: Detected [9]	RO
INV06	Modulation Mode Select Bit	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode [10]	RW
INV07	Software Trigger Select Bit	A transfer trigger is generated when this bit is set to 1. When the INV06 bit is 1, another trigger to the dead time timer is also generated. This bit is read as 0	RW

Notes:

1. Set this register after setting the PRC1 bit in the PRCR register to 1 (write enabled). Also, rewrite bits INV00 to INV02 and INV06 while timers A1, A2, A4, and B2 are stopped.
2. This bit is enabled when the INV11 bit in the INVC1 register is 1 (three-phase mode 1). When the INV11 bit is 0 (three-phase mode 0), the ICTB2 counter increments by one each time timer B2 underflows irrespective of the INV00 and INV01 bit settings.
3. Set the ICTB2 register before setting the INV01 bit to 1. Also, set the TA1S bit in the TABSR register to 1 before the initial timer B2 underflow occurs.
4. When the INV00 bit is 1, the first interrupt occurs when timer B2 underflows n-1 times (n is the value set in the ICTB2 counter). Subsequent interrupts occur every n times timer B2 underflows.
5. Set the INV02 bit to 1 to operate the dead time timer, U-, V-, and W-phase output control circuits, and the ICTB2 counter.
6. After setting the INV02 bit to 1, pins should be configured first by the IOBC register then by the output function select registers.
7. When the INV02 bit is set to 1 and the INV03 bit is set to 0, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$, even when they are assigned to other peripheral functions, become high-impedance.
8. The INV03 bit becomes 0 when any of the following occurs:
   - Reset
   - Signals of both the high and low sides are simultaneously switched to active when the INV04 bit is 1.
   - The INV03 bit is set to 0 by a program.
   - The $\overline{NMI}$ pin goes from high to low when the PM24 bit in the PM2 register is 1 (NMI enabled).
9. This bit cannot be set to 1 by a program. Set the INV04 bit to 0 to set this bit to 0.
10. When the INV06 bit is 1, set the INV11 bit in the INVC1 register to 0 (three-phase mode 0) and the PWCON bit in the TB2SC register to 0 (timer B2 register reloaded when timer B2 underflows).

Figure 17.2 INVC0 Register

Three-phase PWM Control Register 1 [1]



Symbol	Address	Reset Value
INVC1	0309h	0000 0000b

Bit Symbol	Bit Name	Function	RW
INV10	Timers A1, A2, and A4 Start Trigger Select Bit	0: The underflow of timer B2 1: The underflow of timer B2 and a write operation to the TB2 register	RW
INV11	Timers A1-1, A2-1, and A4-1 Control Bit	0: Three-phase mode 0 [2, 3] 1: Three-phase mode 1	RW
INV12	Dead Time Timer Count Source Select Bit	0: f1 1: f1 divided-by-2	RW
INV13	Carrier Wave Detection Flag [4]	0: Timer A1 reload control signal is 0 1: Timer A1 reload control signal is 1	RO
INV14	Active Level Control Bit	0: Active low output 1: Active high output	RW
INV15	Dead Time Disable Bit	0: Enables dead time 1: Disables dead time	RW
INV16	Dead Time Timer Trigger Select Bit	0: Falling edge of a one-shot pulse of timer (A4, A1, and A2) [5] 1: Rising edge of the three-phase output shift register (phases U, V, and W)	RW
— (b7)	Reserved	Should be written with 0	RW

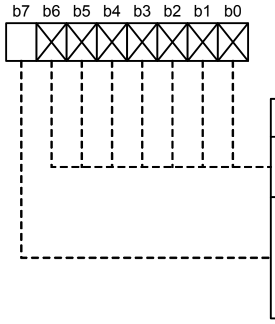
Notes:

1. Set this register after setting the PRC1 bit in the PRCR register to 1 (write enabled). Also, rewrite this register while timers A1, A2, A4, and B2 are stopped.
2. Set the INV11 bit to 0 when the INV06 bit in the INVC0 register is 1 (sawtooth wave modulation mode).
3. Set the PWCON bit in the TB2SC register to 0 (timer B2 register reloaded if timer B2 underflows) when the INV11 bit is 0.
4. This bit setting is enabled when the INV06 bit is 0 (triangular wave modulation mode) and the INV11 bit is 1.
5. Set the INV16 bit to 1 when the following conditions are all met:
   - The INV15 bit is 0.
   - The Dij bit has a different value from the DiBj bit whenever the INV03 bit is 1 (enables the three-phase motor control timer output); the high- and low-side output signals always have inverse levels on periods other than dead time (i = U, V, or W; j = 0, 1).

   Set the INV16 bit to 0 when the conditions above are not met.

Figure 17.3 INVC1 Register

Three-phase Output Buffer Control Register [1]



Symbol	Address	Reset Value
IOBC	40097h	0XXX XXXXb

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	No register bits; should be written with 0 and read as undefined value		—
TBSOUT	Three-phase Output Pin Select Bit [2]	0: Use pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of ports P7 and P8 1: Use pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of port P3	RW

Notes:

1. Set this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. Set this bit after setting the INV02 bit in the INVC0 register to 1. Then, set the output function select register of corresponding port. When the INV03 bit in the INVC0 register is 0, output pins for the three-phase motor control timers become high-impedance by the output enable control of output buffers. However, the output enable cannot be controlled only by the output function select register when more than two ports are assigned for output. Thus, a three-state output buffer should be selected using the TBSOUT bit.

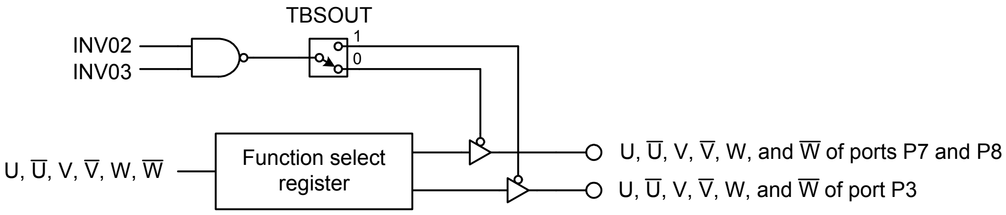
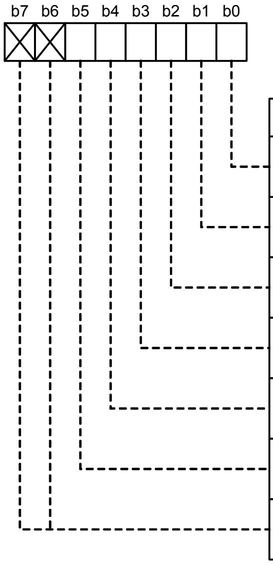

TBSOUT
INV02
INV03
1
0
U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$
Function select register
U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of ports P7 and P8
U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of port P3

Figure 17.4 IOBC Register

Three-phase Output Buffer Register i (i = 0, 1) [(1)]



Symbol: IDB0, IDB1 Address: 030Ah, 030Bh Reset Value: XX11 1111b

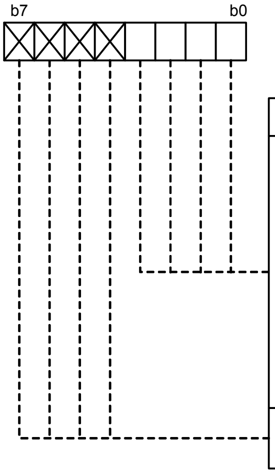
Bit Symbol	Bit Name	Function	RW
DUi	U-phase Output Buffer i	These bits should be written with an output level of the three-phase output shift register. The written value is reflected in each turn-on signal as follows: 0: Active (ON) 1: Inactive (OFF) The bits are read as the value of the three-phase output shift register	RW
DUBi	$\overline{U}$-phase Output Buffer i		RW
DVi	V-phase Output Buffer i		RW
DVBi	$\overline{V}$-phase Output Buffer i		RW
DWi	W-phase Output Buffer i		RW
DWBi	$\overline{W}$-phase Output Buffer i		RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. Values of registers IDB0 and IDB1 are transferred to the three-phase output shift register by a transfer trigger. The initial output signal level of each phase is determined by the value written in the IDB0 register after the transfer trigger occurs. Then the output signal level is determined by the value written in the IDB1 register on the falling edge of a one-shot pulse of timers A1, A2, and A4.

Figure 17.5 Registers IDB0 and IDB1

Timer B2 Interrupt Generating Frequency Set Counter (1, 2, 3)



Symbol	Address	Reset Value
ICTB2	030Dh	Undefined

Function	Setting Range	RW
- When the INV01 bit is 0 (ICTB2 counter increments each time timer B2 underflows), a timer B2 interrupt request is generated every nth times timer B2 underflows - When the INV01 bit is 1 (ICTB2 counter increments when the timer A1 reload control signal is set to 0 or 1 and timer B2 underflows), a timer B2 interrupt request is generated every nth times timer B2 underflows when the timer A1 reload control signal is 0 or 1 (n = setting value)	01h to 0Fh	WO
No register bits; should be written with 0		—

Notes:

1. Use the MOV instruction to set the ICTB2 register.
2. When the INV01 bit in the INVC0 register is 1, set this register while the TB2S bit in the TABSR register is 0 (timer B2 count stops). Although it can be set even when the TB2S bit is 1 (timer B2 count starts) when the INV01 bit is 0, do not set this register when timer B2 underflows.
3. When the INV00 bit in the INVC0 register is set to 1, the first interrupt occurs when timer B2 underflows n-1 times. Subsequent interrupts occur every n times timer B2 underflows.
   (n = setting value of the ICTB2 counter)

Figure 17.6 ICTB2 Register

17.1 Modulation Modes of Three-phase Motor Control Timers

The three-phase motor control timers support two modulation modes: triangular wave modulation mode and sawtooth wave modulation mode. The triangular wave modulation mode has two modes: three-phase mode 0 and three-phase mode 1. Table 17.2 lists bit settings and characteristics of each mode.

Table 17.2 Modulation Modes

Item	Triangular Wave Modulation Mode		Sawtooth Wave Modulation Mode
	Three-phase mode 0	Three-phase mode 1	(Three-phase mode 0)
Bit settings	INV06 is 0, INV11 is 0, PWCON is 0	INV06 is 0, INV11 is 1	INV06 is 1, INV11 is 0, PWCON is 0
Waveform	Triangular wave		Sawtooth wave
Registers TA11, TA21, and TA41	Not used	Used	Not used
Timing to transfer data from registers IDB0 and IDB1 to the three-phase output shift register	Only once when a transfer trigger [1] occurs after setting registers IDB0 and IDB1		Whenever a transfer trigger [1] occurs
Timing to trigger the dead time timer when the INV16 bit is 0	On the falling edge of a one-shot pulse of timers A1, A2, and A4		When a transfer trigger occurs, or on the falling edge of a one-shot pulse of timers A1, A2, and A4
Bits INV00 and INV01 in the INVC0 register	Disabled. The ICTB2 counter increments each time timer B2 underflows, irrespective of the INV00 and INV01 bit settings	Enabled	Disabled. The ICTB2 counter increments each time timer B2 underflows, irrespective of the INV00 and INV01 bit settings
INV13 bit	Disabled	Enabled	Disabled

Note:

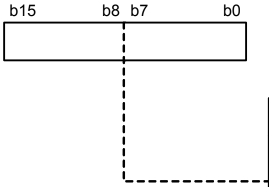
1. The transfer trigger is a timer B2 underflow, a write operation to the INV07 bit, or a write operation to the TB2 register when the INV10 bit is 1.

17.2 Timer B2

Timer B2, which operates in timer mode, is used for carrier wave control in the three-phase motor control timers.

Figures 17.7 and 17.8 show registers TB2 and TB2MR in this function, respectively. Figure 17.9 shows the TB2SC register which switches timing to change the carrier wave frequency in three-phase mode 1.

Timer B2 Register [1]



Symbol	Address	Reset Value
TB2	0355h-0354h	Undefined

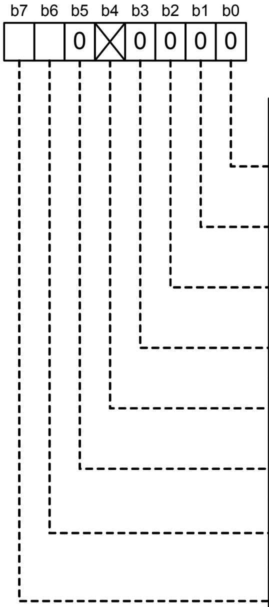
Function	Setting Range	RW
Divides the count source by n+1. Starts timers A1, A2, and A4 each time an underflow occurs (n = setting value)	0000h to FFFFh	RW

Note:

1. A 16-bit read/write access to this register should be performed.

Figure 17.7 TB2 Register When Using Three-phase Motor Control Timers

Timer B2 Mode Register

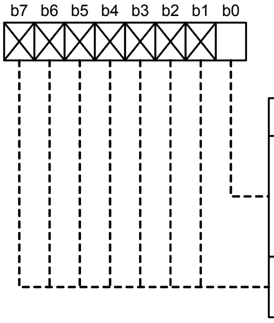


Symbol	Address	Reset Value
TB2MR	035Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	Should be written with 00b (timer mode) when using the three-phase motor control timers	RW
TMOD1			RW
MR0	Disabled when using the three-phase motor control timers. Should be written with 0 and read as undefined value		RW
MR1			RW
MR2	No register bit; should be written with 0 and read as undefined value		—
MR3	Disabled when using the three-phase motor control timers. Should be written with 0 and read as undefined value		—
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Figure 17.8 TB2MR Register When Using Three-phase Motor Control Timers

Timer B2 Special Mode Register



Symbol	Address	Reset Value
TB2SC	035Eh	XXXX XXX0b

Bit Symbol	Bit Name	Function	RW
PWCON	Timer B2 Reload Timing Switching Bit [1]	0: The underflow of timer B2 1: The underflow of timer B2 when the reload control signal for timer A1 is 0	RW
— (b7-b1)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. Set this bit to 0 when the INV11 bit is 0 (three-phase mode 0) or the INV06 bit is 1 (sawtooth wave modulation mode).

Figure 17.9 TB2SC Register

17.3 Timers A4, A1, and A2

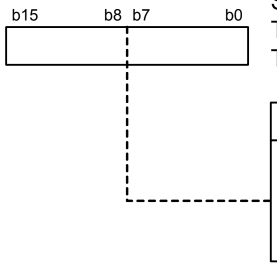
Timers A4, A1, and A2 are used for three-phase PWM output (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$) control when using the three-phase motor control timers.

These timers should be operated in one-shot timer mode. Every time timer B2 underflows, a trigger is input to timers A4, A1, and A2 to generate a one-shot pulse. If the values of registers TA4, TA1, and TA2 are rewritten every time a timer B2 interrupt occurs, the duty cycle of the PWM waveform can be varied.

In three-phase mode 1, the value of registers TAi and TAi-1 is alternately reloaded to the counter at each timer B2 interrupt, which halves the timer B2 interrupt frequency (i = 4, 1, 2).

Figure 17.10 shows registers TA1, TA2, TA4, TA11, TA21, and TA41 in the three-phase motor control timers. Figure 17.11 shows registers TA1MR, TA2MR, and TA4MR in this function. Figures 17.12 and 17.13 show registers TRGSR and TABSR, respectively, in this function.

Timer Ai/Timer Ai-1 Registers (i = 1, 2, 4) (1 to 6)



Symbol	Address	Reset Value
TA1, TA2, TA4	0349h-0348h, 034Bh-034Ah, 034Fh-034Eh	Undefined
TA11, TA21, TA41	0303h-0302h, 0305h-0304h, 0307h-0306h	Undefined

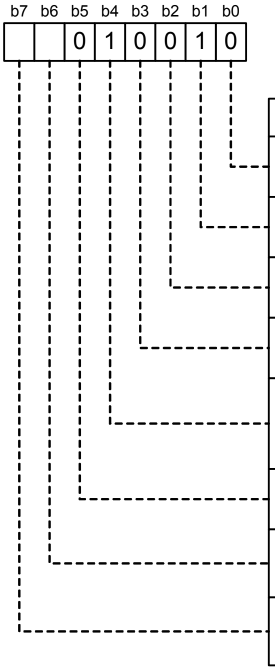
Function	Setting Range	RW
The timer stops when the nth count source is counted after a start trigger is generated. The output signal for each phase is switched when timers A1, A2, and A4 stop (n = setting value)	0000h to FFFFh	WO

Notes:

1. A 16-bit write access to these registers should be performed.
2. When these registers are set to 0000h, the counter does not start, and no timer Ai interrupt request is generated.
3. Use the MOV instruction to set these registers.
4. When the INV15 bit in the INVC1 register is 0 (enables dead time), the turn-on output signal is switched to its active state with a delay. It switches when the dead time timer stops.
5. When the INV11 bit in the INVC1 register is 0 (three-phase mode 0), the value of the TAi register is transferred to the reload register by a timer Ai start trigger. When the INV11 bit is 1 (three-phase mode 1), first the value of the TAi1 register is transferred to the reload register by a timer Ai start trigger. Then the value of the TAi register is transferred by the next timer Ai start trigger. After that, the values of registers TAi1 and TAi are alternately transferred to the reload register.
6. These registers should not be written when timer B2 underflows.

Figure 17.10 Registers TA1, TA2, TA4, TA11, TA21, and TA41

Timer Ai Mode Register (i = 1, 2, 4)

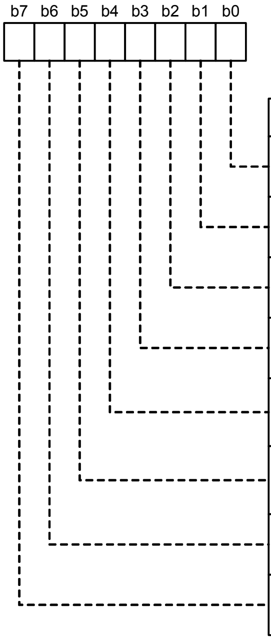


Symbol	Address	Reset Value
TA1MR, TA2MR, TA4MR	0357h, 0358h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	Should be written with 10b (one-shot timer mode) when using the three-phase motor control timers	RW
TMOD1			RW
MR0	Reserved	Should be written with 0	RW
MR1	External Trigger Select Bit	Should be written with 0 when using the three-phase motor control timers	RW
MR2	Trigger Select Bit	Should be written with 1 (selected by the TRGSR register) when using the three-phase motor control timers	RW
MR3	Should be written with 0 when using the three-phase motor control timers		RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Figure 17.11 Registers TA1MR, TA2MR, and TA4MR When Using Three-phase Motor Control Timers

Trigger Select Register



Symbol TRGSR Address 0343h Reset Value 0000 0000b

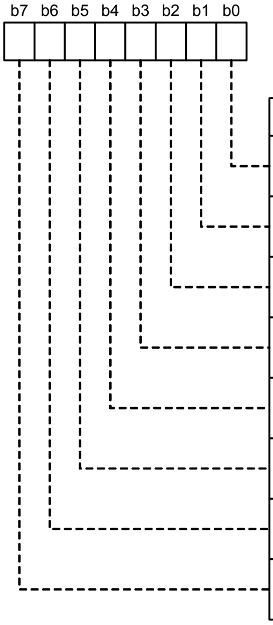
Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 Event/Trigger Select Bit	Should be set to 01b (the underflow of TB2) to use the V-phase output control circuit	RW
TA1TGH			RW
TA2TGL	Timer A2 Event/Trigger Select Bit	Should be set to 01b (the underflow of TB2) to the use W-phase output control circuit	RW
TA2TGH			RW
TA3TGL	Timer A3 Event/Trigger Select Bit	b5 b4 0 0 : Select the input to the TA3IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA2 (1) 1 1 : Select the overflow of TA4 (1)	RW
TA3TGH			RW
TA4TGL	Timer A4 Event/Trigger Select Bit	Should be set to 01b (the underflow of TB2) to the use U-phase output control circuit	RW
TA4TGH			RW

Note:

1. The timer overflows or underflows.

Figure 17.12 TRGSR Register in Three-phase Motor Control Timers

Count Start Register



Symbol TABSR Address 0340h Reset Value 0000 0000b

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 Count Start Bit	0: Stop counter 1: Start counter	RW
TA1S	Timer A1 Count Start Bit	0: Stop counter 1: Start counter	RW
TA2S	Timer A2 Count Start Bit	0: Stop counter 1: Start counter	RW
TA3S	Timer A3 Count Start Bit	0: Stop counter 1: Start counter	RW
TA4S	Timer A4 Count Start Bit	0: Stop counter 1: Start counter	RW
TB0S	Timer B0 Count Start Bit	0: Stop counter 1: Start counter	RW
TB1S	Timer B1 Count Start Bit	0: Stop counter 1: Start counter	RW
TB2S	Timer B2 Count Start Bit	0: Stop counter 1: Start counter	RW

Figure 17.13 TABSR Register

17.4 Simultaneous Conduction Prevention and Dead Time Timer

The three-phase motor control timers offer two ways to avoid shoot-through, which occurs when high-side and low-side transistors are simultaneously turned on.

One is “simultaneous turn-on signal output disable function”. This function prevents high-side and low-side transistors from being inadvertently switched to active due to events like program errors. The other is by the use of dead time timers. A dead time timer delays the turn-on of one transistor in order to ensure that an adequate time (the dead time) passes after the other is turned off.

To disable simultaneous turn-on output signals, the INV04 bit in the INVC0 register should be set to 1. If outputs for any pair of phases (U and $\overline{U}$, V and $\overline{V}$, or W and $\overline{W}$) are simultaneously switched to an active state, every three-phase motor control output pin becomes high-impedance. Figure 17.14 shows an example of output waveform when simultaneous turn-on signal output is disabled.

To enable the dead time timer, the INV15 bit in the INVC1 register should be set to 0. The DTT register determines the dead time. Figure 17.15 shows the DTT register and Figure 17.16 shows an example of output waveform on using dead time timer.

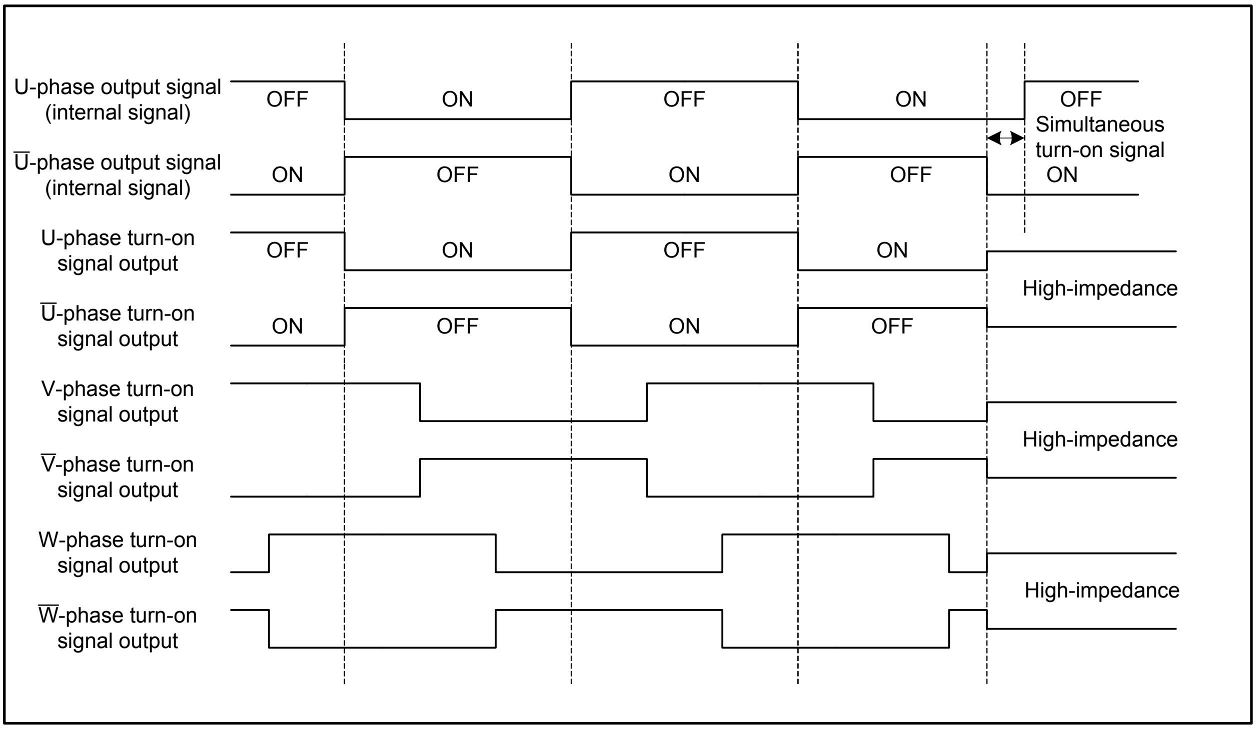
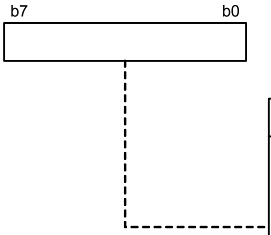

U-phase output signal (internal signal)
OFF
ON
OFF
ON
OFF
Simultaneous turn-on signal
$\overline{U}$-phase output signal (internal signal)
ON
OFF
ON
OFF
ON
U-phase turn-on signal output
OFF
ON
OFF
ON
High-impedance
$\overline{U}$-phase turn-on signal output
ON
OFF
ON
OFF
V-phase turn-on signal output
High-impedance
$\overline{V}$-phase turn-on signal output
W-phase turn-on signal output
High-impedance
$\overline{W}$-phase turn-on signal output

Figure 17.14 Output Waveform When Simultaneous Turn-on Signal Output is Disabled

Dead Time Timer [1, 2]



Symbol
DTT

Symbol	Address	Reset Value
DTT	030Ch	Undefined

Function	Setting Range	RW
The dead time timer is a one-shot timer that delays the timing for a turn-on signal to be switched to its active state preventing a simultaneous conduction of high-side and low-side transistors. The timer stops when counting a count source n times after a start trigger occurs (n = setting value) [3]	01h to FFh	WO

Notes:

1. Use the MOV instruction to set this register.
2. This register setting is enabled when the INV15 bit in the INVC1 register is 0 (enables dead time). No dead time can be set when the INV15 bit is 1 (disables dead time).
3. The trigger and count source should be selected using bits INV16 and INV12 in the INVC1 register, respectively.

Figure 17.15 DTT Register

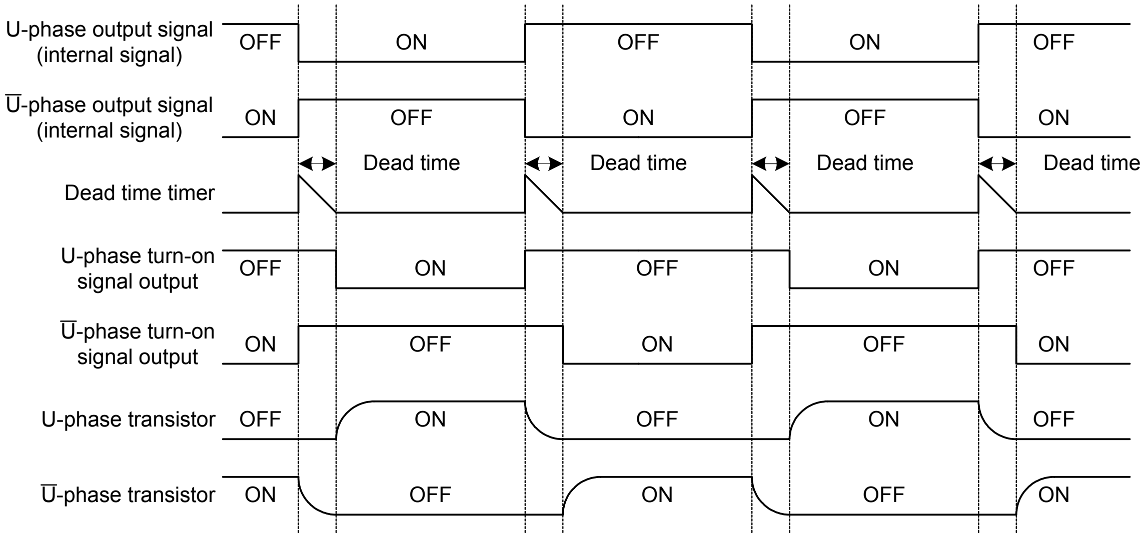
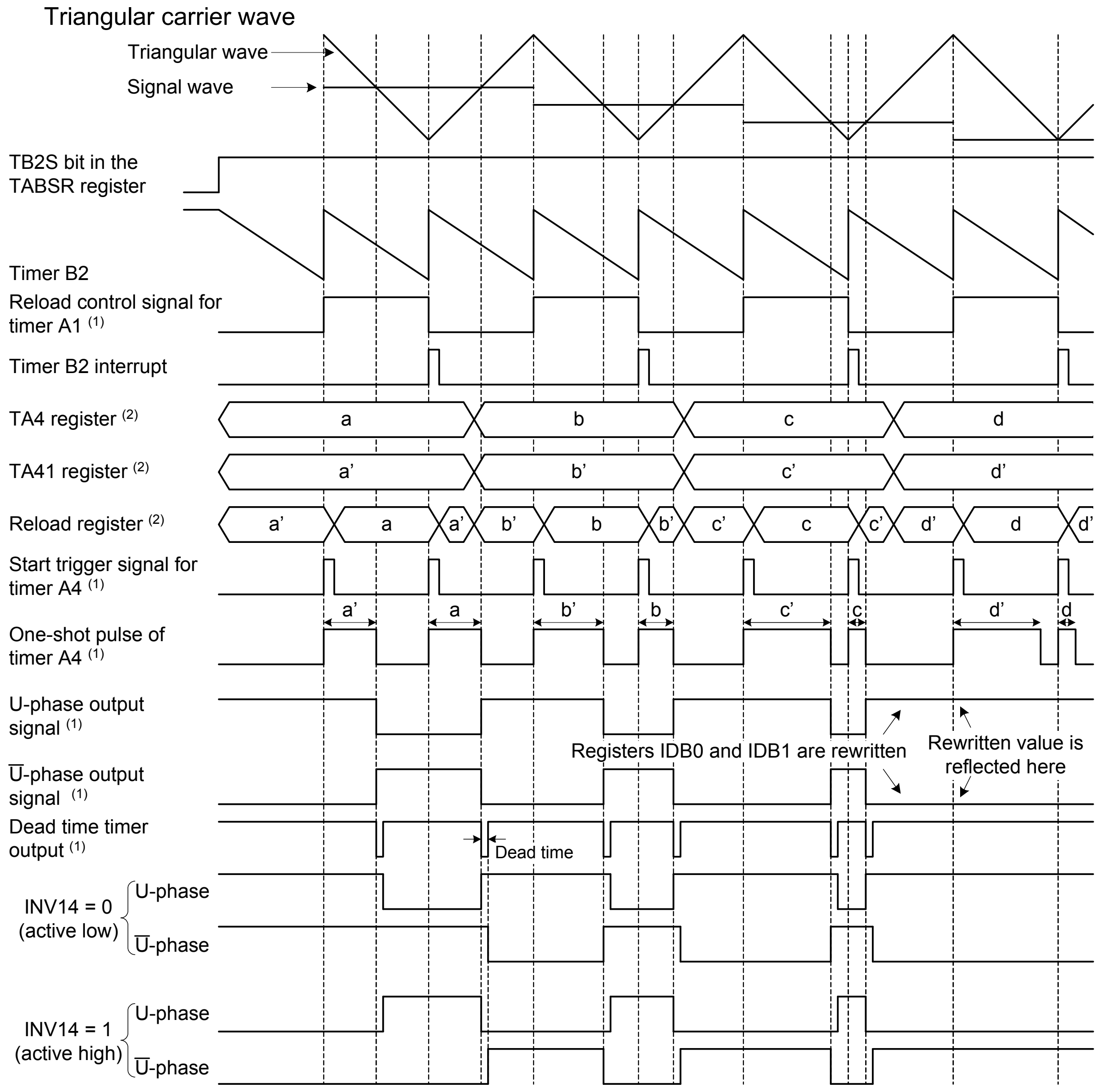

U-phase output signal (internal signal)
OFF
ON
OFF
ON
OFF
$\overline{U}$-phase output signal (internal signal)
ON
OFF
ON
OFF
ON
Dead time
Dead time
Dead time
Dead time
Dead time timer
U-phase turn-on signal output
OFF
ON
OFF
ON
OFF
$\overline{U}$-phase turn-on signal output
ON
OFF
ON
OFF
ON
U-phase transistor
OFF
ON
OFF
ON
OFF
$\overline{U}$-phase transistor
ON
OFF
ON
OFF
ON

Figure 17.16 Output Waveform When Using Dead Time Timer

17.5 Three-phase Motor Control Timer Operation

Figures 17.17 and 17.18 show an operation example of triangular wave modulation and sawtooth wave modulation, respectively.


Triangular carrier wave
Triangular wave
Signal wave
TB2S bit in the TABSR register
Timer B2
Reload control signal for timer A1 (1)
Timer B2 interrupt
TA4 register (2)
a
b
c
d
TA41 register (2)
a’
b’
c’
d’
Reload register (2)
a’
a
a’
b’
b
b’
c’
c
c’
d’
d
d’
Start trigger signal for timer A4 (1)
a’
a
b’
b
c’
c
d’
d
One-shot pulse of timer A4 (1)
U-phase output signal (1)
Registers IDB0 and IDB1 are rewritten
Rewritten value is reflected here
U̅-phase output signal (1)
Dead time timer output (1)
Dead time
INV14 = 0 (active low)
U-phase
U̅-phase
INV14 = 1 (active high)
U-phase
U̅-phase

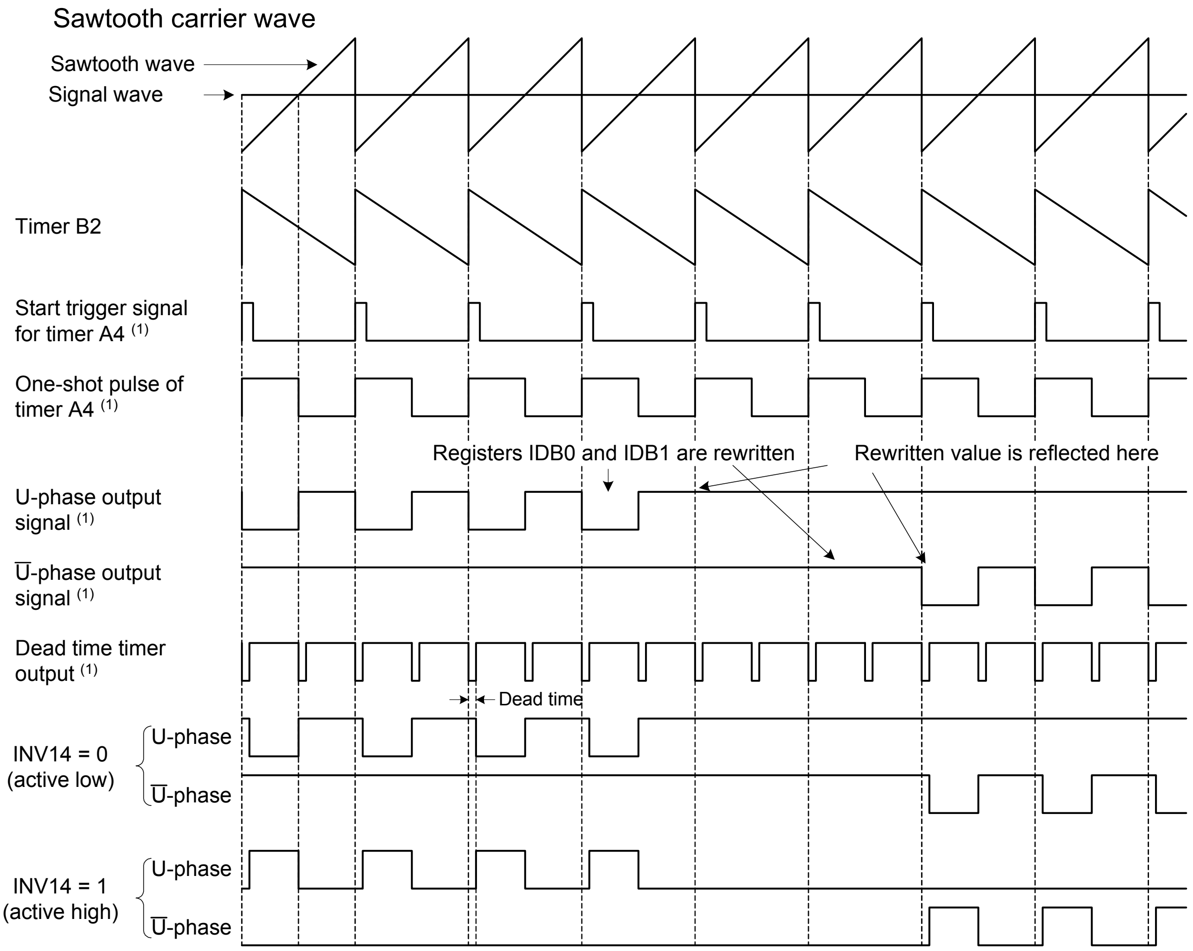
This figure applies when INVC0 = 00XX11XXb (X varies depending on each system) and INVC1 = 010XXXX0b.
PWM output may vary as follows:

(A) When INV11 = 1 (three-phase mode 1)
- INV01 = 0 and ICTB2 = 2h (timer B2 interrupt occurs every second time timer B2 underflows), or
  INV01 = 1, INV00 = 1, and ICTB2 = 1h (timer B2 interrupt occurs every time timer B2 underflows when the reload control signal for timer A1 is 1)
- The setting of registers TA4 and TA41 are varied every time a timer B2 interrupt occurs,
  Default value: TA41 = a’, TA4 = a
  On the first timer B2 interrupt: TA41 = b’, TA4 = b; the second time: TA41 = c’, TA4 = c
- Default value of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1
  On the third time: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 0

(B) When INV11 = 0 (three-phase mode 0)
- INV01 = 0 and ICTB2 = 1h (timer B2 interrupt occurs every time timer B2 underflows)
- TA4 register setting is varied every time a timer B2 interrupt occurs,
  Default value: TA4 = a’
  On the first timer B2 interrupt: TA4 = a; 2nd time: TA4 = b’; 3rd time: TA4 = b; 4th time: TA4 = c’; 5th time: TA4 = c
- Default value of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1
  On the sixth time: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 0

Notes:
1. Internal signal. Refer to the block diagram of three-phase motor control timers.
2. Applicable when the INV11 bit in the INVC1 register is 1 (three-phase mode 1).

Figure 17.17 Triangular Wave Modulation Operation


Sawtooth carrier wave
Sawtooth wave
Signal wave
Timer B2
Start trigger signal for timer A4 (1)
One-shot pulse of timer A4 (1)
Registers IDB0 and IDB1 are rewritten
Rewritten value is reflected here
U-phase output signal (1)
U̅-phase output signal (1)
Dead time timer output (1)
Dead time
INV14 = 0 (active low)
U-phase
U̅-phase
INV14 = 1 (active high)
U-phase
U̅-phase

This figure applies when INVC0 = 01XX110Xb (X varies depending on each system) and INVC1 = 000XXX00b.
This bit setting is applicable to turn-on control with a phase shift of 120 degrees.

PWM output may vary as follows:

Default value of registers IDB0 and IDB1: DU0 = 0, DUB0 = 1, DU1 = 1, DUB1 = 1
On the third timer B2 interrupt: DU0 = 1, DUB0 = 1, DU1 = 1, DUB1 = 1
On the fifth timer B2 interrupt: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 1

Note:

1. Internal signal. Refer to the block diagram of three-phase motor control timers.

Figure 17.18 Sawtooth Wave Modulation Operation

17.6 Notes on Three-phase Motor Control Timers

17.6.1 Shutdown

- When a low signal is applied to the $\overline{\text{NMI}}$ pin with the following bit settings, pins TA1OUT, TA2OUT, and TA4OUT become high-impedance: the PM24 bit in the PM2 register is 1 (NMI enabled), the INV02 bit in the INVC0 register is 1 (three-phase motor control timers used), and the INV03 bit is 1 (three-phase motor control timer output enabled).

17.6.2 Register Setting

- Do not write to the TAi1 register before and after timer B2 underflows (i = 1, 2, 4). Before writing to the TAi1 register, read the TB2 register to verify that sufficient time remains until timer B2 underflows. Then, immediately write to the TAi1 register so no interrupt handling is performed during this write procedure. If the TB2 register indicates little time remains until the underflow, write to the TAi1 register after timer B2 underflows.

18. Serial Interface

The serial interface consists of 11 channels: UART0 to UART10.
Each channel has an exclusive timer to generate the transmit/receive clock and operates independently.
Figures 18.1 and 18.2 show block diagrams of UART0 to UART6 and UART7 to UART10, respectively.
UARTi supports the following modes:

- Synchronous serial interface mode (for UART0 to UART10)
- Asynchronous serial interface mode (UART mode) (for UART0 to UART10)
- Special mode 1 (I^{2}C mode) (for UART0 to UART6)
- Special mode 2 (for UART0 to UART6)
- Special mode 4 (Bus collision detection: IE mode) (optional) [1] (for UART0 to UART6)

Figures 18.3 to 18.20 show registers associated with UARTi (i = 0 to 10).
Refer to the tables listing each mode for registers and pin settings.

Note:
1. Contact a Renesas Electronics sales office to use the optional features.

Table 18.1 Comparison of UART0 to UART10 Functions

Mode/Function		UART0 to UART6	UART7 to UART10
Synchronous serial interface mode		Available	Available
	Serial data logic inversion	Available	Not available
UART mode		Available	Available
	CTS/RTS function selection	Available	Available
	TXD and RXD I/O polarity selection	Available	Not available
Special mode 1 (I^{2}C mode)		Available	Not available
Special mode 2		Available	Not available
Special mode 4 (IE mode) (optional) [1]		Available	Not available
Pins TXD and RXD output mode		Push-pull output, N-channel open drain output programmable by port function select registers	Push-pull output, N-channel open drain output programmable by port function select registers

Note:
1. Contact a Renesas Electronics sales office to use the optional features.

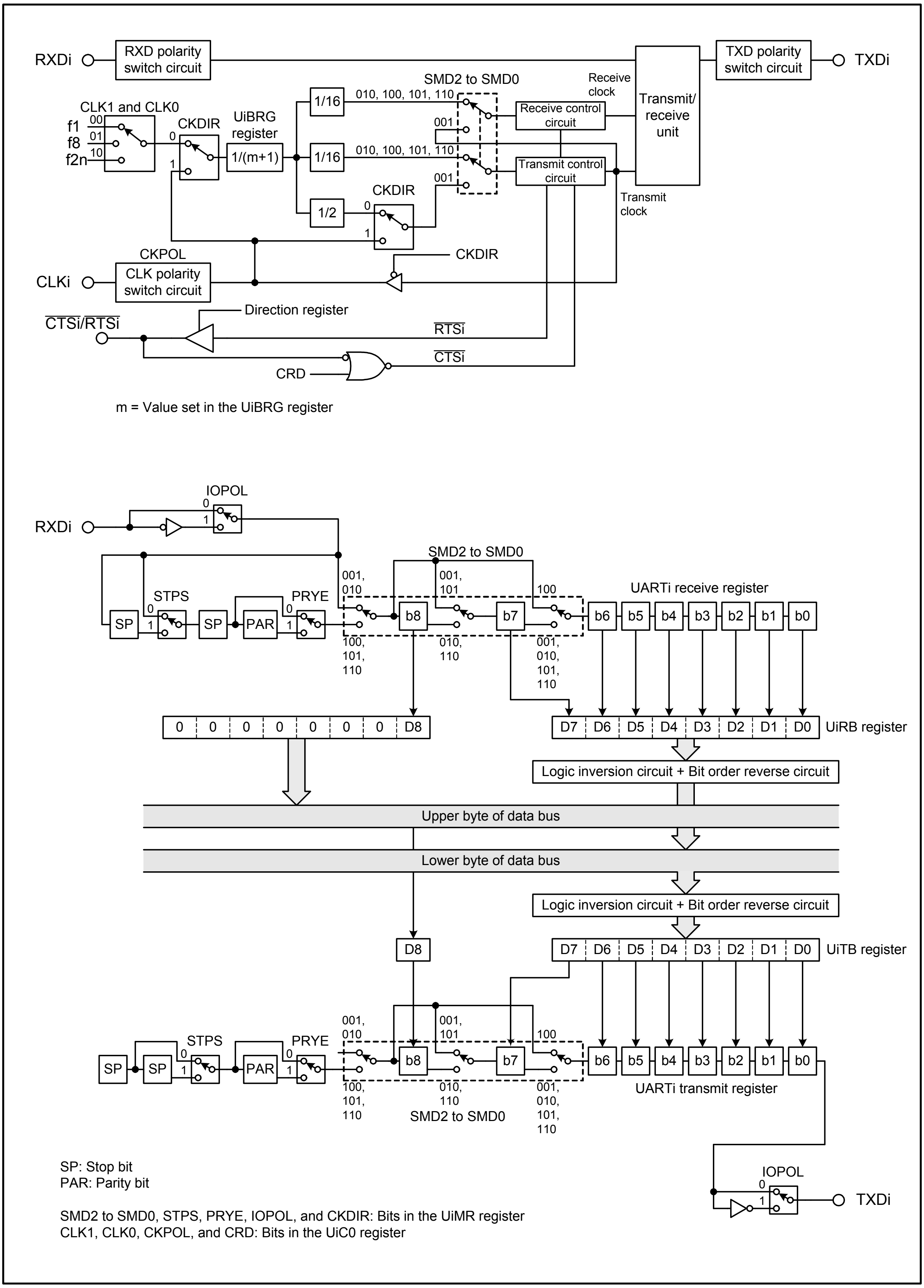


Figure 18.1 UARTi Block Diagram (i = 0 to 6)

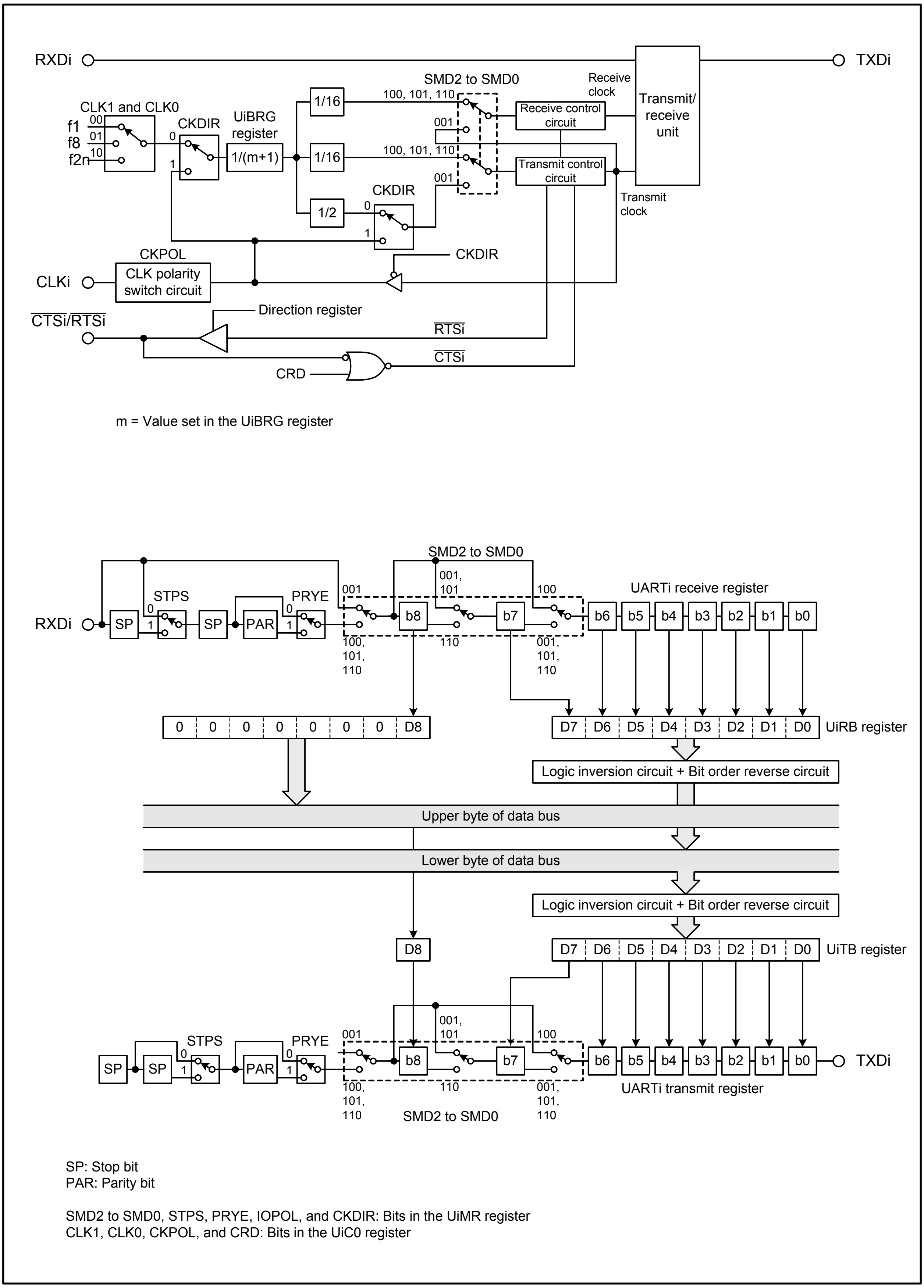
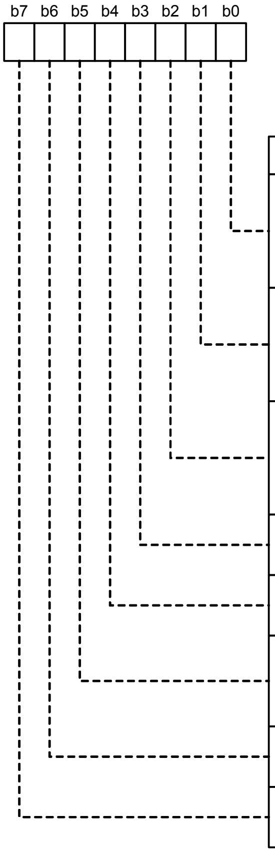


Figure 18.2 UARTi Block Diagram (i = 7 to 10)

UARTi Transmit/Receive Mode Register (i = 0 to 6)

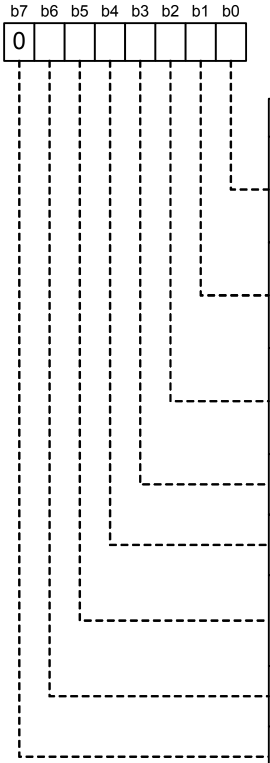


Symbol	Address	Reset Value
U0MR to U3MR	0368h, 02E8h, 0338h, 0328h	0000 0000b
U4MR to U6MR	02F8h, 01C8h, 01D8h	0000 0000b

Bit Symbol	Bit Name	Function	RW
SMD0	Serial Interface Mode Select Bit	b2 b1 b0 0 0 0 : Serial interface disabled 0 0 1 : Synchronous serial interface mode 0 1 0 : I^{2}C mode 1 0 0 : UART mode, 7-bit character length 1 0 1 : UART mode, 8-bit character length 1 1 0 : UART mode, 9-bit character length Only use the combinations listed above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/External Clock Select Bit	0: Internal clock 1: External clock	RW
STPS	Stop Bit Length Select Bit	0: 1 stop bit 1: 2 stop bits	RW
PRY	Odd/Even Parity Select Bit	Enabled when the PRYE bit is 1 0: Odd parity 1: Even parity	RW
PRYE	Parity Enable Bit	0: Parity disabled 1: Parity enabled	RW
IOPOL	TXD, RXD Input/Output Polarity Switch Bit	0: Not inverted 1: Inverted	RW

Figure 18.3 Registers U0MR to U6MR

UARTi Transmit/Receive Mode Register (i = 7 to 10)



Symbol
U7MR to U10MR

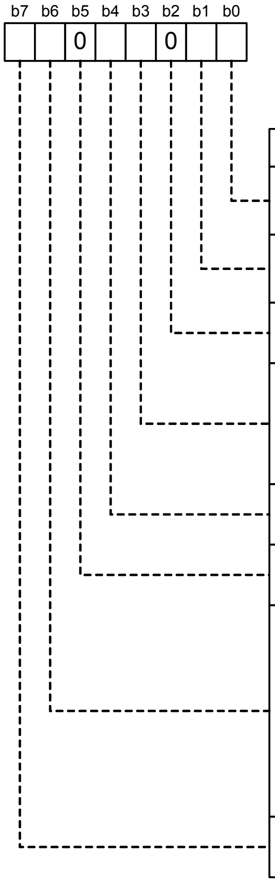
Address
01E0h, 01E8h, 40300h, 40308h

Reset Value
0000 0000b

Bit Symbol	Bit Name	Function	RW
SMD0	Serial Interface Mode Select Bit	b2 b1 b0 0 0 0 : Serial interface disabled 0 0 1 : Synchronous serial interface mode 1 0 0 : UART mode, 7-bit character length 1 0 1 : UART mode, 8-bit character length 1 1 0 : UART mode, 9-bit character length Only use the combinations listed above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/External Clock Select Bit	0: Internal clock 1: External clock	RW
STPS	Stop Bit Length Select Bit	0: 1 stop bit 1: 2 stop bits	RW
PRY	Odd/Even Parity Select Bit	Enabled when the PRYE bit is 1 0: Odd parity 1: Even parity	RW
PRYE	Parity Enable Bit	0: Parity disabled 1: Parity enabled	RW
— (b7)	Reserved	Should be written with 0	RW

Figure 18.4 Registers U7MR to U10MR

UARTi Transmit/Receive Control Register 0 (i = 0 to 6)



Symbol	Address	Reset Value
U0C0 to U3C0	036Ch, 02ECh, 033Ch, 032Ch	0000 1000b
U4C0 to U6C0	02FCh, 01CCh, 01DCh	0000 1000b

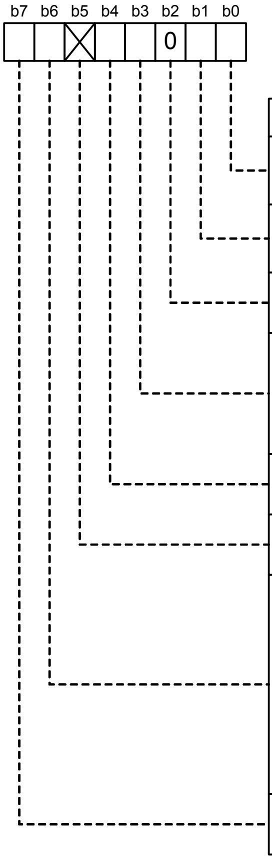
Bit Symbol	Bit Name	Function	RW
CLK0	UiBRG Count Source Select Bit	b1 b0 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : Do not use this combination	RW
CLK1			RW
— (b2)	Reserved	Should be written with 0	RW
TXEPT	Transmit Shift Register Empty Flag	0: Data held in the transmit shift register (transmission in progress) 1: No data held in the transmit shift register (transmission completed)	RO
CRD	CTS Function Disable Bit	0: CTS function enabled 1: CTS function disabled	RW
— (b5)	Reserved	Should be written with 0	RW
CKPOL	CLK Polarity Select Bit	0: Output transmit data on the falling edge of the transmit/receive clock and input receive data on the rising edge 1: Output transmit data on the rising edge of the transmit/receive clock and input receive data on the falling edge	RW
UFORM	Bit Order Select Bit [1]	0: LSB first 1: MSB first	RW

Note:

1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode) or 101b (UART mode, 8-bit character length). It should be set to 1 when bits SMD2 to SMD0 are set to 010b (I^2C mode) and should be set to 0 when they are set to 100b (UART mode, 7-bit character length) or 110b (UART mode, 9-bit character length).

Figure 18.5 Registers U0C0 to U6C0

UARTi Transmit/Receive Control Register 0 (i = 7 to 10)



Symbol	Address	Reset Value
U7C0 to U10C0	01E4h, 01ECh, 40304h, 4030Ch	00X0 1000b

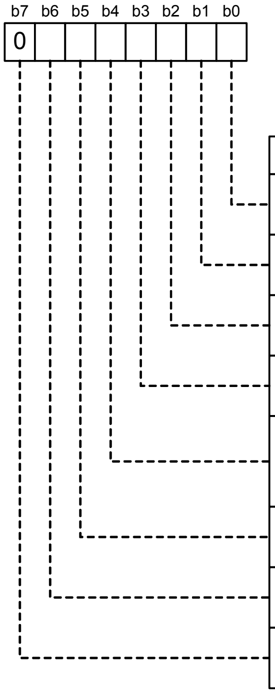
Bit Symbol	Bit Name	Function	RW
CLK0	UiBRG Count Source Select Bit	b1 b0 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : Do not use this combination	RW
CLK1			RW
— (b2)	Reserved	Should be written with 0	RW
TXEPT	Transmit Shift Register Empty Flag	0: Data held in the transmit shift register (transmission in progress) 1: No data held in the transmit shift register (transmission completed)	RO
CRD	CTS Function Disable Bit	0: CTS function enabled 1: CTS function disabled	RW
— (b5)	No register bit; should be written with 0 and read as undefined value		—
CKPOL	CLK Polarity Select Bit	0: Output transmit data on the falling edge of the transmit/receive clock and input receive data on the rising edge 1: Output transmit data on the rising edge of the transmit/receive clock and input receive data on the falling edge	RW
UFORM	Bit Order Select Bit [1]	0: LSB first 1: MSB first	RW

Note:

1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode) or 101b (UART mode, 8-bit character length). It should be set to 0 when they are set to 100b (UART mode, 7-bit character length) or 110b (UART mode, 9-bit character length).

Figure 18.6 Registers U7C0 to U10C0

UARTi Transmit/Receive Control Register 1 (i = 0 to 6)



Symbol	Address	Reset Value
U0C1 to U3C1	036Dh, 02EDh, 033Dh, 032Dh	0000 0010b
U4C1 to U6C1	02FDh, 01CDh, 01DDh	0000 0010b

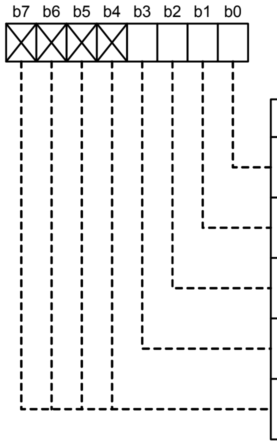
Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0: Transmission disabled 1: Transmission enabled	RW
TI	Transmit Buffer Empty Flag	0: Data held in the UiTB register 1: No data held in the UiTB register	RO
RE	Receive Enable Bit	0: Reception disabled 1: Reception enabled	RW
RI	Receive Complete Flag	0: No data held in the UiRB register 1: Data held in the UiRB register	RO
UiIRS	UARTi Transmit Interrupt Source Select Bit	0: Transmit buffer is empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
UiRRM	UARTi Continuous Receive Mode Enable Bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
UiLCH	Logic Inversion Select Bit (1)	0: Data is not logic inverted 1: Data is logic inverted	RW
— (b7)	Reserved	Should be written with 0	RW

Note:

1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode), 100b (UART mode, 7-bit character length), or 101b (UART mode, 8-bit character length). Set this bit to 0 when bits SMD2 to SMD0 are set to 010b (I^2C mode) or 110b (UART mode, 9-bit character length).

Figure 18.7 Registers U0C1 to U6C1

UARTi Transmit/Receive Control Register 1 (i = 7 to 10)

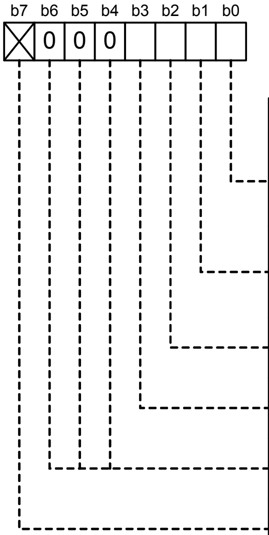


Symbol	Address	Reset Value
U7C1 to U10C1	01E5h, 01EDh, 40305h, 4030Dh	XXXX 0010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0: Transmission disabled 1: Transmission enabled	RW
TI	Transmit Buffer Empty Flag	0: Data held in the UiTB register 1: No data held in the UiTB register	RO
RE	Receive Enable Bit	0: Reception disabled 1: Reception enabled	RW
RI	Receive Complete Flag	0: No data held in the UiRB register 1: Data held in the UiRB register	RO
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		—

Figure 18.8 Registers U7C1 to U10C1

UART7, UART8 Transmit/Receive Control Register 2



Symbol
U78CON

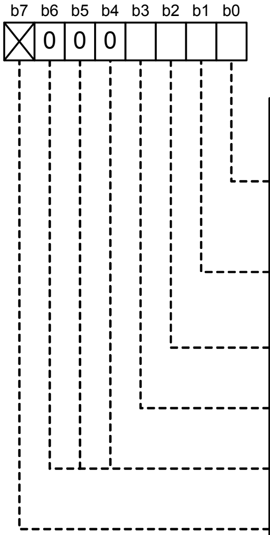
Address
01F0h

Reset Value
X000 0000b

Bit Symbol	Bit Name	Function	RW
U7IRS	UART7 Transmit Interrupt Source Select Bit	0: Transmit buffer is empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
U8IRS	UART8 Transmit Interrupt Source Select Bit	0: Transmit buffer is empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
U7RRM	UART7 Continuous Receive Mode Enable Bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
U8RRM	UART8 Continuous Receive Mode Enable Bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
— (b6-b4)	Reserved	Should be written with 0	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Figure 18.9 U78CON Register

UART9, UART10 Transmit/Receive Control Register 2



Symbol
U910CON

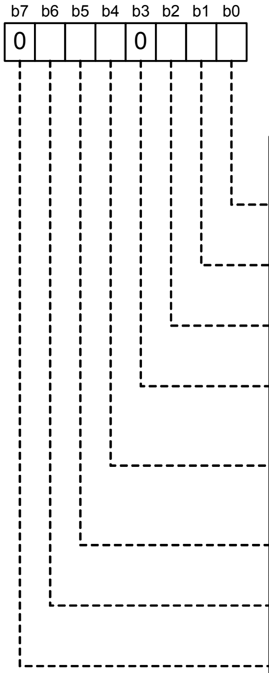
Address
40310h

Reset Value
X000 0000b

Bit Symbol	Bit Name	Function	RW
U9IRS	UART9 Transmit Interrupt Source Select Bit	0: Transmit buffer is empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
U10IRS	UART10 Transmit Interrupt Source Select Bit	0: Transmit buffer is empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
U9RRM	UART9 Continuous Receive Mode Enable Bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
U10RRM	UART10 Continuous Receive Mode Enable Bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
— (b6-b4)	Reserved	Should be written with 0	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Figure 18.10 U910CON Register

UARTi Special Mode Register (i = 0 to 6)



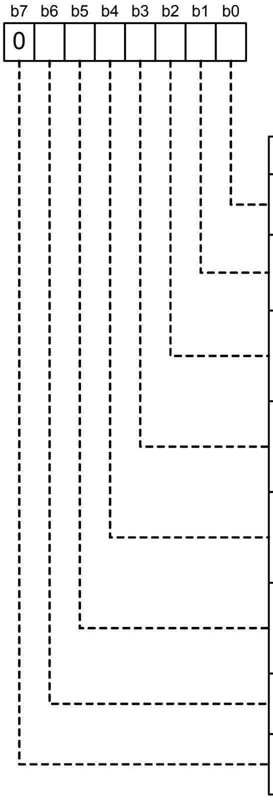
Symbol	Address	Reset Value
U0SMR to U3SMR	0367h, 02E7h, 0337h, 0327h	0000 0000b
U4SMR to U6SMR	02F7h, 01C7h, 01D7h	0000 0000b

Bit Symbol	Bit Name	Function	RW
IICM	I^2C Mode Select Bit [1]	0: Mode other than I^2C mode 1: I^2C mode	RW
ABC	Arbitration Lost Detection Flag Control [1]	0: Update every bit 1: Update every byte	RW
BBS	Bus Busy Flag [1, 2]	0: Detect stop condition 1: Detect start condition (bus busy)	RW
— (b3)	Reserved	Should be written with 0	RW
ABSCS	Bus Collision Detect Sampling Clock Select Bit [3]	0: Rising edge of the transmit/receive clock 1: Underflow of timer Aj (j = 0, 3, 4) [4]	RW
ACSE	Transmit Enable Bit Auto-reset to Zero Select Bit [3]	0: No auto-reset to zero 1: Auto-reset to zero at bus collision	RW
SSS	Transmit Start Condition Select Bit [3]	0: No relation with RXDi 1: Synchronized with RXDi	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. This bit is used in I^2C mode.
2. The BBS bit can only be set to 0. Writing 1 to this bit has no effect.
3. This bit is used in IE mode.
4. UART0: timer A3 underflow signal, UART1: timer A4 underflow signal
   UART2: timer A0 underflow signal, UART3: timer A3 underflow signal
   UART4: timer A4 underflow signal, UART5: timer A3 underflow signal
   UART6: timer A4 underflow signal

Figure 18.11 Registers U0SMR to U6SMR

UARTi Special Mode Register 2 (i = 0 to 6)



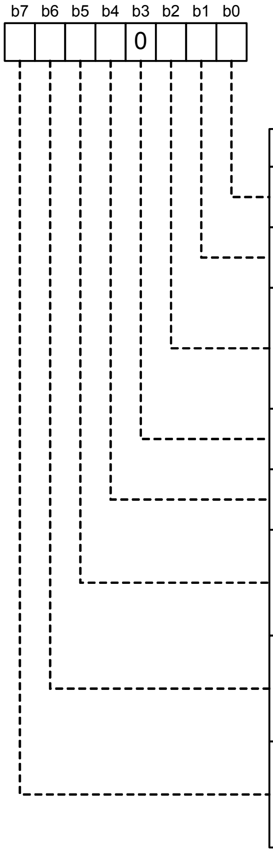
Symbol	Address	Reset Value
U0SMR2 to U3SMR2	0366h, 02E6h, 0336h, 0326h	0000 0000b
U4SMR2 to U6SMR2	02F6h, 01C6h, 01D6h	0000 0000b

Bit Symbol	Bit Name	Function	RW
IICM2	I²C Mode Select Bit 2	0: Use ACK/NACK interrupt 1: Use transmit/receive interrupt	RW
CSC	Clock Synchronization Bit (1)	0: Clock synchronization disabled 1: Clock synchronization enabled	RW
SWC	SCL Wait Auto Insert Bit (2)	0: No wait-state/wait-state cleared 1: Hold the SCLi pin low after the eighth bit is received	RW
ALS	SDA Output Auto Stop Bit (1)	When an arbitration lost is detected, 0: Do not stop the SDAi output 1: Stop the SDAi output	RW
STC	UARTi Auto Initialize Bit (2)	When a start condition is detected, 0: Do not initialize the circuit 1: Initialize the circuit	RW
SWC2	SCL Wait Output Bit 2 (1)	0: Output the transmit/receive clock at the SCLi pin 1: Hold the SCLi pin low	RW
SDHI	SDA Output Stop Bit (2)	0: Output data 1: Stop the output (high-impedance)	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. This bit is used in master mode of I²C mode.
2. This bit is used in slave mode of I²C mode.

Figure 18.12 Registers U0SMR2 to U6SMR2

UARTi Special Mode Register 3 (i = 0 to 6)



Symbol	Address	Reset Value
U0SMR3 to U3SMR3	0365h, 02E5h, 0335h, 0325h	0000 0000b
U4SMR3 to U6SMR3	02F5h, 01C5h, 01D5h	0000 0000b

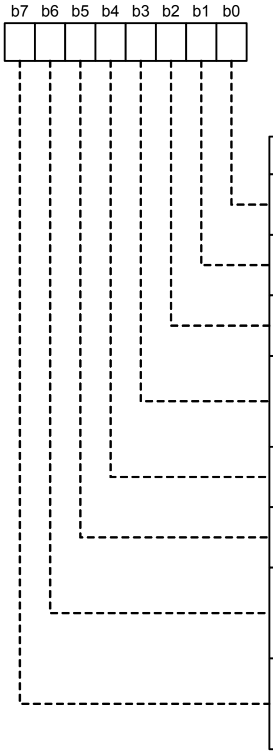
Bit Symbol	Bit Name	Function	RW
SSE	$\overline{SS}$ Pin Function Enable Bit [1, 2]	0: SS function disabled 1: SS function enabled	RW
CKPH	Clock-phase Set Bit	0: No clock delay 1: Clock delayed	RW
DINC	Serial Input Pin Set Bit [1]	0: Select the TXDi/RXDi pin (master mode) 1: Select the STXDi/SRXDi pin (slave mode)	RW
— (b3)	Reserved	Should be written with 0	RW
ERR	Mode Fault Flag [1]	0: No mode fault detected 1: Mode fault detected [3]	RW
DL0	SDAi Digital Delay Time Set Bit [4, 5]	Based on the baud rate generator count source, the SDAi output is delayed as follows: b7 b6 b5 0 0 0 : No delay 0 0 1 : 1 to 2 cycles 0 1 0 : 2 to 3 cycles 0 1 1 : 3 to 4 cycles 1 0 0 : 4 to 5 cycles 1 0 1 : 5 to 6 cycles 1 1 0 : 6 to 7 cycles 1 1 1 : 7 to 8 cycles	RW
DL1			RW
DL2			RW

Notes:

1. This bit is used in special mode 2.
2. Set the CRD bit in the UiC0 register to 1 (CTS function disabled) to use the SS function.
3. The ERR bit can only be set to 0. Writing 1 to this bit has no effect.
4. Bits DL2 to DL0 in I^{2}C mode generate a digital delay for the SDAi output. Set these bits to 000b in all modes other than I^{2}C mode.
5. When an external clock is selected, a delay of approximately 100 ns is added.

Figure 18.13 Registers U0SMR3 to U6SMR3

UARTi Special Mode Register 4 (i = 0 to 6)



Symbol	Address	Reset Value
U0SMR4 to U3SMR4	0364h, 02E4h, 0334h, 0324h	0000 0000b
U4SMR4 to U6SMR4	02F4h, 01C4h, 01D4h	0000 0000b

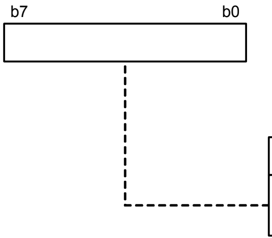
Bit Symbol	Bit Name	Function	RW
STAREQ	Start Condition Generate Bit (1)	0: Clear 1: Start (2)	RW
RSTAREQ	Restart Condition Generate Bit (1)	0: Clear 1: Start (2)	RW
STPREQ	Stop Condition Generate Bit (1)	0: Clear 1: Start (2)	RW
STSPSEL	SCL, SDA Output Select Bit (1)	0: Select serial I/O circuit 1: Select start condition/stop condition generation circuit (3)	RW
ACKD	ACK Data Bit (4)	0: ACK 1: NACK	RW
ACKC	ACK Data Output Enable Bit (4)	0: Serial data output 1: ACK data output	RW
SCLHI	SCL Output Stop Bit (1)	When a stop condition is detected, 0: Do not stop SCLi output 1: Stop SCLi output	RW
SWC9	SCL Wait Auto Insert Bit 3 (4)	0: No wait-state/wait-state cleared 1: Hold the SCLi pin low after the ninth bit is received	RW

Notes:

1. This bit is used in master mode of I^2C mode. It can be set to 1 when the IICM bit in the UiSMR register is 1 (I^2C mode).
2. This bit becomes 0 when the condition is generated. The setting remains 1 when the condition is incomplete.
3. Set the STSPSEL bit to 1 after setting the STAREQ, RSTAREQ, or STPREQ bit to 1.
4. This bit is used in slave mode of I^2C mode. It can be set to 1 when the IICM bit in the UiSMR register is 1 (I^2C mode).

Figure 18.14 Registers U0SMR4 to U6SMR4

UARTi Bit Rate Register (i = 0 to 10) (1, 2, 3)



Symbol	Address	Reset Value
U0BRG to U3BRG	0369h, 02E9h, 0339h, 0329h	Undefined
U4BRG to U7BRG	02F9h, 01C9h, 01D9h, 01E1h	Undefined
U8BRG to U10BRG	01E9h, 40301h, 40309h	Undefined

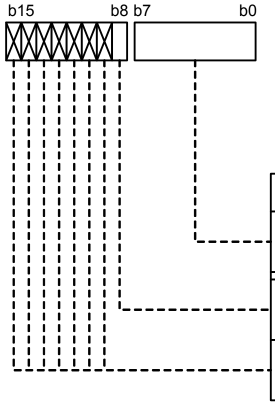
Function	Setting Range	RW
The UiBRG register divides the count source by n+1 (n = setting value)	00h to FFh	WO

Notes:

1. Set bits CLK1 and CLK0 in the UiC0 register before rewriting this register.
2. Use the MOV instruction to set this register.
3. Write this register while no data is being transmitted/received.

Figure 18.15 Registers U0BRG to U10BRG

UARTi Transmit Buffer Register (i = 0 to 10) [1]



Symbol	Address	Reset Value
U0TB to U2TB	036Bh-036Ah, 02EBh-02EAh, 033Bh-033Ah	Undefined
U3TB to U5TB	032Bh-032Ah, 02FBh-02FAh, 01CBh-01CAh	Undefined
U6TB to U8TB	01DBh-01DAh, 01E3h-01E2h, 01EBh-01EAh	Undefined
U9TB, U10TB	40303h-40302h, 4030Bh-4030Ah	Undefined

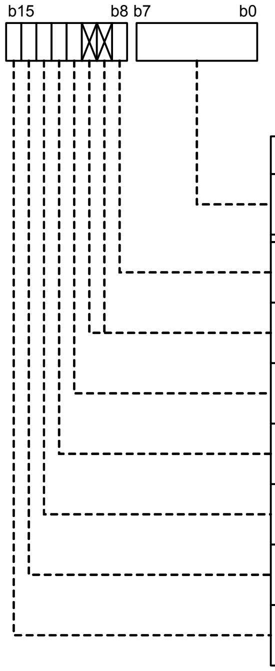
Bit Symbol	Function	RW
— (b7-b0)	Data (D7 to D0) to be transmitted	WO
— (b8)	Data (D8) to be transmitted	WO
— (b15-b9)	No register bits; should be written with 0	—

Note:

1. Use the MOV instruction to set this register.

Figure 18.16 Registers U0TB to U10TB

UARTi Receive Buffer Register (i = 0 to 6)



Symbol	Address	Reset Value
U0RB to U2RB	036Fh-036Eh, 02EFh-02EEh, 033Fh-033Eh	Undefined
U3RB to U5RB	032Fh-032Eh, 02FFh-02FEh, 01CFh-01CEh	Undefined
U6RB	01DFh-01DEh	Undefined

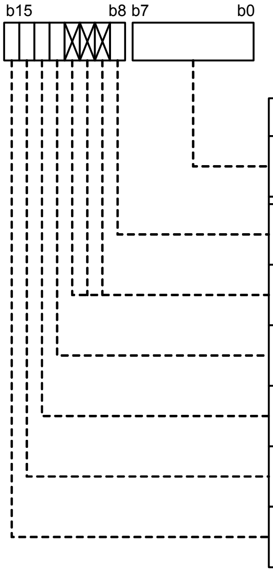
Bit Symbol	Bit Name	Function	RW
— (b7-b0)	—	Data (D7 to D0) received	RO
— (b8)	—	Data (D8) received	RO
— (b10-b9)	No register bits; should be written with 0 and read as 0		—
ABT	Arbitration Lost Detection Flag [1]	0: Not detected (win) 1: Detected (lose)	RW
OER	Overrun Error Flag [2]	0: No overrun error occurred 1: Overrun error occurred	RO
FER	Framing Error Flag [2, 3]	0: No framing error occurred 1: Framing error occurred	RO
PER	Parity Error Flag [2, 3]	0: No parity error occurred 1: Parity error occurred	RO
SUM	Error Sum Flag [2, 3]	0: No error occurred 1: Error occurred	RO

Notes:

1. The ABT bit can only be set to 0.
2. Bits OER, FER, PER, and SUM become 0 when bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled) or the RE bit in the UiC1 register is set to 0 (reception disabled). When bits OER, FER, and PER all become 0, the SUM bit also becomes 0. Bits FER and PER become 0 when the lower byte in the UiRB register is read.
3. When bits SMD2 to SMD0 are 001b (synchronous serial interface mode) or 010b (I^{2}C mode), these error flags are disabled and read as an undefined value.

Figure 18.17 Registers U0RB to U6RB

UARTi Receive Buffer Register (i = 7 to 10)



Symbol	Address	Reset Value
U7RB, U8RB	01E7h-01E6h, 01EFh-01EEh	Undefined
U9RB, U10RB	40307h-40306h, 4030Fh-4030Eh	Undefined

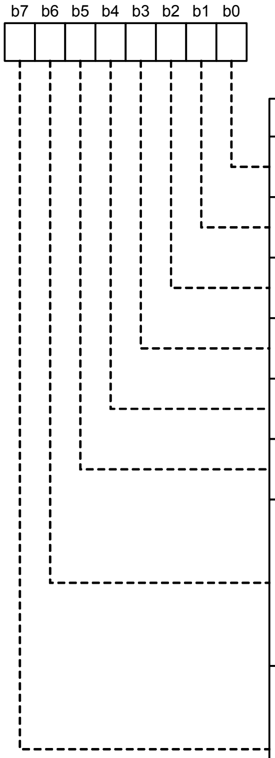
Bit Symbol	Bit Name	Function	RW
— (b7-b0)	—	Data (D7 to D0) received	RO
— (b8)	—	Data (D8) received	RO
— (b11-b9)	No register bits; should be written with 0 and read as 0		—
OER	Overrun Error Flag (1)	0: No overrun error occurred 1: Overrun error occurred	RO
FER	Framing Error Flag (1, 2)	0: No framing error occurred 1: Framing error occurred	RO
PER	Parity Error Flag (1, 2)	0: No parity error occurred 1: Parity error occurred	RO
SUM	Error Sum Flag (1, 2)	0: No error occurred 1: Error occurred	RO

Notes:

1. Bits OER, FER, PER, and SUM become 0 when bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled) or the RE bit in the UiC1 register is set to 0 (reception disabled). When bits OER, FER, and PER all become 0, the SUM bit also becomes 0. Bits FER and PER become 0 when the lower byte in the UiRB register is read.
2. When bits SMD2 to SMD0 are 001b (synchronous serial interface mode) or 010b (I^2C mode), these error flags are disabled and read as an undefined value.

Figure 18.18 Registers U7RB to U10RB

External Interrupt Request Source Select Register 0



Symbol IFSR0 | Address 4406Fh | Reset Value 0000 0000b

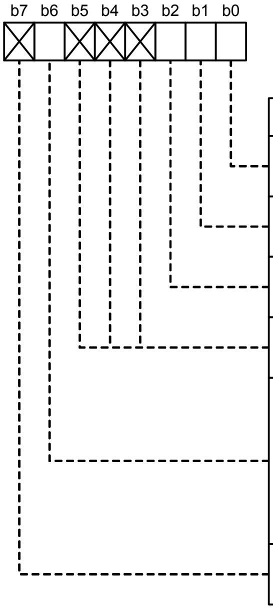
Bit Symbol	Bit Name	Function	RW
IFSR00	$\overline{\text{INT0}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR01	$\overline{\text{INT1}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR02	$\overline{\text{INT2}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR03	$\overline{\text{INT3}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR04	$\overline{\text{INT4}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR05	$\overline{\text{INT5}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR06	UART0/UART3 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART3 1: Bus collision, start condition detection, stop condition detection in UART0	RW
IFSR07	UART1/UART4 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART4 1: Bus collision, start condition detection, stop condition detection in UART1	RW

Note:

1. Set this bit to 0 to select the level sensitive input as trigger. To set this bit to 1, set the POL bit in the corresponding INTiIC register to 0 (falling edge) (i = 0 to 5).

Figure 18.19 IFSR0 Register

External Interrupt Request Source Select Register 1



Symbol	Address	Reset Value
IFSR1	4406Dh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
IFSR10	$\overline{\text{INT6}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
IFSR11	$\overline{\text{INT7}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
IFSR12	$\overline{\text{INT8}}$ Pin Polarity Select Bit [(1)]	0: One edge 1: Both edges	RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
IFSR16	UART5/UART6 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART5 1: Bus collision, start condition detection, stop condition detection in UART6	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Note:

1. Set this bit to 0 to select the level sensitive input as trigger. To set this bit to 1, set the POL bit in the corresponding INTiIC register (i = 6 to 8) to 0 (falling edge).

Figure 18.20 IFSR1 Register

18.1 Synchronous Serial Interface Mode

The synchronous serial interface mode allows data transmission/reception synchronized with the transmit/receive clock. Table 18.2 lists specifications of synchronous serial interface mode.

Table 18.2 Synchronous Serial Interface Mode Specifications

Item	Specification
Data format	8-bit character length
Transmit/receive clock	• The CKDIR bit in the UiMR register is 0 (internal clock) (i = 0 to 10): $\frac{fx}{2(m+1)}$ fx = f1, f8, f2n; m: UiBRG register setting value, 00h to FFh • The CKDIR bit is 1 (external clock): input to the CLKi pin
Transmit/receive control	CTS function enabled, RTS function enabled, or CTS/RTS function disabled
Transmit start conditions	The conditions for starting data transmission are as follows [1]: • The TE bit in the UiC1 register is 1 (transmission enabled) • The TI bit in the UiC1 register is 0 (data held in the UiTB register) • Input level at the $\overline{\text{CTSi}}$ pin is low when the CTS function is selected
Receive start conditions	The conditions for starting data reception are as follows [1]: • The RE bit in the UiC1 register is 1 (reception enabled) • The TE bit in the UiC1 register is 1 (transmission enabled) • The TI bit in the UiC1 register is 0 (data held in the UiTB register) • Input level at the $\overline{\text{CTSi}}$ pin is low when the CTS function is selected
Interrupt request generating timing	In transmit interrupt, one of the following conditions can be selected by setting the UiIRS bit in registers U0C1 to U6C1, U78CON, and U910CON: • The UiIRS bit is 0 (transmit buffer is empty): when data is transferred from the UiTB register to the UARTi transmit register (when the transmission has started) • The UiIRS bit is 1 (transmission is completed): when data transmission from the UARTi transmit register is completed In receive interrupt, • When data is transferred from the UARTi receive register to the UiRB register (when the reception is completed)
Error detection	Overrun error [2] This error occurs when the seventh bit of the next data is received before the UiRB register is read
Other functions	• CLK polarity Rising or falling edge of the transmit/receive clock for output and input of transmit/receive data • Bit order selection LSB first or MSB first • Continuous receive mode Data reception is enabled by a read access to the UiRB register • Serial data logic inversion (UART0 to UART6) This function logically inverses transmit/receive data

Notes:

1. When selecting an external clock, the following preconditions should be met:
   - The CLKi pin is held high when the CKPOL bit in the UiC0 register is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge).
   - The CLKi pin is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge).
2. The UiRB register is undefined when an overrun error occurs. The IR bit in the SiRIC register does not change to 1 (interrupt requested).

Tables 18.3 and 18.4 list register settings. When UARTi operating mode is selected, a high is output at the TXDi pin until transmission starts (the TXDi pin is high-impedance when the N-channel open drain output is selected) (i = 0 to 10).

Figures 18.21 and 18.22 show examples of transmit and receive operations in synchronous serial interface mode, respectively.

Table 18.3 Register Settings in Synchronous Serial Interface Mode (for UART0 to UART6)

Register	Bits	Function
UiMR	7 to 4	Set the bits to 0000b
	CKDIR	Select either an internal clock or external clock
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Select a transmit/receive clock polarity
	5	Set the bit to 0
	CRD	Select CTS function enabled or disabled
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	7	Set the bit to 0
	UiLCH	Set the bit to 1 to use logic inversion
	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select a source for the UARTi transmit interrupt
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
UiSMR	7 to 0	Set the bits to 00h
UiSMR2	7 to 0	Set the bits to 00h
UiSMR3	7 to 0	Set the bits to 00h
UiSMR4	7 to 0	Set the bits to 00h
UiBRG	7 to 0	Set the bit rate
IFS0	IFS06	Select input pins for CLK3, RXD3, and $\overline{\text{CTS3}}$
	IFS03 and IFS02	Select input pins for CLK6, RXD6, and $\overline{\text{CTS6}}$
UiTB	7 to 0	Set the data to be transmitted
UiRB	OER	Overrun error flag
	7 to 0	Received data is read

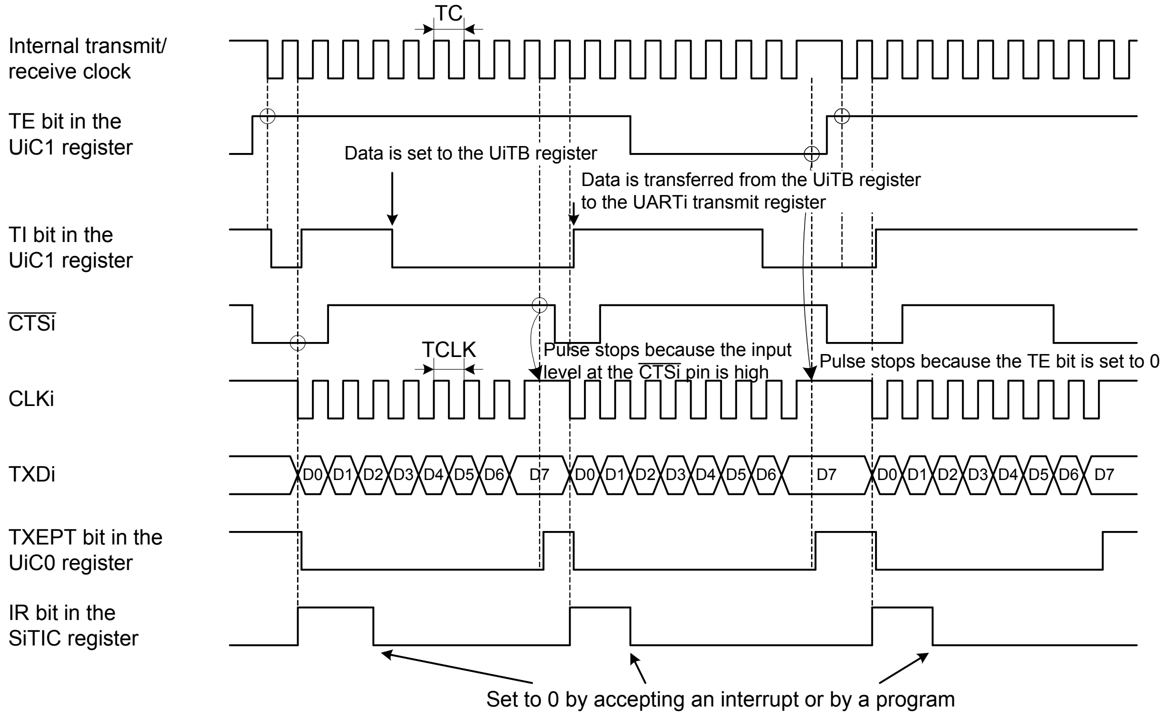
i = 0 to 6

Table 18.4 Register Settings in Synchronous Serial Interface Mode (for UART7 to UART10)

Register	Bits	Function
UiMR	7 to 4	Set the bits to 0000b
	CKDIR	Select an internal clock or external clock
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Select a transmit/receive clock polarity
	5	Set the bit to 0
	CRD	Select CTS function enabled or disabled
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
U78CON	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select an interrupt source for UARTi transmit
U910CON	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select an interrupt source for UARTi transmit
IFS0	IFS05	Select input pins for CLK7, RXD7, and $\overline{\mathrm{CTS7}}$
	IFS04	Select input pins for CLK8, RXD8, and $\overline{\mathrm{CTS8}}$
IFS3	IFS35	Select input pins for CLK10, RXD10, and $\overline{\mathrm{CTS10}}$
	IFS34	Select input pins for CLK9, RXD9, and $\overline{\mathrm{CTS9}}$
UiBRG	7 to 0	Set the bit rate
UiTB	7 to 0	Set the data to be transmitted
UiRB	OER	Overrun error flag
	7 to 0	Received data can be read

i = 7 to 10

Transmit timing (when selecting an internal clock)

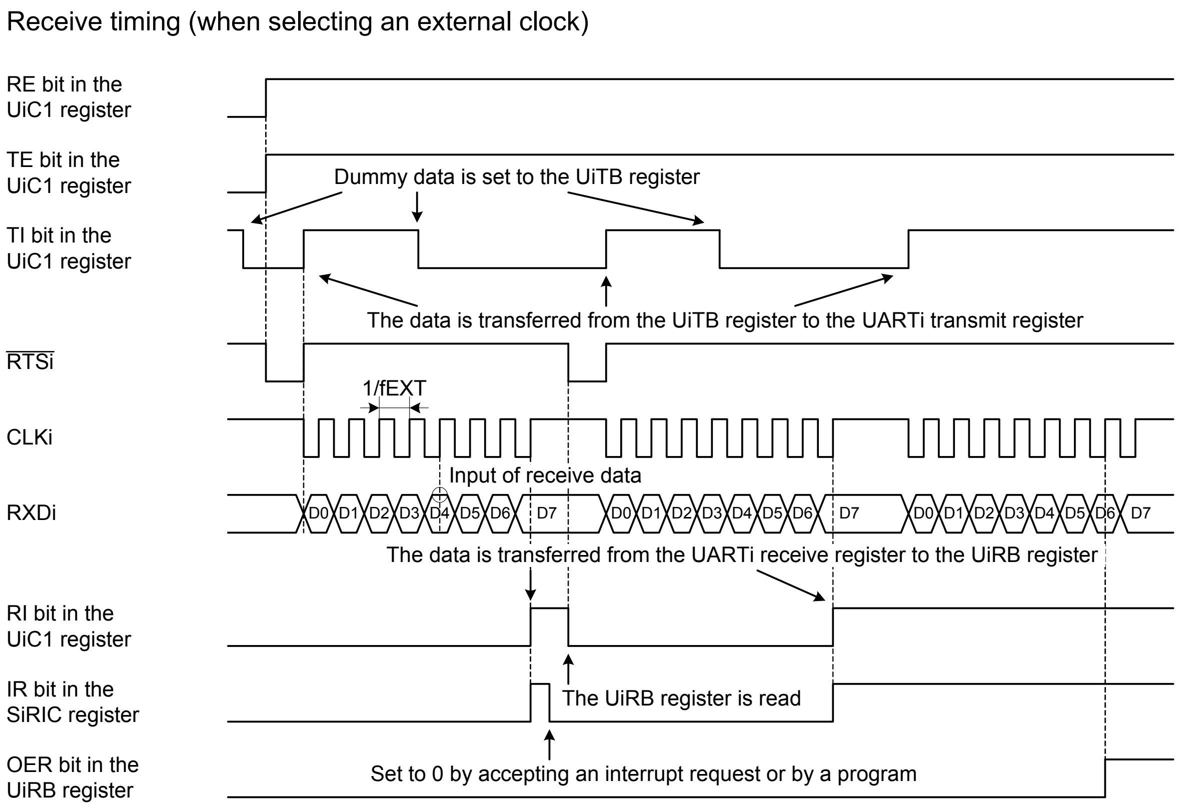

TC
Internal transmit/ receive clock
TE bit in the UiC1 register
Data is set to the UiTB register
Data is transferred from the UiTB register to the UARTi transmit register
TI bit in the UiC1 register
CTSi
TCLK
Pulse stops because the input level at the CTSi pin is high
Pulse stops because the TE bit is set to 0
CLKi
TXDi
D0 D1 D2 D3 D4 D5 D6 D7
TXEPT bit in the UiC0 register
IR bit in the SiTIC register
Set to 0 by accepting an interrupt or by a program

This figure applies under the following conditions:

- The CKDIR bit in the UiMR register is 0 (internal clock).
- The CRD bit in the UiC0 register is 0 (CTS function enabled).
- The CKPOL bit in the UiC0 register is 0 (output transmit data on the falling edge of the transmit/receive clock).
- The UiIRS bit in registers UiC1, U78CON, and U910CON is 0 (an interrupt request is generated when the transmit buffer is empty).

TC = TCLK = 2(m + 1)/fx
fx: UiBRG count source frequency (f1, f8, or f2n)
m: Value setting in the UiBRG register

Figure 18.21 Transmit Operation in Synchronous Serial Interface Mode


Receive timing (when selecting an external clock)
RE bit in the UiC1 register
TE bit in the UiC1 register
Dummy data is set to the UiTB register
TI bit in the UiC1 register
The data is transferred from the UiTB register to the UARTi transmit register
RTSi
1/fEXT
CLKi
Input of receive data
RXDi
D0 D1 D2 D3 D4 D5 D6 D7
The data is transferred from the UARTi receive register to the UiRB register
RI bit in the UiC1 register
IR bit in the SiRIC register
The UiRB register is read
OER bit in the UiRB register
Set to 0 by accepting an interrupt request or by a program

This figure applies under the following conditions:

- The CKDIR bit in the UiMR register is 1 (external clock).
- The CKPOL bit in the UiC0 register is 0 (input receive data on the rising edge of the transmit/receive clock).

fEXT: External clock frequency

The following conditions should be met while an input level at the CLKi pin before receiving data is high:

- The TE bit in the UiC1 register is 1 (transmission enabled).
- The RE bit in the UiC1 register is 1 (reception enabled).
- Write of dummy data to the UiTB register.

Figure 18.22 Receive Operation in Synchronous Serial Interface Mode

18.1.1 Reset Procedure on Transmit/Receive Error

When a transmit/receive error occurs in synchronous serial interface mode, follow the procedures below to perform a reset:

A. Reset procedure for the UiRB register (i = 0 to 10)

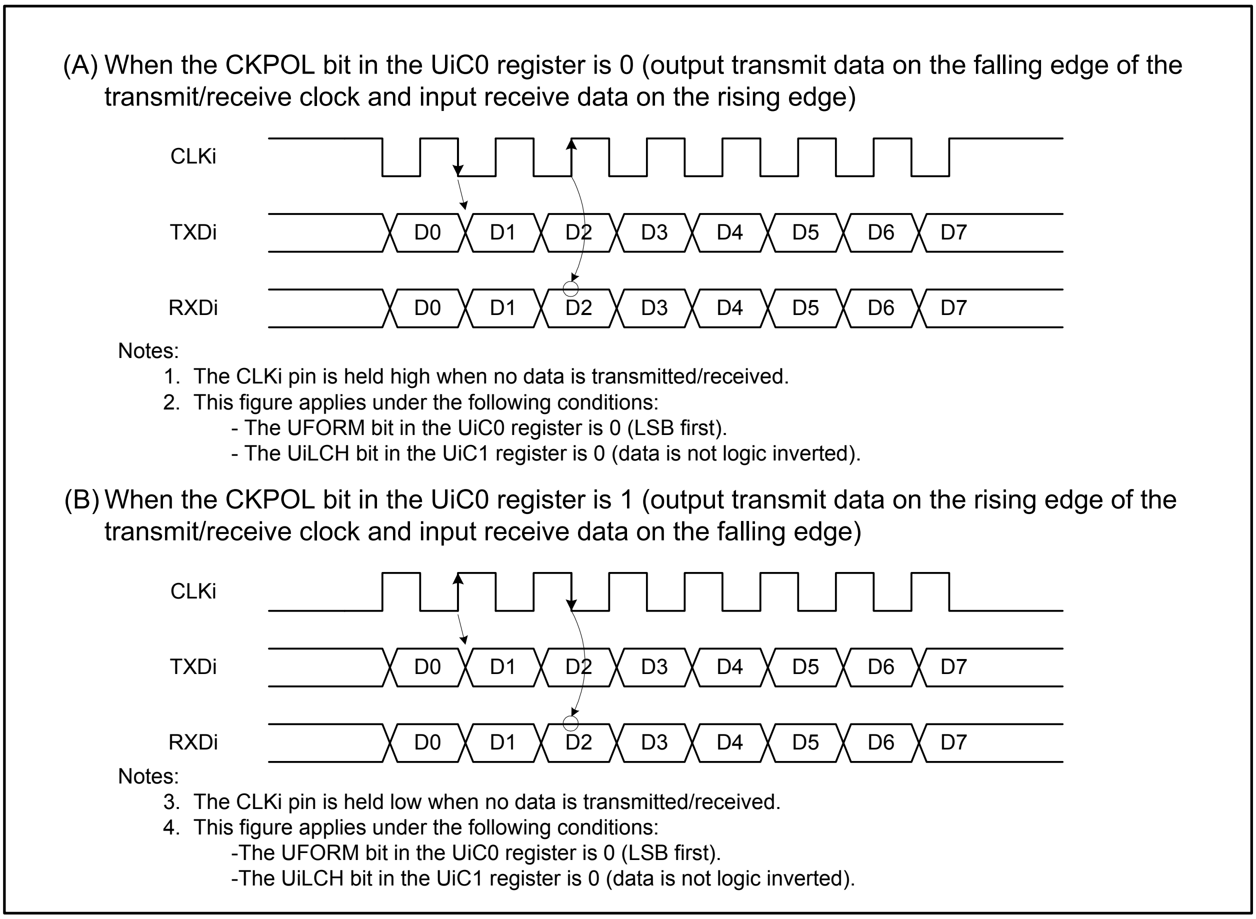
(1) Set the RE bit in the UiC1 register to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 to 001b (synchronous serial interface mode).
(4) Set the RE bit in the UiC1 register to 1 (reception enabled).

B. Reset procedure for the UiTB register

(1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(2) Set bits SMD2 to SMD0 to 001b (synchronous serial interface mode).
(3) Irrespective of its status, set the TE bit in the UiC1 register to 1 (transmission enabled).

18.1.2 CLK Polarity

As shown in Figure 18.23, the polarity of the transmit/receive clock is selected using the CKPOL bit in the UiC0 register (i = 0 to 10).



(A) When the CKPOL bit in the UiC0 register is 0 (output transmit data on the falling edge of the transmit/receive clock and input receive data on the rising edge)

Notes:
1. The CLKi pin is held high when no data is transmitted/received.
2. This figure applies under the following conditions:
   - The UFORM bit in the UiC0 register is 0 (LSB first).
   - The UiLCH bit in the UiC1 register is 0 (data is not logic inverted).

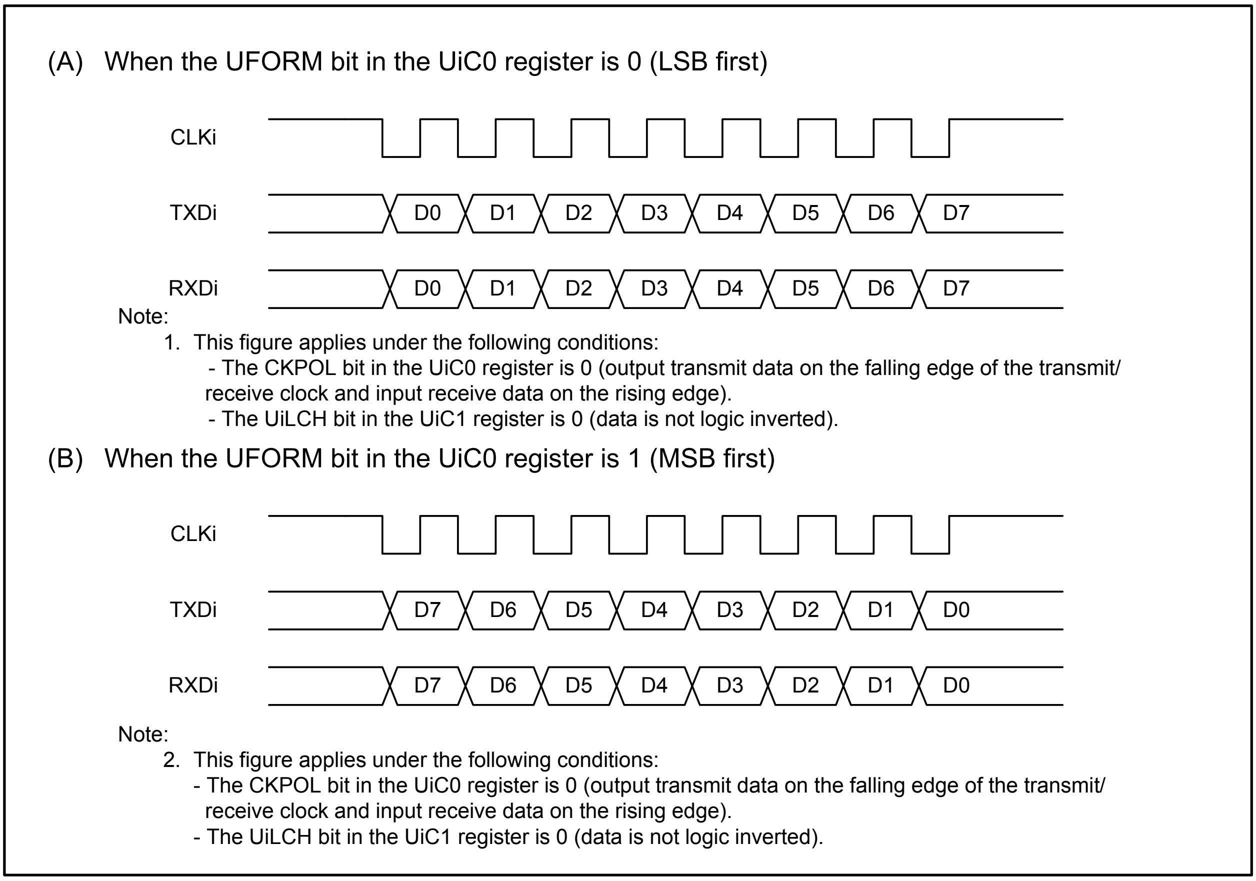
(B) When the CKPOL bit in the UiC0 register is 1 (output transmit data on the rising edge of the transmit/receive clock and input receive data on the falling edge)

Notes:
3. The CLKi pin is held low when no data is transmitted/received.
4. This figure applies under the following conditions:
   -The UFORM bit in the UiC0 register is 0 (LSB first).
   -The UiLCH bit in the UiC1 register is 0 (data is not logic inverted).

Figure 18.23 Transmit/Receive Clock Polarity (i = 0 to 10)

18.1.3 LSB First and MSB First Selection

As shown in Figure 18.24, the bit order is selected by setting the UFORM bit in the UiC0 register (i = 0 to 10).



(A) When the UFORM bit in the UiC0 register is 0 (LSB first)

Note:

1. This figure applies under the following conditions:
   - The CKPOL bit in the UiC0 register is 0 (output transmit data on the falling edge of the transmit/ receive clock and input receive data on the rising edge).
   - The UiLCH bit in the UiC1 register is 0 (data is not logic inverted).

(B) When the UFORM bit in the UiC0 register is 1 (MSB first)

Note:

2. This figure applies under the following conditions:
   - The CKPOL bit in the UiC0 register is 0 (output transmit data on the falling edge of the transmit/ receive clock and input receive data on the rising edge).
   - The UiLCH bit in the UiC1 register is 0 (data is not logic inverted).

Figure 18.24 Bit Order (i = 0 to 10)

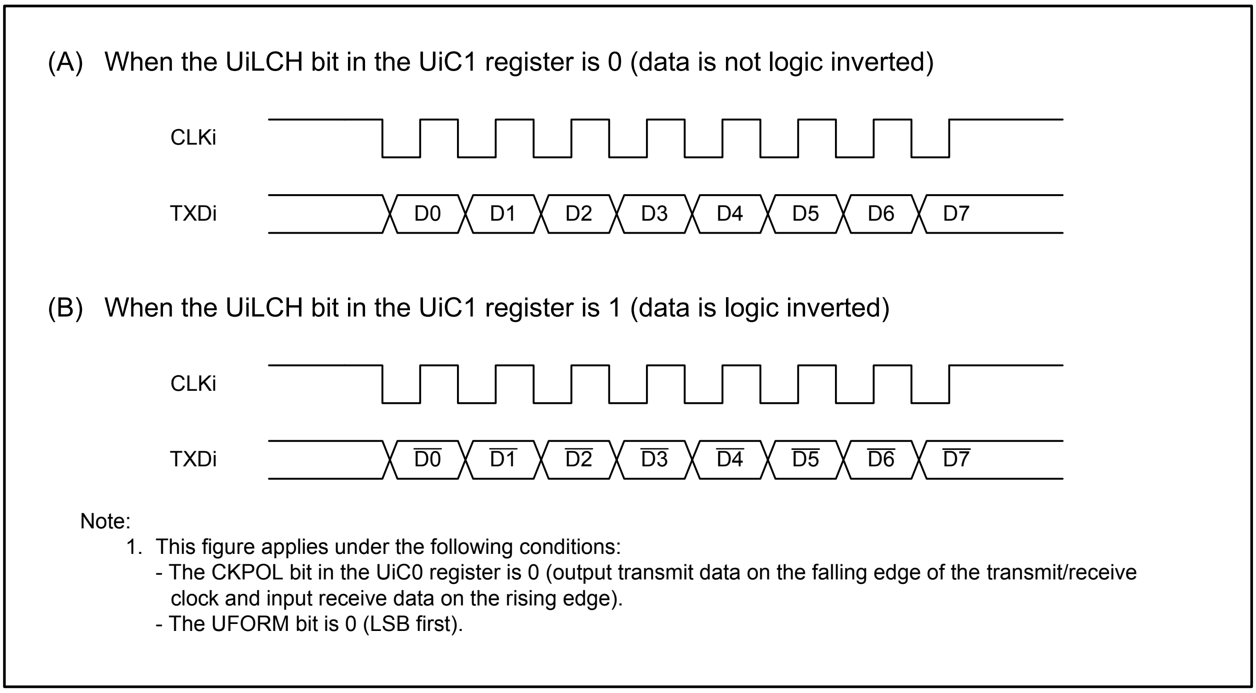
18.1.4 Continuous Receive Mode

In continuous receive mode, data reception is automatically enabled by a read access to the receive buffer register without writing dummy data to the transmit buffer register. To start data reception, however, dummy data is required to read the receive buffer register.

When the UiRRM bit in registers U0C1 to U6C1, the U78CON register, and the U910CON register is set to 1 (continuous receive mode enabled), the TI bit in the UiC1 register becomes 0 (data held in the UiTB register) by a read access to the UiRB register (i = 0 to 10). In this UiRRM bit setting, no dummy data should be written to the UiTB register.

18.1.5 Serial Data Logic Inversion

When the UiLCH bit in the UiC1 register is 1 (data is logic inverted), the logical value written in the UiTB register is inverted before being transmitted (i = 0 to 6). The UiRB register is read as logic-inverted receive data. Figure 18.25 shows the logic inversion of serial data.



Note:

1. This figure applies under the following conditions:
   - The CKPOL bit in the UiC0 register is 0 (output transmit data on the falling edge of the transmit/receive clock and input receive data on the rising edge).
   - The UFORM bit is 0 (LSB first).

Figure 18.25 Serial Data Logic Inversion (i = 0 to 6)

18.1.6 CTS/RTS Function

CTS function controls data transmission using the $\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ pin (i = 0 to 10). When an input level at the pin becomes low, data transmission starts. If the input level changes to high during transmission, the transmission of the next data is stopped.

In synchronous serial interface mode, the transmitter is required to operate even during the receive operation. If CTS function is enabled, the input level at the $\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ pin should be low to start data reception as well.

RTS function indicates receiver status using the $\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ pin. When data reception is ready, the output level at the pin becomes low. It becomes high on the first falling edge of the CLKi pin.

18.2 Asynchronous Serial Interface Mode (UART Mode)

The UART mode enables data transmission/reception synchronized with an internal clock generated by a trigger on the falling edge of the start bit. Table 18.5 lists specifications of UART mode.

Table 18.5 UART Mode Specifications

Item	Specification
Data format	• Start bit: 1-bit • Data bit (data character): 7-bit, 8-bit, or 9-bit • Parity bit: odd, even, or none • Stop bit: 1-bit or 2-bit
Transmit/receive clock	• The CKDIR bit in the UiMR register is 0 (internal clock) (i = 0 to 10): $\frac{fx}{16(m+1)}$ fx = f1, f8, f2n; m: UiBRG register setting value, 00h to FFh • The CKDIR bit is 1 (external clock) $\frac{fEXT}{16(m+1)}$ $fEXT$: Clock applied to the CLKi pin
Transmit/receive control	CTS function enabled, RTS function enabled, or CTS/RTS function disabled
Transmit start conditions	The conditions for starting data transmission are as follows: • The TE bit in the UiC1 register is 1 (transmission enabled) • The TI bit in the UiC1 register is 0 (data held in the UiTB register) • Input level at the $\overline{\text{CTSi}}$ pin is low when CTS function is selected
Receive start conditions	The conditions for starting data reception are as follows: • The RE bit in the UiC1 register is 1 (reception enabled) • The start bit is detected
Interrupt request generating timing	In transmit interrupt, one of the following conditions can be selected by setting the UiIRS bit in registers U0C1 to U6C1, the U78CON register, and the U910CON register: • The UiIRS bit is 0 (transmit buffer is empty): when data is transferred from the UiTB register to the UARTi transmit register (when the transmission has started) • The UiIRS bit is 1 (transmission is completed): when data transmission from the UARTi transmit register is completed In receive interrupt, • When data is transferred from the UARTi receive register to the UiRB register (when reception is completed)
Error detection	• Overrun error [1] This error occurs when 1 bit prior to the stop bit (when 1 stop bit length is selected) or the first stop bit (when 2 stop bit length is selected) of the next data is received before the UiRB register is read • Framing error This error occurs when the required number of stop bits is not detected • Parity error This error occurs when an even number of 1's in parity and character bits is detected while the odd number is set, or vice versa. The parity should be enabled • Error sum flag This flag becomes 1 when any of overrun error, framing error, or parity error occurs
Other functions	• Bit order selection LSB first or MSB first • Serial data logic inversion This function logically inverses transmit/receive data. The start bit and stop bit are not inverted • TXD/RXD I/O polarity switching The output level from the TXD pin and the input level to the RXD pin are inverted. All I/O levels are inverted

Note:

1. The UiRB register is undefined when an overrun error occurs. The IR bit in the SiRIC register does not change to 1 (interrupt requested).

Tables 18.6 and 18.7 list register settings. When UARTi operating mode is selected, a high is output at the TXDi pin until transmission starts (the TXDi pin is high-impedance when the N-channel open drain output is selected) (i = 0 to 10). Figures 18.26 and 18.27 show examples of transmit operations in UART mode. Figure 18.28 shows an example of receive operation.

Table 18.6 Register Settings in UART Mode (UART0 to UART6)

Register	Bits	Function
UiMR	IOPOL	Select I/O polarity of pins TXD and RXD
	PRY and PRYE	Select parity enabled or disabled, and odd or even
	STPS	Select a stop bit length
	CKDIR	Select an internal clock or external clock
	SMD2 to SMD0	Set the bits to 100b in 7-bit character length
		Set the bits to 101b in 8-bit character length
		Set the bits to 110b in 9-bit character length
UiC0	UFORM	Select LSB first or MSB first in 8-bit character length. Set the bit to 0 in 7-bit or 9-bit character length
	CKPOL	Set the bit to 0
	5	Set the bit to 0
	CRD	Select CTS function enabled or disabled
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	7	Set the bit to 0
	UiLCH	Set the bit to 1 to use logic inversion
	UiRRM	Set the bit to 0
	UiIRS	Select an interrupt source for UARTi transmission
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission
UiSMR	7 to 0	Set the bits to 00h
UiSMR2	7 to 0	Set the bits to 00h
UiSMR3	7 to 0	Set the bits to 00h
UiSMR4	7 to 0	Set the bits to 00h
UiBRG	7 to 0	Set the bit rate
IFS0	IFS06	Select input pins for CLK3, RXD3, and $\overline{\text{CTS3}}$
	IFS03 and IFS02	Select input pins for CLK6, RXD6, and $\overline{\text{CTS6}}$
UiTB	8 to 0	Set the data to be transmitted (1)
UiRB	OER, FER, PER, and SUM	Error flag
	8 to 0	Received data is read (1)

i = 0 to 6

Note:

1. The bits used are as follows: 7-bit character length: bits 6 to 0
8-bit character length: bits 7 to 0
9-bit character length: bits 8 to 0

Table 18.7 Register Settings in UART Mode (UART7 to UART10)

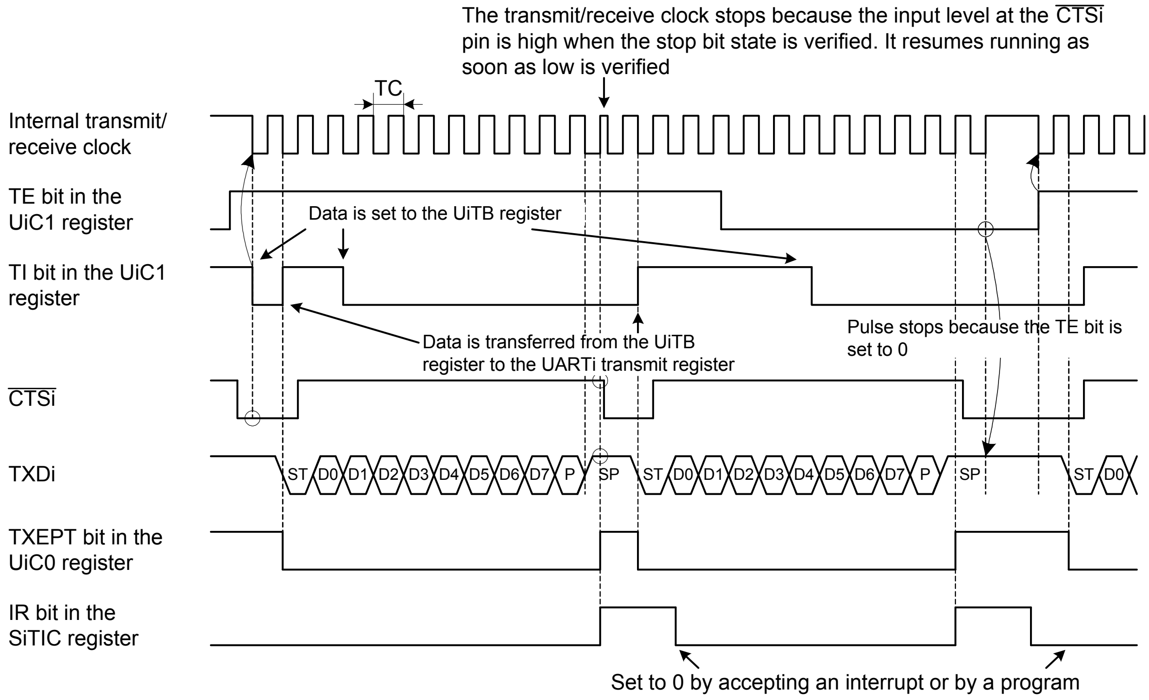
Register	Bits	Function
UiMR	PRY and PRYE	Select parity enabled or disabled, and odd or even
	STPS	Select a stop bit length
	CKDIR	Select an internal clock or external clock
	SMD2 to SMD0	Set the bits to 100b in 7-bit character length
		Set the bits to 101b in 8-bit character length
		Set the bits to 110b in 9-bit character length
UiC0	UFORM	Select LSB first or MSB first in 8-bit character length. Set the bit to 0 in 7-bit or 9-bit character length
	CKPOL	Set the bit to 0
	5	Set the bit to 0
	CRD	Select CTS function enabled or disabled
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission
U78CON	UiRRM	Set the bit to 0
	UiIRS	Select an interrupt source for UARTi transmission
U910CON	UiRRM	Set the bit to 0
	UiIRS	Select an interrupt source for UARTi transmission
UiBRG	7 to 0	Set the bit rate
IFS0	IFS05	Select input pins for CLK7, RXD7, and $\overline{\text{CTS7}}$
	IFS04	Select input pins for CLK8, RXD8, and $\overline{\text{CTS8}}$
IFS3	IFS35	Select input pins for CLK10, RXD10, and $\overline{\text{CTS10}}$
	IFS34	Select input pins for CLK9, RXD9, and $\overline{\text{CTS9}}$
UiTB	8 to 0	Set the data to be transmitted (1)
UiRB	OER, FER, PER, and SUM	Error flag
	8 to 0	Received data is read (1)

i = 7 to 10

Note:

1. The bits used are as follows: 7-bit character length: bits 6 to 0
   8-bit character length: bits 7 to 0
   9-bit character length: bits 8 to 0

Example of data transmit timing when the character length is 8-bit (parity enabled, 1 stop bit)


The transmit/receive clock stops because the input level at the CTSi pin is high when the stop bit state is verified. It resumes running as soon as low is verified
TC
Internal transmit/ receive clock
TE bit in the UiC1 register
Data is set to the UiTB register
TI bit in the UiC1 register
Data is transferred from the UiTB register to the UARTi transmit register
Pulse stops because the TE bit is set to 0
CTSi
TXDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0
TXEPT bit in the UiC0 register
IR bit in the SiTIC register
Set to 0 by accepting an interrupt or by a program

ST: Start bit
P: Parity bit
SP: Stop bit

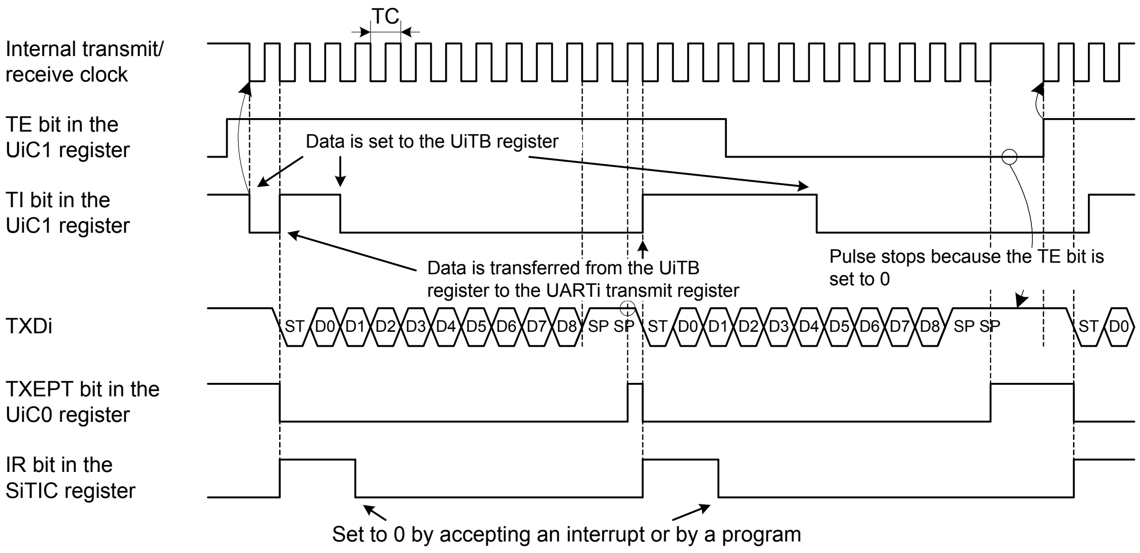
This figure applies under the following conditions:

- The PRYE bit in the UiMR register is 1 (parity enabled).
- The STPS bit in the UiMR register is 0 (1 stop bit).
- The CRD bit in the UiC0 register is 0 (CTS function enabled).
- The UiIRS bit in the UiC1 register is 1 (an interrupt request is generated when transmission is comepleted).

TC = 16 (m + 1) / fx or 16 (m + 1) / fEXT
fx: UiBRG count source frequency (f1, f8, or f2n)
fEXT: UiBRG count source frequency (external clock)
m: Value set in the UiBRG register

Figure 18.26 Transmit Operation in UART Mode (1/2) (i = 0 to 10)

Example of data transmit timing when the character length is 9-bit (parity disabled, 2 stop bits)


TC
Internal transmit/ receive clock
TE bit in the UiC1 register
TI bit in the UiC1 register
TXDi
TXEPT bit in the UiC0 register
IR bit in the SiTIC register
Data is set to the UiTB register
Data is transferred from the UiTB register to the UARTi transmit register
Pulse stops because the TE bit is set to 0
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0
Set to 0 by accepting an interrupt or by a program

ST: Start bit
P: Parity bit
SP: Stop bit

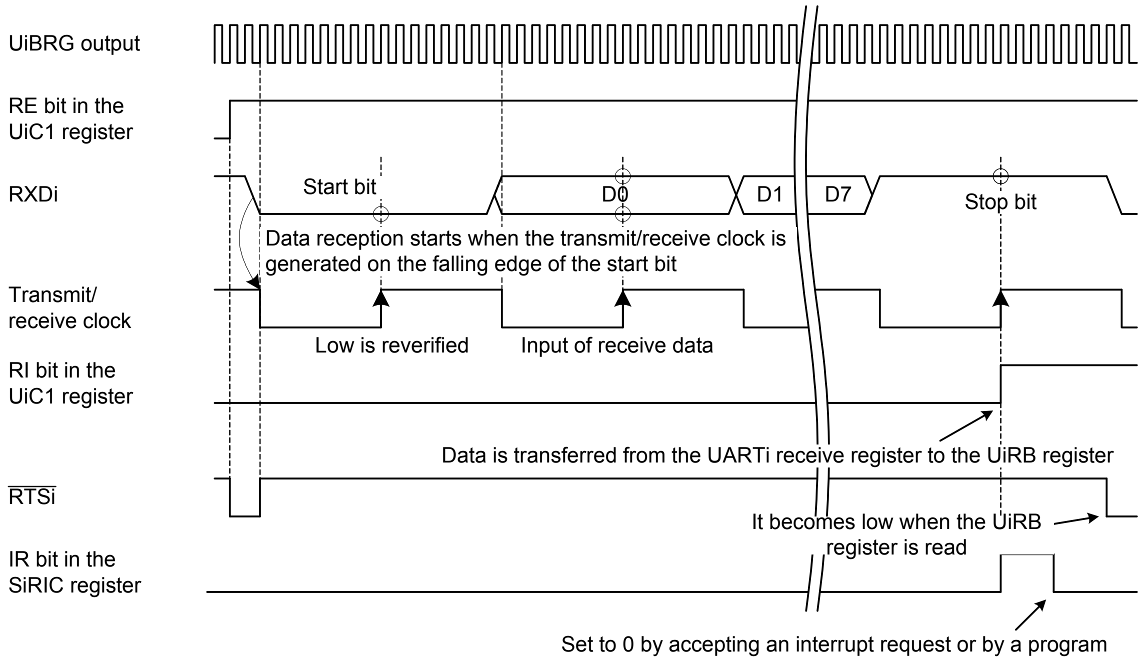
This figure applies under the following conditions:

- The PRYE bit in the UiMR register is 0 (parity disabled).
- The STPS bit in the UiMR register is 1 (2 stop bits).
- The CRD bit in the UiC0 register is 1 (CTS function disabled).
- The UiIRS bit in the UiC1 register is 0 (an interrupt request is generated when the transmit buffer is empty).

TC = 16 (m + 1) / fx or 16 (m + 1) / fEXT
fx: UiBRG count source frequency (f1, f8, or f2n)
fEXT: UiBRG count source frequency (external clock)
m: Value set in the UiBRG register

Figure 18.27 Transmit Operation in UART Mode (2/2) (i = 0 to 10)

Example of data receive timing when the character length is 8-bit (parity disabled, 1 stop bit)


UiBRG output
RE bit in the UiC1 register
RXDi
Start bit
D0
D1
D7
Stop bit
Data reception starts when the transmit/receive clock is generated on the falling edge of the start bit
Transmit/ receive clock
Low is reverified
Input of receive data
RI bit in the UiC1 register
Data is transferred from the UARTi receive register to the UiRB register
RTSi
It becomes low when the UiRB register is read
IR bit in the SiRIC register
Set to 0 by accepting an interrupt request or by a program

This figure applies under the following conditions:
- The PRYE bit in the UiMR register is 0 (parity disabled).
- The STPS bit in the UiMR register is 0 (1 stop bit).

Figure 18.28 Receive Operation in UART Mode (i = 0 to 10)

18.2.1 Bit Rate

In UART mode, the bit rate is a clock frequency which is divided by a setting value of the UiBRG register and again divided by 16 (i = 0 to 10). Table 18.8 lists an example of bit rate setting.

Table 18.8 Bit Rate Setting

Bit Rate (bps)	Count Source of BRG	Peripheral Clock: 30 MHz		Peripheral Clock: 32 MHz	
		Setting value of BRG: n	Actual bit rate (bps)	Setting value of BRG: n	Actual bit rate (bps)
1200	f8	194 (C2h)	1202	207 (CHh)	1202
2400	f8	97 (61h)	2392	103 (67h)	2404
4800	f8	48 (30h)	4783	51 (33h)	4808
9600	f1	194 (C2h)	9615	207 (CFh)	9615
14400	f1	129 (81h)	14423	138 (8Ah)	14388
19200	f1	97 (61h)	19133	103 (67h)	19231
28800	f1	64 (40h)	28846	68 (44h)	28986
31250	f1	59 (3Bh)	31250	63 (3Fh)	31250
38400	f1	48 (30h)	38265	51 (33h)	38462
51200	f1	36 (24h)	50676	38 (26h)	51282

18.2.2 Reset Procedure on Transmit/Receive Error

When a transmit/receive error occurs in UART mode, follow the procedure below to perform a reset:

A. Reset procedure for the UiRB register (i = 0 to 10)

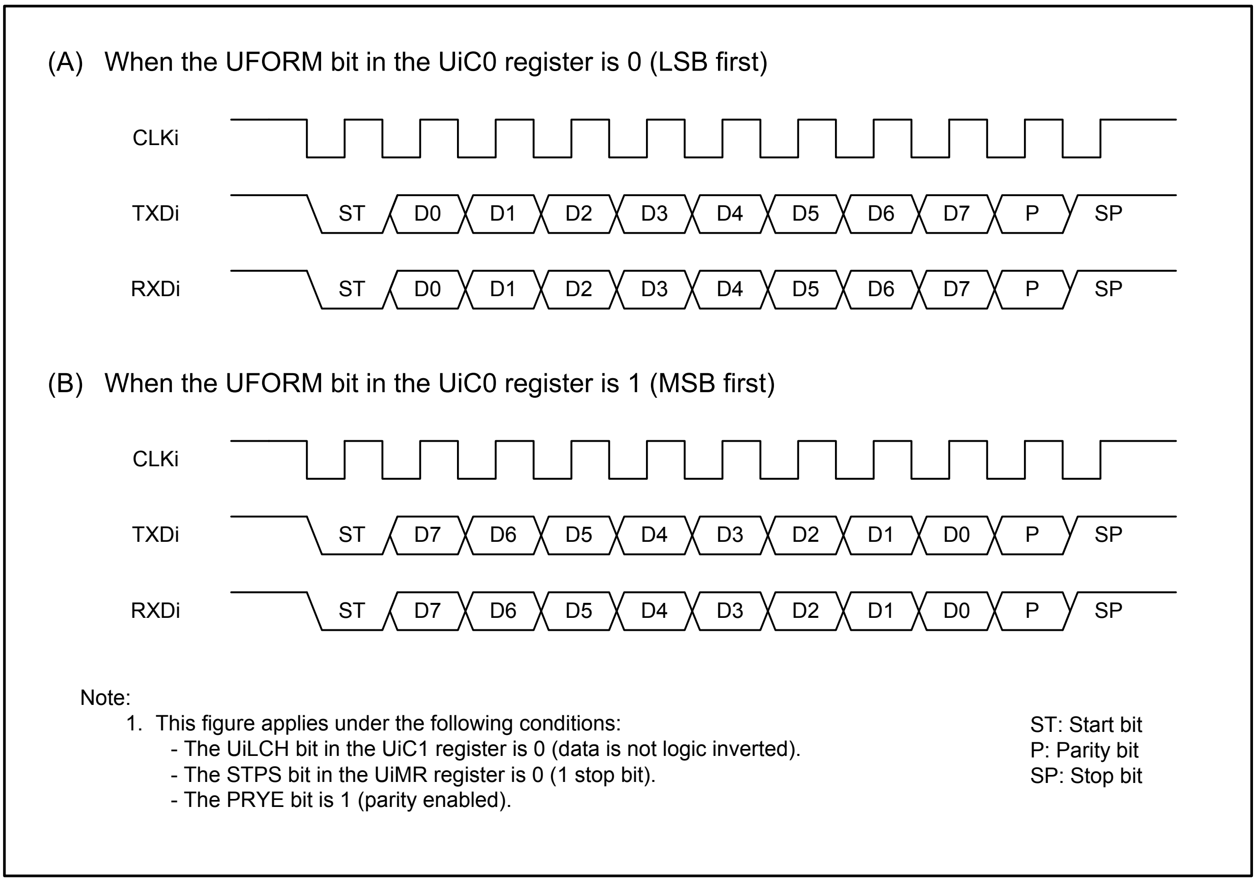
(1) Set the RE bit in the UiC1 register to 0 (reception disabled).
(2) Set the RE bit in the UiC1 register to 1 (reception enabled).

B. Reset procedure for the UiTB register

(1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(2) Set again bits SMD2 to SMD0 to either of 001b, 101b, or 110b.
(3) Irrespective of its status, set the TE bit in the UiC1 register to 1 (transmission enabled).

18.2.3 LSB First and MSB First Selection

As shown in Figure 18.29, the bit order is selected by setting the UFORM bit in the UiC0 register (i = 0 to 10). This function is available when the character length is 8-bit.



(A) When the UFORM bit in the UiC0 register is 0 (LSB first)

(B) When the UFORM bit in the UiC0 register is 1 (MSB first)

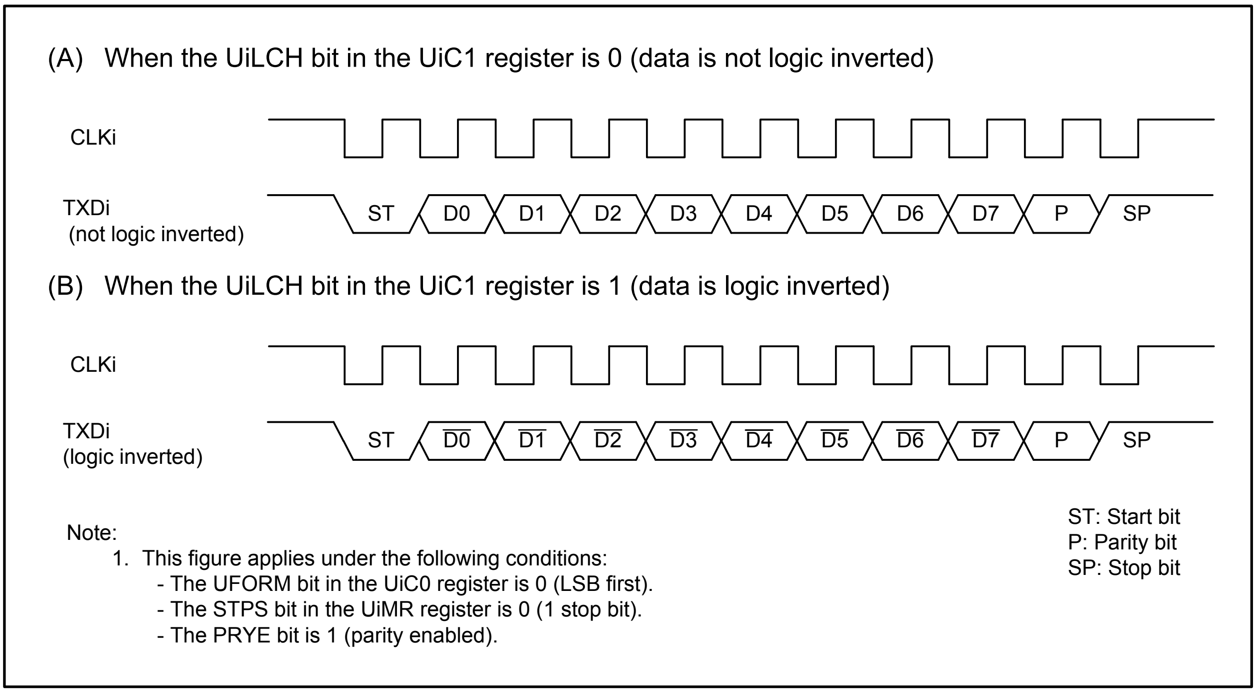
Note:
1. This figure applies under the following conditions:
- The UiLCH bit in the UiC1 register is 0 (data is not logic inverted).
- The STPS bit in the UiMR register is 0 (1 stop bit).
- The PRYE bit is 1 (parity enabled).

ST: Start bit
P: Parity bit
SP: Stop bit

Figure 18.29 Bit Order (i = 0 to 10)

18.2.4 Serial Data Logic Inversion

When the UiLCH bit in the UiC1 register is 1 (data is logic inverted), the logical value written in the UiTB register is inverted before being transmitted (i = 0 to 6). The UiRB register is read as logic-inverted receive data. The parity bit is not inverted. Figure 18.30 shows the logic inversion of serial data.



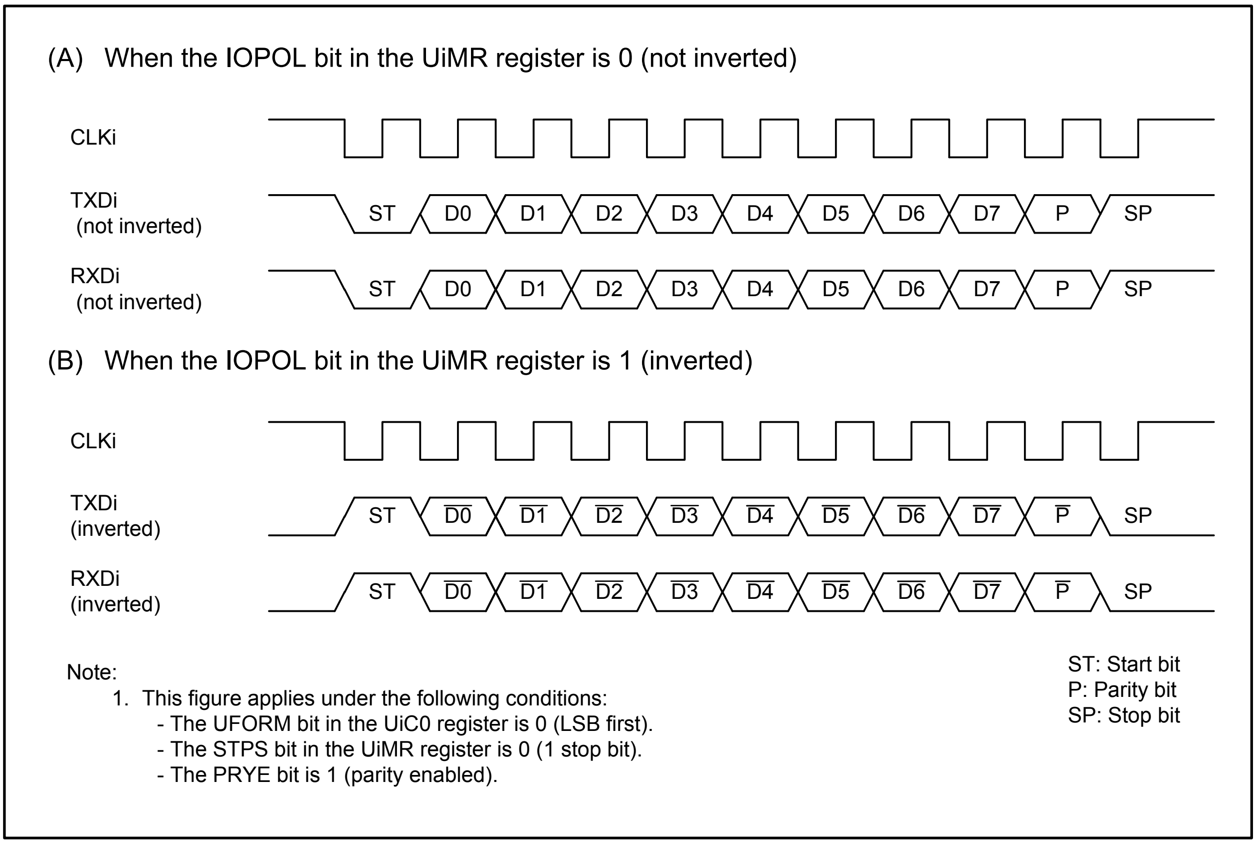
Note:

1. This figure applies under the following conditions:
   - The UFORM bit in the UiC0 register is 0 (LSB first).
   - The STPS bit in the UiMR register is 0 (1 stop bit).
   - The PRYE bit is 1 (parity enabled).

Figure 18.30 Serial Data Logic Inversion (i = 0 to 6)

18.2.5 TXD and RXD I/O Polarity Inversion

The output level at the TXD pin and the input level at the RXD pin are inverted by this function. All I/O data levels, including the start bit, stop bit, and parity bit are inverted by setting the IOPOL bit in the UiMR register to 1 (inverted) (i = 0 to 6). Figure 18.31 shows TXD and RXD I/O polarity inversion.



(A) When the IOPOL bit in the UiMR register is 0 (not inverted)

CLKi
TXDi (not inverted)
RXDi (not inverted)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP

(B) When the IOPOL bit in the UiMR register is 1 (inverted)

CLKi
TXDi (inverted)
RXDi (inverted)
ST $\overline{D0}$ $\overline{D1}$ $\overline{D2}$ $\overline{D3}$ $\overline{D4}$ $\overline{D5}$ $\overline{D6}$ $\overline{D7}$ $\overline{P}$ SP

Note:
1. This figure applies under the following conditions:
   - The UFORM bit in the UiC0 register is 0 (LSB first).
   - The STPS bit in the UiMR register is 0 (1 stop bit).
   - The PRYE bit is 1 (parity enabled).

ST: Start bit
P: Parity bit
SP: Stop bit

Figure 18.31 TXD and RXD I/O Polarity Inversion (i = 0 to 6)

18.2.6 CTS/RTS Function

CTS function controls data transmission using the $\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ pin (i = 0 to 10). When an input level at the pin becomes low, data transmission starts. If the input level changes to high during transmit operation, transmission of the next data is stopped.

RTS function indicates receiver status using the $\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ pin. When the MCU is ready to receive data, the output level at the pin becomes low. It becomes high on the first falling edge of the CLKi pin.

18.3 Special Mode 1 (I^2C Mode)

This mode uses an I^2C-typed interface for communication. Table 18.9 lists specifications of the I^2C mode.

Table 18.9 I^2C Mode Specifications

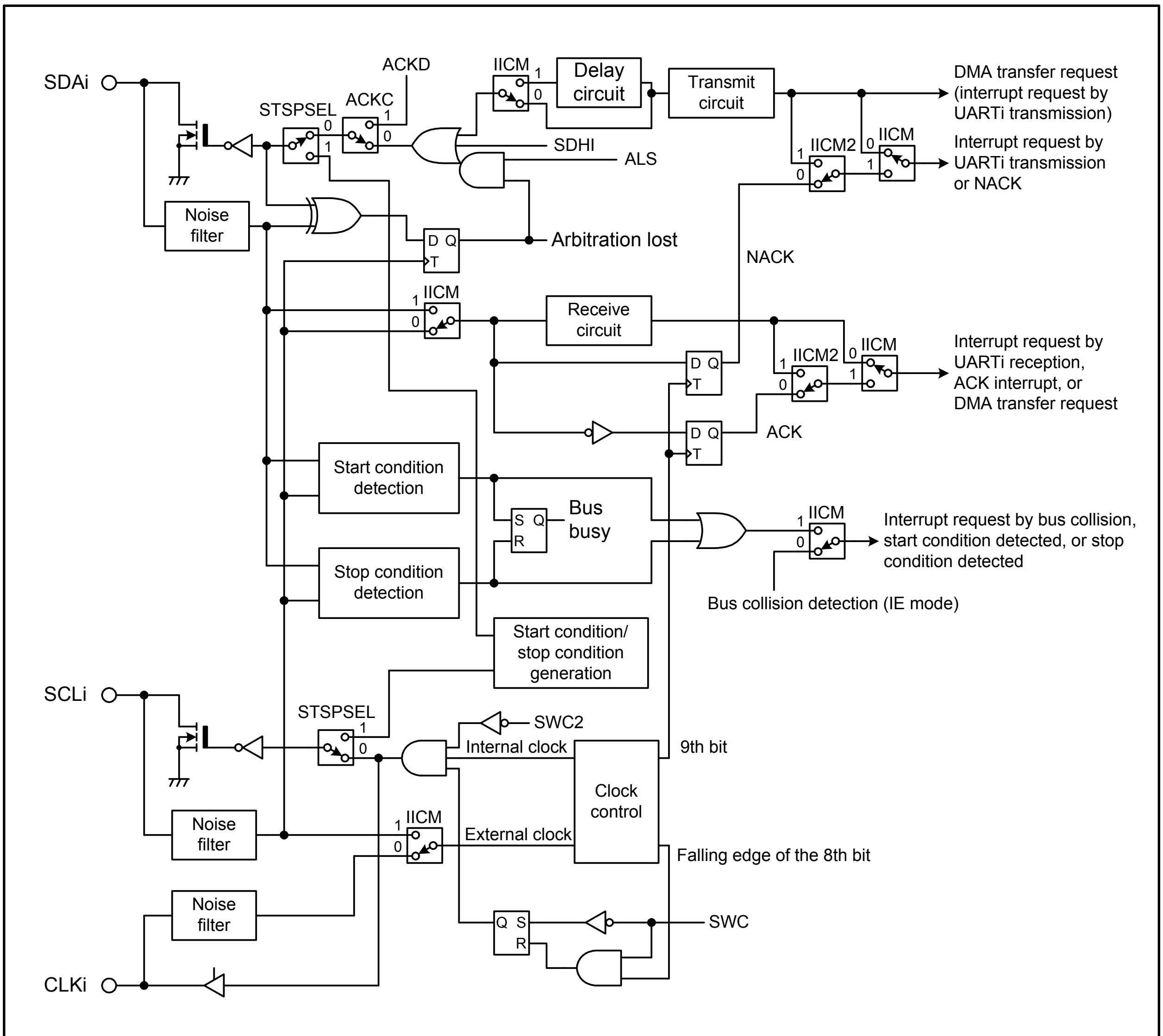
Item	Specification
Data format	8-bit character length
Transmit/receive clock	In master mode • The CKDIR bit in the UiMR register is 0 (internal clock) (i = 0 to 6): $\frac{fx}{2(m+1)}$ fx = f1, f8, f2n m: UiBRG register setting value, 00h to FFh In slave mode • The CKDIR bit is 1 (external clock): input to the SCLi pin
Transmit start conditions	The conditions for starting data transmission are as follows (1): • The TE bit in the UiC1 register is 1 (transmission enabled) • The TI bit in the UiC1 register is 0 (data held in the UiTB register)
Receive start conditions	The conditions for starting data reception are as follows (1): • The RE bit in the UiC1 register is 1 (reception enabled) • The TE bit in the UiC1 register is 1 (transmission enabled) • The TI bit in the UiC1 register is 0 (data held in the UiTB register)
Interrupt request generating timing	When any of the following is detected: start condition, stop condition, NACK (not-acknowledge), or ACK (acknowledge)
Error detection	Overrun error (2) This error occurs when the eighth bit of the next data is received before the UiRB register is read
Other functions	• Arbitration lost Update timing of the ABT bit in the UiRB register can be selected • SDAi digital delay No digital delay or two to eight cycles of digital delay of UiBRG count source • Clock phase setting Clock delayed or no clock delay

Notes:

1. When an external clock is selected, the conditions should be met while the external clock signal is held high.
2. The UiRB register is undefined when an overrun error occurs. The IR bit in the SiRIC register does not change to 1 (interrupt requested).

Table 18.10 lists register settings in I^2C mode, and Tables 18.11 and 18.12 list I^2C mode functions. Figure 18.32 shows a block diagram of I^2C mode, and Figure 18.33 shows timings for the transfer to the UiRB register and the interrupt (i = 0 to 6).

As shown in Tables 18.11 and 18.12, UARTi enters this mode when bits SMD2 to SMD0 in the UiMR register are set to 010b, and the IICM bit in the UiSMR register is set to 1 (i = 0 to 6). Since a transmit signal at the SDAi pin is output via the delay circuit, it changes after the SCLi pin is stably held low.


SDAi
ACKD
IICM
1
0
Delay circuit
Transmit circuit
DMA transfer request (interrupt request by UARTi transmission)
STSPSEL
ACKC
SDHI
ALS
IICM2
Interrupt request by UARTi transmission or NACK
Noise filter
D Q
T
Arbitration lost
NACK
Receive circuit
Interrupt request by UARTi reception, ACK interrupt, or DMA transfer request
ACK
Start condition detection
S Q
R
Bus busy
Interrupt request by bus collision, start condition detected, or stop condition detected
Stop condition detection
Bus collision detection (IE mode)
Start condition/ stop condition generation
SCLi
SWC2
Internal clock
9th bit
Clock control
External clock
Falling edge of the 8th bit
Q S
R
SWC
CLKi

IICM: Bit in the UiSMR register
IICM2, SWC, ALS, SWC2, and SDHI: Bits in the UiSMR2 register
STSPSEL, ACKD, and ACKC: Bits in the UiSMR4 register

Figure 18.32 I^{2}C Mode Block Diagram (i = 0 to 6)

Table 18.10 Register Settings in I²C Mode (i = 0 to 6)

Register	Bits	Function	
		Master	Slave
UiMR	IOPOL	Set the bit to 0	
	CKDIR	Set the bit to 0	Set the bit to 1
	SMD2 to SMD0	Set the bit to 010b	
UiC0	7 to 4	Set the bits to 1001b	
	TXEPT	Transmit register empty flag	
	2	Set the bit to 0	
	CLK1 and CLK0	Select a count source for the UiBRG register	Disabled
UiC1	7 to 5	Set the bits to 000b	
	UiIRS	Set the bit to 1	
	RI	Receive complete flag	
	RE	Set the bit to 1 to enable data reception	
	TI	Transmit buffer empty flag	
	TE	Set the bit to 1 to enable data transmission/reception	
UiSMR	7 to 3	Set the bits to 00000b	
	BBS	Bus busy flag	
	ABC	Select an arbitration lost detection timing	Disabled
	IICM	Set the bit to 1	
UiSMR2	7	Set the bit to 0	
	SDHI	Set the bit to 1 to disable the SDA output	
	SWC2	Set the bit to 1 to hold the SCL output at a forcible low	
	STC	Set the bit to 0	Set the bit to 1 to reset UARTi by detecting the start condition
	ALS	Set the bit to 1 to stop the output at the SDAi pin to detect an arbitration lost	Set the bit to 0
	SWC	Set the bit to 1 to hold a low output at the SCLi pin after receiving the eighth bit of the clock	
	CSC	Set the bit to 1 to enable clock synchronization	Set the bit to 0
	IICM2	Refer to Tables 18.11 and 18.12	
UiSMR3	DL2 to DL0	Set the digital delay value of SDAi	
	4 to 2	Set the bit to 000b	
	CKPH	Refer to Tables 18.11 and 18.12	
	SSE	Set the bit to 0	
UiSMR4	SWC9	Set the bit to 0	Set the bit to 1 to hold a low output at the SCLi pin after receiving the ninth bit of the clock
	SCLHI	Set the bit to 1 to stop the SCL output to detect stop condition	Set the bit to 0
	ACKC	Set the bit to 1 for ACK data output	
	ACKD	Select ACK or NACK	
	STSPSEL	Set the bit to 1 when any condition is output	Set the bit to 0
	STPREQ	Set the bit to 1 to generate a stop condition	Set the bit to 0
	RSTAREQ	Set the bit to 1 to generate a restart condition	Set the bit to 0
	STAREQ	Set the bit to 1 to generate a start condition	Set the bit to 0
UiBRG	7 to 0	Set the bit rate	Disabled
IFSR0	IFSR06 and IFSR07	Select a UART as interrupt source	
IFSR1	IFSR16	Select a UART as interrupt source	
IFS0	IFS06	Select input pins for SCL3 and SDA3	
	IFS03 and IFS02	Select input pins for SCL6 and SDA6	
UiTB	8	Set the bit to 1 when transmitting. Set the bit to the value of the ACK bit when receiving	
	7 to 0	Set the data to be transmitted when transmitting. Set the register to FFh when receiving	
UiRB	OER	Overrun error flag	
	ABT	Arbitration lost detection flag	Disabled
	8	D0 is loaded immediately after a receive interrupt occurs. ACK or NACK is loaded after a transmit interrupt occurs	
	7 to 0	D7 to D1 are read immediately after a receive interrupt occurs. D7 to D0 are read after a transmit interrupt occurs	

Table 18.11 I^{2}C Mode Functions (i = 0 to 6) (1/2)

Function	Synchronous Serial Interface Mode (SMD2 to SMD0 are 001b, IICM is 0)	I^{2}C Mode (SMD2 to SMD0 are 010b, IICM is 1)			
		IICM2 is 0 (ACK/NACK interrupt)		IICM2 is 1 (Transmit/receive interrupt)	
		CKPH is 0 (No clock delay)	CKPH is 1 (Clock delayed)	CKPH is 0 (No clock delay)	CKPH is 1 (Clock delayed)
Source of software interrupt numbers 6 and 39 to 41 [1] (refer to Figure 18.33)	—	Start condition or stop condition detection (refer to Table 18.13)			
Source of software interrupt numbers 2, 4, 17, 19, 33, 35, and 37 [1] (refer to Figure 18.33)	UARTi transmission: Transmission started or completed (selected using the UiIRS register)	NACK detection: Rising edge of the ninth bit of SCLi		UARTi transmission: Rising edge of the ninth bit of SCLi	UARTi transmission: Falling edge of the ninth bit of SCLi
Source of software interrupt numbers 3, 5, 18, 20, 34, 36, and 38 [1] (refer to Figure 18.33)	UARTi reception: Receiving at eighth bit CKPOL is 0 (rising edge) CKPOL is 1 (falling edge)	ACK detection: Rising edge of the ninth bit of SCLi		UARTi reception: Falling edge of the eighth bit of SCLi	
Data transfer timing from the UART receive register to the UiRB register	CKPOL is 0 (rising edge) CKPOL is 1 (falling edge)	Rising edge of the ninth bit of SCLi		Falling edge of the eighth bit of SCLi	Falling edge of the eighth bit and rising edge of the ninth bit of SCLi
UARTi transmit output delay	No delay	Delayed			
Pins P6_3, P6_7, P7_0, P7_3, P7_6, P9_2, P9_6, P11_0, P12_0, P15_0, and P15_4	TXDi output	SDAi I/O			
Pins P6_2, P6_6, P7_1, P7_5, P8_0, P9_1, P9_7, P11_2, P12_2, P15_2, and P15_5	RXDi input	SCLi I/O			
Pins P6_1, P6_5, P7_2, P7_4, P7_7, P9_0, P9_5, P11_1, P12_1, P15_1, and P15_6	Select CLKi input or output	— (Not used in I^{2}C mode)			

Note:

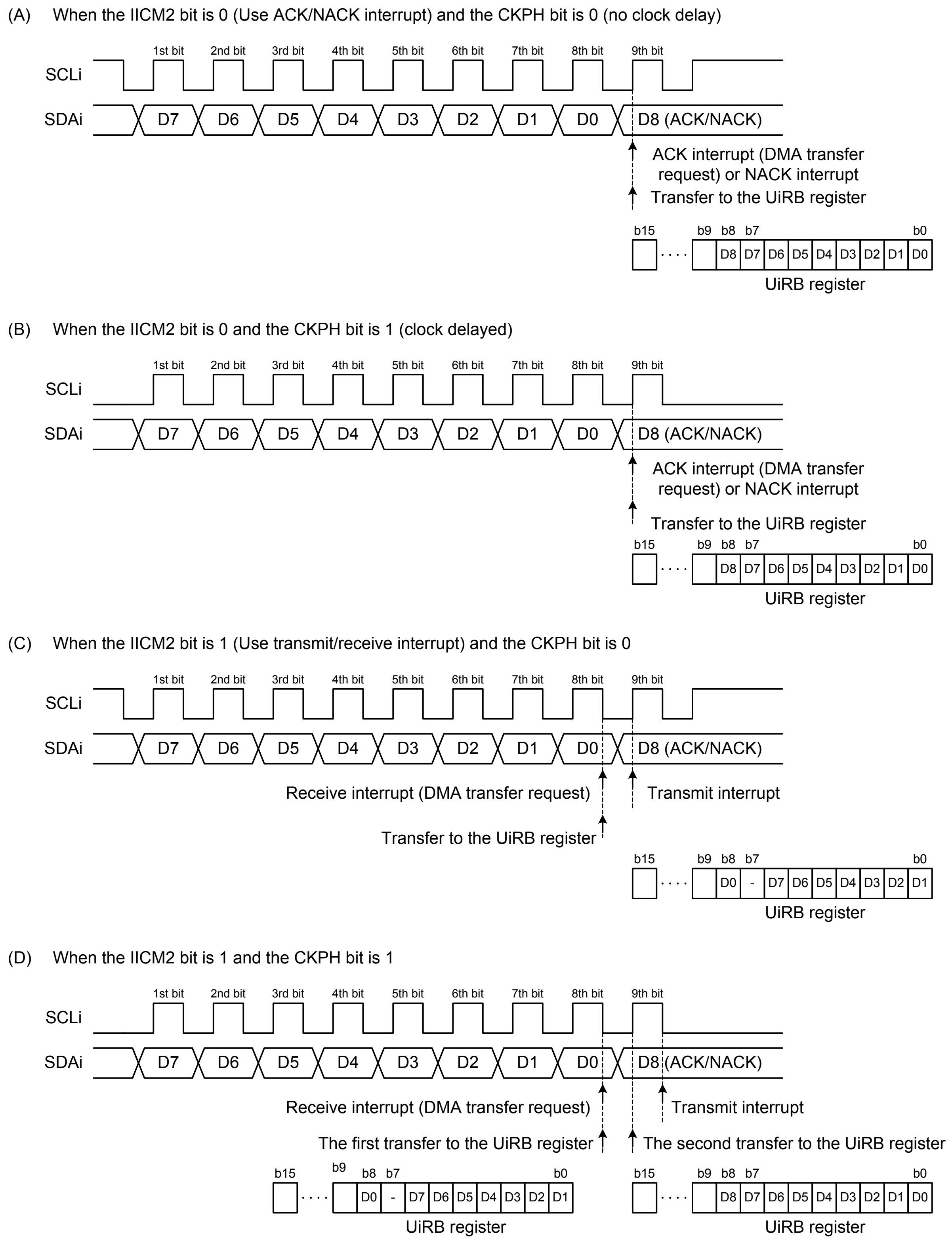
1. Steps to change an interrupt source are as follows:
   (1) Disable the interrupt of the corresponding software interrupt number.
   (2) Change the source of interrupt.
   (3) Set the IR bit of the corresponding software interrupt number to 0 (no interrupt requested).
   (4) Set bits ILVL2 to ILVL0 of the corresponding software interrupt number.

Table 18.12 I^{2}C Mode Functions (i = 0 to 6) (2/2)

Function	Synchronous Serial Interface Mode (SMD2 to SMD0 are 001b, IICM is 0)	I^{2}C Mode (SMD2 to SMD0 are 010b, IICM is 1)			
		IICM2 is 0 (ACK/NACK interrupt)		IICM2 is 1 (Transmit/receive interrupt)	
		CKPH is 0 (No clock delay)	CKPH is 1 (Clock delayed)	CKPH is 0 (No clock delay)	CKPH is 1 (Clock delayed)
Read level at pins RXDi and SCLi	Readable irrespective of the port direction bit				
Default output value at the SDAi pin	—	High (Value set in the port Pi register if the I/O port is selected by output function select registers (i = 0 to 7))			
SCLi default and end values	—	High	Low	High	Low
DMA source (refer to Figure 18.33)	UARTi reception	ACK detection		UARTi reception: Falling edge of the eighth bit of SCLi	
Store received data	The first to eighth bits of received data are stored into bits 0 to 7 in the UiRB register	The first to eighth bits of received data are stored into bits 7 to 0 in the UiRB register		The first to seventh bits of received data are stored into bits 6 to 0 in the UiRB register and the eighth bit is stored into bit 8	Same as on the left column on the first data storing. (1) The first to eighth bits of received data are stored into 7 to 0 bits in the UiRB register and the ninth bit is stored into bit 8 on the second data storing (2)
Read received data	The UiRB register status is read as it is			Bits 6 to 0 in the UiRB register are read as bits 7 to 1 and bit 8 is read as bit 0	Same as on the left column on the first read. (1) The UiRB register status is read as it is on the second read (2)

Notes:

1. The first data transfer to the UiRB register starts on the rising edge of the eighth bit of SCLi.
2. The second data transfer to the UiRB register starts on the rising edge of the ninth bit of SCLi.


(A) When the IICM2 bit is 0 (Use ACK/NACK interrupt) and the CKPH bit is 0 (no clock delay)
1st bit
2nd bit
3rd bit
4th bit
5th bit
6th bit
7th bit
8th bit
9th bit
SCLi
SDAi
D7
D6
D5
D4
D3
D2
D1
D0
D8 (ACK/NACK)
ACK interrupt (DMA transfer request) or NACK interrupt
Transfer to the UiRB register
b15
b9
b8
b7
b0
D8 D7 D6 D5 D4 D3 D2 D1 D0
UiRB register
(B) When the IICM2 bit is 0 and the CKPH bit is 1 (clock delayed)
ACK interrupt (DMA transfer request) or NACK interrupt
Transfer to the UiRB register
UiRB register
(C) When the IICM2 bit is 1 (Use transmit/receive interrupt) and the CKPH bit is 0
Receive interrupt (DMA transfer request)
Transmit interrupt
Transfer to the UiRB register
D0 - D7 D6 D5 D4 D3 D2 D1
UiRB register
(D) When the IICM2 bit is 1 and the CKPH bit is 1
Receive interrupt (DMA transfer request)
Transmit interrupt
The first transfer to the UiRB register
The second transfer to the UiRB register
UiRB register

This figure applies under the following condition:
- The CKDIR bit in the UiMR register is 0 (internal clock).

Figure 18.33 Timings for the Transfer and Interrupt to the UiRB Register (i = 0 to 6)

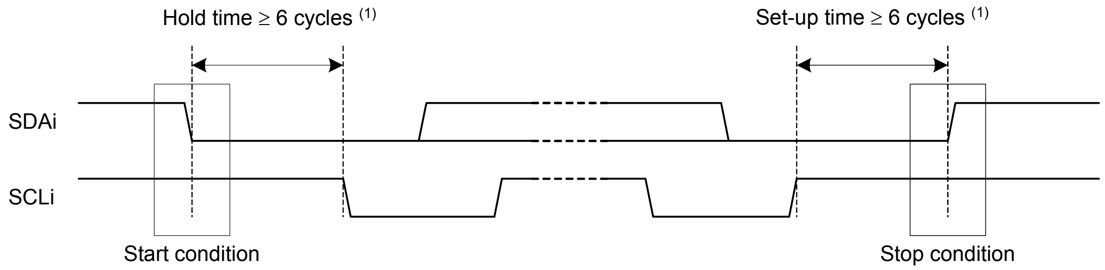
18.3.1 Start Condition and Stop Condition Detection

The start condition and stop condition are detected by their respective detectors.

The start condition detection interrupt request is generated by a high-to-low transition at the SDAi pin while the SCLi pin is held high (i = 0 to 6). The stop condition detection interrupt request is generated by a low-to-high transition at the SDAi pin while the SCLi pin is held high.

The start condition detection interrupt shares interrupt control registers and vectors with the stop condition detection interrupt. The BBS bit in the UiSMR register determines which interrupt is requested.

To detect a start condition or stop condition, both set-up and hold times require at least six cycles of the peripheral clock (f1) as shown in Figure 18.34. To meet the condition for the Fast-mode specification, f1 must be at least 10 MHz.



Note:
1. These are cycles of the peripheral clock (f1).

Figure 18.34 Start Condition and Stop Condition Detection Timing (i = 0 to 6)

18.3.2 Start Condition and Stop Condition Generation

The start condition, restart condition, and stop condition are generated by bits STAREQ, RSTAREQ, and STPREQ in the UiSMR4 register, respectively (i = 0 to 6). To output a start condition, set the STSPSEL bit in the UiSMR4 register to 1 (select start condition/stop condition generation circuit) after setting the STAREQ bit to 1 (start). To output a restart condition or stop condition, set the STSPSEL bit to 1 after setting RSTAREQ bit or STPREQ bit to 1, respectively.

Table 18.13 and Figure 18.35 show the functions of the STSPSEL bit.

Table 18.13 STSPSEL Bit Functions

Function	STSPSEL is 0	STSPSEL is 1
Start condition and stop condition generation	Output is provided by the program with port (no auto generation by hardware)	Start condition or stop condition is output according to the STAREQ, RSTAREQ, or STPREQ bit
Start condition and stop condition interrupt request generating timing	When start condition or stop condition is detected	When start condition or stop condition generation is completed

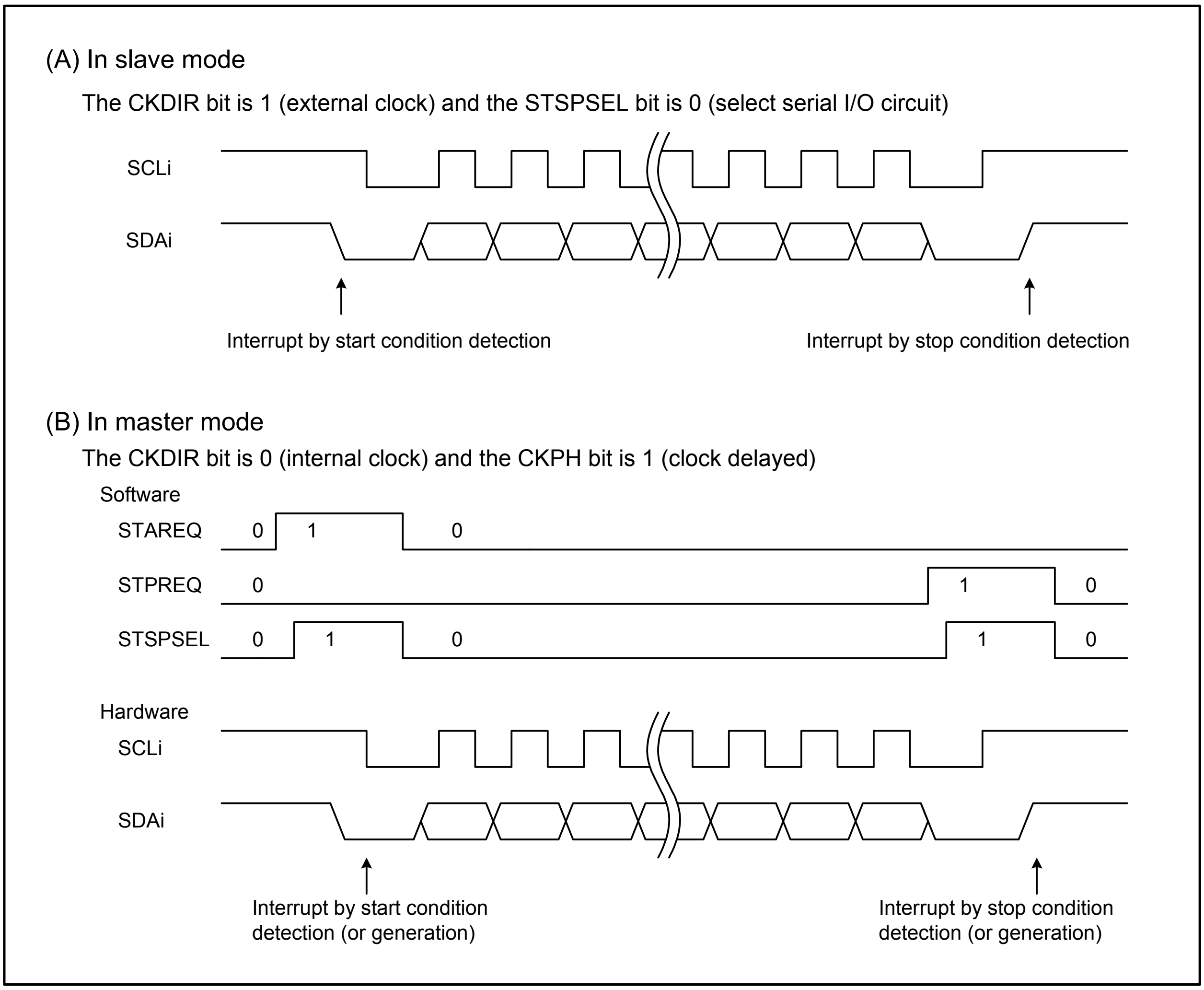

(A) In slave mode
The CKDIR bit is 1 (external clock) and the STSPSEL bit is 0 (select serial I/O circuit)
SCLi
SDAi
Interrupt by start condition detection
Interrupt by stop condition detection
(B) In master mode
The CKDIR bit is 0 (internal clock) and the CKPH bit is 1 (clock delayed)
Software
STAREQ
0
1
0
STPREQ
0
1
0
STSPSEL
0
1
0
1
0
Hardware
SCLi
SDAi
Interrupt by start condition detection (or generation)
Interrupt by stop condition detection (or generation)

Figure 18.35 STSPSEL Bit Functions (i = 0 to 6)

18.3.3 Arbitration

On the rising edge of the SCLi, the MCU compares the transmit data with the data input from the SDAi pin. If no match is found, the MCU performes arbitration by stopping the SDAi output.

The update timing for the ABT bit in the UiRB register is selected by setting the ABC bit in the UiSMR register (i = 0 to 6).

When the ABC bit is 0 (update every bit), the ABT bit becomes 1 (detected (lose)) as soon as a data discrepancy is detected. If not detected, the ABT bit becomes 0 (not detected (win)). When the ABC bit is 1 (update every byte), the ABT bit becomes 1 on the falling edge of the eighth bit of the SCLi if any discrepancy is detected. In this ABC bit setting, set the ABT bit to 0 to start the next 1-byte transfer.

When the ALS bit in the UiSMR2 register is 1 (stop the SDAi output), the SDAi pin becomes high-impedance as the ABT bit becomes 1 when an arbitration lost occurs.

18.3.4 SCL Control and Clock Synchronization

Data transmission/reception in I^{2}C mode uses the transmit/receive clock as shown in Figure 18.33. The clock speed increase makes it difficult to secure the required time for ACK generation and data transmit procedure. I^{2}C mode supports a function of wait-state insertion to secure this required time and a function of clock synchronization with a wait-state inserted by other devices.

The SWC bit in the UiSMR2 register is used to insert a wait-state for ACK generation (i = 0 to 6). When the SWC bit is 1 (hold the SCLi pin low after the eighth bit is received), the SCLi pin is held low on the falling edge of the eighth bit of the SCLi. When the SWC bit is 0 (no wait-state/wait-state cleared), the SCLi line is released.

When the SWC2 bit in the UiSMR2 register is 1 (hold the SCLi pin low), the SCLi pin is forced low even during transmission or reception. When the SWC2 bit is 0 (output the transmit/receive clock at the SCLi pin), the SCLi line is released to output the transmit/receive clock.

The SWC9 bit in the UiSMR4 register is used to insert a wait-state for checking received acknowledge bits. While the CKPH bit in the UiSMR3 register is 1 (clock delayed), when the SWC9 bit is set to 1 (hold the SCLi pin low after the ninth bit is received), the SCLi pin is held low on the falling edge of the ninth bit of the SCLi. When the SWC9 bit is set to 0 (no wait-state/wait-state cleared), the SCLi line is released.

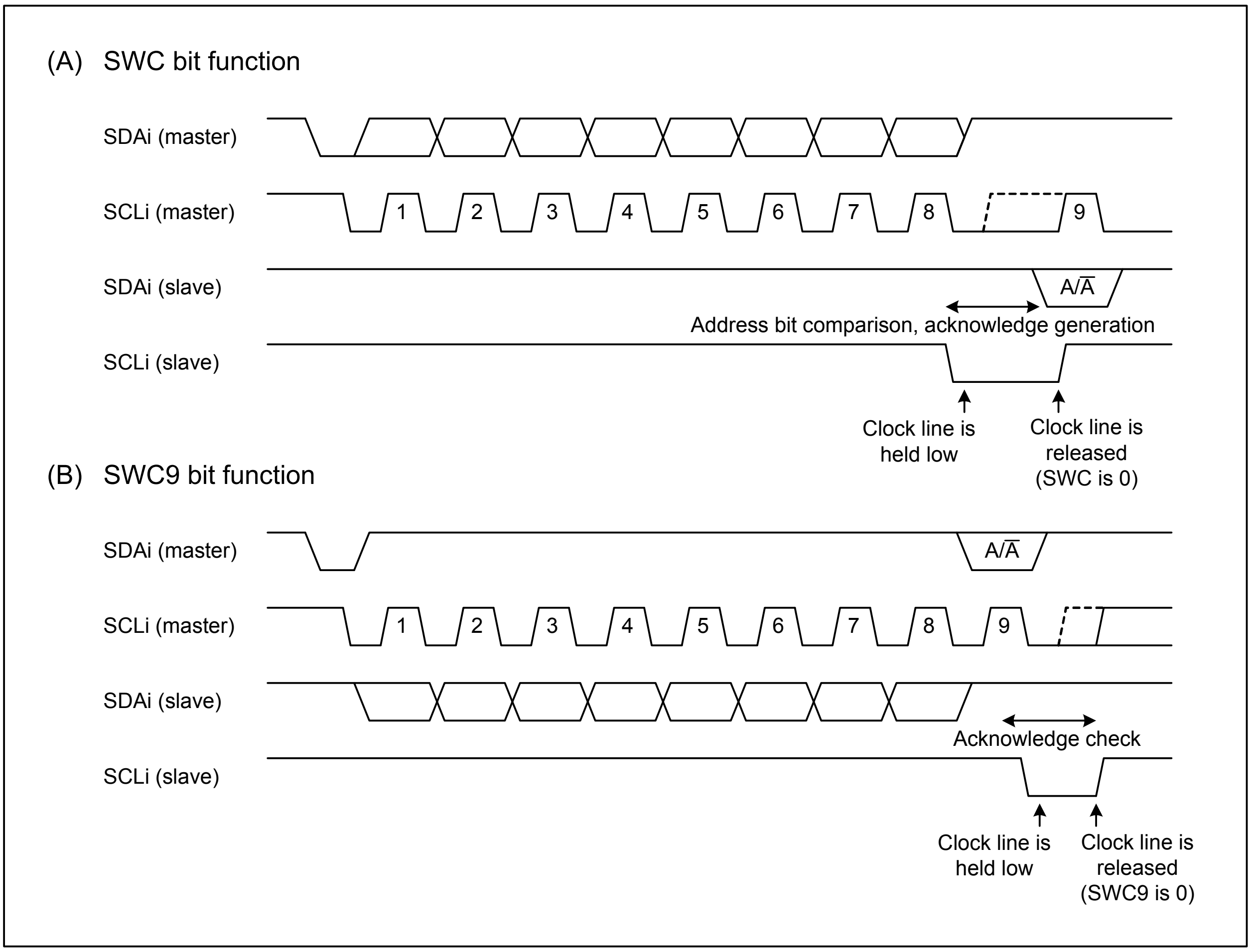


Figure 18.36 Wait-state Insertion Using the SWC or SWC9 Bit (i = 0 to 6)

The CSC bit in the UiSMR2 register is used to synchronize an internally generated clock with the clock applied to the SCLi pin. For example, if a wait-state is inserted from another device, the two clocks are not synchronized. While the CSC bit is 1 (clock synchronization enabled) and the internal clock is held high, when a high at the SCLi pin changes to low, the internal clock becomes low in order to reload the value of the UiBRG register and to resume counting. While the SCLi pin is held low, when the internal clock changes from low to high, the count is stopped until the SCLi pin becomes high. That is, the UARTi transmit/receive clock is the logical AND of the internal clock and the SCLi. The synchronized period starts from one clock prior to the first synchronized clock and ends when the ninth clock is completed. The CSC bit can be set to 1 only when the CKDIR bit in the UiMR register is 0 (internal clock).

The SCLHI bit in the UiSMR4 register is used to leave the SCLi pin open when another master generates a stop condition while the master is in transmit/receive operation. If the SCLHI bit is set to 1 (stop SCLi output), the SCLi pin is open (the pin is high-impedance) when a stop condition is detected and the clock output is stopped.

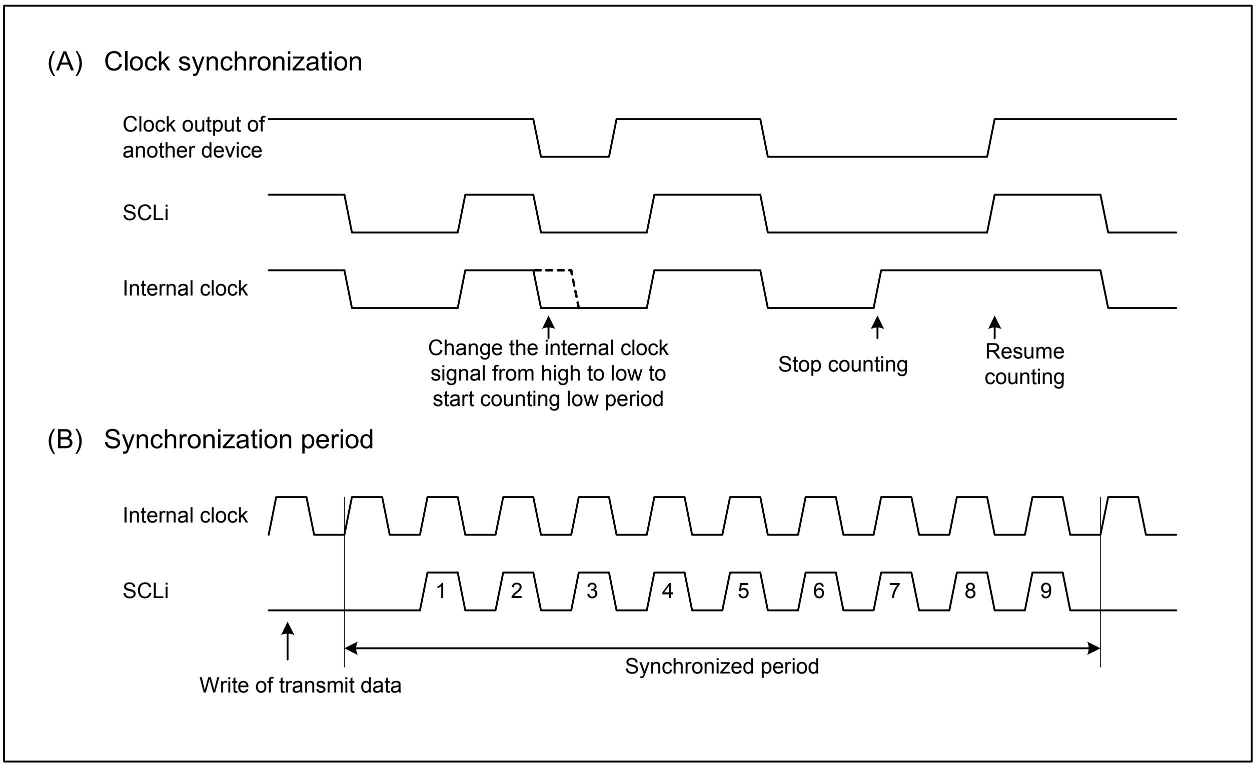

(A) Clock synchronization
Clock output of another device
SCLi
Internal clock
Change the internal clock signal from high to low to start counting low period
Stop counting
Resume counting
(B) Synchronization period
Internal clock
SCLi
1
2
3
4
5
6
7
8
9
Write of transmit data
Synchronized period

Figure 18.37 Clock Synchronization (i = 0 to 6)

18.3.5 SDA Output

Values set to bits 8 to 0 (D8 to D0) in the UiTB register are output starting from D7 to D0, and lastly D8, which is a bit for the acknowledge signal (i = 0 to 6). When transmitting, D8 should be set to 1 to free the bus. When receiving, D8 should be set to ACK or NACK.

Bits DL2 to DL0 in the UiSMR3 register set a delay time of the SDAi on the falling edge of the SCLi. Based on the UiBRG count source, the delay time can be selected from zero cycles (no delay) or two to eight cycles.

The SDAi pin can be high-impedance at any given time once the SDHI bit in the UiSMR2 register is set to 1 (stop the output). Output at the SDAi pin is low if an I/O port is selected for the SDAi and the pin is specified as the output port after selecting I^2C mode. In this case, if the SDHI bit is 1, the SDAi pin becomes high-impedance.

When the SDHI bit is rewritten while the SCLi pin is held high, a start condition or stop condition is generated. When it is rewritten immediately before the rising edge of SCLi, arbitration lost may be accidently detected. Therefore, the SDHI bit should be rewritten so the SDAi pin level changes while the SCLi pin is low.

18.3.6 SDA Input

When the IICM2 bit in the UiSMR2 register is 0 (use ACK/NACK interrupt), the first 8 bits of received data (D7 to D0) are stored into bits 7 to 0 in the UiRB register and the ninth bit (ACK/NACK) is stored into bit 8 (i = 0 to 6).

When the IICM2 bit is 1, the first 7 bits of received data (D7 to D1) are stored into bits 6 to 0 in the UiRB register and eighth bit (D0) is stored into bit 8.

If the IICM2 bit is 1 and the CKPH bit in the UiSMR3 register is 1 (clock delayed), the same data that is set when the IICM2 bit is 0 can be read. To read this data, read the UiRB register after data in the ninth bit is latched on the rising edge of the SCLi.

18.3.7 Acknowledge

When data is to be received in master mode, ACK is output after 8 bits are received by setting the UiTB register to 00FFh as dummy data. When the STSPSEL bit in the UiSMR4 register is 0 (select serial I/O circuit) and the ACKC bit is 1 (ACK data output), the value of the ACKD bit is output at the SDAi pin (i = 0 to 6).

If the IICM2 bit is 0, a NACK interrupt request is generated when the SDAi pin is high on the rising edge of the ninth bit of the SCLi. An ACK interrupt request is generated when the SDAi pin is low.

When the DMA request source is “UARTi receive interrupt request or ACK interrupt request”, the DMA transfer starts when an ACK is detected.

18.3.8 Initialization of Transmit/Receive Operation Reset

When the CKDIR bit in the UiMR register is 1 (external clock), the STC bit in the UiSMR2 register is 1 (initialize the circuit), and a start condition is detected, the following three operations are performed (i = 0 to 6):

- The transmit register is reset and the UiTB register value is transferred to the transmit register. New data transmission starts on the falling edge of the first bit of the next SCLi as transmit clock. The transmit register value before the reset is output at the SDAi pin in the period from the falling edge of the SCLi until the first data output.
- The receive register is reset and the new data reception starts on the falling edge of the first bit of the next SCLi.
- The SWC bit in the UiSMR2 register becomes 1 (hold the SCLi pin low after the eighth bit is received).

The TI bit in the UiC1 register does not change when using this function to start the UARTi transmission/reception.

18.4 Special Mode 2

Special mode 2 enables serial communication between one or multiple masters and multiple slaves. The $\overline{\text{SSi}}$ input pin controls serial bus communication (i = 0 to 6). Table 18.14 lists specifications of special mode 2.

Table 18.14 Special Mode 2 Specifications

Item	Specification
Data format	8-bit character length
Transmit/receive clock	• The CKDIR bit in the UiMR register is set to 0 (internal clock) (i = 0 to 6): $\frac{fx}{2(m+1)}$ fx = f1, f8, f2n m: UiBRG register setting value, 00h to FFh • The CKDIR bit is set to 1 (external clock): input to the CLKi pin
Transmit/receive control	SS function
Transmit start conditions	The conditions for starting data transmission are as follows (1): • The TE bit in the UiC1 register is 1 (transmission enabled) • The TI bit in the UiC1 register is 0 (data held in the UiTB register)
Receive start conditions	The conditions for starting data reception are as follows (1): • The RE bit in the UiC1 register is 1 (reception enabled) • The TE bit in the UiC1 register is 1 (transmission enabled) • The TI bit in the UiC1 register is 0 (data held in the UiTB register)
Interrupt request generating timing	In transmit interrupt, one of the following conditions can be selected by setting the UiIRS bit in registers U0C1 to U6C1: • The UiIRS bit is 0 (transmit buffer is empty): when data is transferred from the UiTB register to the UARTi transmit register (when the transmission has started) • The UiIRS bit is 1 (transmission is completed): when data transmission from the UARTi transmit register is completed In receive interrupt, • When data is transferred from the UARTi receive register to the UiRB register (when the reception is completed)
Error detection	Overrun error (2) This error occurs when the seventh bit of the next data has been received before reading the UiRB register
Other functions	• CLK polarity Rising or falling edge of the transmit/receive clock for transfer data input and output • Bit order selection LSB first or MSB first • Continuous receive mode Data reception is enabled by a read access to the UiRB register • Serial data logic inversion This function logically inverses transmit/receive data • Clock phase selection One of four combinations of transmit/receive clock polarity and phases • $\overline{\text{SSi}}$ input pin function Output pin can be high-impedance when the $\overline{\text{SSi}}$ pin is high

Notes:

1. When selecting an external clock, the following preconditions should be met:
   - The CLKi pin is held high when the CKPOL bit in the UiC0 register is 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge).
   - The CLKi pin is held low when the CKPOL bit is 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge).
2. The UiRB register is undefined when an overrun error occurs. The IR bit in the SiRIC register does not change to 1 (interrupt requested).

Table 18.15 lists register settings in special mode 2.

Table 18.15 Register Settings in Special Mode 2 (i = 0 to 6)

Register	Bits	Function
UiMR	7 to 4	Set the bits to 0000b
	CKDIR	Set the bit to 0 in master mode and set it to 1 in slave mode
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Clock phase can be set by the combination of bits CKPOL and CKPH in the UiSMR3 register
	5	Set the bit to 0
	CRD	Set the bit to 1
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	7 and 6	Set the bits to 00b
	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select a source for UARTi transmit interrupt
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
UiSMR	7 to 0	Set the bits to 00h
UiSMR2	7 to 0	Set the bits to 00h
UiSMR3	7 to 5	Set the bits to 000b
	ERR	Mode fault flag
	3	Set the bit to 0
	DINC	Set to 0 in master mode and set to 1 in slave mode
	CKPH	Clock phase can be set by the combination of bits CKPH and CKPOL in the UiC0 register
	SSE	Set the bit to 1
UiSMR4	7 to 0	Set the bits to 00h
UiBRG	7 to 0	Set the bit rate
IFS0	IFS06	Select input pins for CLK3, RXD3, SRXD3, and $\overline{\text{SS3}}$
	IFS03 and IFS02	Select input pins for CLK6, RXD6, SRXD6, and $\overline{\text{SS6}}$
UiTB	7 to 0	Set the data to be transmitted
UiRB	OER	Overrun error flag
	7 to 0	Received data is read

18.4.1 $\overline{\text{SSi}}$ Input Pin Function (i = 0 to 6)

Special mode 2 is selected by setting the SSE bit in the UiSMR3 register to 1 (SS function enabled). The $\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$/$\overline{\text{SSi}}$ pin functions as $\overline{\text{SSi}}$ input.

The DINC bit in the UiSMR3 register determines which MCU performs as a master or slave.

When multiple MCUs perform as masters (multi-master system), the $\overline{\text{SSi}}$ pin setting determines which master MCU is active and when.

18.4.1.1 SS Function in Slave Mode

When the DINC bit is 1 (slave mode) while input at the $\overline{\text{SSi}}$ pin is high, the STXDi pin becomes high-impedance and the clock input at the CLKi pin is ignored. When input at the $\overline{\text{SSi}}$ pin is low, the clock input is valid and serial data is output from the STXDi pin to enable serial communication.

18.4.1.2 SS Function in Master Mode

When the DINC bit is 0 (master mode) while input at the $\overline{\text{SSi}}$ pin is high, which means there is the only one master MCU or no other master MCU is active, the MCU as master starts communication. The master provides the transmit/receive clock output at the CLKi pin. When input at the $\overline{\text{SSi}}$ pin is low, which means that there are more masters, pins TXDi and CLKi become high-impedance. This error is called a mode fault. It can be verified using the ERR bit in the UiSMR3 register. The ongoing data transmission/reception does not stop even if a mode fault occurs. To stop transmission/reception, bits SMD2 to SMD0 in the UiMR register should be set to 000b (serial interface disabled).

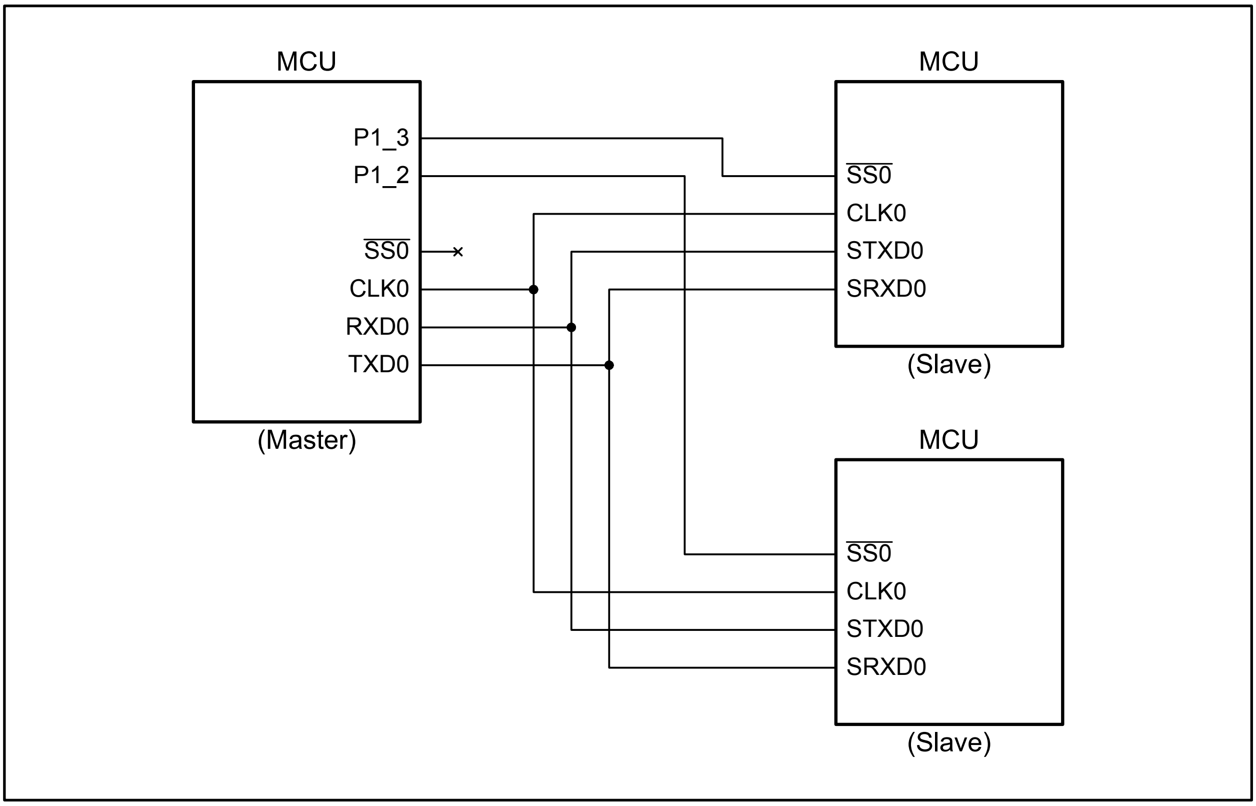


Figure 18.38 Serial Bus Communication Control with the $\overline{\text{SSi}}$ Pin

18.4.2 Clock Phase Setting

The CKPH bit in the UiSMR3 register and the CKPOL bit in the UiC0 register select one of four combinations of transmit/receive clock polarity and serial clock phase (i = 0 to 6).

The transmit/receive clock phase and polarity should be identical for the master device and the communicating slave device.

18.4.2.1 Transmit/Receive Timing in Master Mode

When the DINC bit is 0 (master mode), the CKDIR bit in the UiMR register should be set to 0 (internal clock) to generate the clock. Figure 18.39 shows transmit/receive timing of each clock phase.

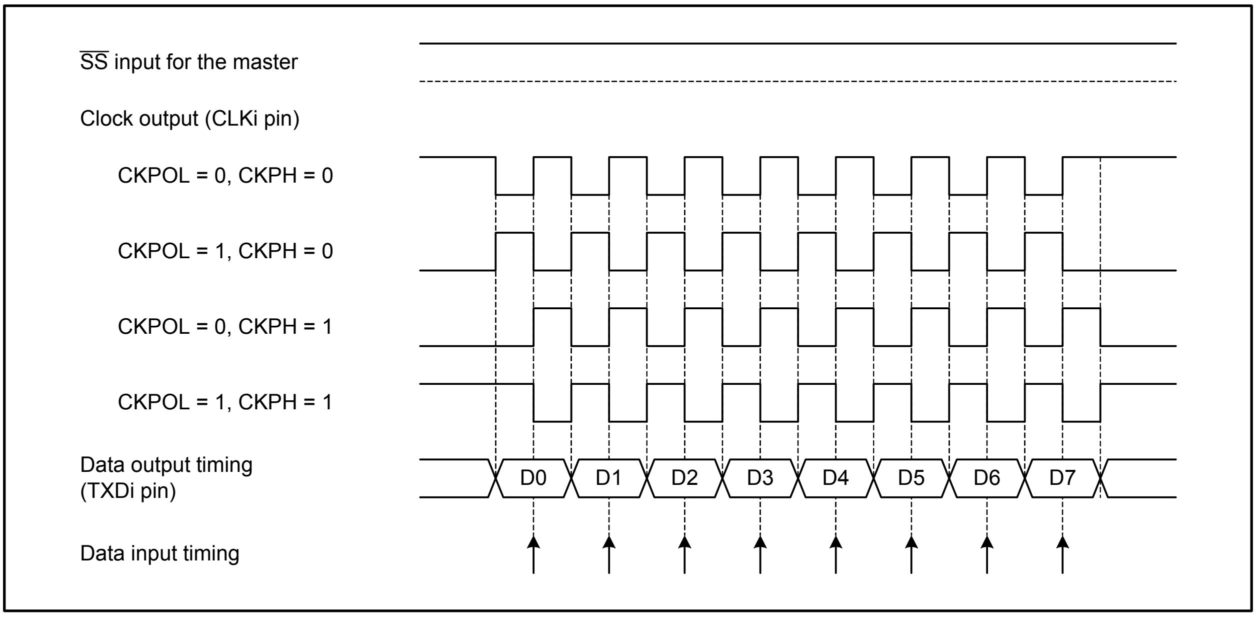


Figure 18.39 Transmit/Receive Timing in Master Mode

18.4.2.2 Transmit/Receive Timing in Slave Mode

When the DINC bit is 1 (slave mode), the CKDIR bit in the UiMR register should be set to 1 (external clock).

When the CKPH bit is 0 (no clock delay) while input at the $\overline{\text{SSi}}$ pin is high, the STXDi pin becomes high-impedance. When input at the $\overline{\text{SSi}}$ pin is low, the conditions for data transmission are all met, but output is undefined. Then the data transmission/reception starts synchronizing with the clock. Figure 18.40 shows the transmit/receive timing.

When the CKPH bit is 1 (clock delayed) while input at the $\overline{\text{SSi}}$ pin is high, the STXDi pin becomes high-impedance. When input at the $\overline{\text{SSi}}$ pin is low, the first data is output. Then the data transmission starts synchronizing with the clock. Figure 18.41 shows the transmit/receive timing.

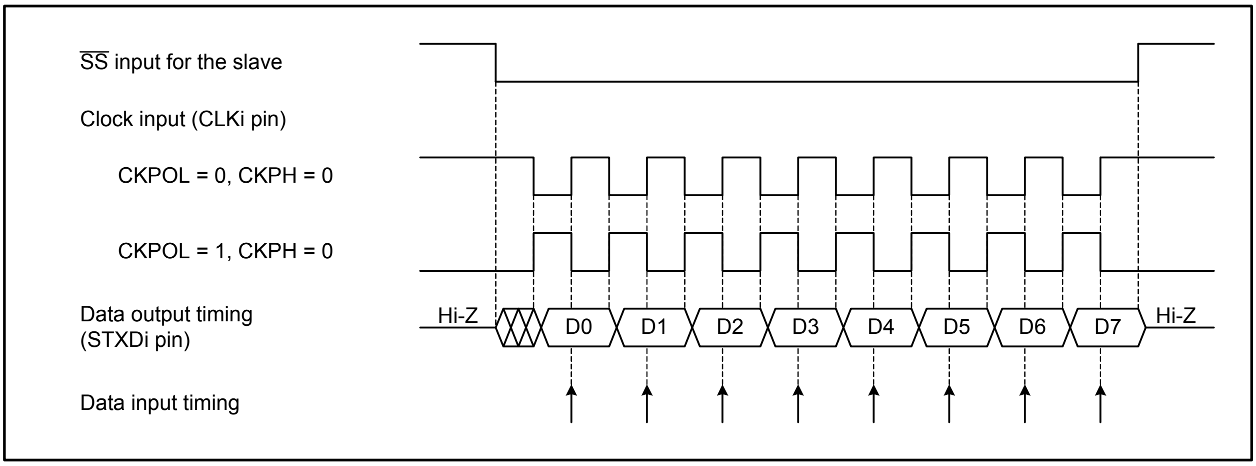


Figure 18.40 Transmit/Receive Timing in Slave Mode (CKPH = 0)

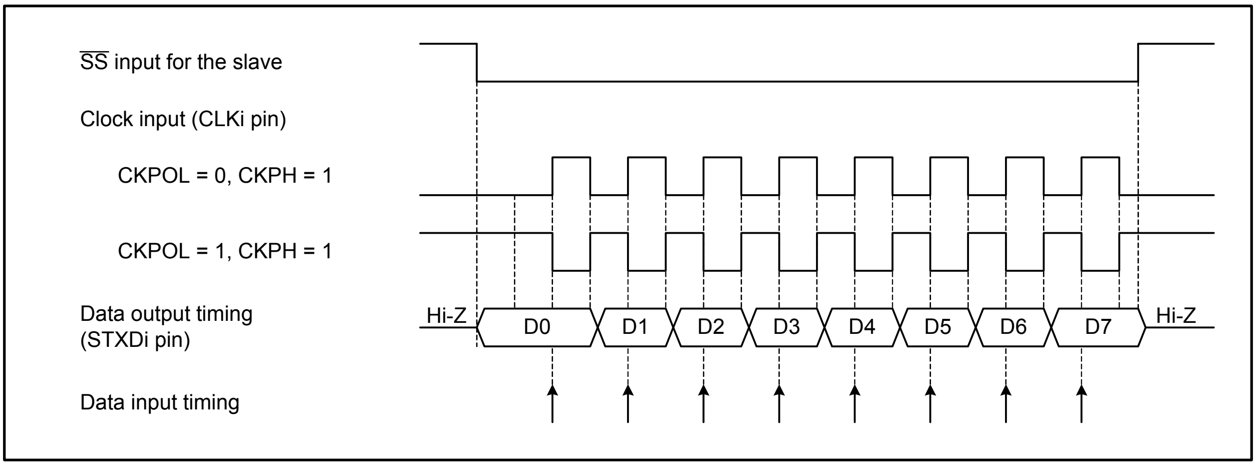


Figure 18.41 Transmit/Receive Timing in Slave Mode (CKPH = 1)

18.5 Notes on Serial Interface

18.5.1 Changing the UiBRG Register (i = 0 to 10)

- Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When these bits are changed, the UiBRG register must be set again.
- When a clock is input immediately after the UiBRG register is set to 00h, the counter may become FFh. In this case, it requires extra 256 clocks to reload 00h to the register. Once 00h is reloaded, the counter performs the operation without dividing the count source according to the setting.

18.5.2 Synchronous Serial Interface Mode

18.5.2.1 Selecting an External Clock

- If an external clock is selected, the following conditions must be met while the external clock is held high when the CKPOL bit in the UiC0 register is 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge), or while the external clock is held low when the CKPOL bit is 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge) (i = 0 to 10):
  - The TE bit in the UiC1 register is 1 (transmission enabled).
  - The RE bit in the UiC1 register is 1 (reception enabled). This bit setting is not required when only transmitting.
  - The TI bit in the UiC1 register is 0 (data held in the UiTB register).

18.5.2.2 Receive Operation

- In synchronous serial interface mode, the transmit/receive clock is controlled by the transmit control circuit. Set UARTi-associated registers for a transmit operation, even if the MCU is used only for receive operation (i = 0 to 10). Dummy data is output from the TXDi pin while receiving when the TXDi pin is set to output mode.

- When data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data held in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. Then, the OER bit in the UiRB register becomes 1 (overrun error occurred). In this case, the UiRB register becomes undefined. If an overrun error occurs, the IR bit in the SiRIC register does not change to 1.

18.5.3 Special Mode 1 (I^2C Mode)

- To generate a start condition, stop condition, or restart condition, set the STSPSEL bit in the UiSMR4 register to 0 (i = 0 to 6). Then, wait at least a half clock cycle of the transmit/receive clock to change the condition generate bits (STAREQ, RSTAREQ, or STPREQ bit) from 0 to 1.

18.5.4 Reset Procedure on Communication Error

Operations which result in communication errors such as rewriting function select registers during transmission/reception should not be performed. Follow the procedure below to reset the internal circuit once the communication error occurs in the following cases: when the operation above is performed by a receiver or transmitter or when a bit slip is caused by noise.

A. Synchronous Serial Interface Mode

(1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled) (i = 0 to 10).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode).
(4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

B. UART Mode

(1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 100b (UART mode, 7-bit character length), 101b (UART mode, 8-bit character length), or 110b (UART mode, 9-bit character length).
(4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

19. A/D Converter

The A/D converter consists of one 10-bit successive approximation A/D converter with a capacitive coupling amplifier.

A/D converted results are stored in the A/D registers corresponding to selected pins. Results are stored in the AD00 register only when the DMAC operating mode is enabled.

When the A/D converter is not in use, power consumption can be reduced by setting the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). This bit setting enables the power supply from the VREF pin to the resistor ladder to stop.

Table 19.1 lists specifications of the A/D converter. Figure 19.1 shows a block diagram of the A/D converter.

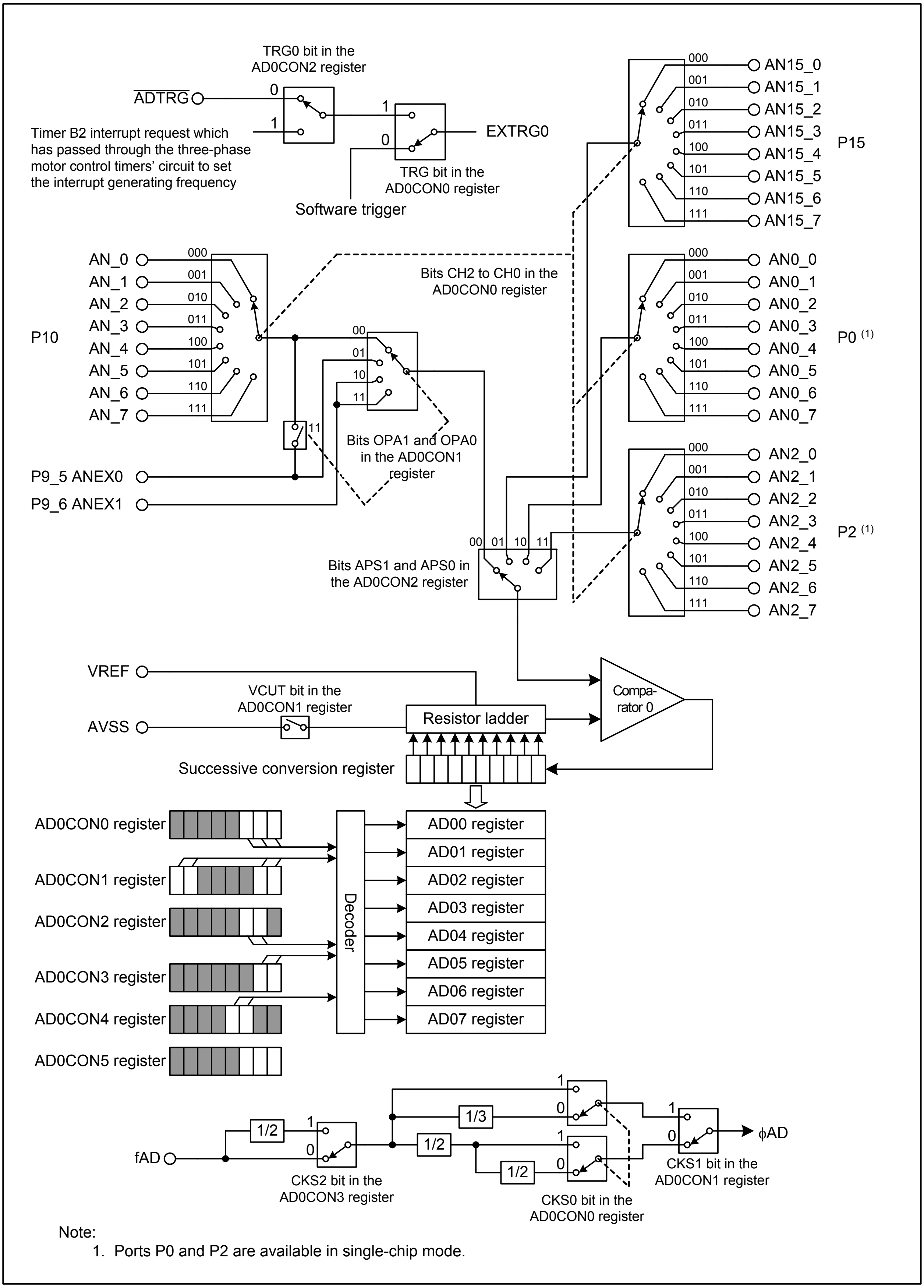
Figures 19.2 to 19.8 show registers associated with the A/D converter.

Table 19.1 A/D Converter Specifications

Item	Specification
A/D conversion method	Capacitance-based successive approximation
Analog input voltage (1)	0 V to AVCC (VCC)
Operating clock, ϕAD (2)	fAD, fAD divided by 2, fAD divided by 3, fAD divided by 4, fAD divided by 6, or fAD divided by 8
Resolution	8 bits or 10 bits
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweet mode 0, repeat sweep mode 1, multi-port single sweep mode, multi-port repeat sweep mode 0, and self test mode
Analog input pins (3)	34 8 pins each for AN, AN0, AN2, and AN15 2 function-extended input pins (ANEX0 and ANEX1)
A/D conversion start conditions	• Software trigger The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program • External trigger (retrigger is enabled) An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program • Hardware trigger (retrigger is enabled) • Generation of a timer B2 interrupt request which has passed through the circuit to set an interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program
Conversion rates per pin	• Without sample and hold function 49 ϕAD cycles at 8-bit resolution 59 ϕAD cycles at 10-bit resolution including 2 ϕAD cycles for sampling time • With sample and hold function 28 ϕAD cycles at 8-bit resolution 33 ϕAD cycles at 10-bit resolution including 3 ϕAD cycles for sampling time

Notes:

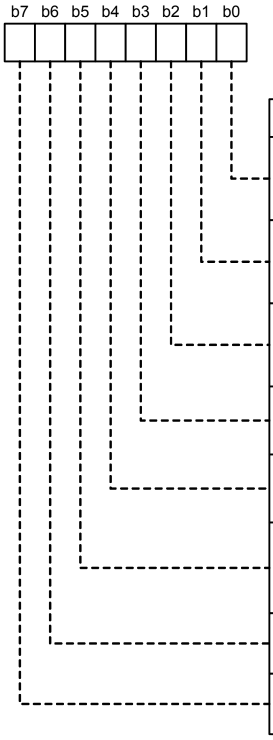
1. The analog input voltage is not dependent on whether the sample and hold function is enabled or disabled.
2. The ϕAD frequency should be as follows:
   - When VCC = 4.2 to 5.5 V, 16 MHz or below
   - When VCC = 3.0 to 4.2 V, 10 MHz or below
   - When not using the sample and hold function, 250 kHz or above
   - When using the sample and hold function, 1 MHz or above
3. When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1 should be VCC or lower.



Note:
1. Ports P0 and P2 are available in single-chip mode.

Figure 19.1 A/D Converter Block Diagram

A/D0 Control Register 0 [1]



Symbol
AD0CON0 | Address
0396h | Reset Value
0000 0000b

Symbol	Address	Reset Value
AD0CON0	0396h	0000 0000b

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit [2, 3, 4]	b2 b1 b0 0 0 0 : ANi_0 0 0 1 : ANi_1 0 1 0 : ANi_2 0 1 1 : ANi_3 1 0 0 : ANi_4 1 0 1 : ANi_5 1 1 0 : ANi_6 1 1 1 : ANi_7 (i = no value, 0, 2, 15)	RW
CH1			RW
CH2			RW
MD0	A/D Operating Mode Select Bit 0 [2, 5, 6]	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or 1	RW
MD1			RW
TRG	Trigger Select Bit	0: Software trigger 1: External trigger or hardware trigger [7]	RW
ADST	A/D Conversion Start Bit	0: A/D conversion stopped 1: A/D conversion started [7]	RW
CKS0	Frequency Select Bit	(See Note 8)	RW

Notes:

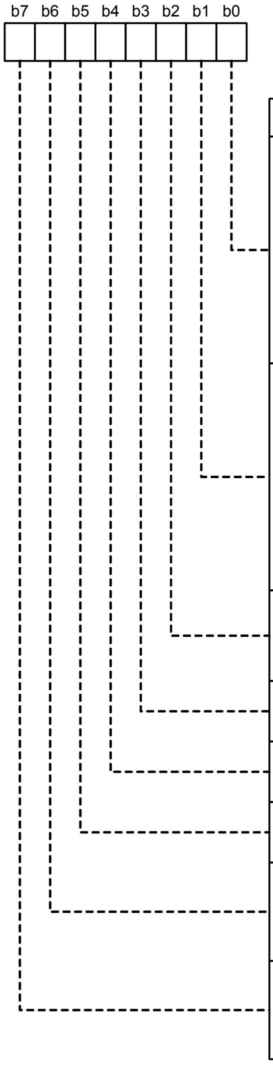
1. When this register is rewritten during an A/D conversion, the converted result is undefined.
2. Set the analog input pins again after changing the A/D operating mode.
3. This bit setting is enabled in one-shot mode or repeat mode.
4. Select a port from AN, AN0, AN2, or AN15 by using bits APS1 and APS0 in the AD0CON2 register.
5. When the MSS bit in the AD0CON3 register is 1 (multi-port sweep mode enabled), set bits MD1 and MD0 to 10b for multi-port single sweep mode and 11b for multi-port repeat sweep mode 0.
6. Set bits MD1 and MD0 to 10b or 11b when the MSS bit in the AD0CON3 register is 1.
7. To use the external trigger or the hardware trigger, select the source of the trigger by setting the TRG0 bit in the AD0CON2 register. Then set the ADST bit to 1 after setting the TRG bit to 1.
8. The ϕAD frequency should be as follows: 16 MHz or below when VCC = 5 V,
   10 MHz or below when VCC = 3.3 V

   The ϕAD frequency is selected from the combination of bits CKS0, CKS1, and CKS2 shown as below:

CKS2 Bit in the AD0CON3 Register	CKS0 Bit in the AD0CON0 Register	CKS1 Bit in the AD0CON1 Register	ϕAD
0	0	0	fAD divided by 4
		1	fAD divided by 3
	1	0	fAD divided by 2
		1	fAD
1	0	0	fAD divided by 8
		1	fAD divided by 6

Figure 19.2 AD0CON0 Register

A/D0 Control Register 1 (1)



Symbol	Address	Reset Value
AD0CON1	0397h	0000 0000b

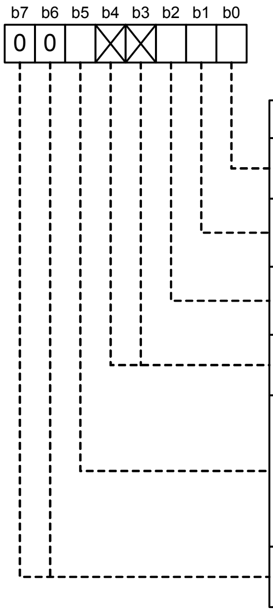
Bit Symbol	Bit Name	Function	RW
SCAN0	A/D Sweep Pin Select Bit (2, 3)	In single sweep mode or repeat sweep mode 0 b1 b0 0 0 : ANi_0, ANi_1 0 1 : ANi_0 to ANi_3 1 0 : ANi_0 to ANi_5 1 1 : ANi_0 to ANi_7 In repeat sweep mode 1 (4) b1 b0 0 0 : ANi_0 0 1 : ANi_0, ANi_1 1 0 : ANi_0 to ANi_2 1 1 : ANi_0 to ANi_3 In multi-port single sweep mode or multi-port repeat sweep mode 0 b1 b0 1 1 : ANi_0 to ANi_7 (5) (i = no value, 0, 2, 15)	RW
SCAN1			RW
MD2	A/D Operating Mode Select Bit 1	0: Mode other than repeat sweep mode 1 1: Repeat sweep mode 1(6)	RW
BITS	8/10-bit Mode Select Bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency Select Bit	(See Note 7)	RW
VCUT	VREF Connection Bit (8)	0: VREF disconnected (9) 1: VREF connected (10)	RW
OPA0	External Op-Amp Connect Mode Bit (11, 12)	b7 b6 0 0 : No use of ANEX0 or ANEX1 pin (convert input at pins ANi_0 to ANi_7) 0 1 : Convert input at the ANEX0 pin 1 0 : Convert input at the ANEX1 pin 1 1 : External op-amp connected	RW
OPA1			RW

Notes:

1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. This bit setting is enabled in single sweep mode, repeat sweep mode 0, repeat sweep mode 1, multi-port single sweep mode, or multi-port repeat sweep mode 0.
3. Select a port from AN, AN0, AN2, or AN15 by using bits APS1 and APS0 in the AD0CON2 register.
4. These pins are commonly used in A/D conversion when the MD2 bit is set to 1.
5. Set bits SCAN1 and SCAN0 to 11b in multi-port single sweep mode or multi-port repeat sweep mode 0.
6. When the MSS bit in the AD0CON3 register is 1 (multi-port sweep mode enabled), set the MD2 bit to 0.
7. Refer to the note on the CKS0 bit in the AD0CON0 register.
8. This bit controls the reference voltage to the A/D converter. It does not affect VREF performance of the D/A converter.
9. Do not set the VCUT bit to 0 during A/D conversion.
10. When the VCUT bit is switched from 0 to 1, wait at least 1 μs before starting A/D conversion.
11. Bits OPA1 and OPA0 can be set to 01b or 10b only in one-shot mode or repeat mode. Set them to 00b or 11b in other modes.
12. Set bits OPA1 and OPA0 to 00b when the MSS bit in the AD0CON3 register is 1 (multi-port sweep mode enabled).

Figure 19.3 AD0CON1 Register

A/D0 Control Register 2 [1]



Symbol	Address	Reset Value
AD0CON2	0394h	XX0X X000b

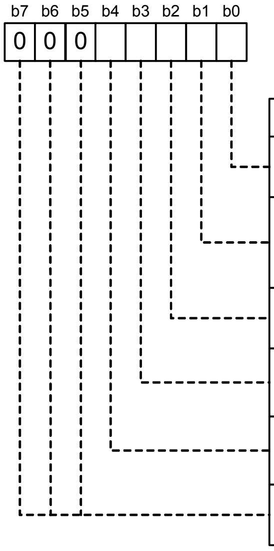
Bit Symbol	Bit Name	Function	RW
SMP	A/D Conversion Method Select Bit	0: Without sample and hold function 1: With sample and hold function	RW
APS0	Analog Input Port Select Bit [2, 3]	b2 b1 0 0 : AN_0 to AN_7, ANEX0, ANEX1 0 1 : AN15_0 to AN15_7 1 0 : AN0_0 to AN0_7 1 1 : AN2_0 to AN2_7	RW
APS1			RW
— (b4-b3)	No register bits; should be written with 0 and read as undefined value		—
TRG0	External Trigger Request Source Select Bit	0: Select $\overline{\text{ADTRG}}$ pin 1: Select a timer B2 interrupt request (after counting the ICTB2 register) in the three-phase motor control timers	RW
— (b7-b6)	Reserved	Should be written with 0 and read as undefined value	RW

Notes:

1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. Set bits APS1 and APS0 to 01b when the MSS bit in the AD0CON3 register is 1 (multi-port sweep mode enabled).
3. These bits can be set to 10b or 11b in single-chip mode only.

Figure 19.4 AD0CON2 Register

A/D0 Control Register 3 (1, 2)



Symbol AD0CON3

Address 0395h

Reset Value XXXX X000b

Bit Symbol	Bit Name	Function	RW
DUS	DMAC Operating Mode Select Bit (3)	0: DMAC operating mode disabled 1: DMAC operating mode enabled (4, 5)	RW
MSS	Multi-port Sweep Mode Select Bit	0: Multi-port sweep mode disabled 1: Multi-port sweep mode enabled (3, 6)	RW
CKS2	Frequency Select Bit	(See Note 7)	RW
MSF0	Multi-port Sweep Status Flag (8)	b4 b3 0 0 : AN_0 to AN_7 0 1 : AN15_0 to AN15_7 1 0 : AN0_0 to AN0_7 1 1 : AN2_0 to AN2_7	RO
MSF1			RO
— (b7-b5)	Reserved	Set to 0. The read value is undefined	RW

Notes:

1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. This register may be read incorrectly during A/D conversion. It should be read or written after the A/D converter stops operating.
3. To set the MSS bit to 1, the DUS bit should be also set to 1.
4. When the DUS bit is set to 1, all A/D converted results are stored into the AD00 register.
5. Configure DMAC when it is used to transfer converted results.
6. To set the MSS bit to 1:
   - Set the MD2 bit in the AD0CON1 register to 0 (mode other than repeat sweep mode 1).
   - Set bits APS1 and APS0 in the AD0CON2 register to 01b (AN15_0 to AN15_7).
   - Set bits OPA1 and OPA0 in the AD0CON1 register to 00b (no use of ANEX0 or ANEX1).
7. Refer to the note on the CKS0 bit in the AD0CON0 register.
8. This bit setting is enabled when the MSS bit is set to 1. The read value is undefined when the MSS bit is set to 0.

Figure 19.5 AD0CON3 Register

A/D0 Control Register 4 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0			0	0

Symbol
AD0CON4

Address
0392h

Reset Value
XXXX 00XXb

Bit Symbol	Bit Name	Function	RW
— (b1-b0)	Reserved	Should be written with 0 and read as undefined value	RW
MPS10	Multi-port Sweep Port Select Bit (2)	b3 b2 0 0 : (Note 3) 0 1 : AN_0 to AN_7, AN15_0 to AN15_7 1 0 : AN_0 to AN_7, AN0_0 to AN0_7 1 1 : AN_0 to AN_7, AN2_0 to AN2_7	RW
MPS11			RW
— (b7-b4)	Reserved	Should be written with 0 and read as undefined value	RW

Notes:
1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. Bits MPS11 and MPS10 can be set to 10b or 11b in single-chip mode only.
3. When the MSS bit in the AD0CON3 register is 0 (multi-port sweep mode disabled), set bits MSP11 and MPS10 to 00b. When it is 1 (multi-port sweep mode enabled), set them to any value other than 00b.

Figure 19.6 AD0CON4 Register

A/D0 Control Register 5 (1, 2, 3)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0				

Symbol
AD0CON5

Address
0393h

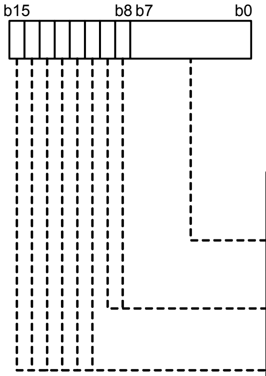
Reset Value
0000 0000b

Bit Symbol	Bit Name	Function	RW
ASTEN	Self Test and Open-circuit Detection Assist Setting Bit	0: Assist disabled 1: Assist enabled (4)	RW
ASTSEL	Self Test/Open-circuit Detection Assist Select Bit	0: Open-circuit detection assist 1: Self test assist	RW
ASTMTD	Assist Mode Select Bit	0: Pull-up 1: Pull-down	RW
— (b3)	Reserved	If necessary, set to 0	RO
— (b5-b4)	Reserved	Should be written with 0	RW
— (b7-b6)	Reserved	If necessary, set to 0	RO

Notes:
1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. An 8-bit read access to this register should be performed.
3. Appropriate evaluation for each system should be performed since A/D converted result for open-circuit detection varies with external circuits.
4. To use the self test assist or open-circuit detection assist, set the SMP bit in the AD0CON2 register to 1 (with sample and hold function).

Figure 19.7 AD0CON5 Register

A/D0 Register i (i = 0 to 7) (1 to 4)



Symbol	Address	Reset Value
AD00, AD01	0381h-0380h, 0383h-0382h	0000 0000 XXXX XXXXb
AD02, AD03	0385h-0384h, 0387h-0386h	0000 0000 XXXX XXXXb
AD04, AD05	0389h-0388h, 038Bh-038Ah	0000 0000 XXXX XXXXb
AD06, AD07	038Dh-038Ch, 038Fh-038Eh	0000 0000 XXXX XXXXb

Bit Symbol	Function	RW
— (b7-b0)	The lower byte in an A/D converted result	RO
— (b9-b8)	In 10-bit mode: 2 upper bits in an A/D converted result In 8-bit mode: These bits are read as 0	RO
— (b15-b10)	These bits are read as 0	RO

Notes:

1. If this register is read by a program while the DMAC is configured to transfer converted results, the value is undefined.
2. The register value written while the A/D converter stops operating is undefined.
3. Only the AD00 register is available when the DUS bit in the AD0CON3 register is 1 (DMAC operating mode enabled). Other registers are undefined.
4. When a converted result is transferred by DMAC at 10-bit mode, the DMAC should be set for a 16-bit transfer.

Figure 19.8 Registers AD00 to AD07

19.1 Mode Descriptions

19.1.1 One-shot Mode

In one-shot mode, the analog voltage applied to a selected pin is converted into a digital code only once. Table 19.2 lists specifications of one-shot mode.

Table 19.2 One-shot Mode Specifications

Item	Specification
Function	Converts the analog voltage applied to a pin into a digital code only once. The pin is selected by setting bits CH2 to CH0 in the AD0CON0 register, bits OPA1 and OPA0 in the AD0CON1 register, and bits APS1 and APS0 in the AD0CON2 register
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set TRG0 in the AD0CON2 register to select external trigger request source. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected) • The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When A/D conversion is completed, an interrupt request is generated
Input pin to be selected	One pin is selected from among AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) Configure the DMAC (refer to 13. “DMAC”). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program

19.1.2 Repeat Mode

In repeat mode, the analog voltage applied to a selected pin is repeatedly converted into a digital code. Table 19.3 lists specifications of repeat mode.

Table 19.3 Repeat Mode Specifications

Item	Specification
Function	Converts the analog voltage input to a pin into a digital code repeatedly. The pin is selected by setting bits CH2 to CH0 in the AD0CON0 register, bits OPA1 and OPA0 in the AD0CON1 register, and bits APS1 and APS0 in the AD0CON2 register
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set TRG0 in the AD0CON2 register to select external trigger request source. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled), no interrupt request is generated. When the DUS bit is 1 (DMAC operating mode enabled), each time A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	One pin is selected from among AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) • When the converted result is transferred by DMAC Configure the DMAC (refer to 13. “DMAC”). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program • When the converted result is transferred by a program Read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0

19.1.3 Single Sweep Mode

In single sweep mode, the analog voltage applied to selected pins is converted one-by-one into a digital code. Table 19.4 lists specifications of single sweep mode.

Table 19.4 Single Sweep Mode Specifications

Item	Specification
Function	Converts the analog voltage input to a set of pins into a digital code one-by-one. The pins are selected by setting bits SCAN1 and SCAN0 in the AD0CON1 register and bits APS1 and APS0 in the AD0CON2 register
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set TRG0 in the AD0CON2 register to select external trigger request source. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected) • The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) when a sweep is completed, an interrupt request is generated. When the DUS bit is 1 (DMAC operating mode enabled), each time A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	Selected from a group of 2 pins (ANi_0 and ANi_1), 4 pins (ANi_0 to ANi_3), 6 pins (ANi_0 to ANi_5), or 8 pins (ANi_0 to ANi_7) (i = no value, 0, 2, 15)
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) Configure the DMAC (refer to 13. “DMAC”). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program

19.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, the analog voltage applied to selected pins is repeatedly converted into a digital code. Table 19.5 lists specifications of repeat sweep mode 0.

Table 19.5 Repeat Sweep Mode 0 Specifications

Item	Specification
Function	Converts the analog voltage input to a set of pins into a digital code repeatedly. The pins are selected by setting bits SCAN1 and SCAN0 in the AD0CON1 register and APS1 and APS0 in the AD0CON2 register
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set TRG0 in the AD0CON2 register to select external trigger request source. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled), no interrupt request is generated. When the DUS bit is 1 (DMAC operating mode enabled), each time A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	Selected from a group of 2 pins (ANi_0 and ANi_1), 4 pins (ANi_0 to ANi_3), 6 pins (ANi_0 to ANi_5), or 8 pins (ANi_0 to ANi_7) (i = no value, 0, 2, 15)
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) • When the converted result is transferred by DMAC Configure the DMAC (refer to 13. “DMAC”). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program • When the converted result is transferred by a program Read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0

19.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, the analog voltage applied to eight selected pins including one to four prioritized pins is repeatedly converted into a digital code. Table 19.6 lists specifications of repeat sweep mode 1.

Table 19.6 Repeat Sweep Mode 1 Specifications

Item	Specification
Function	The analog voltage applied to eight selected pins including one to four prioritized pins is repeatedly converted into a digital code. The prioritized pins are selected by setting bits SCAN1 and SCAN0 in the AD0CON1 register and bits APS1 and APS0 in the AD0CON2 register For example, when AN_0 is selected, the A/D conversion is performed in the following order: AN_0→AN_1→AN_0→AN_2→AN_0→AN_3•••
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set TRG0 in the AD0CON2 register to select external trigger request source. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. Retrigger is invalid. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled), no interrupt request is generated. When the DUS bit is 1 (DMAC operating mode enabled), each time A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	8 (ANi_0 to ANi_7) (i = no value, 0, 2, 15)
Prioritized pin(s)	Selected from a group of 1 pin (ANi_0), 2 pins (ANi_0 and ANi_1), 3 pins (ANi_0 to ANi_2), or 4 pins (ANi_0 to ANi_3)
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) • When the converted result is transferred by DMAC Configure the DMAC (refer to 13. “DMAC”). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program • When the converted result is transferred by a program Read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0

19.1.6 Multi-port Single Sweep Mode

In multi-port single sweep mode, the analog voltage applied to 16 selected pins is converted one-by-one into a digital code. The DUS bit in the AD0CON3 register should be set to 1 (DMAC operating mode enabled). Table 19.7 lists specifications of multi-port single sweep mode.

Table 19.7 Multi-port Single Sweep Mode Specifications

Item	Specification
Function	Converts the analog voltage input to a set of 16 selected pins into a digital code one-by-one in the following order: AN_0 to AN_7→ANi_0 to ANi_7 (i = 0, 2, 15) The 16 pins are selected by setting bits MPS11 and MPS10 in the AD0CON4 register For example, when bits MPS11 and MPS10 are set to 10b (AN_0 to AN_7, AN0_0 to AN0_7), the analog voltage is converted into a digital code in the following order: AN_0→AN_1→AN_2→AN_3→AN_4→AN_5→AN_6→AN_7→AN0_0→•••→AN0_6→AN0_7
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set TRG0 in the AD0CON2 register to select external trigger request source. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected) • The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	Every time A/D conversion is completed (set the DUS bit to 1)
Analog voltage input pins	A combination of pin group is selected from AN_0 to AN_7→AN15_0 to AN15_7, AN_0 to AN_7→AN0_0 to AN0_7, or AN_0 to AN_7→AN2_0 to AN2_7
Reading A/D converted result	Set the DUS bit to 1 and configure the DMAC (refer to 13. “DMAC”). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program

19.1.7 Multi-port Repeat Sweep Mode 0

In multi-port repeat sweep mode 0, the analog voltage applied to 16 selected pins is repeatedly converted into a digital code. The DUS bit in the AD0CON3 register should be set to 1 (DMAC operating mode enabled). Table 19.8 lists specifications of multi-port repeat sweep mode 0.

Table 19.8 Multi-port Repeat Sweep Mode 0 Specifications

Item	Specification
Function	Converts the analog voltage input to a set of 16 selected pins into a digital code repeatedly in the following order: AN_0 to AN_7→ANi_0 to ANi_7 (i = 0, 2, 15) The 16 pins are selected by setting bits MPS11 and MPS10 in the AD0CON4 register For example, when bits MPS11 and MPS10 are set to 10b (AN_0 to AN_7, AN0_0 to AN0_7),the analog voltage is converted into a digital code repeatedly in the following order: AN_0→AN_1→AN_2→AN_3→AN_4→AN_5→AN_6→AN_7→AN0_0→•••→ AN0_6→AN0_7
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set TRG0 in the AD0CON2 register to select external trigger request source. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	Every time A/D conversion is completed (set the DUS bit to 1)
Analog voltage input pins	A combination of pin group is selected from AN_0 to AN_7→AN15_0 to AN15_7, AN_0 to AN_7→AN0_0 to AN0_7, or AN_0 to AN_7→AN2_0 to AN2_7
Reading A/D converted result	Set the DUS bit to 1 and configure the DMAC (refer to 13. “DMAC”). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program

19.2 Functions

19.2.1 Resolution Selection

Resolution is selected by setting the BITS bit in the AD0CON1 register. When the BITS bit is set to 1 (10-bit precision), the A/D converted result is stored into bits 9 to 0 in the AD0i register (i = 0 to 7). When the BITS bit is set to 0 (8-bit precision), the result is stored into bits 7 to 0 in the AD0i register.

19.2.2 Sample and Hold Function

This function improves the conversion rate per pin to 28 ϕAD cycles at 8-bit resolution and 33 ϕAD cycles for 10-bit resolution. This function is available in all operating modes and is enabled by setting the SMP bit in the AD0CON2 register to 1 (with sample and hold function). Start A/D conversion after setting the SMP bit.

19.2.3 Trigger Selection

A trigger to start A/D conversion is specified by the combination of TRG bit in the AD0CON0 register and TRG0 in the AD0CON2 register. Table 19.9 lists the settings of the trigger selection.

Table 19.9 Trigger Selection Settings

Bit and Setting		Trigger
AD0CON0 register	AD0CON2 register	
TRG = 0	—	Software trigger The ADST bit in the AD0CON0 register is set to 1
TRG = 1 (1, 2)	TRG0 = 0	External trigger Falling edge of a signal applied to the $\overline{\text{ADTRG}}$ pin
	TRG0 = 1	Hardware trigger Generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers

Notes:

1. A/D conversion starts when a trigger is generated while the ADST bit is 1 (A/D conversion started).
2. When an external trigger or a hardware trigger is generated during A/D conversion, the A/D converter aborts the operation in progress. Then, it restarts the operation.

19.2.4 DMAC Operating Mode

DMAC operating mode can be used in all operating modes. DMAC operating mode is highly recommended when the A/D converter is in multi-port single sweep mode or multi-port repeat sweep mode 0. When the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled), all A/D converted results are stored in the AD00 register. The DMAC transfers the data from the AD00 register to a given memory space every time A/D conversion is completed at a pin. 8-bit DMA transfer should be selected for 8-bit resolution. For 10-bit resolution, 16-bit DMA transfer should be selected. Refer to 13. “DMAC” for details.

19.2.5 Function-extended Analog Input Pins

In one-shot mode and repeat mode, pins ANEX0 and ANEX1 can be used as analog input pins by setting bits OPA1 and OPA0 in the AD0CON1 register (refer to Table 19.10). The A/D converted results of pins ANEX0 and ANEX1 are stored into registers AD00 and AD01, respectively. However, when the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled), all results are stored into the AD00 register.

To use function-extended analog input pins, bits APS1 and APS0 in the AD0CON2 register should be set to 00b (AN0 to AN7, ANEX0, ANEX1 function as analog input ports) and the MSS bit in the AD0CON3 register to 0 (multi-port sweep mode disabled).

Table 19.10 Function-extended Analog Input Pin Settings

AD0CON1 Register		ANEX0	ANEX1
OPA1	OPA0		
0	0	Not used	Not used
0	1	Analog input	Not used
1	0	Not used	Analog input
1	1	Output to an external op-amp	Input from an external op-amp

19.2.6 External Operating Amplifier (Op-Amp) Connection Mode

In external op-amp connection mode, multiple analog inputs can be amplified by one external op-amp using function-extended analog input pins ANEX0 and ANEX1.

When bits OPA1 and OPA0 in the AD0CON1 register are 11b (external op-amp connected), the voltage applied to pins AN0 to AN7 is output from the ANEX0 pin. This output signal should be amplified by an external op-amp and applied to the ANEX1 pin.

The analog voltage applied to the ANEX1 pin is converted into a digital code. The converted result is stored in the corresponding AD0i register (i = 0 to 7). The conversion rate varies with the response of the external op-amp. Note that the ANEX0 pin should not be connected to the ANEX1 pin directly.

To use external op-amp connection mode, set bits APS1 and APS0 in the AD0CON2 register to 00b.

Figure 19.9 shows an example of an external op-amp connection.

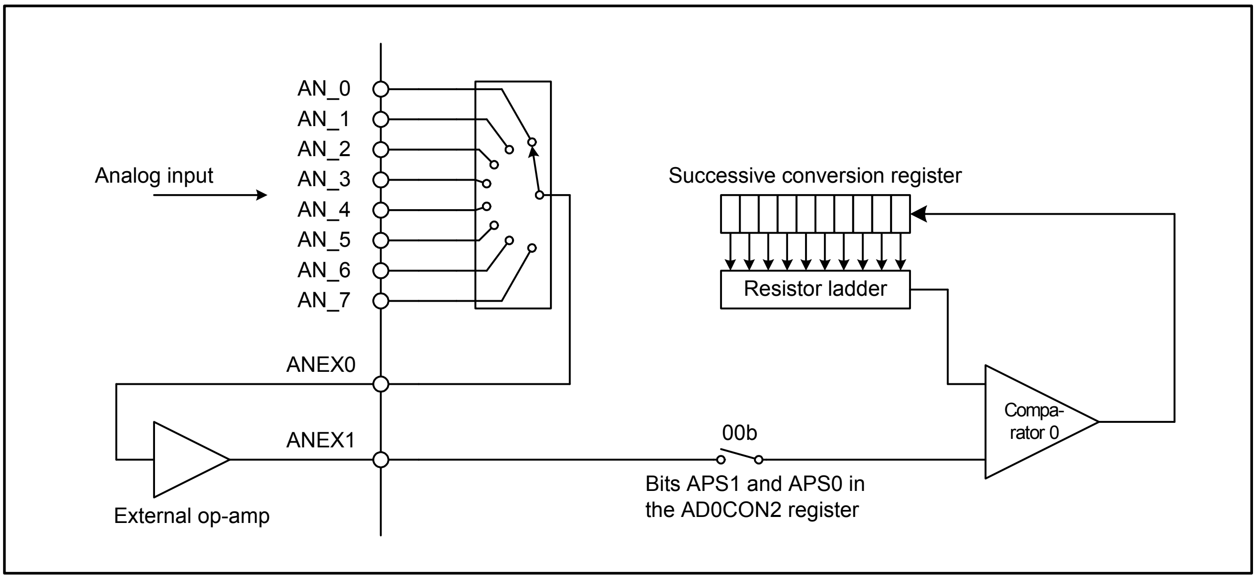


Figure 19.9 External Op-Amp Connection

19.2.7 Self Test/Open-circuit Detection Assist

This function enables the MCU to detect open-circuit in analog input pins. It also enables it to perform a self test.

Figure 19.10 shows a block diagram of open-circuit detection assist circuit.

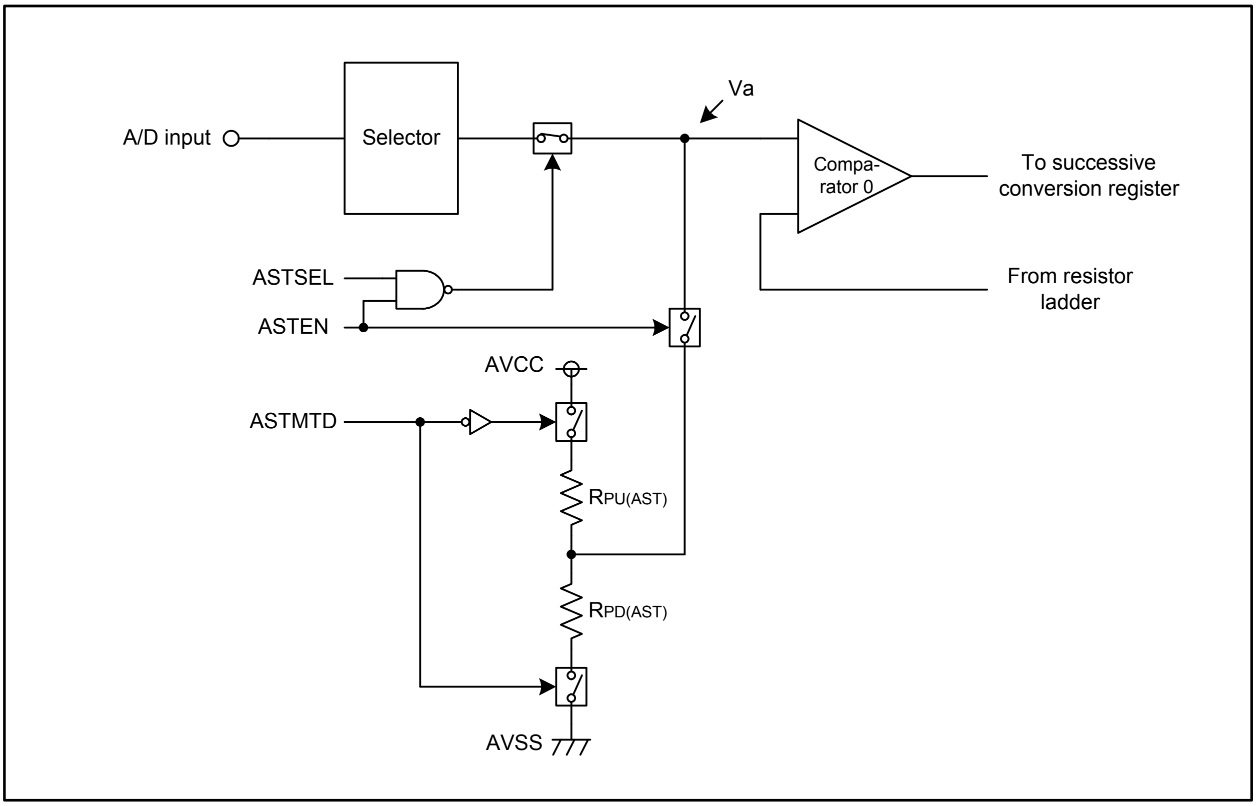


Figure 19.10 Open-circuit Detection Assist Circuit

To detect open-circuit, the ASTSEL bit in the AD0CON5 register should be set to 0 (open-circuit detection assist). Va in Figure 19.10 above is a voltage with a value between AVCC and the A/D applied voltage in the following bit settings: the ASTEN bit is 1 (assist enabled), the ASTSEL bit is 0, and the ASTMTD bit is 0 (pull-up). If the A/D input pin is open, Va is almost equal to AVCC. When the ASTEN bit is 1, the ASTSEL bit is 0, and the ASTMTD bit is 1 (pull-down), Va is between AVSS and the A/D applied voltage. If the A/D input pin is open, Va is almost AVSS. That is, the A/D input pin is considered open if the result value of A/D conversion is almost the maximum/minimum voltage.

To enable the self test function, the ASTSEL bit in the AD0CON5 register should be set to 1 (self test assist). Va is almost equal to AVCC in the bit settings as the ASTEN bit is 1, the ASTSEL bit is 1, and the ASTMTD bit is 0. When the ASTEN bit is 1, the ASTSEL bit is 1, and the ASTMTD bit is 1, Va is almost AVSS. That is, if the result value of A/D conversion is almost the maximum/minimum voltage in each bit setting, the A/D converter is considered to be functioning normally.

19.2.8 Power Saving

When the A/D converter is not in use, power consumption can be reduced by setting the VCUT bit in the AD0CON1 to 0 (VREF disconnected). With this bit setting, the reference voltage input pin (VREF) can be disconnected from the resistor ladder, which enables the power supply from the VREF to the resistor ladder to stop.

To use the A/D converter, set the VCUT bit to 1 (VREF connected) and wait at least 1 μs before setting the ADST bit in the AD0CON0 register to 1 (A/D conversion started). Bits ADST and VCUT should not be set to 1 simultaneously. The VCUT bit should not be set to 0 during A/D conversion.

The VCUT bit does not affect VREF performance of the D/A converter (refer to Figure 19.11).

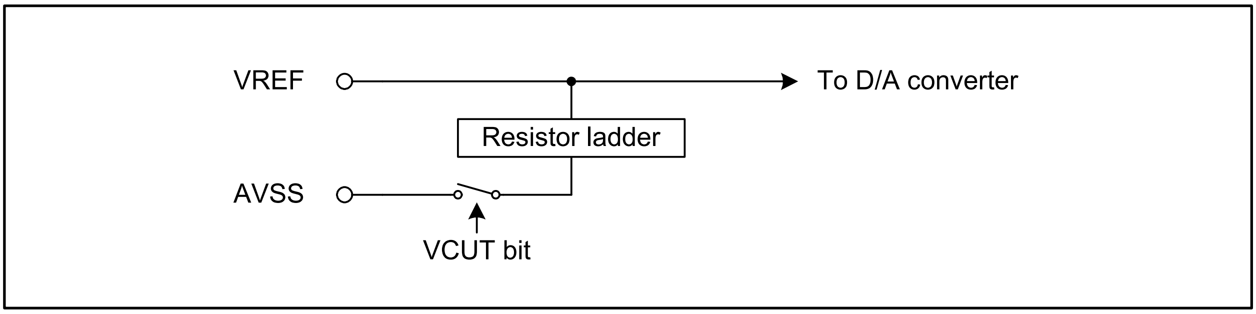


Figure 19.11 Power Supply by VCUT Bit

19.2.9 Output Impedance of Sensor Equivalent Circuit under A/D Conversion

Figure 19.12 shows an analog input pin and external sensor equivalent circuit.

To perform A/D conversion correctly, the internal capacitor (C) charging, shown in Figure 19.12, should be completed within the specified period. This period, called the sampling time, is 2 ϕAD cycles for conversion without the sample and hold function and 3 ϕAD cycles for conversion with this function.

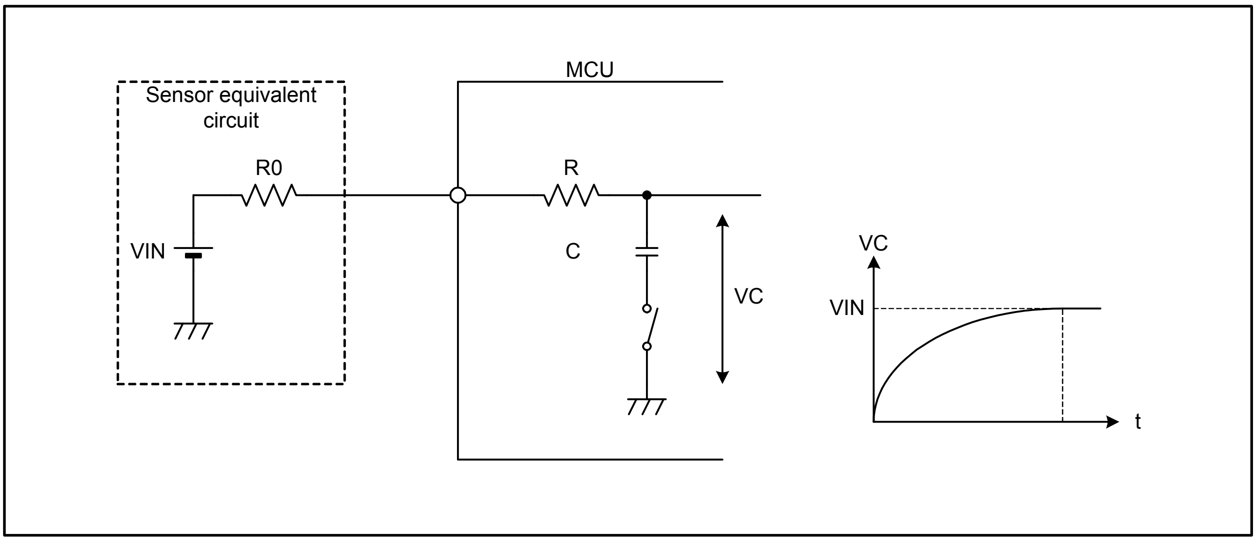


Figure 19.12 Analog Input Pin and External Sensor Equivalent Circuitry

The voltage between pins (VC) is expressed as follows:

$$VC = VIN\left\{1 - e^{-\frac{t}{C(R0+R)}}\right\}$$

When t = T and the precision (error) is x or less,

$$VC = VIN - \frac{x}{y}VIN = VIN\left(1 - \frac{x}{y}\right)$$

Thus, output impedance of the sensor equivalent circuit (R0) is determined by the following formulas:

$$e^{-\frac{T}{C(R0+R)}} = \frac{x}{y}$$

$$-\frac{T}{C(R0+R)} = \ln\frac{x}{y}$$

$$R0 = -\frac{T}{C\ln\frac{x}{y}} - R$$

where:
T[s] = Sampling time
R0[Ω] = Output impedance of the sensor equivalent circuit
VC = Potential difference between edges of capacitor C
R[Ω] = Internal resistance of the MCU
x[LSB] = Precision (error) of the A/D converter
y[step] = Resolution of the A/D converter (1024 steps at 10-bit mode, 256 steps at 8-bit mode)

When ϕAD = 10 MHz, the A/D conversion mode is 10-bit resolution with the sample and hold function, the output impedance (R0) with the precision (error) of 0.1 LSB or less is determined by the following formula:
Using T = 0.3 μs, R = 2.0 kΩ (reference value), C = 6.5 pF (reference value), x = 0.1, y = 1024,

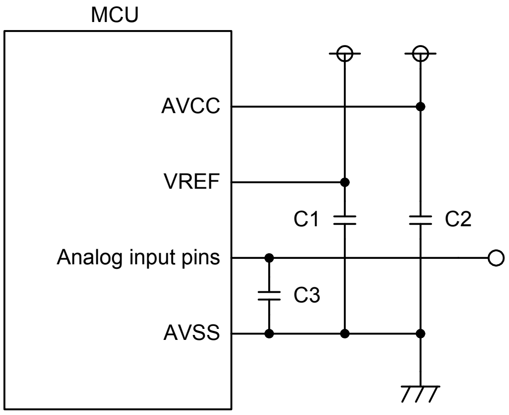
$$R0 = -\frac{0.3 \times 10^{-6}}{6.5 \times 10^{-12} \times 1\text{n}\frac{0.1}{1024}} - 2.0 \times 10^{3}$$
$$= 2998$$

Thus, the allowable output impedance of the sensor equivalent circuit (R0), making the precision (error) of 0.1 LSB or less, should be less than 3 kΩ.
The actual error, however, is the value of absolute precision added to the 0.1 LSB mentioned above.

19.3 Notes on A/D Converter

19.3.1 Notes on Designing Boards

- Three capacitors should be placed between the AVSS pin and pins such as AVCC, VREF, and analog inputs (AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, and AN15_0 to AN15_7) to avoid erroneous operations caused by noise or latchup, and to reduce conversion errors. Figure 19.13 shows an example of pin configuration for A/D converter.



Notes:

1. C1 ≥ 0.47 μF, C2 ≥ 0.47 μF, and C3 ≥ 100 pF (reference values)
2. The traces for the capacitor and the MCU should be as short and wide as physically possible.

Figure 19.13 Pin Configuration for the A/D Converter

- Do not use AN_4 to AN_7 for analog input if the key input interrupt is to be used. Otherwise, a key input interrupt request occurs when the A/D input voltage becomes VIL or lower.

- When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1 should be VCC or lower.

19.3.2 Notes on Programming

- The following registers should be written while A/D conversion is stopped. That is, before a trigger occurs: AD0CON0 (except the ADST bit), AD0CON1, AD0CON2, AD0CON3, AD0CON4, and AD0CON5.

- When the VCUT bit in the AD0CON1 register is changed from 0 (VREF connected) to 1 (VREF disconnected), wait for at least 1 μs before starting A/D conversion. When not performing A/D conversion, set the VCUT bit to 0 to reduce power consumption.

- Set the port direction bit for the pin to be used as an analog input pin to 0 (input). Set the ASEL bit of the corresponding port function select register to 1 (port is used as A/D input).

- When the TRG bit in the AD0CON0 register is 1 (external trigger or hardware trigger), set the corresponding port direction bit (PD9_7 bit) for the $\overline{\text{ADTRG}}$ pin to 0 (input).

- The ϕAD frequency should be 16 MHz or lower when VCC is 4.2 to 5.5 V, and 10 MHz or lower when VCC is 3.0 to 4.2 V. It should be 1 MHz or higher when the sample and hold function is enabled. If not, it should be 250 kHz or higher.

- When A/D operating mode (bits MD1 and MD0 in the AD0CON0 register or the MD2 bit in the AD0CON1 register) has been changed, reselect analog input pins by setting bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register.

- If the AD0i register is read when the A/D converted result is stored to the register, the stored value may have an error (i = 0 to 7). Read the AD0i register after A/D conversion is completed.
  In one-shot mode or single sweep mode, read the AD0i register after the IR bit in the AD0IC register becomes 1 (interrupt requested).
  In repeat mode, repeat sweep mode 0, or repeat sweep mode 1, an interrupt request can be generated each time A/D conversion is completed when the DUS bit in the AD0CON3 register is 1 (DMAC operating mode enabled). Similar to the other modes above, read the AD00 register after the IR bit in the AD0IC register becomes 1 (interrupt requested).

- When an A/D conversion is halted by setting the ADST bit in the AD0CON0 register to 0, the converted result is undefined. In addition, the unconverted AD0i register may also become undefined. Consequently, the AD0i register should not be used just after A/D conversion is halted.

- External triggers cannot be used in DMAC operating mode. When the DMAC is configured to transfer converted results, do not read the AD00 register by a program.

- While in single sweep mode, if A/D conversion is halted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion is stopped), an interrupt request may be generated even though the sweep is not completed. To halt A/D conversion, disable interrupts before setting the ADST bit to 0.

20. D/A Converter

The MCU has two separate 8-bit R-2R resistor ladder D/A converters.

Digital code is converted to an analog voltage when a value is written to the corresponding DAi register (i = 0, 1). The DAiE bit in the DACON register determines whether the D/A conversion result is output or not. Set the DAiE bit to 1 (output enabled) to output the converted value. This bit setting disables a pull-up resistor for the corresponding port.

Analog voltage to be output (V) is calculated based on the value (n) set in the DAi register (n is a decimal number).

$$V = \frac{VREF \times n}{256} \qquad \text{(n = 0 to 255)}$$

$VREF$: reference voltage

Table 20.1 lists specifications of the D/A converter. Figure 20.1 shows a block diagram of the D/A converter. Figures 20.2 and 20.3 show registers associated with the D/A converter. Figure 20.4 shows a D/A converter equivalent circuit.

When the D/A converter is not used, set the DAi register to 00h and the DAiE bit to 0 (output disabled).

Table 20.1 D/A Converter Specifications

Item	Specification
D/A conversion method	R-2R resistor ladder
Resolution	8 bits
Analog output pins	2 channels

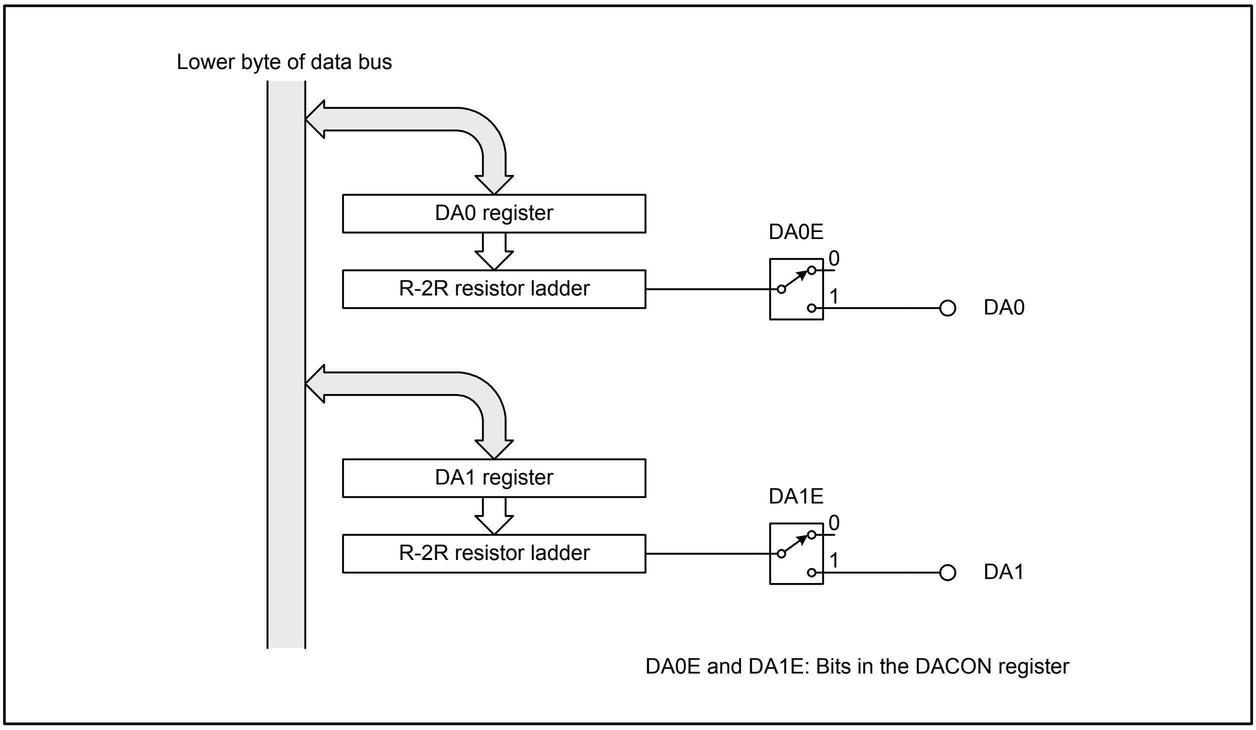
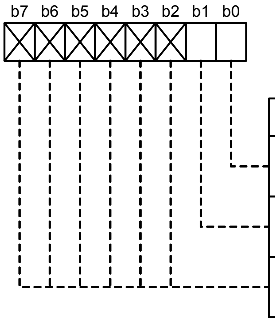


Figure 20.1 D/A Converter Block Diagram

D/A Control Register



Symbol DACON

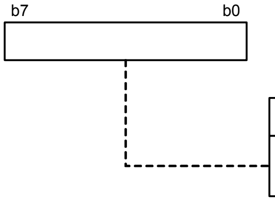
Address 039Ch

Reset Value XXXX XX00b

Bit Symbol	Bit Name	Function	RW
DA0E	D/A0 Output Enable Bit	0: Output disabled 1: Output enabled	RW
DA1E	D/A1 Output Enable Bit	0: Output disabled 1: Output enabled	RW
— (b7-b2)	No register bits; should be written with 0 and read as undefined value		—

Figure 20.2 DACON Register

D/A Register i (i = 0, 1)



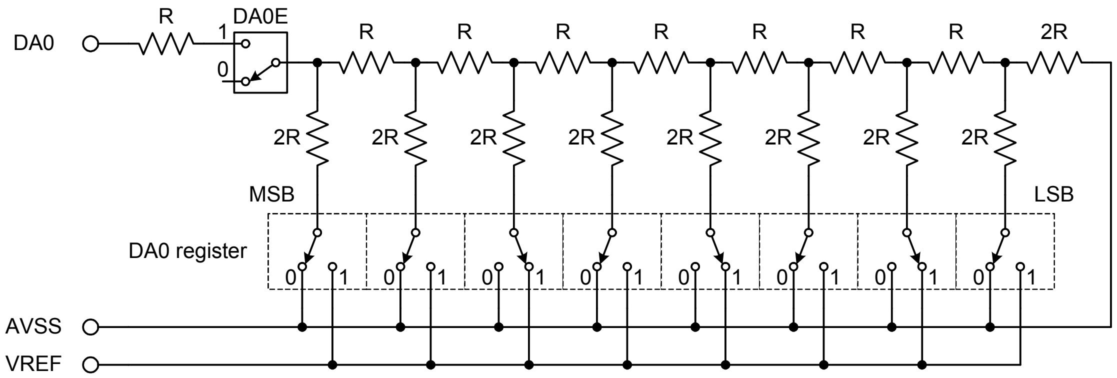
Symbol DA0, DA1

Address 0398h, 039Ah

Reset Value Undefined

Function	Setting Range	RW
Output value by the D/A conversion	00h to FFh	RW

Figure 20.3 Registers DA0 and DA1



Notes:

1. This figure applies when the DA0 register is 2Ah.
2. This circuitry also applies to D/A converter 1.
3. To reduce power consumption when the D/A converter is not in use, set the DAiE bit to 0 (output disabled) and the DAi register to 00h to prevent the current from flowing into the R-2R resistor ladder (i = 0, 1).

Figure 20.4 D/A Converter Equivalent Circuitry

21. CRC Calculator

The Cyclic Redundancy Check (CRC) calculator is used for detecting errors in data blocks. A generator polynomial of CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) generates the CRC.

The CRC is a 16-bit code generated for a given set of blocks of 8-bit data. It is set in the CRCD register every time 1-byte data is written to the CRCIN register after a default value is set to the CRCD register.

Figure 21.1 shows a block diagram of the CRC calculator. Figures 21.2 and 21.3 show registers associated with the CRC. Figure 21.4 shows an example of the CRC calculation.

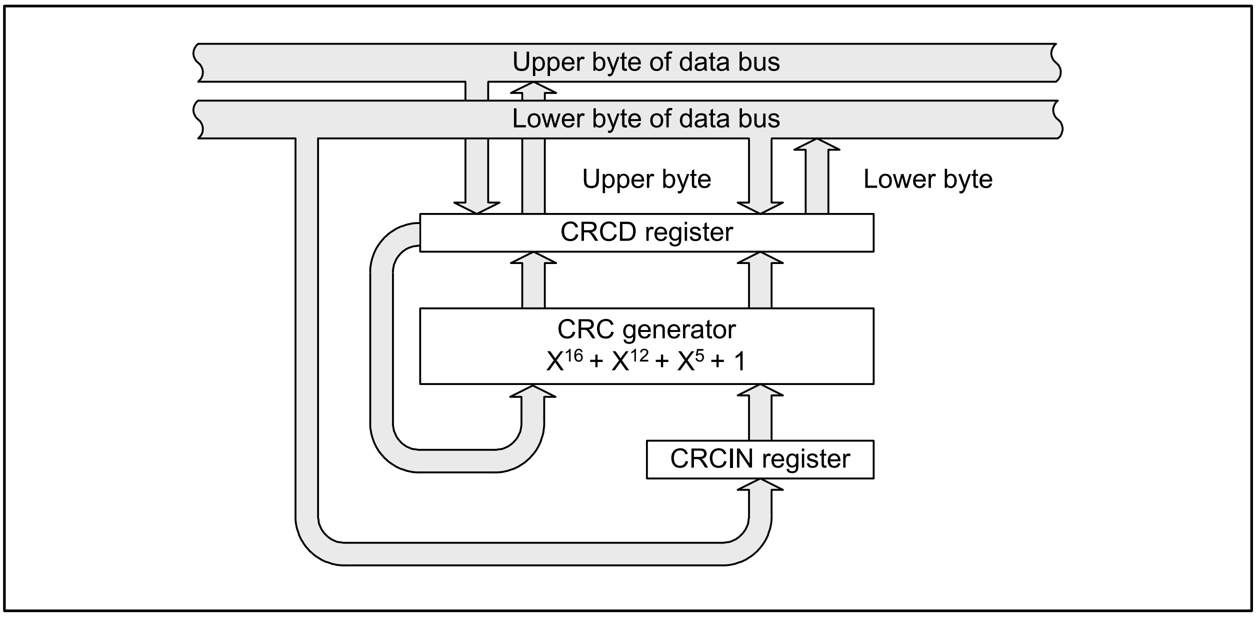
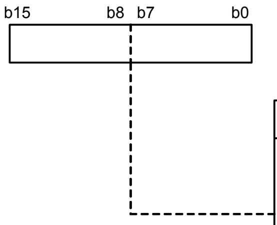


Figure 21.1 CRC Calculator Block Diagram

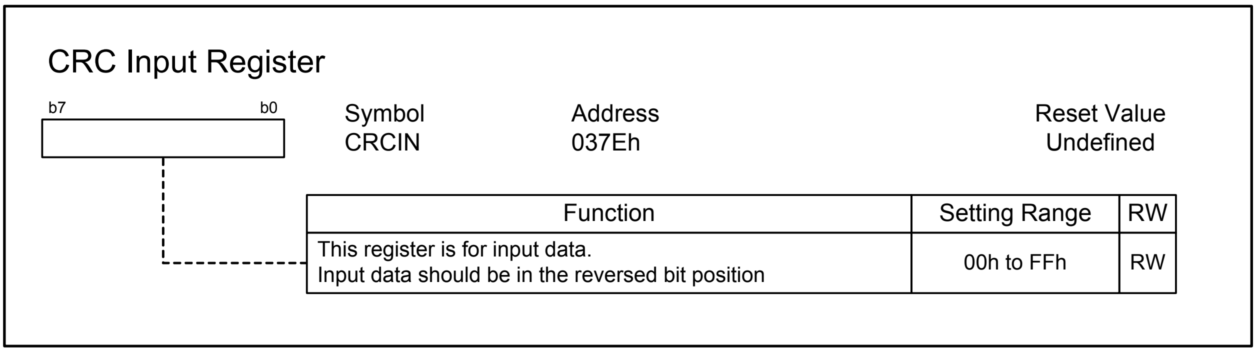
CRC Data Register



Symbol	Address	Reset Value
CRCD	037Dh-037Ch	Undefined

Function	Setting Range	RW
The CRC calculation result is stored in the CRCD register. When a default value in a reversed bit position is set in this register and then data in reversed bit position is written to the CRCIN register, the CRC in the reversed bit position is read from this register	0000h to FFFFh	RW

Figure 21.2 CRCD Register

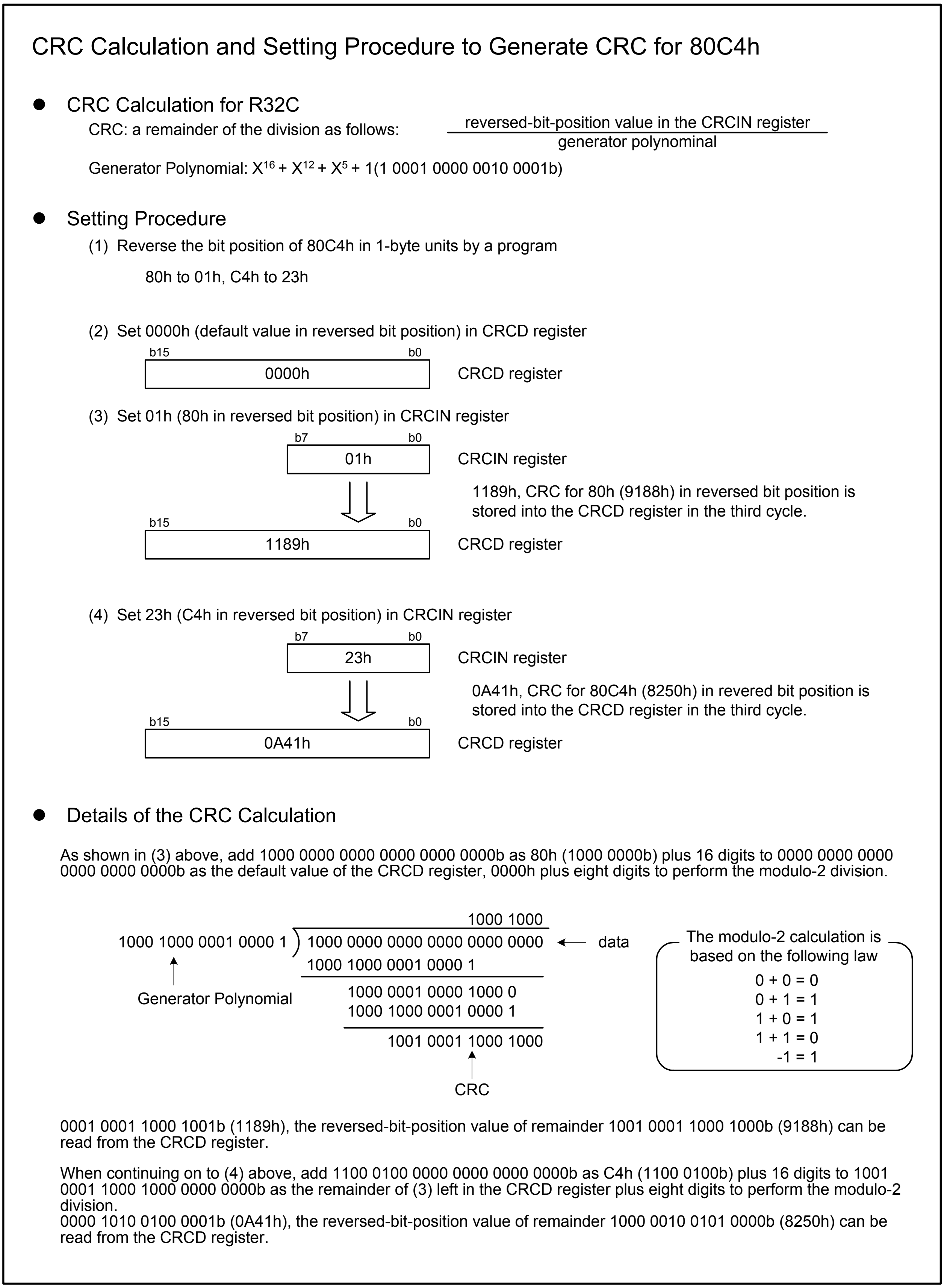


CRC Input Register

Symbol	Address	Reset Value
CRCIN	037Eh	Undefined

Function	Setting Range	RW
This register is for input data. Input data should be in the reversed bit position	00h to FFh	RW

Figure 21.3 CRCIN Register



CRC Calculation and Setting Procedure to Generate CRC for 80C4h

- CRC Calculation for R32C

CRC: a remainder of the division as follows: $\dfrac{\text{reversed-bit-position value in the CRCIN register}}{\text{generator polynominal}}$

Generator Polynomial: $X^{16} + X^{12} + X^5 + 1$(1 0001 0000 0010 0001b)

- Setting Procedure

(1) Reverse the bit position of 80C4h in 1-byte units by a program

80h to 01h, C4h to 23h

(2) Set 0000h (default value in reversed bit position) in CRCD register

(3) Set 01h (80h in reversed bit position) in CRCIN register

1189h, CRC for 80h (9188h) in reversed bit position is stored into the CRCD register in the third cycle.

(4) Set 23h (C4h in reversed bit position) in CRCIN register

0A41h, CRC for 80C4h (8250h) in revered bit position is stored into the CRCD register in the third cycle.

- Details of the CRC Calculation

As shown in (3) above, add 1000 0000 0000 0000 0000 0000b as 80h (1000 0000b) plus 16 digits to 0000 0000 0000 0000 0000 0000b as the default value of the CRCD register, 0000h plus eight digits to perform the modulo-2 division.

```
                                        1000 1000
1000 1000 0001 0000 1 ) 1000 0000 0000 0000 0000 0000  <-- data
                        1000 1000 0001 0000 1
                        ---------------------------
                             1000 0001 0000 1000 0
                             1000 1000 0001 0000 1
                             ----------------------
                                  1001 0001 1000 1000
```

The modulo-2 calculation is based on the following law

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
-1 = 1

0001 0001 1000 1001b (1189h), the reversed-bit-position value of remainder 1001 0001 1000 1000b (9188h) can be read from the CRCD register.

When continuing on to (4) above, add 1100 0100 0000 0000 0000 0000b as C4h (1100 0100b) plus 16 digits to 1001 0001 1000 1000 0000 0000b as the remainder of (3) left in the CRCD register plus eight digits to perform the modulo-2 division.
0000 1010 0100 0001b (0A41h), the reversed-bit-position value of remainder 1000 0010 0101 0000b (8250h) can be read from the CRCD register.

Figure 21.4 CRC Calculation

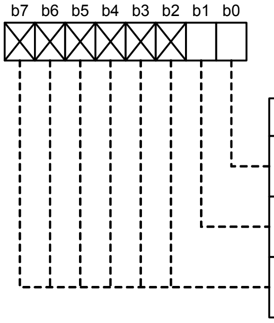
22. X-Y Conversion

X-Y conversion rotates a 16 × 16-bit matrix data 90 degrees or reverses the bit position of 16-bit data.

X-Y conversion is set using the XYC register shown in Figure 22.1.

Data is written to the write-only XiR registers and converted data is read from the read-only YjR register (i = 0 to 15; j = 0 to 15). These registers are allocated to the same address. Figures 22.2 and 22.3 show registers XiR and YjR, respectively. A write/read access to registers XiR and YjR should be performed in 16-bit units from an even address. 8-bit access operation results are undefined.

X-Y Control Register

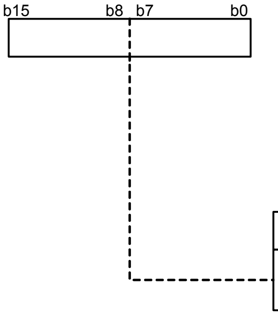


Symbol	Address	Reset Value
XYC	02E0h	XXXX XX00b

Bit Symbol	Bit Name	Function	RW
XYC0	Read Mode Set Bit	0: Data rotation 1: No data rotation	RW
XYC1	Write Mode Set Bit	0: No bit position reverse 1: Bit position reverse	RW
— (b7-b2)	No register bits; should be written with 0 and read as undefined value		—

Figure 22.1 XYC Register

Xi Register (i = 0 to 15) (1)



Symbol	Address	Reset Value
X0R to X2R	02C1h-02C0h, 02C3h-02C2h, 02C5h-02C4h	Undefined
X3R to X5R	02C7h-02C6h, 02C9h-02C8h, 02CBh-02CAh	Undefined
X6R to X8R	02CDh-02CCh, 02CFh-02CEh, 02D1h-02D0h	Undefined
X9R to X11R	02D3h-02D2h, 02D5h-02D4h, 02D7h-02D6h	Undefined
X12R to X14R	02D9h-02D8h, 02DBh-02DAh, 02DDh-02DCh	Undefined
X15R	02DFh-02DEh	Undefined

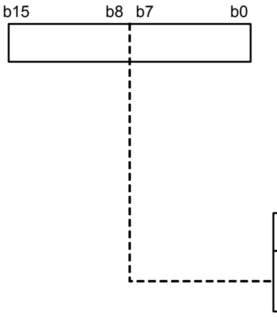
Function	Setting Range	RW
Input data for X-Y conversion	0000h to FFFFh	WO

Note:

1. A 16-bit write access to this register should be performed.

Figure 22.2 Registers X0R to X15R

Yj Register (j = 0 to 15) (1)



Symbol	Address	Reset Value
Y0R to Y2R	02C1h-02C0h, 02C3h-02C2h, 02C5h- 02C4h	Undefined
Y3R to Y5R	02C7h-02C6h, 02C9h-02C8h, 02CBh-02CAh	Undefined
Y6R to Y8R	02CDh-02CCh, 02CFh-02CEh, 02D1h-02D0h	Undefined
Y9R to Y11R	02D3h-02D2h, 02D5h-02D4h, 02D7h-02D6h	Undefined
Y12R to Y14R	02D9h-02D8h, 02DBh-02DAh, 02DDh-02DCh	Undefined
Y15R	02DFh-02DEh	Undefined

Function	RW
Result of X-Y conversion	RO

Note:

1. A 16-bit read access to this register should be performed.

Figure 22.3 Registers Y0R to Y15R

22.1 Data Conversion When Reading

Set the XYC0 bit in the XYC register to select a read mode for the YjR register. When the XYC0 bit is 0 (data rotation), bit j in the corresponding registers X0R to X15R is automatically read upon reading the YjR register (j = 0 to 15).

More concretely, upon reading bit i (i = 0 to 15) in the Y0R register, the data of bit 0 in the XiR register is read. That is, the read data of bit 0 in the Y15R register means the data of bit 15 in the X0R register and the data of bit 15 in the Y0R register is identical to that of bit 0 in the X15R register.

Figure 22.4 shows the conversion table when the XYC0 bit is 0 and Figure 22.5 shows an example of X-Y conversion.

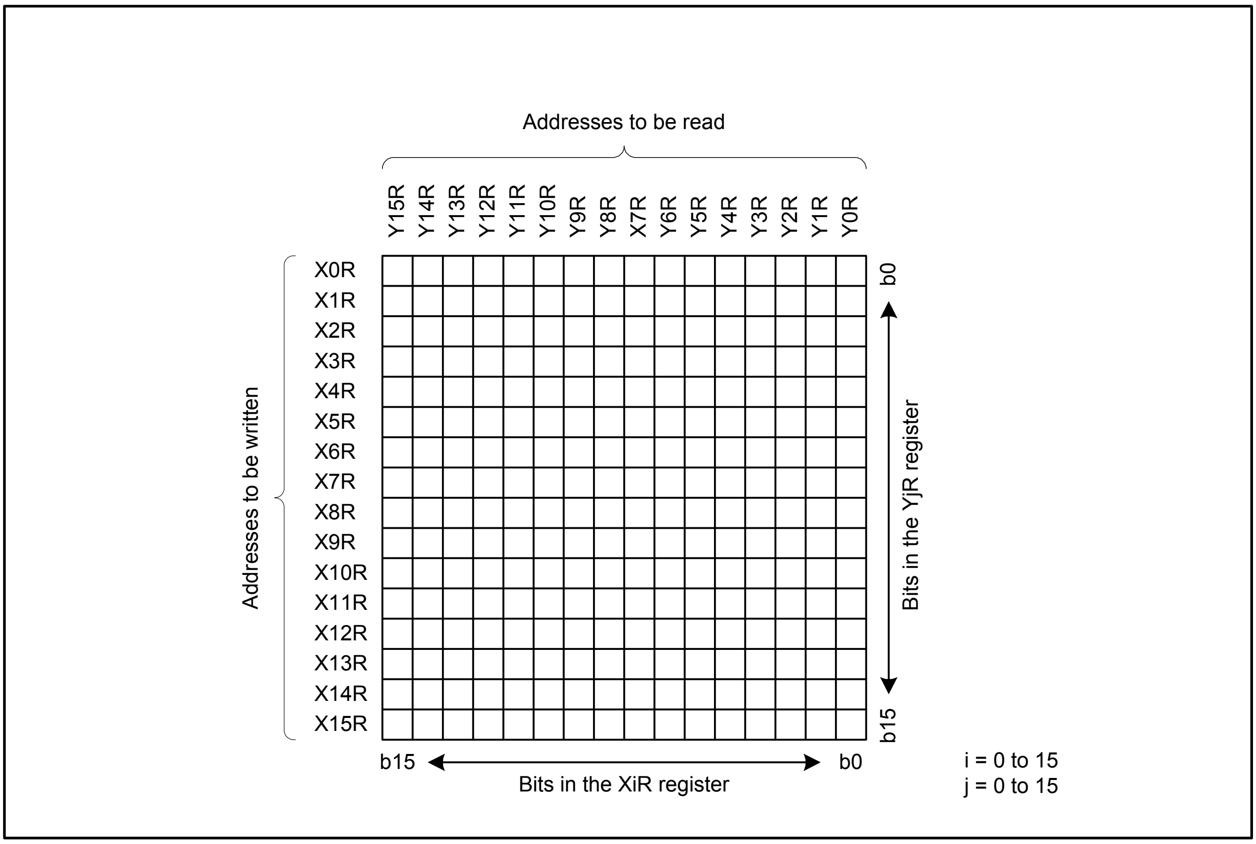


Figure 22.4 Conversion Table (XYC0 Bit is 0)

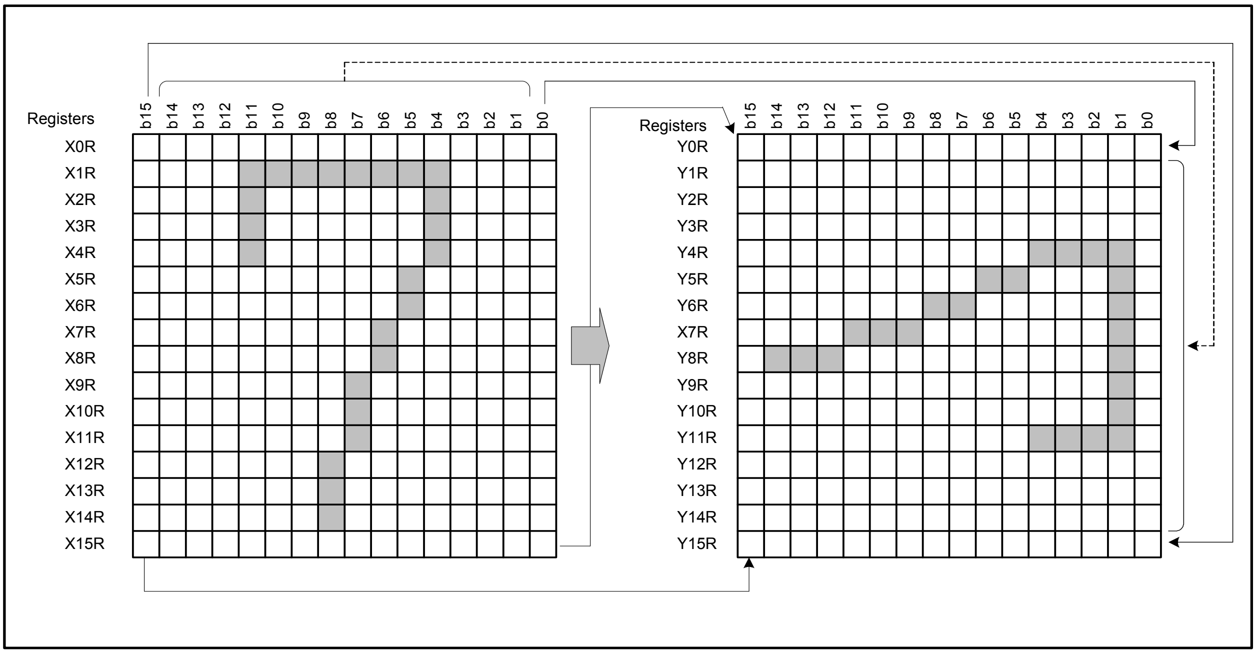


Figure 22.5 X-Y Conversion

When the XYC0 bit is set to 1 (no data rotation), the data of each bit in the YjR register is identical to that written in the XiR register. Figure 22.6 shows the conversion table when the XYC0 bit is set to 1.

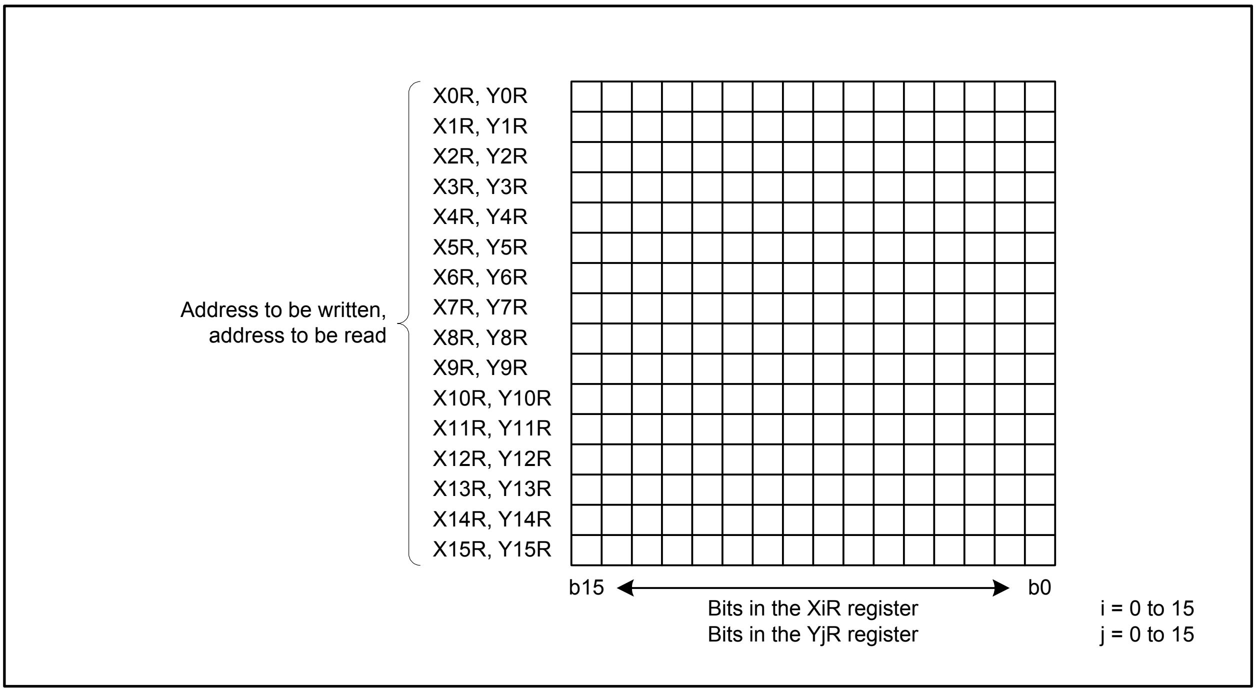


Figure 22.6 Conversion Table (XYC0 Bit is 1)

22.2 Data Conversion When Writing

Set the XYC1 bit in the XYC register to select a write mode for the XiR register.
When the XYC1 bit is set to 0 (no bit position reverse), the data is written in order. When it is set to 1 (bit position reverse), the data is written in reversed order. Figure 22.7 shows the conversion table when the XYC1 bit is set to 1.

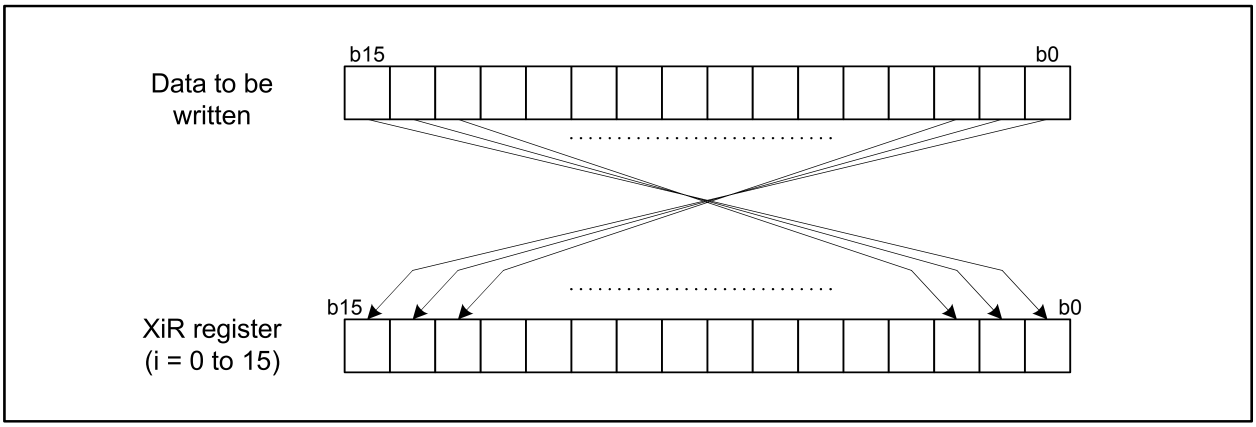


Figure 22.7 Conversion Table (XYC1 Bit is 1)

23. Intelligent I/O

The intelligent I/O is a multifunctional I/O port for time measurement, waveform generation, variable character length synchronous serial interface, and IEBus.

It consists of three groups each of which has one free-running 16-bit base timer and eight 16-bit registers for time measurement or waveform generation.

Table 23.1 lists the functions and channels of the intelligent I/O.

Table 23.1 Intelligent I/O Functions and Channels

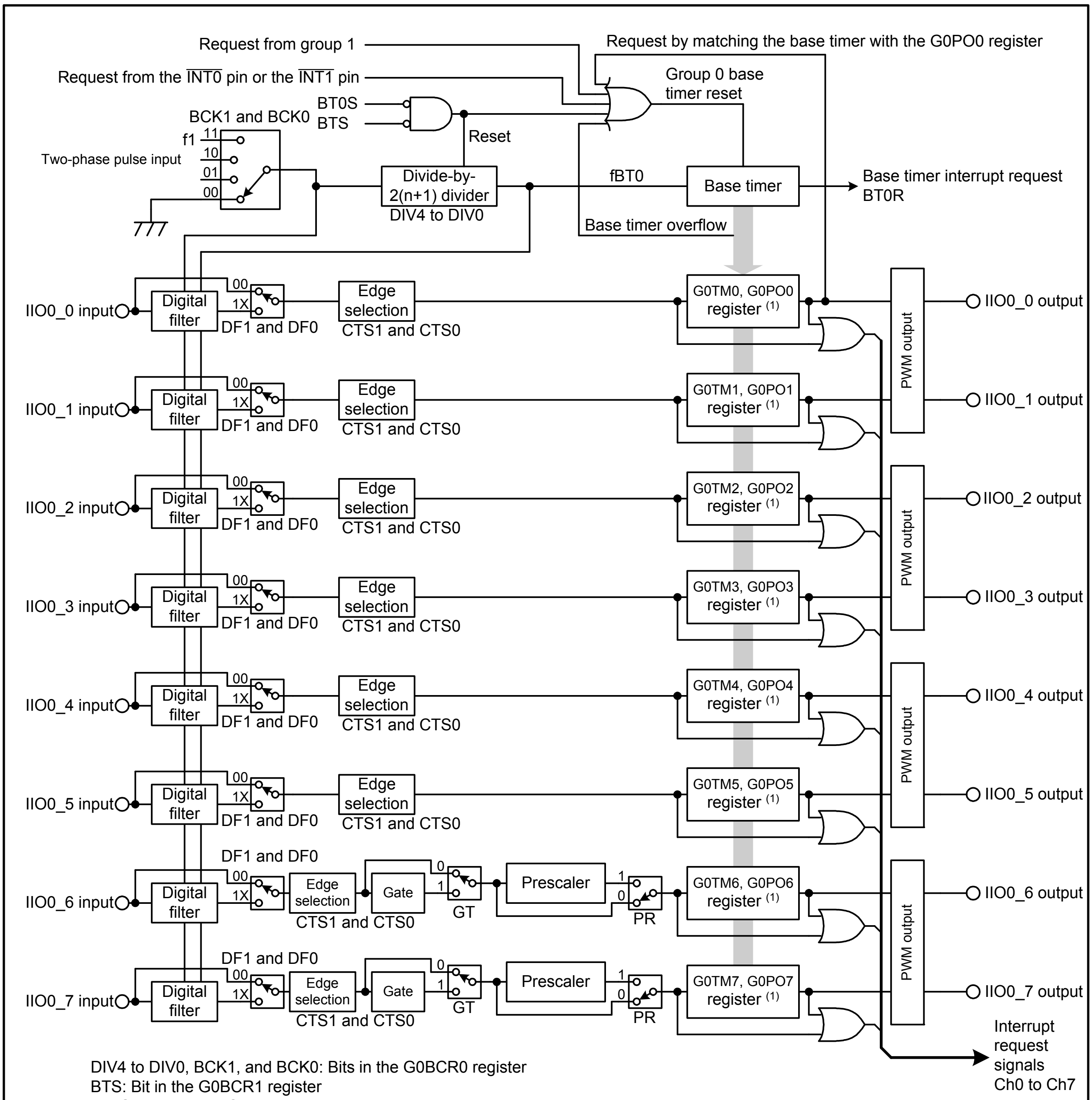
Functions		Group 0	Group 1	Group 2
Time measurement [(1)]	Digital filter	8 channels	8 channels	Not available
	Prescaler	2 channels	2 channels	
	Gating	2 channels	2 channels	
Waveform generation [(1)]	Single-phase waveform output mode	8 channels	8 channels	8 channels
	Inverted waveform output mode	8 channels	8 channels	8 channels
	SR waveform output mode	8 channels	8 channels	8 channels
	Bit modulation PWM mode	Not available	Not available	8 channels
	RTP mode			8 channels
	Parallel RTP mode			8 channels
Serial interface	Variable character length synchronous serial interface mode	Not available	Not available	Available
	IEBus mode (optional [(2)])			

Notes:

1. The time measurement and waveform generation functions share a pin.
2. Contact a Renesas Electronics sales office to use the optional features.

Each channel can be individually assigned for time measurement or waveform generation function.

Figures 23.1 to 23.3 show block diagrams of the intelligent I/O.

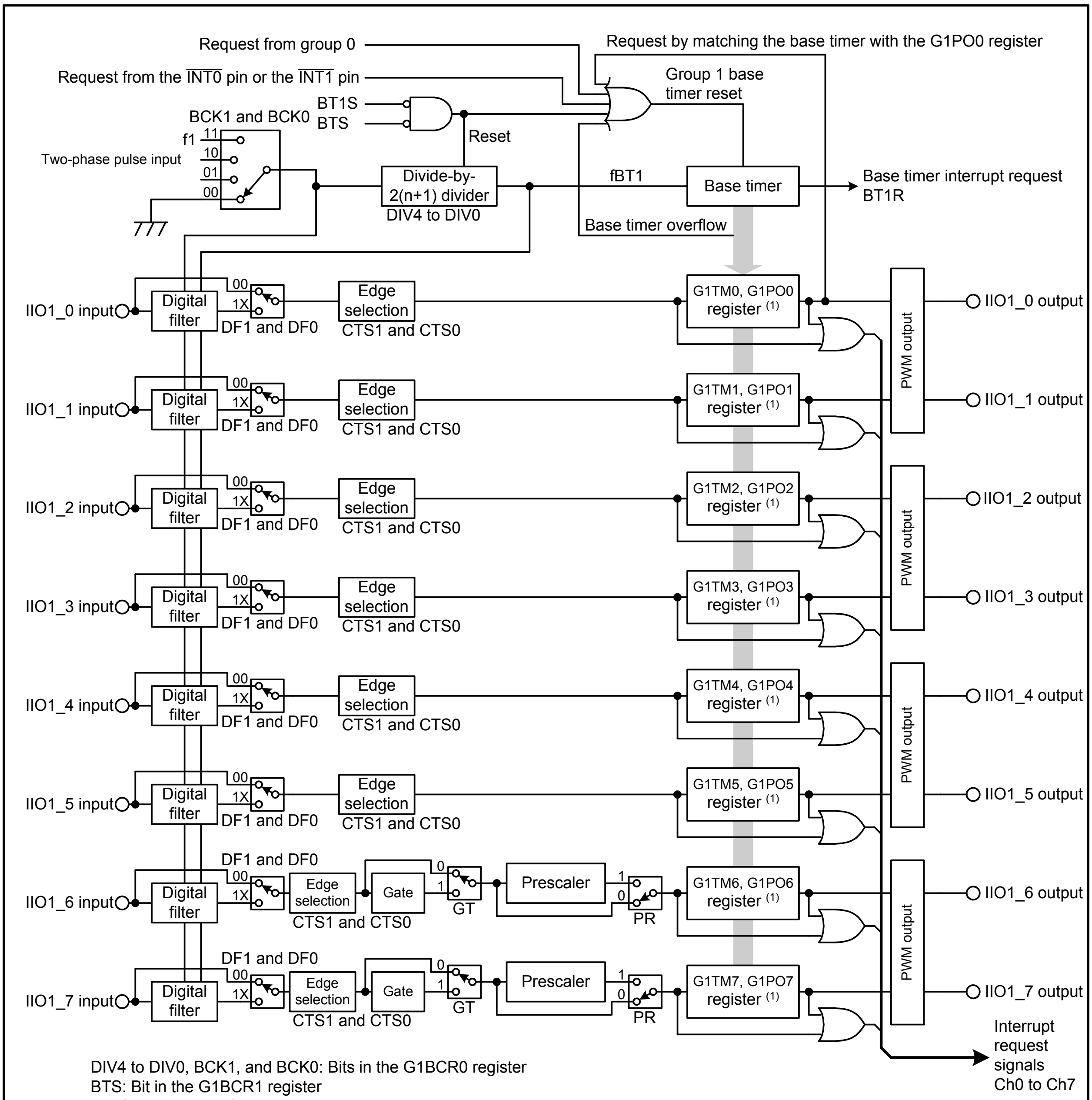


DIV4 to DIV0, BCK1, and BCK0: Bits in the G0BCR0 register
BTS: Bit in the G0BCR1 register
BT0S: Bit in the BTSR register
CTS1, CTS0, DF1, DF0, GT, and PR: Bits in the G0TMCRj register
BT0R: Bit in the IIO7IR register

Note:

1. Each register is placed in a reset state after the clock is provided via the G0BCR0 register.

Figure 23.1 Intelligent I/O Group 0 Block Diagram (j = 0 to 7)

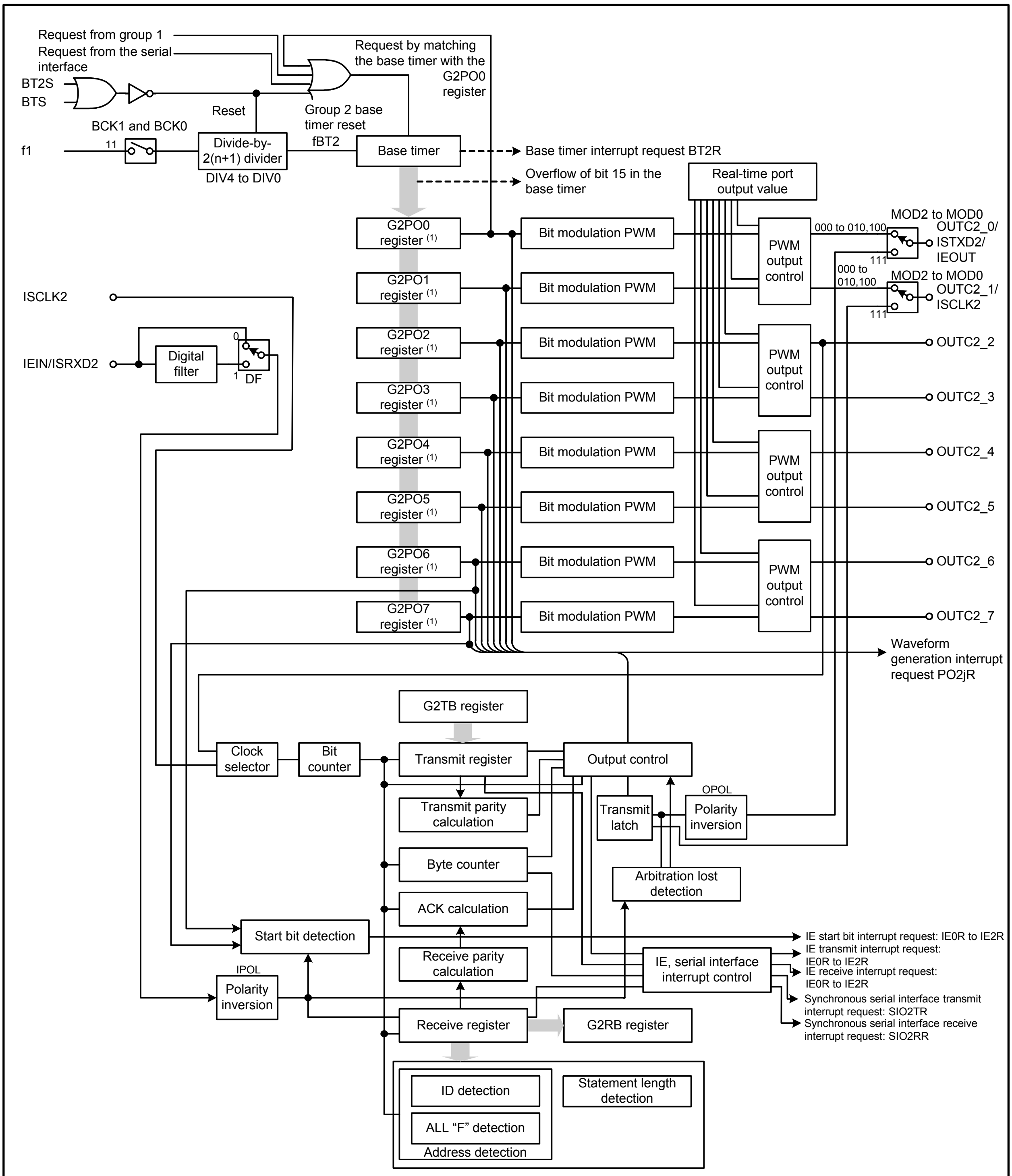


DIV4 to DIV0, BCK1, and BCK0: Bits in the G1BCR0 register
BTS: Bit in the G1BCR1 register
BT1S: Bit in the BTSR register
CTS1, CTS0, DF1, DF0, GT, and PR: Bits in the G1TMCRj register
BT1R: Bit in the IIO4IR register

Note:

1. Each register is placed in a reset state after the clock is provided via the G1BCR0 register.

Figure 23.2 Intelligent I/O Group 1 Block Diagram (j = 0 to 7)



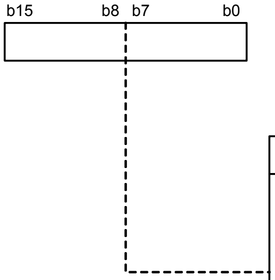
DIV4 to DIV0, BCK1, and BCK0: Bits in the G2BCR0 register
BTS: Bit in the G2BCR1 register
BT2S: Bit in the BTSR register
OPOL and IPOL: Bits in the G2CR register
DF: Bit in the IECR register
MOD2 to MOD0: Bits in the G2POCRj register
BT2R, PO2jR, IE0R to IE2R, SIO2TR, and SIO2RR: Bits in registers IIO3IR and IIO5IR to IIO11IR

Note:
1. Each register is placed in a reset state after the clock is provided via the G2BCR0 register.

Figure 23.3 Intelligent I/O Group 2 Block Diagram (j = 0 to 7)

Figures 23.4 to 23.17 show registers associated with the intelligent I/O base timer, time measurement, and waveform generation (for registers associated with the serial interface, refer to Figures 23.33 to 23.40).

Group i Base Timer Register (i = 0 to 2) (1)



Symbol	Address	Reset Value
G0BT, G1BT	01A1h-01A0h, 0121h-0120h	Undefined
G2BT	0161h-0160h	Undefined

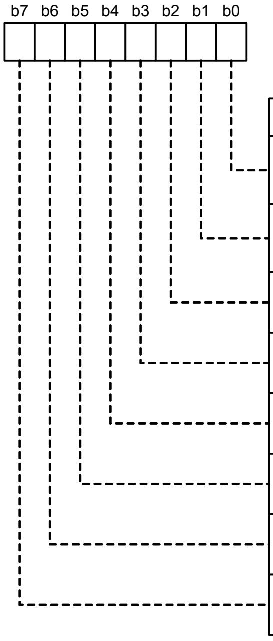
Function	Setting Range	RW
- While the base timer is running, this register indicates the value of base timer; when a value is written, the counter immediately starts counting from this value. The register is set to 0000h when the base timer is reset - While the base timer is being reset, this register is set to 0000h; the register is read as undefined value; no value can be set (2)	0000h to FFFFh	RW

Notes:

1. The GiBT register reflects the base timer value after a delay of a half fBTi cycle.
2. The base timer stops only when bits BCK1 and BCK0 in the GiBCR0 register are set to 00b (clock stopped). However, the base timer can be in a “no-counting” state, holding the value 0000h, by setting the BTiS bit in the BTSR register and the BTS bit in the GiBCR1 register to 0 (reset the base timer). When either of these bits is set to 1 (start counting), this state is cleared and the base timer starts counting.

Figure 23.4 Registers G0BT to G2BT

Group i Base Timer Control Register 0 (i = 0 to 2)



Symbol	Address	Reset Value
G0BCR0 to G2BCR0	01A2h, 0122h, 0162h	0000 0000b

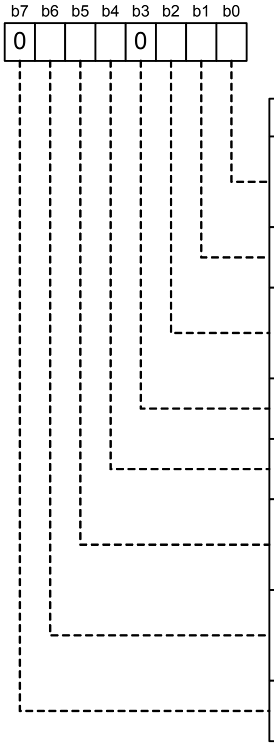
Bit Symbol	Bit Name	Function	RW
BCK0	Count Source Select Bit	b1 b2 0 0 : Clock stopped 0 1 : Do not use this combination 1 0 : Two-phase pulse signal input [1] 1 1 : f1	RW
BCK1			RW
DIV0	Count Source Divide Ratio Select Bit	Divide the count source by 2(n+1). The count source is not divided when n = 31 (n = 0 to 31). b6 b5 b4 b3 b2 0 0 0 0 0 : divide-by-2 (n = 0) 0 0 0 0 1 : divide-by-4 (n = 1) 0 0 0 1 0 : divide-by-6 (n = 2) : 1 1 1 1 0 : divide-by-62 (n = 30) 1 1 1 1 1 : no division (n = 31)	RW
DIV1			RW
DIV2			RW
DIV3			RW
DIV4			RW
IT	Base Timer Interrupt Source Select Bit	0: Overflow of bit 15 or bit 9 1: Overflow of bit 14	RW

Note:

1. This bit setting is enabled only when bits UD1 and UD0 in the GjBCR1 register are set to 10b (two-phase pulse signal processing mode) (j = 0, 1). Bits BCK1 and BCK0 should not be set to 10b in other modes or in group 2.

Figure 23.5 Registers G0BCR0 to G2BCR0

Group i Base Timer Control Register 1 (i = 0, 1)



Symbol	Address	Reset Value
G0BCR1, G1BCR1	01A3h, 0123h	0000 0000b

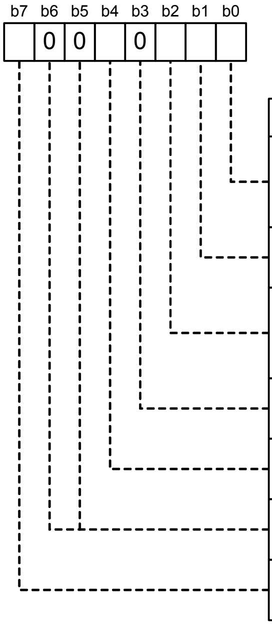
Bit Symbol	Bit Name	Function	RW
RST0	Base Timer Reset Source Select Bit 0	0: No reset 1: Synchronization with another base timer reset [1]	RW
RST1	Base Timer Reset Source Select Bit 1	0: No reset 1: Match with the GiPO0 register [2]	RW
RST2	Base Timer Reset Source Select Bit 2	0: No reset 1: Low signal input into the $\overline{\text{INT0}}$/$\overline{\text{INT1}}$ pin [3]	RW
— (b3)	Reserved	Should be written with 0	RW
BTS	Base Timer Start Bit [4, 5]	0: Reset the base timer 1: Start counting	RW
UD0	Increment/Decrement Control Bit	b6 b5 0 0 : Increment mode 0 1 : Increment/decrement mode 1 0 : Two-phase pulse signal processing mode [6] 1 1 : Do not use this combination	RW
UD1			RW
— (b7)	Reserved	Should be written with 0	RW

Notes:

1. The group 0 base timer is reset by synchronizing with the reset of group 1 base timer, and vice versa.
2. The base timer is reset after two fBTi clock cycles when the base timer value matches the GiPO0 register setting. When the RST1 bit is 1, the value of the GiPOj register used for waveform generation should be smaller than that of the GiPO0 register (j = 1 to 7).
3. The base timer is reset by an input of low signal to the external interrupt input pin selected for the UDiZ signal by the IFS2 register.
4. To start base timer group 0 and 1 individually, the BTS bit should be set to 1 after setting the BTkS bit in the BTSR register to 0 (reset the base timer) (k = 0, 1).
5. To start the base timers of multiple groups simultaneously, the BTSR register should be used. The BTS bit should be set to 0.
6. In two-phase pulse signal processing mode, the base timer is not reset, even if the RST1 bit is 1, if the timer counter decrements after two clock cycles when the base timer value matches the GiPO0 register.

Figure 23.6 Registers G0BCR1 and G1BCR1

Group 2 Base Timer Control Register 1



Symbol	Address	Reset Value
G2BCR1	0163h	0000 0000b

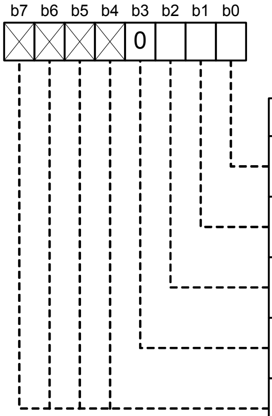
Bit Symbol	Bit Name	Function	RW
RST0	Base Timer Reset Source Select Bit 0	0: No reset 1: Synchronization with group1 base timer reset	RW
RST1	Base Timer Reset Source Select Bit 1	0: No reset 1: Match with the G2PO0 register (1)	RW
RST2	Base Timer Reset Source Select Bit 2	0: No reset 1: Reset request from the serial interface	RW
— (b3)	Reserved	Should be written with 0	RW
BTS	Base Timer Start Bit (2, 3)	0: Reset the base timer 1: Start counting	RW
— (b6-b5)	Reserved	Should be written with 0	RW
PRP	Parallel Real-time Port Select Bit (4)	0: RTP output mode 1: Parallel RTP output mode	RW

Notes:

1. The base timer is reset after two fBT2 clock cycles if the base timer value matches the G2PO0 register setting. When the RST1 bit is set to 1, the value of G2POj register used for waveform generation or the serial interface should be smaller than that of the G2PO0 register (j = 1 to 7).
2. To start the group 2 base timer, the BTS bit should be set to 1 after setting the BT2S bit in the BTSR register to 0 (reset the base timer).
3. To start the base timers of multiple groups simultaneously, the BTSR register should be used. The BTS bit should be set to 0.
4. This bit setting is enabled when the RTP bit in the G2POCRi register is set to 1 (real-time port used).

Figure 23.7 G2BCR1 Register

Base Timer Start Register [1, 2]



Symbol	Address	Reset Value
BTSR	0164h	XXXX 0000b

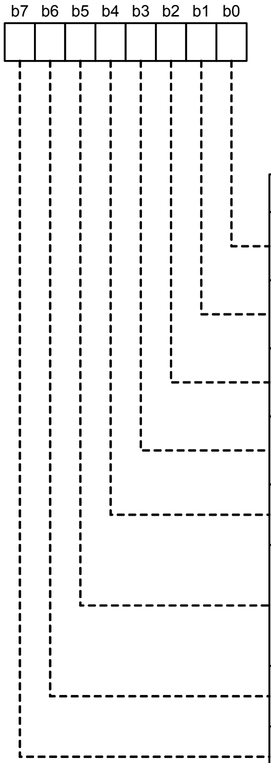
Bit Symbol	Bit Name	Function	RW
BT0S	Group 0 Base Timer Start Bit	0: Reset the base timer 1: Start counting	RW
BT1S	Group 1 Base Timer Start Bit	0: Reset the base timer 1: Start counting	RW
BT2S	Group 2 Base Timer Start Bit	0: Reset the base timer 1: Start counting	RW
— (b3)	Reserved	Should be written with 0	RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value.		—

Notes:

1. The following initial bit and register settings for the intelligent I/O should be performed:
   (1) Set the G2BCR0 register to provide the clock to the group 2 base timer.
   (2) Set all bits BT0S to BT2S to 0.
   (3) Set other registers associated with the intelligent I/O.
   The BTiS bit allows the base timers of two or all groups to start counting simultaneously (i = 0 to 2). To start counting individually, the BTiS bit should be set to 0 and the BTS bit in the GiBCR1 register should be used.
2. Perform the following procedure to start the base timers of multiple groups simultaneously:
   -Bits BCK1 to BCK0 and bits DIV4 to DIV0 in the GiBCR0 register to be used should be set identically (more than one of i = 0 to 2).
   -After bits BCK1 to BCK0 or bits DIV4 to DIV0 are changed, use the following procedure to start the base timers twice:
   (1) Set the BTiS bit to 1.
   (2) Set the BTiS bit to 0 after one fBTi clock cycle.
   (3) Set the BTiS bit to 1 again after one additional fBTi clock cycle.

Figure 23.8 BTSR Register

Group i Time Measurement Control Register j (i = 0, 1; j = 0 to 7)



Symbol	Address	Reset Value
G0TMCR0 to G0TMCR3	0198h, 0199h, 019Ah, 019Bh	0000 0000b
G0TMCR4 to G0TMCR7	019Ch, 019Dh, 019Eh, 019Fh	0000 0000b
G1TMCR0 to G1TMCR3	0118h, 0119h, 011Ah, 011Bh	0000 0000b
G1TMCR4 to G1TMCR7	011Ch, 011Dh, 011Eh, 011Fh	0000 0000b

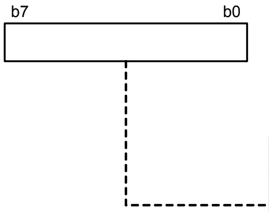
Bit Symbol	Bit Name	Function	RW
CTS0	Time Measurement Trigger Select Bit	b1 b0 0 0 : No time measurement 0 1 : Rising edge 1 0 : Falling edge 1 1 : Both edges	RW
CTS1			RW
DF0	Digital Filter Select Bit	b3 b2 0 0 : No digital filter used 0 1 : Do not use this combination 1 0 : fBTi 1 1 : f1	RW
DF1			RW
GT	Gating Select Bit [1]	0: Gating disabled 1: Gating enabled	RW
GOC	Gating Clear Select Bit [1, 2]	0: Gating not cleared 1: Gating cleared when the base timer matches the GiPOk register (k = j - 2)	RW
GSC	Gating Clear Bit [1, 2]	Gating is cleared by setting this bit to 1	RW
PR	Prescaler Select Bit [1]	0: Prescaler disabled 1: Prescaler enabled	RW

Notes:

1. These functions are available in registers GiTMCR6 and GiTMCR7. Bits 4 to 7 in registers GiTMCR0 to GiTMCR5 should be set to 0.
2. These bit settings are enabled when the GT bit is 1.

Figure 23.9 Registers G0TMCR0 to G0TMCR7 and G1TMCR0 to G1TMCR7

Group i Time Measurement Prescaler Register j (i = 0, 1; j = 6, 7)



Symbol	Address	Reset Value
G0TPR6, G0TPR7	01A4h, 01A5h	00h
G1TPR6, G1TPR7	0124h, 0125h	00h

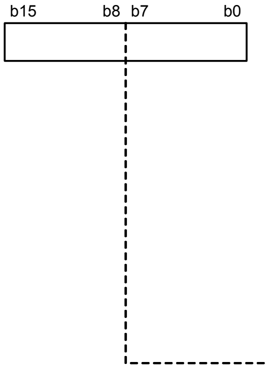
Function	Setting Range	RW
Time measurement is executed whenever a trigger input is counted by n+1 (n = setting value) [1]	00h to FFh	RW

Note:

1. The first prescaler, after the PR bit in the GiTMCRj register is changed from 0 (prescaler disabled) to 1 (prescaler enabled), may be divided by n rather than n+1. The subsequent prescaler is divided by n+1.

Figure 23.10 Registers G0TPR6, G0TPR7, G1TPR6, and G1TPR7

Group i Time Measurement Register j (i = 0, 1; j = 0 to 7)

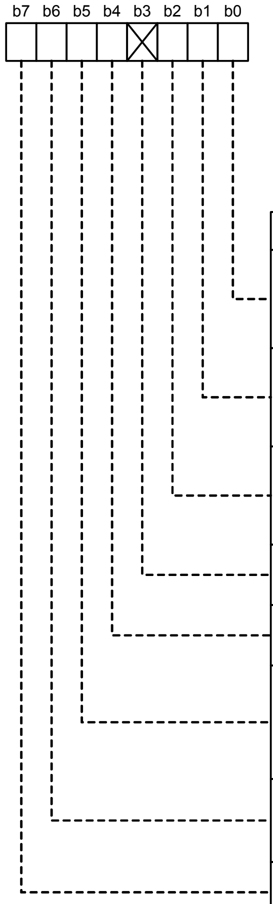


Symbol	Address	Reset Value
G0TM0, G0TM1	0181h-0180h, 0183h-0182h	Undefined
G0TM2, G0TM3	0185h-0184h, 0187h-0186h	Undefined
G0TM4, G0TM5	0189h-0188h, 018Bh-018Ah	Undefined
G0TM6, G0TM7	018Dh-018Ch, 018Fh-018Eh	Undefined
G1TM0, G1TM1	0101h-0100h, 0103h-0102h	Undefined
G1TM2, G1TM3	0105h-0104h, 0107h-0106h	Undefined
G1TM4, G1TM5	0109h-0108h, 010Bh-010Ah	Undefined
G1TM6, G1TM7	010Dh-010Ch, 010Fh-010Eh	Undefined

Function	Setting Range	RW
The base timer value is stored every measurement timing	—	RO

Figure 23.11 Registers G0TM0 to G0TM7 and G1TM0 to G1TM7

Group i Waveform Generation Control Register j (i = 0, 1; j = 0 to 7)



Symbol	Address	Reset Value
G0POCR0	0190h	0000 X000b
G0POCR1 to G0POCR3	0191h, 0192h, 0193h	0X00 X000b
G0POCR4 to G0POCR7	0194h, 0195h, 0196h, 0197h	0X00 X000b
G1POCR0	0110h	0000 X000b
G1POCR1 to G1POCR3	0111h, 0112h, 0113h	0X00 X000b
G1POCR4 to G1POCR7	0114h, 0115h, 0116h, 0117h	0X00 X000b

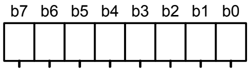
Bit Symbol	Bit Name	Function	RW
MOD0	Operating Mode Select Bit	b2 b1 b0 0 0 0 : Single-phase waveform output mode 0 0 1 : SR waveform output mode [1] 0 1 0 : Inverted waveform output mode 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
MOD1			RW
MOD2			RW
— (b3)	No register bit; should be written with 0 and read as undefined value		—
IVL	Default Output Value Select Bit [2]	0: Output low as default value 1: Output high as default value	RW
RLD	GiPOj Register Value Reload Timing Select Bit	Reload the value into the GiPOj register 0: On a write access 1: When the base timer is reset	RW
BTRE	Base Timer Reset Enable Bit [3]	Reset the base timer when 0: Bit 15 overflows 1: Bit 9 overflows [4]	RW
INV	Output Level Inversion Select Bit [5]	0: Do not invert the output level 1: Invert the output level	RW

Notes:

1. This bit setting is enabled only for even channels. In SR waveform output mode, the corresponding odd channel (the next channel after an even channel) setting is ignored. Waveforms are only output from even channels.
2. The setting value is output by a write operation to the IVL bit when the FSCj bit in the GiFS register is 0 (select the waveform generation) and the IFEj bit in the GiFE register is 1 (enable the channel j function).
3. This bit is available only in the GiPOCR0 register. Set bit 6 in registers GiPOCR1 to GiPOCR7 to 0.
4. To set the BTRE bit to 1, set bits BCK1 and BCK0 in the GiBCR0 register to 11b (f1) and bits UD1 and UD0 in the GiBCR1 register to 00b (increment mode).
5. The output level inversion is the final step in the waveform generation process. When the INV bit is 1, high is output by setting the IVL bit to 0, and vice versa.

Figure 23.12 Registers G0POCR0 to G0POCR7 and G1POCR0 to G1POCR7

Group 2 Waveform Generation Control Register j (j = 0 to 7)



Symbol	Address	Reset Value
G2POCR0 to G2POCR3	0150h, 0151h, 0152h, 0153h	0000 0000b
G2POCR4 to G2POCR7	0154h, 0155h, 0156h, 0157h	0000 0000b

Bit Symbol	Bit Name	Function	RW
MOD0	Operating Mode Select Bit [1]	b2 b1 b0 0 0 0 : Single waveform output mode 0 0 1 : SR waveform output mode [2] 0 1 0 : Inverted waveform output mode 0 1 1 : Do not use this combination 1 0 0 : Bit modulation PWM output mode 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Use an output for the serial interface [3]	RW
MOD1			RW
MOD2			RW
PRT	Parallel Real-time Port Output Trigger Select Bit [4]	0: Not triggered by matching the base timer with registers G2PO0 to G2PO7 1: Triggered by matching the base timer with registers G2PO0 to G2PO7	RW
IVL	Default Output Value Select Bit	0: Output low as default value 1: Output high as default value	RW
RLD	G2POj Register Value Reload Timing Select Bit	0: Reload the value into the G2POj register on a write access 1: Reload the value into the G2POj register when the base timer is reset	RW
RTP	Real-time Port Select Bit	0: No real-time port function used 1: Use RTP output mode or parallel RTP output mode	RW
INV	Output Level Inversion Select Bit [5]	0: Do not invert the output level 1: Invert the output level	RW

Notes:

1. When the RTP bit is set to 1, the settings of bits MOD2 to MOD0 are disabled.
2. This bit setting is enabled only for even channels. In SR waveform output mode, the corresponding odd channel (the next channel after an even channel) setting is ignored. Waveforms are only output from even channels.
3. This bit setting is enabled only for channels 0 and 1 of group 2. To use the ISTXD2 or IEOUT pin as an output, set bits MOD2 to MOD0 in the G2POCR0 register to 111b. To use the ISCLK2 pin, set the same bits in the G2POCR1 register to 111b. This bit setting should only be performed with channels 0 and 1.
4. This bit setting is enabled when the RTP bit is 1 and the PRP bit in the G2BCR1 register is 1 (parallel RTP output mode).
5. The output level inversion is the final step in the waveform generation process. When the INV bit is 1, high is output by setting the IVL bit to 0, and vice versa.

Figure 23.13 Registers G2POCR0 to G2POCR7

Group i Waveform Generation Register j (i = 0 to 2; j = 0 to 7)

b15 b8 b7 b0

Symbol	Address	Reset Value
G0PO0 to G0PO2	0181h-0180h, 0183h-0182h, 0185h-0184h	Undefined
G0PO3 to G0PO5	0187h-0186h, 0189h-0188h, 018Bh-018Ah	Undefined
G0PO6, G0PO7	018Dh-018Ch, 018Fh-018Eh	Undefined
G1PO0 to G1PO2	0101h-0100h, 0103h-0102h, 0105h-0104h	Undefined
G1PO3 to G1PO5	0107h-0106h, 0109h-0108h, 010Bh-010Ah	Undefined
G1PO6, G1PO7	010Dh-010Ch, 010Fh-010Eh	Undefined
G2PO0 to G2PO2	0141h-0140h, 0143h-0142h, 0145h-0144h	Undefined
G2PO3 to G2PO5	0147h-0146h, 0149h-0148h, 014Bh-014Ah	Undefined
G2PO6, G2PO7	014Dh-014Ch, 014Fh-014Eh	Undefined

Function	Setting Range	RW
- When the RLD bit in the GiPOCRj register is 0, the value is reloaded into the GiPOj register immediately after being written and is reflected in the output waveform - When the RLD bit is 1, the value is reloaded when the base timer is reset. The register indicates the written value until the value is reloaded	0000h to FFFFh	RW

Figure 23.14 Registers G0PO0 to G0PO7, G1PO0 to G1PO7, and G2PO0 to G2PO7

Group i Function Select Register (i = 0, 1)

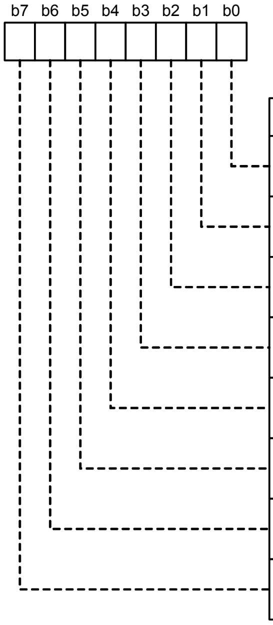
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
G0FS, G1FS	01A7h, 0127h	0000 0000b

Bit Symbol	Bit Name	Function	RW
FSC0	Channel 0 Time Measurement/Waveform Generation Select Bit	0: Select the waveform generation 1: Select the time measurement	RW
FSC1	Channel 1 Time Measurement/Waveform Generation Select Bit		RW
FSC2	Channel 2 Time Measurement/Waveform Generation Select Bit		RW
FSC3	Channel 3 Time Measurement/Waveform Generation Select Bit		RW
FSC4	Channel 4 Time Measurement/Waveform Generation Select Bit		RW
FSC5	Channel 5 Time Measurement/Waveform Generation Select Bit		RW
FSC6	Channel 6 Time Measurement/Waveform Generation Select Bit		RW
FSC7	Channel 7 Time Measurement/Waveform Generation Select Bit		RW

Figure 23.15 Registers G0FS and G1FS

Group i Function Enable Register (i = 0 to 2)



Symbol
G0FE to G2FE

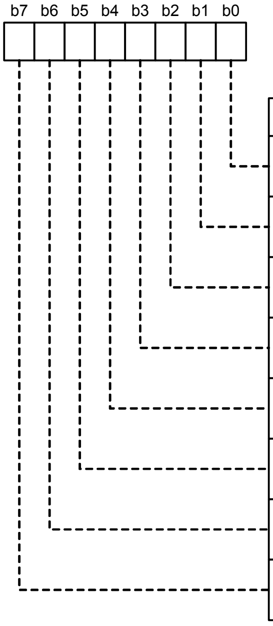
Address
01A6h, 0126h, 0166h

Reset Value
0000 0000b

Bit Symbol	Bit Name	Function	RW
IFE0	Channel 0 Function Enable Bit	0: Disable the channel j function 1: Enable the channel j function (j = 0 to 7)	RW
IFE1	Channel 1 Function Enable Bit		RW
IFE2	Channel 2 Function Enable Bit		RW
IFE3	Channel 3 Function Enable Bit		RW
IFE4	Channel 4 Function Enable Bit		RW
IFE5	Channel 5 Function Enable Bit		RW
IFE6	Channel 6 Function Enable Bit		RW
IFE7	Channel 7 Function Enable Bit		RW

Figure 23.16 Registers G0FE to G2FE

Group 2 RTP Output Buffer Register



Symbol
G2RTP

Address
0167h

Reset Value
0000 0000b

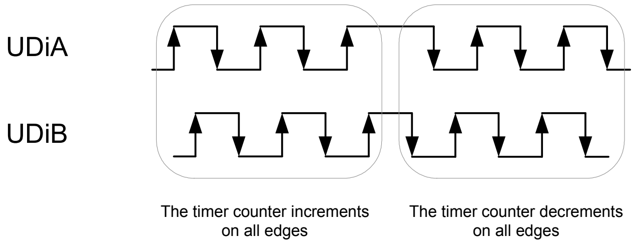
Bit symbol	Bit Name	Function	RW
RTP0	Channel 0 RTP Output Buffer	0: Output a low level 1: Output a high level	RW
RTP1	Channel 1 RTP Output Buffer		RW
RTP2	Channel 2 RTP Output Buffer		RW
RTP3	Channel 3 RTP Output Buffer		RW
RTP4	Channel 4 RTP Output Buffer		RW
RTP5	Channel 5 RTP Output Buffer		RW
RTP6	Channel 6 RTP Output Buffer		RW
RTP7	Channel 7 RTP Output Buffer		RW

Figure 23.17 G2RTP Register

23.1 Base Timer for Groups 0 to 2

The base timer is a free-running counter that counts an internally generated count source. Table 23.2 lists specifications of the base timer. Figures 23.4 to 23.17 show registers associated with the base timer. Figure 23.18 shows a block diagram of the base timer. Figures 23.19, 23.20, and 23.21 show operation examples of the base timer for groups 0 and 1 in increment mode, increment/decrement mode, and two-phase pulse signal processing mode, respectively.

Table 23.2 Base Timer Specifications (i = 0 to 2)

Item	Specification
Count source (fBTi)	f1 divided by 2(n+1) for groups 0 to 2, two-phase pulse input divided by 2(n+1) for groups 0 and 1 n: setting value using bits DIV4 to DIV0 in the GiBCR0 register n = 0 to 31; however no division when n = 31
Count operations	• Increment • Increment/decrement • Two-phase pulse signal processing
Count start conditions	• To start each base timer individually, The BTS bit in the GiBCR1 register is 1 (start counting) • To start the base timers of multiple groups simultaneously, The BTiS bit in the BTSR register is 1 (start counting)
Count stop condition	The BTiS bit in the BTSR register and the BTS bit in the GiBCR1 register are 0 (reset the base timer)
Reset conditions	• The base timer value matches the GiPO0 register setting • An input of low signal into the external interrupt pin ($\overline{\text{INT0}}$ or $\overline{\text{INT1}}$) as follows: for group 0: selected using bits IFS23 and IFS22 in the IFS2 register for group 1: selected using bits IFS27 and IFS26 in the IFS2 register • The overflow of bit 15 or bit 9 in the base timer • The base timer reset request from the communication functions (group 2)
Reset value	0000h
Interrupt request	When the BTiR bit in the interrupt request register becomes 1 (interrupt requested) by the overflow of bit 9, 14, or 15 in the base timer (refer to Figure 11.12)
Read from base timer	• The GiBT register indicates a counter value while the base timer is running • The GiBT register is undefined while the base timer is being reset
Write to base timer	When a value is written while the base timer is running, the timer counter immediately starts counting from this value. No value can be written while the base timer is being reset
Other functions	• Increment/decrement mode for groups 0 and 1 The base timer starts counting when the BTS or BTiS bit is set to 1. When the base timer reaches FFFFh, it starts decrementing. When the RST1 bit in the GiBCR1 register is 1 (the base timer is reset by matching with the GiPO0 register), the timer counter starts decrementing two counts after the base timer value matches the GiPO0 register setting. When the timer counter reaches 0000h, it starts incrementing again (refer to Figure 23.20). • Two-phase pulse signal processing mode for groups 0 and 1 Two-phase pulse signals at pins UDiA and UDiB are counted (refer to Figure 23.21).  UDiA UDiB The timer counter increments on all edges The timer counter decrements on all edges

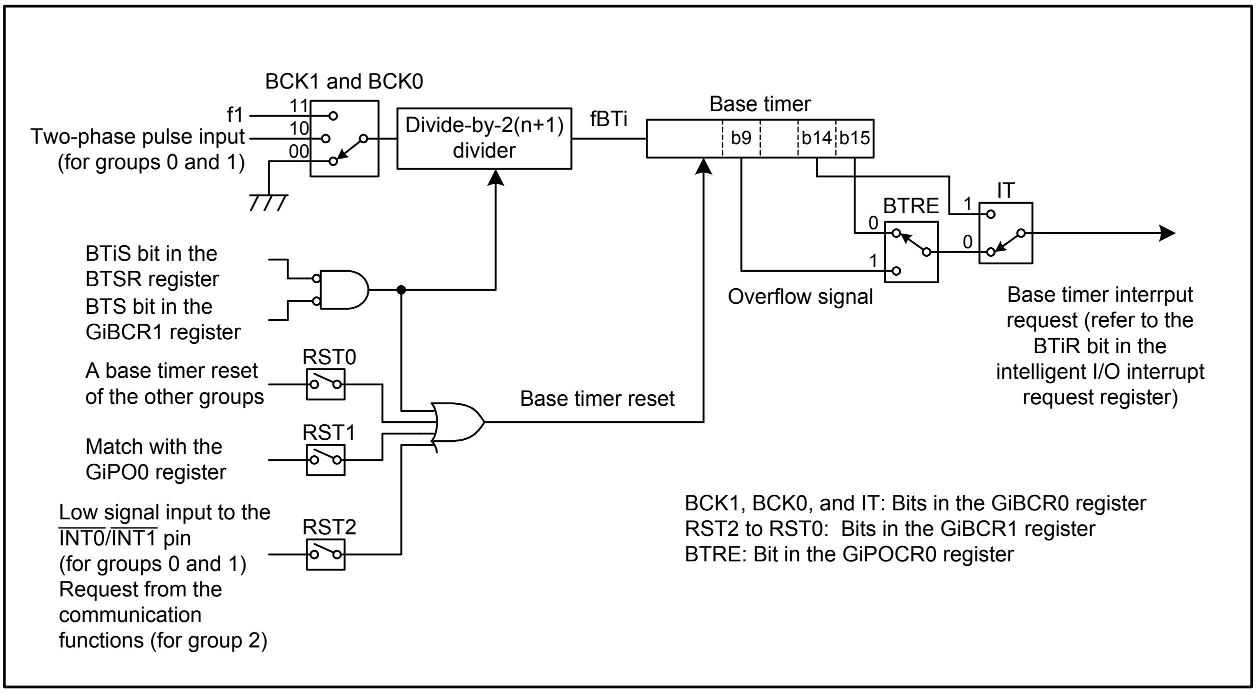

BCK1 and BCK0
f1
Two-phase pulse input
(for groups 0 and 1)
11
10
00
Divide-by-2(n+1) divider
fBTi
Base timer
b9
b14
b15
BTRE
IT
0
1
1
0
Overflow signal
Base timer interrput request (refer to the BTiR bit in the intelligent I/O interrupt request register)
BTiS bit in the BTSR register
BTS bit in the GiBCR1 register
A base timer reset of the other groups
RST0
Base timer reset
Match with the GiPO0 register
RST1
Low signal input to the INT0/INT1 pin (for groups 0 and 1)
Request from the communication functions (for group 2)
RST2
BCK1, BCK0, and IT: Bits in the GiBCR0 register
RST2 to RST0: Bits in the GiBCR1 register
BTRE: Bit in the GiPOCR0 register

Figure 23.18 Base Timer Block Diagram (i = 0 to 2)

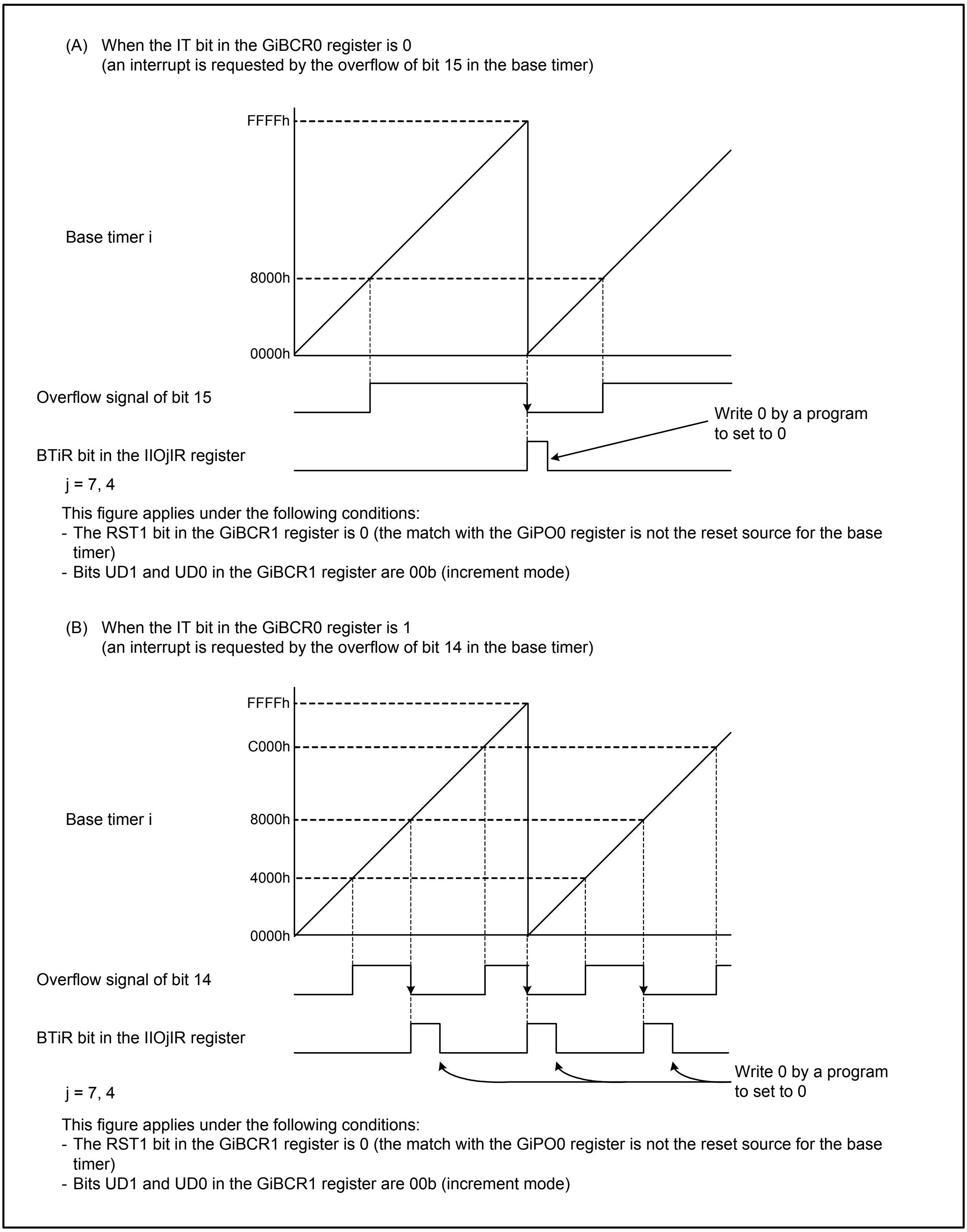
Table 23.3 Base Timer Associated Register Settings (Common Settings for Time Measurement, Waveform Generation, and Serial Interface) (i = 0 to 2)

Register	Bits	Function
G2BCR0	—	Provide an operating clock to the BTSR register. Set to 0111 1111b
BTSR	—	Set to 0000 0000b
GiBCR0	BCK1 and BCK0	Select a count source
	DIV4 to DIV0	Select a count source divide ratio
	IT	Select a base timer interrupt source
GiBCR1	RST2 to RST0	Select a timing for base timer reset
	BTS	Use this bit when each base timer individually starts counting
	UD1 and UD0	Select a count mode in groups 0 and 1
GiPOCR0	BTRE	Select a source for base timer reset
GiBT	—	Read or write the base timer value

The following register settings are required to set the RST1 bit to 1 (the base timer is reset by matching with the GiPO0 register).

GiPOCR0	MOD2 to MOD0	Set to 000b (single-phase waveform output mode)
GiPO0	—	Set the reset cycle
GiFS	FSC0	Set the bit to 0 (select the waveform generation)
GiFE	IFE0	Set the bit to 1 (channel operation starts)

Bit configurations and functions vary by group.



(A) When the IT bit in the GiBCR0 register is 0
(an interrupt is requested by the overflow of bit 15 in the base timer)

j = 7, 4

This figure applies under the following conditions:

- The RST1 bit in the GiBCR1 register is 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)

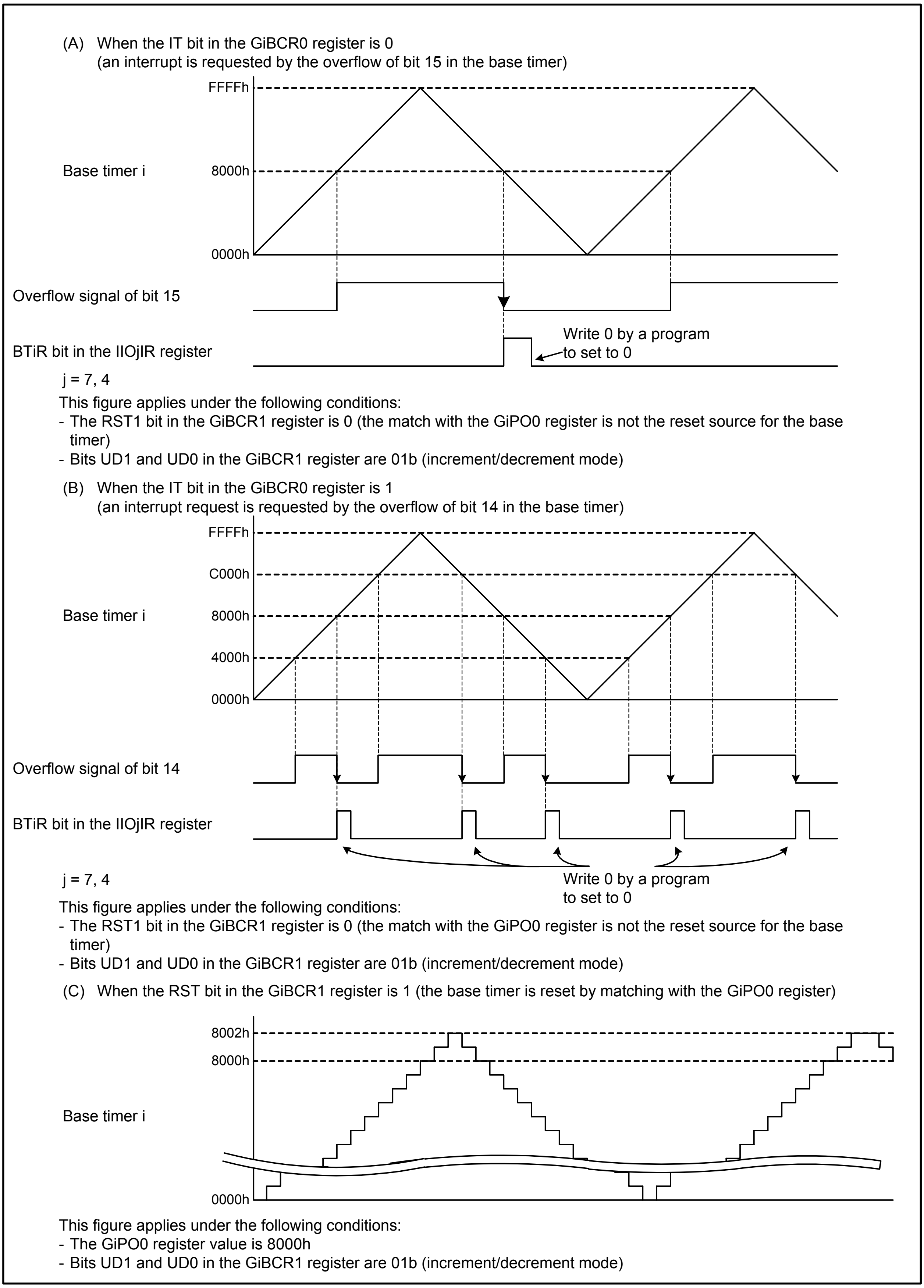
(B) When the IT bit in the GiBCR0 register is 1
(an interrupt is requested by the overflow of bit 14 in the base timer)

j = 7, 4

This figure applies under the following conditions:

- The RST1 bit in the GiBCR1 register is 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)

Figure 23.19 Base Timer Increment Mode (i = 0, 1) (for Groups 0 and 1)



(A) When the IT bit in the GiBCR0 register is 0
(an interrupt is requested by the overflow of bit 15 in the base timer)

j = 7, 4

This figure applies under the following conditions:
- The RST1 bit in the GiBCR1 register is 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are 01b (increment/decrement mode)

(B) When the IT bit in the GiBCR0 register is 1
(an interrupt request is requested by the overflow of bit 14 in the base timer)

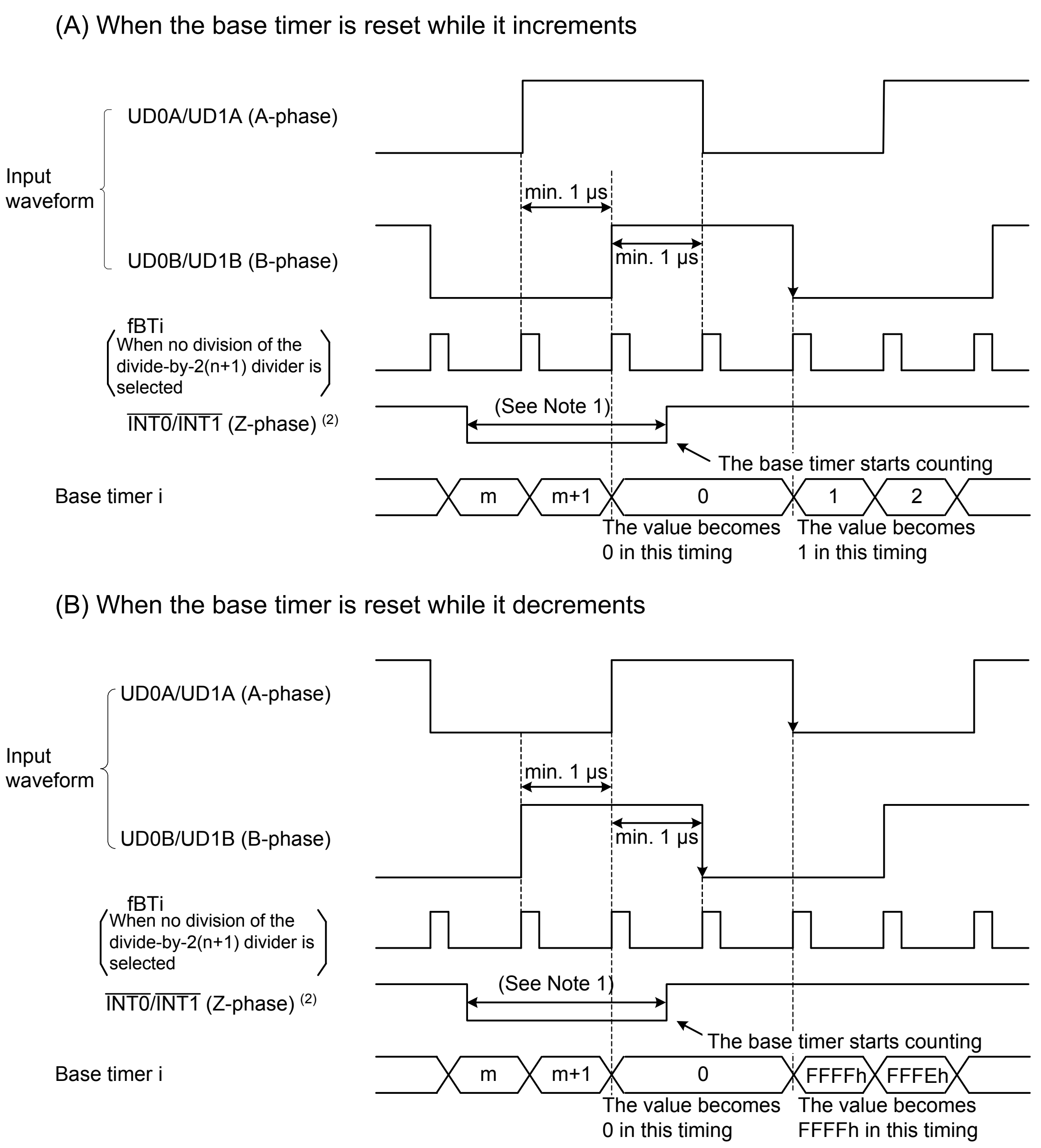
j = 7, 4

This figure applies under the following conditions:
- The RST1 bit in the GiBCR1 register is 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are 01b (increment/decrement mode)

(C) When the RST bit in the GiBCR1 register is 1 (the base timer is reset by matching with the GiPO0 register)

This figure applies under the following conditions:
- The GiPO0 register value is 8000h
- Bits UD1 and UD0 in the GiBCR1 register are 01b (increment/decrement mode)

Figure 23.20 Base Timer Increment/Decrement (i = 0, 1) (for Groups 0 and 1)


(A) When the base timer is reset while it increments
Input waveform
UD0A/UD1A (A-phase)
UD0B/UD1B (B-phase)
min. 1 μs
min. 1 μs
fBTi
When no division of the divide-by-2(n+1) divider is selected
INT0/INT1 (Z-phase) (2)
(See Note 1)
The base timer starts counting
Base timer i
m
m+1
0
1
2
The value becomes 0 in this timing
The value becomes 1 in this timing
(B) When the base timer is reset while it decrements
Input waveform
UD0A/UD1A (A-phase)
UD0B/UD1B (B-phase)
min. 1 μs
min. 1 μs
fBTi
When no division of the divide-by-2(n+1) divider is selected
INT0/INT1 (Z-phase) (2)
(See Note 1)
The base timer starts counting
Base timer i
m
m+1
0
FFFFh
FFFEh
The value becomes 0 in this timing
The value becomes FFFFh in this timing

Notes:
1. At least 1.5 fBTi clock cycles are required.
2. Set the RST2 bit in the GiBCR1 register to 1 in two-phase pulse signal processing mode.

Figure 23.21 Base Timer Two-phase Pulse Signal Processing Mode (i = 0, 1) (for Groups 0 and 1)

23.2 Time Measurement (for Groups 0 and 1)

Every time an external trigger is input, the base timer value is stored into the GiTMj register (i = 0, 1; j = 0 to 7). Table 23.4 lists specifications of the time measurement and Table 23.5 lists its register settings. Figures 23.22 and 23.23 show operation examples of the time measurement and Figure 23.24 shows operation examples with the prescaler or gate function.

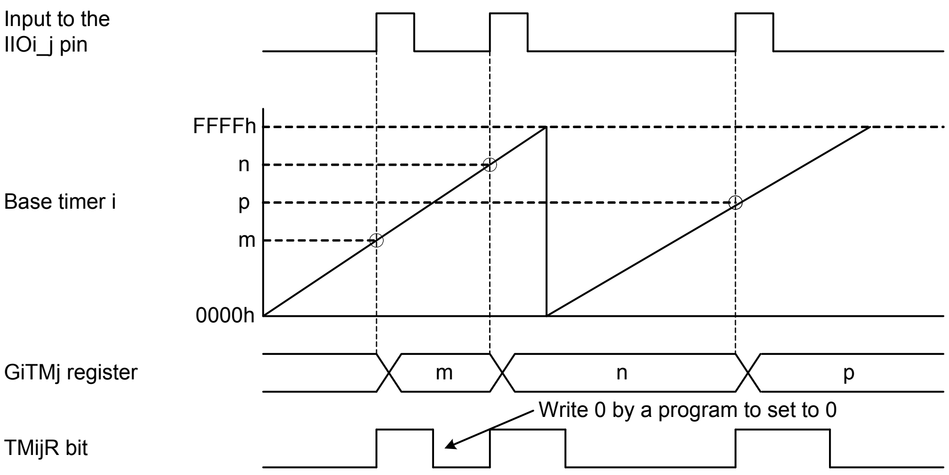
Table 23.4 Time Measurement Specifications (i = 0, 1; j = 0 to 7)

Item	Specification
Time measurement channels	Group 0: Channels 0 to 7 Group 1: Channels 0 to 7
Trigger input polarity	Rising edge, falling edge, or both edges of the IIOi_j pin
Time measurement start condition	The IFEj bit in the GiFE register is 1 (enable the channel j function) while the FSCj bit in the GiFS register is 1 (select the time measurement)
Time measurement stop condition	The IFEj bit is 0 (disable the channel j function)
Time measurement timing	• Without the prescaler: every time a trigger is input • With the prescaler for channels 6 and 7: every [GiTPRk register value + 1] times a trigger is input (k = 6, 7)
Interrupt request	When the TMijR bit in the interrupt request register becomes 1 (interrupt requested) (refer to Figure 11.12)
IIOi_j input pin function	Trigger input
Other functions	• Digital filter The digital filter determines a trigger input level every f1 or fBTi cycle and passes the signals holding the same level during three sequential cycles • Prescaler for channels 6 and 7 Time measurement is executed every [GiTPRk register value + 1] times a trigger is input • Gating for channels 6 and 7 This function disables any trigger input to be accepted after the time measurement by the first trigger input. However, the trigger input can be accepted again if any of following conditions are met while the GOC bit in the GiTMCRk register is 1 (the gating is cleared when the base timer matches the GiPOp register) (p = 4, 5; p = 4 when k = 6; p = 5 when k = 7): • The base timer value matches the GiPOp register setting • The GSC bit in the GiTMCRk register is 1

Table 23.5 Time Measurement (for Groups 0 and 1) Associated Register Settings (i = 0, 1; j = 0 to 7; k = 6, 7)

Register	Bits	Function
GiTMCRj	CTS1 and CTS0	Select a time measurement trigger
	DF1 and DF0	Select a digital filter
	GT, GOC, GSC	Select if the gating is used
	PR	Select if the prescaler is used
GiTPRk	—	Set the prescaler value
GiFS	FSCj	Set the bit to 1 (select the time measurement)
GiFE	IFEj	Set the bit to 1 (enable the channel j function)

Bit configurations and functions vary with channels and groups.
Registers associated with the time measurement should be set after setting the base timer-associated registers.

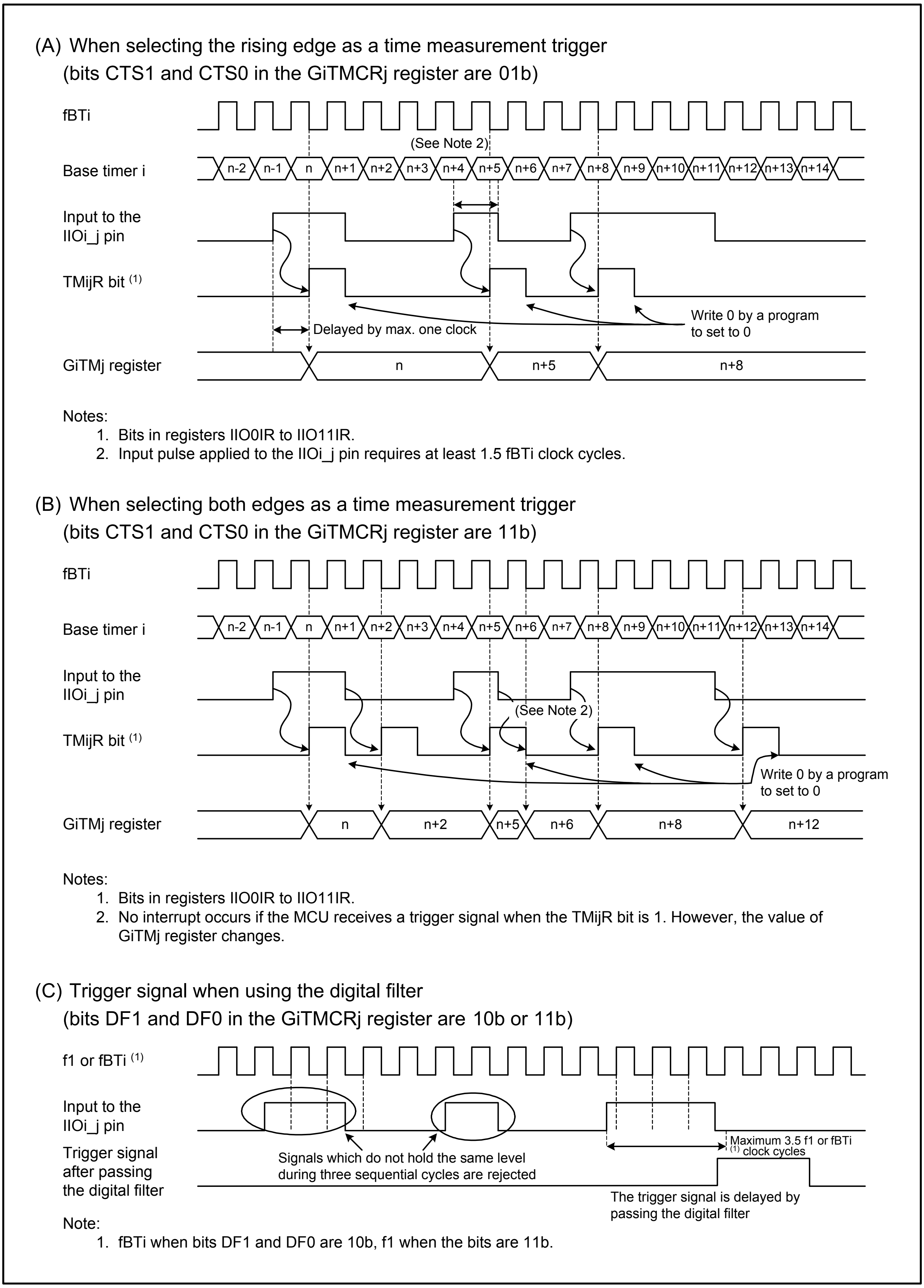


TMijR: Bits in registers IIO0IR to IIO11IR

This figure applies under the following conditions:
- Bits CTS1 and CTS0 in the GiTMCRj register are 01b (rising edge as time measurement trigger), the PR bit is 0 (prescaler disabled), and the GT bit is to 0 (gating disabled)
- Bits RST2 to RST0 in the GiBCR1 register are 000b (reset the base timer) and bits UD1 and UD0 are 00b (increment mode)

When the base timer is reset by matching with the GiPO0 register (bits RST2 to RST0 in the GiBCR1 register are 010b), the base timer becomes 0000h after it reaches the GiPO0 register setting value + 2.

Figure 23.22 Time Measurement Operation (1/2) (i = 0, 1; j = 0 to 7)



(A) When selecting the rising edge as a time measurement trigger
(bits CTS1 and CTS0 in the GiTMCRj register are 01b)

Notes:
1. Bits in registers IIO0IR to IIO11IR.
2. Input pulse applied to the IIOi_j pin requires at least 1.5 fBTi clock cycles.

(B) When selecting both edges as a time measurement trigger
(bits CTS1 and CTS0 in the GiTMCRj register are 11b)

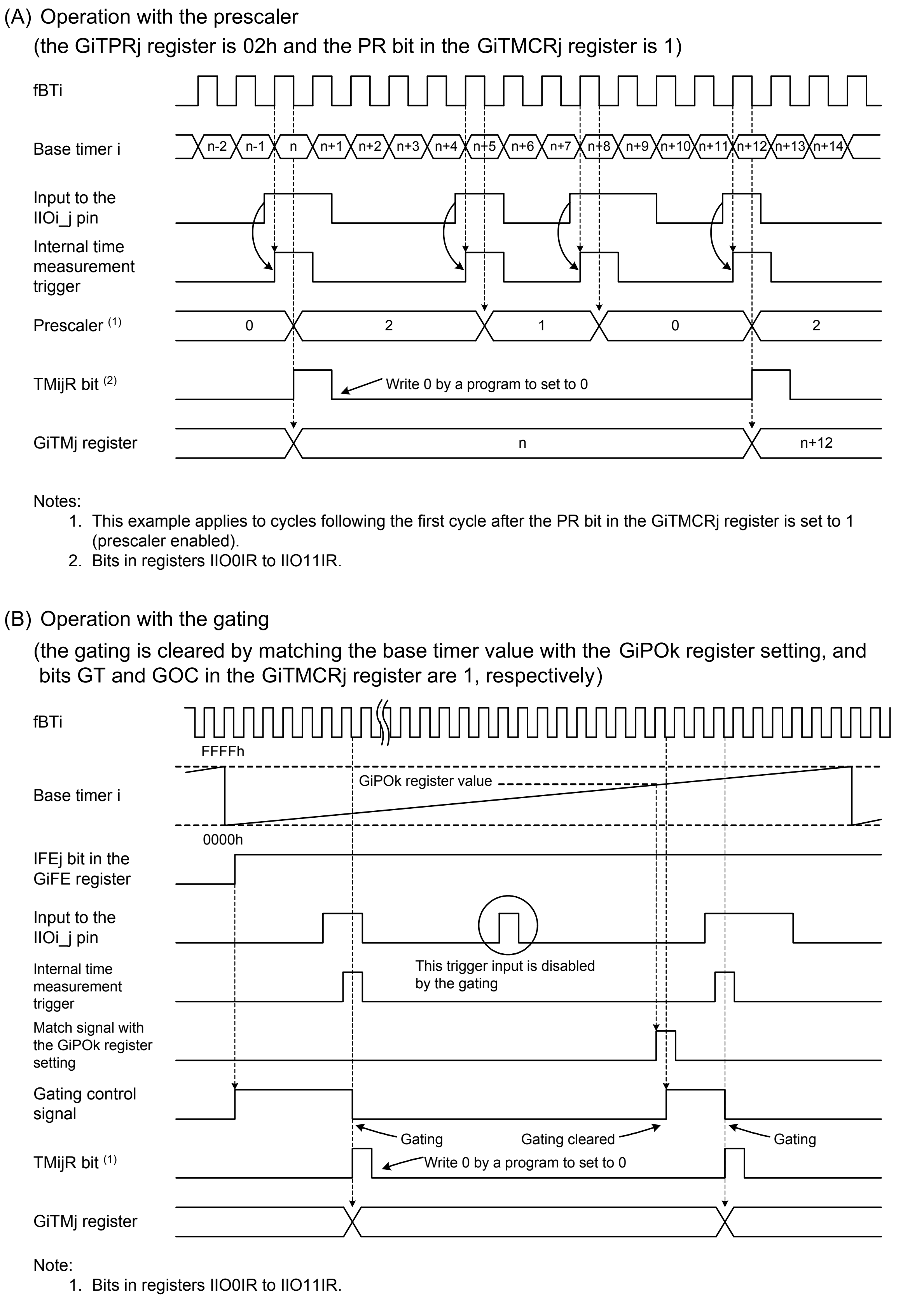
Notes:
1. Bits in registers IIO0IR to IIO11IR.
2. No interrupt occurs if the MCU receives a trigger signal when the TMijR bit is 1. However, the value of GiTMj register changes.

(C) Trigger signal when using the digital filter
(bits DF1 and DF0 in the GiTMCRj register are 10b or 11b)

f1 or fBTi (1)
Input to the IIOi_j pin
Trigger signal after passing the digital filter
Signals which do not hold the same level during three sequential cycles are rejected
Maximum 3.5 f1 or fBTi (1) clock cycles
The trigger signal is delayed by passing the digital filter

Note:
1. fBTi when bits DF1 and DF0 are 10b, f1 when the bits are 11b.

Figure 23.23 Time Measurement Operation (2/2) (i = 0, 1; j = 0 to 7)



(A) Operation with the prescaler

(the GiTPRj register is 02h and the PR bit in the GiTMCRj register is 1)

Notes:

1. This example applies to cycles following the first cycle after the PR bit in the GiTMCRj register is set to 1 (prescaler enabled).
2. Bits in registers IIO0IR to IIO11IR.

(B) Operation with the gating

(the gating is cleared by matching the base timer value with the GiPOk register setting, and bits GT and GOC in the GiTMCRj register are 1, respectively)

fBTi
FFFFh
Base timer i
GiPOk register value
0000h
IFEj bit in the GiFE register
Input to the IIOi_j pin
This trigger input is disabled by the gating
Internal time measurement trigger
Match signal with the GiPOk register setting
Gating control signal
Gating
Gating cleared
Gating
TMijR bit (1)
Write 0 by a program to set to 0
GiTMj register

Note:

1. Bits in registers IIO0IR to IIO11IR.

Figure 23.24 Prescaler and Gate Operations (i = 0, 1; j = 6, 7; k = 4, 5)

23.3 Waveform Generation for Groups 0 to 2

Waveforms are generated when the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0 to 7).

Waveform generation has the following six modes:

- Single-phase waveform output mode for groups 0 to 2
- Inverted waveform output mode for groups 0 to 2
- Set/reset waveform output (SR waveform output) mode for groups 0 to 2
- Bit modulation PWM output mode for group 2
- Real-time port output (RTP output) mode for group 2
- Parallel real-time port output (parallel RTP output) mode for group 2

Table 23.6 lists registers associated with the waveform generation.

Table 23.6 Waveform Generation Associated Register Settings (i = 0 to 2; j = 0 to 7)

Register	Bits	Function
GiPOCRj	MOD2 to MOD0	Select a waveform output mode
	PRT [1]	Set the bit to 1 to use parallel RTP output mode
	IVL	Select a default value
	RLD	Select a timing to reload the value into the GiPOj register
	RTP [1]	Set the bit to 1 to use RTP output mode or parallel RTP output mode. The settings of bits MOD2 to MOD0 are disabled when this bit is set to 1
	INV	Select if output level is inverted
G2BCR1	PRP	Set the bit to 1 to use parallel RTP output mode
GiPOj	—	Set the timing to invert output waveform level
GiFS	FSCj	Set the bit to 0 (select the waveform generation) for groups 0 and 1 only
GiFE	IFEj	Set the bit to 1 (enable the channel j function)
G2RTP	RTP0 to RTP7	Set the RTP output value in RTP output mode or parallel RTP output mode

Bit configurations and functions vary with channels and groups.

Registers associated with the waveform generation should be set after setting the base timer-associated registers.

Note:

1. This bit is available in the G2POCRj register only. Neither the G0POCRj nor G1POCRj register has it.

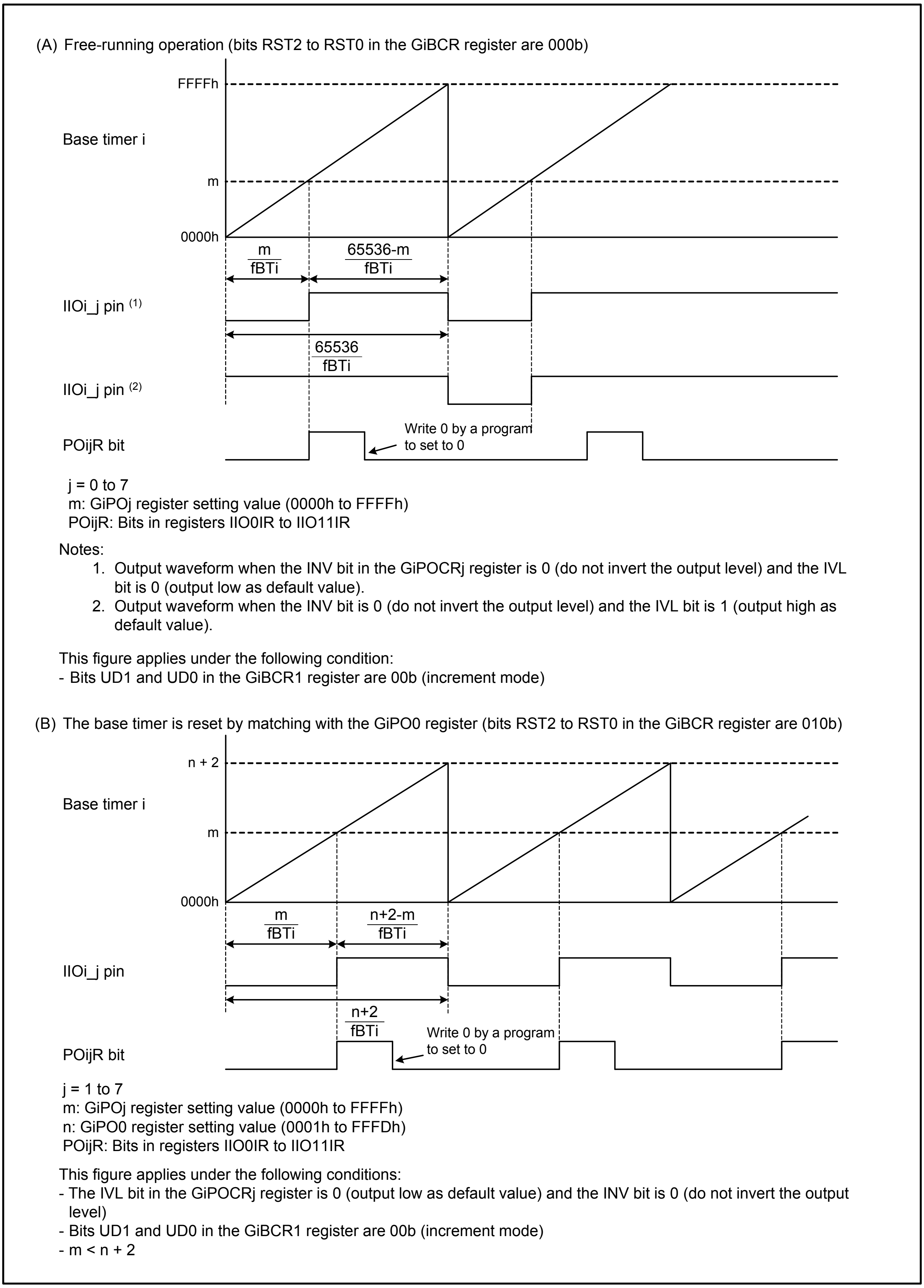
23.3.1 Single-phase Waveform Output Mode (for Groups 0 to 2)

The output level at the IIOi_j pin (or OUTC2_j pin for group 2) becomes high when the base timer value matches the GiPOj register (i = 0 to 2; j = 0 to 7). It switches to low when the base timer reaches 0000h. If the IVL bit in the GiPOCRj register is set to 1 (output high as default value), a high level output is provided when a waveform output starts. If the INV bit is set to 1 (invert the output level), a waveform with an inverted level is output. Refer to Figure 23.25 for details on single-phase waveform mode operation.
Table 23.7 lists specifications of single-phase waveform output mode.

Table 23.7 Single-phase Waveform Output Mode Specifications (i = 0 to 2)

Item	Specification
Output waveform [(1)]	• Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are 000b) Cycle: $\frac{65536}{fBTi}$ Low level width: $\frac{m}{fBTi}$ High level width: $\frac{65536 - m}{fBTi}$ m: GiPOj register setting value (j = 0 to 7), 0000h to FFFFh • The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are 010b) Cycle: $\frac{n + 2}{fBTi}$ Low level width: $\frac{m}{fBTi}$ High level width: $\frac{n + 2 - m}{fBTi}$ m: GiPOj register setting value (j = 1 to 7), 0000h to FFFFh n: GiPO0 register setting value, 0001h to FFFDh If $m \geq n + 2$, the output level is fixed to low
Waveform output start condition [(2)]	The IFEj bit in the GiFE register is 1 (enable the channel j function) (j = 0 to 7)
Waveform output stop condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register becomes 1 (interrupt requested) by matching the base timer value with the GiPOj register setting (refer to Figure 11.12)
IIOi_j output pin (or OUTC2_j pin for group 2) function	Pulse signal output
Other functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and outputs the inverted signal from the IIOi_j pin (or OUTC2_j pin for group 2)

Notes:
1. When the INV bit in the GiPOCRj register is 1 (invert the output level), the high and low widths are inverted.
2. To use channels shared by time measurement and waveform generation, set the FSCj bit in the GiFS register to 0 (select the waveform generation).



(A) Free-running operation (bits RST2 to RST0 in the GiBCR register are 000b)

j = 0 to 7
m: GiPOj register setting value (0000h to FFFFh)
POijR: Bits in registers IIO0IR to IIO11IR

Notes:

1. Output waveform when the INV bit in the GiPOCRj register is 0 (do not invert the output level) and the IVL bit is 0 (output low as default value).
2. Output waveform when the INV bit is 0 (do not invert the output level) and the IVL bit is 1 (output high as default value).

This figure applies under the following condition:
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)

(B) The base timer is reset by matching with the GiPO0 register (bits RST2 to RST0 in the GiBCR register are 010b)

j = 1 to 7
m: GiPOj register setting value (0000h to FFFFh)
n: GiPO0 register setting value (0001h to FFFDh)
POijR: Bits in registers IIO0IR to IIO11IR

This figure applies under the following conditions:
- The IVL bit in the GiPOCRj register is 0 (output low as default value) and the INV bit is 0 (do not invert the output level)
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)
- m < n + 2

Figure 23.25 Single-phase Waveform Output Mode Operation (i = 0 to 2)

23.3.2 Inverted Waveform Output Mode for Groups 0 to 2

The output level at the IIOi_j pin (or OUTC2_j pin for group 2) is inverted every time the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0 to 7).

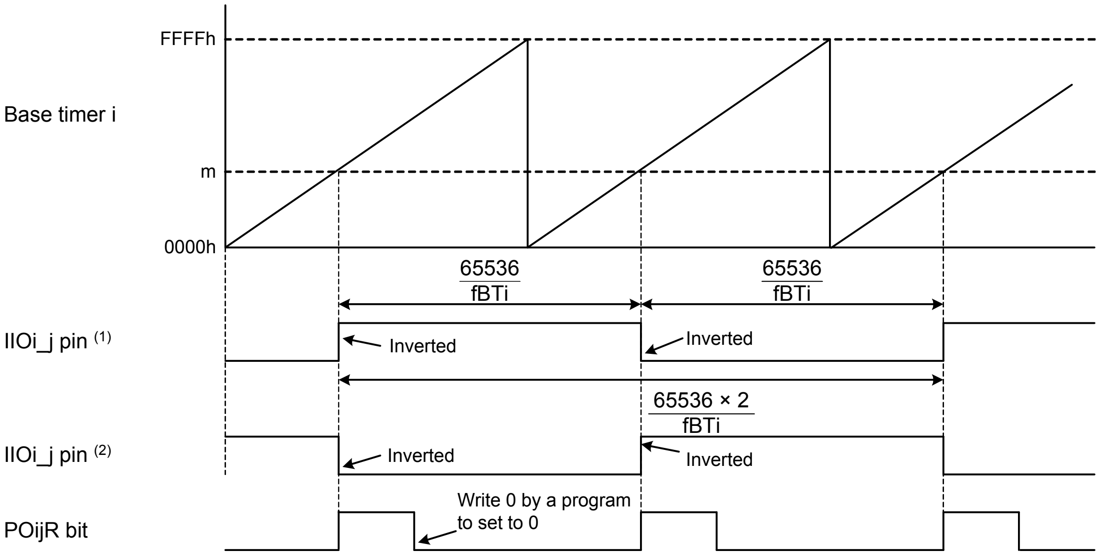
Table 23.8 lists specifications of the inverted waveform output mode. Figure 23.26 shows an example of the inverted waveform output mode operation.

Table 23.8 Inverted Waveform Output Mode Specifications (i = 0 to 2)

Item	Specification
Output waveform	• Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are 000b) Cycle: $\frac{65536 \times 2}{fBTi}$ High or low level width: $\frac{65536}{fBTi}$ m: GiPOj register setting value (j = 0 to 7), 0000h to FFFFh • The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are 010b) Cycle: $\frac{2(n+2)}{fBTi}$ High or low level width: $\frac{n+2}{fBTi}$ n: GiPO0 register setting value, 0001h to FFFDh GiPOj register setting value (j = 1 to 7), 0000h to FFFFh If the GiPOj register setting $\geq n+2$, the output level is not inverted
Waveform output start condition [1]	The IFEj bit in the GiFE register is 1 (enable the channel j function) (j = 0 to 7)
Waveform output stop condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register becomes 1 (interrupt requested) by matching the base timer value with the GiPOj register setting (refer to Figure 11.12)
IIOi_j output pin (or OUTC2_j pin for group 2) function	Pulse signal output
Other functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and outputs the inverted signal from the IIOi_j pin (or OUTC2_j pin for group 2)

Note:

1. To use channels shared by time measurement and waveform generation, set the FSCj bit in the GiFS register to 0 (select the waveform generation).



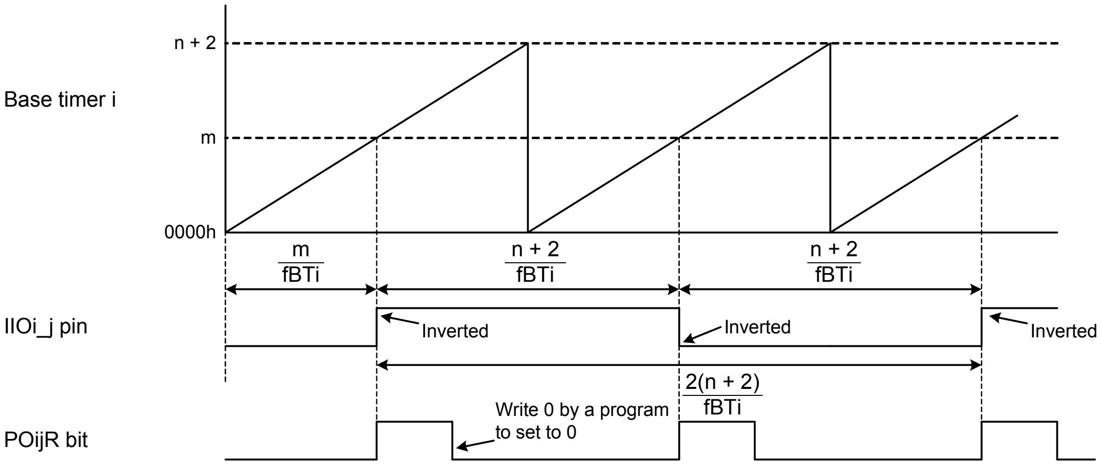
(A) Free-running operation (bits RST2 to RST0 in the GiBCR1 register are 000b)

j = 0 to 7
m: GiPOj register setting value (0000h to FFFFh)
POijR: Bits in registers IIO0IR to IIO11IR

Notes:

1. Output waveform when the INV bit in the GiPOCRj register is 0 (do not invert the output level) and the IVL bit is 0 (output low as default value).
2. Output waveform when the INV bit is 0 (do not invert the output level) and the IVL bit is 1 (output high as default value).

This figure applies under the following condition:
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)



(B) The base timer is reset by matching with the GiPO0 register (bits RST2 to RST0 are 010b)

j = 1 to 7
m: GiPOj register setting value (0000h to FFFFh)
n: GiPO0 register setting value (0001h to FFFDh)
POijR: Bits in registers IIO0IR to IIO11IR

This figure applies under the following conditions:
- The IVL bit in the GiPOCRj register is 0 (output low as default value) and the INV bit is 0 (do not invert the output level)
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)
- m < n + 2

Figure 23.26 Inverted Waveform Output Mode Operation (i = 0 to 2)

23.3.3 Set/Reset Waveform Output Mode (SR Waveform Output Mode) for Groups 0 to 2

The output level at the IIOi_j pin (or OUTC2_j pin for group 2) becomes high when the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0, 2, 4, 6). It becomes low when the base timer value matches the GiPOk register setting or the base timer reaches 0000h (k = j + 1). When the IVL bit in the GiPOCRj register is set to 1 (output high as default value), a high output level is provided when a waveform output starts (j = 0 to 7). When the INV bit is set to 1 (invert the output level), a waveform with inverted level is output. Refer to Figure 23.27 for details on SR waveform mode operation. Tables 23.9 and 23.10 list specifications of SR waveform output mode.

Table 23.9 SR Waveform Output Mode Specifications (i = 0 to 2) (1/2)

Item	Specification
Output waveform [1]	• Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are 000b) (A) $m < n$ High level width: $\frac{n - m}{fBTi}$ Low level width: $\frac{m}{fBTi}$ (See Note 2) + $\frac{65536 - n}{fBTi}$ (See Note 3) (B) $m \geq n$ High level width: $\frac{65536 - m}{fBTi}$ Low level width: $\frac{m}{fBTi}$ m: GiPOj register setting value (j = 0, 2, 4, 6), 0000h to FFFFh n: GiPOk register setting value (k = j + 1), 0000h to FFFFh • The base timer is reset by matching with the GiPO0 register (when bits RST2 to RST0 are 010b) [4] (A) $m < n < p + 2$ High level width: $\frac{n + m}{fBTi}$ Low width: $\frac{m}{fBTi}$ (See Note 2) + $\frac{p + 2 - n}{fBTi}$ (See Note 3) (B) $m < p + 2 \leq n$ High level width: $\frac{p + 2 - m}{fBTi}$ Low level width: $\frac{m}{fBTi}$ (C) $m \geq p + 2$, output level is fixed to low p: GiPO0 register setting value, 0001h to FFFDh m: GiPOj register setting value (j = 2, 4, 6), 0000h to FFFFh n: GiPOk register setting value (k = j + 1), 0000h to FFFFh

Notes:

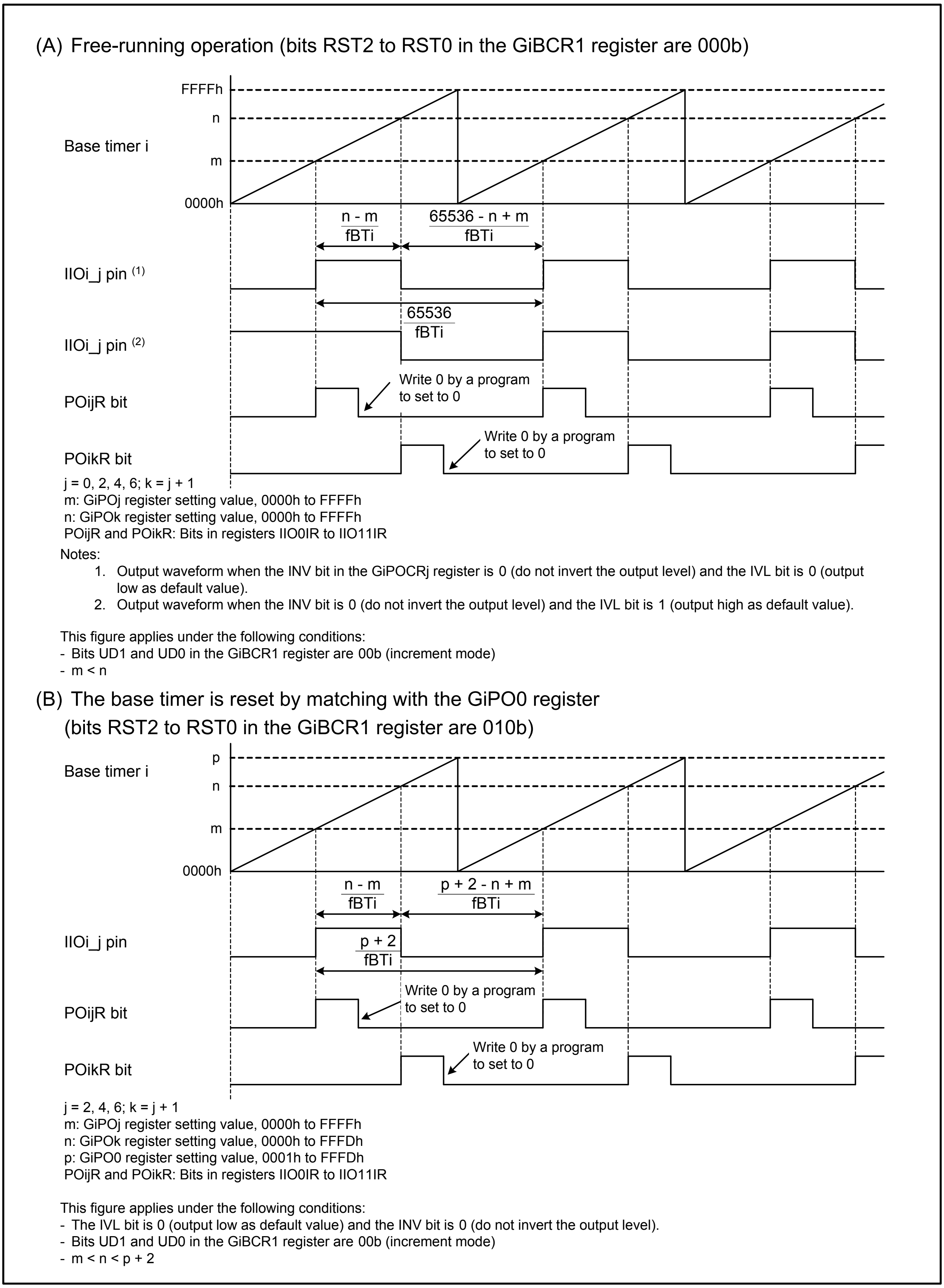
1. When the INV bit in the GiPOCRj register is 1 (invert the output level), the high and low widths are inverted.
2. Output period from a base timer reset until when the output level becomes high.
3. Output period from when the output level becomes low until the next base timer reset.
4. When the GiPO0 register resets the base timer, channel 0 and channel 1 SR waveform generation functions are not available.

Table 23.10 SR Waveform Output Mode Specifications (i = 0 to 2) (2/2)

Item	Specification
Waveform output start condition [1]	The IFEq bit in the GiFE register is 1 (enable the channel q function) (q = 0 to 7)
Waveform output stop condition	The IFEq bit is 0 (disable the channel q function)
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register becomes 1 (interrupt requested) by matching the base timer value with the GiPOj register setting. When the POikR bit becomes 1 (interrupt requested) by matching the base timer value with the GiPOk register setting (refer to Figure 11.12)
IIOi_j output pin (or OUTC2_j pin for group 2) function	Pulse signal output
Other functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and outputs the inverted signal from the IIOi_j pin (or OUTC2_j pin for group 2)

Note:

1. To use channels shared by time measurement and waveform generation, set the FSCj bit in the GiFS register to 0 (select the waveform generation).



(A) Free-running operation (bits RST2 to RST0 in the GiBCR1 register are 000b)

j = 0, 2, 4, 6; k = j + 1
m: GiPOj register setting value, 0000h to FFFFh
n: GiPOk register setting value, 0000h to FFFFh
POijR and POikR: Bits in registers IIO0IR to IIO11IR

Notes:

1. Output waveform when the INV bit in the GiPOCRj register is 0 (do not invert the output level) and the IVL bit is 0 (output low as default value).
2. Output waveform when the INV bit is 0 (do not invert the output level) and the IVL bit is 1 (output high as default value).

This figure applies under the following conditions:
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)
- m < n

(B) The base timer is reset by matching with the GiPO0 register (bits RST2 to RST0 in the GiBCR1 register are 010b)

j = 2, 4, 6; k = j + 1
m: GiPOj register setting value, 0000h to FFFFh
n: GiPOk register setting value, 0000h to FFFDh
p: GiPO0 register setting value, 0001h to FFFDh
POijR and POikR: Bits in registers IIO0IR to IIO11IR

This figure applies under the following conditions:
- The IVL bit is 0 (output low as default value) and the INV bit is 0 (do not invert the output level).
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)
- m < n < p + 2

Figure 23.27 SR Waveform Output Mode Operation (i = 0 to 2)

23.3.4 Bit Modulation PWM Output Mode (for Group 2)

In bit modulation PWM output mode, a PWM output has 16-bit resolution.

Pulses are repeatedly output in a period of 1024 consecutive periods of span t. The period of span t is $\frac{64}{fBT2}$. The 6 upper bits in the G2POj register determine the base low width (j = 0 to 7). The 10 lower bits determine the number of span t, within a period, in which the low width is extended by the minimum resolution bit width, that is, one clock cycle.

When the INV bit is set to 1 (invert the output level), the waveform with an inverted level is output.

Table 23.11 lists specifications of bit modulation PWM output mode. Table 23.12 lists the number of modulated spans and span ts to be extended with the minimum resolution bit width. Figure 23.28 shows an example of bit modulation PWM output mode operation.

Table 23.11 Bit Modulation PWM Output Mode Specifications (j = 0 to 7)

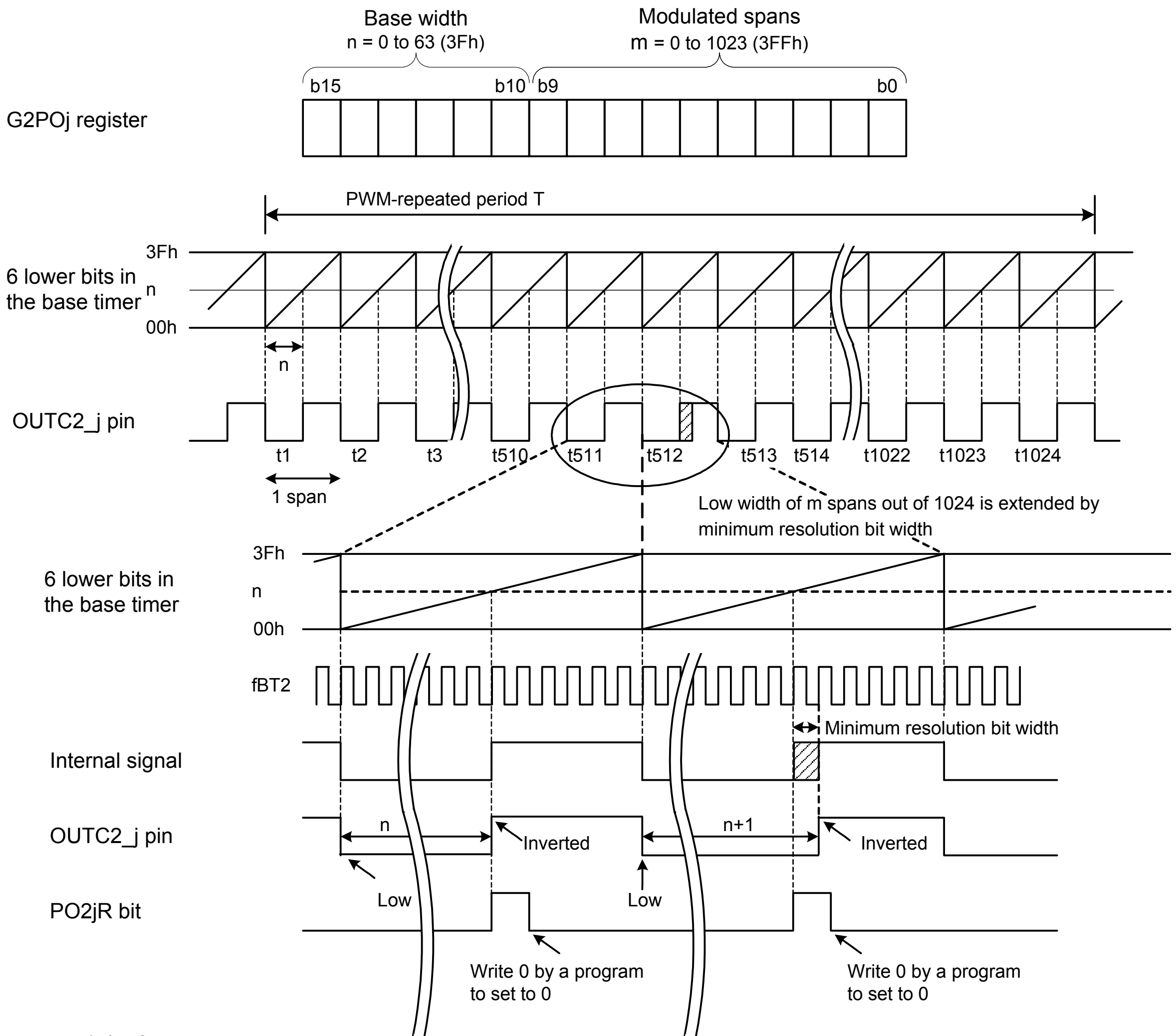
Item	Specification
Output waveform (1,2)	PWM-repeated period T: $\frac{65536}{fBT2}\left(=\frac{64}{fBT2}\times 1024\right)$ Period of span t: $\frac{64}{fBT2}$ Low width: $\frac{n+1}{fBT2}$ of m spans $\frac{n}{fBT2}$ of $(1024-m)$ spans Mean low width: $\frac{1}{fBT2}\times\left(n+\frac{m}{1024}\right)$ n: G2POj register setting value (6 upper bits), 00h to 3Fh m: G2POj register setting value (10 lower bits), 000h to 3FFh
Waveform output start condition	The IFEj bit in the G2FE register is 1 (enable the channel j function)
Waveform output stop condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	When the PO2jR bit in the interrupt request register becomes 1 (interrupt requested) by matching the 6 lower bits of the base timer value with the 6 upper bits of the G2POj register setting (refer to Figure 11.12)
OUTC2_j pin function	Pulse signal output pin
Other functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and outputs the inverted signal from the OUTC2_j pin

Notes:

1. Bits RST2 and RST0 in the G2BCR1 register should be set to 000b to use bit modulation PWM output mode.
2. When the INV bit in the G2POCRj register is set to 1 (invert the output level), the high and low widths are inverted.

Table 23.12 Number of Modulated Spans and Span t Extended Minimum Resolution Bit Width

Modulated Spans	Span ts to be Extended with Minimum Resolution Bit Width
00 0000 0000b	none
00 0000 0001b	t512
00 0000 0010b	t256 and t768
00 0000 0100b	t128, t384, t640, and t896
00 0000 1000b	t64, t192, t320, t448, t576, t704, t832, and t960
:	:
10 0000 0000b	t1, t3, t5, t7, ••• t1019, t1021, and t1023



m = 1, j = 0 to 7
PO2jR: Bits in registers IIO3IR to IIO11IR

This figure above applies under the following condition:
- The IVL bit in the G2POCRj register is 0 (output low as default value) and the INV bit is 0 (do not invert the output level)

Figure 23.28 Bit Modulation PWM Output Mode Operation

23.3.5 Real-time Port Output Mode (RTP Output Mode) (for Group 2)

The OUTC2_j pin outputs the G2RTP register setting value in 1-bit units when the base timer value matches the G2POj register setting (j = 0 to 7). Table 23.13 lists specifications of RTP output mode. Figure 23.29 shows a block diagram of RTP output and Figure 23.30 shows an example of RTP output mode operation.

Table 23.13 RTP Output Mode Specifications (j = 0 to 7)

Item	Specification
Waveform output start condition	The IFEj bit in the G2FE register is 1 (enable the channel j function)
Waveform output stop condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	When the PO2jR bit in the interrupt request register becomes 1 (interrupt requested) by matching the base timer value with the G2POj register setting (0000h to FFFFh [1]) (refer to Figure 11.12)
OUTC2_j pin function	RTP output pin
Other functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and outputs the inverted signal from the OUTC2_j pin

Note:

1. The G2PO0 register should be set to between 0001h and FFFDh to set the base timer value to 0000h (bits RST2 to RST0 are set to 010b) when the base timer value matches the G2PO0 register setting.

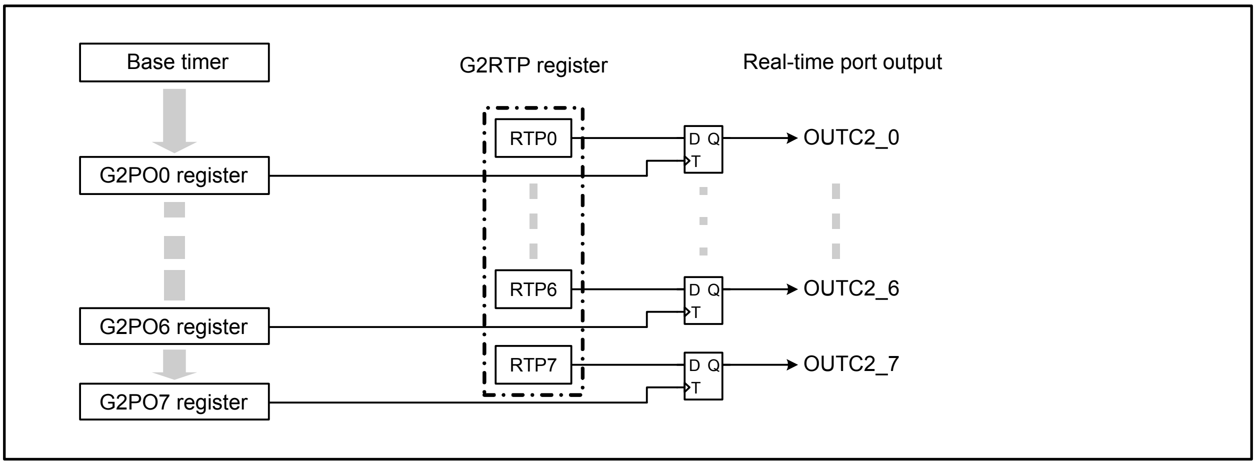
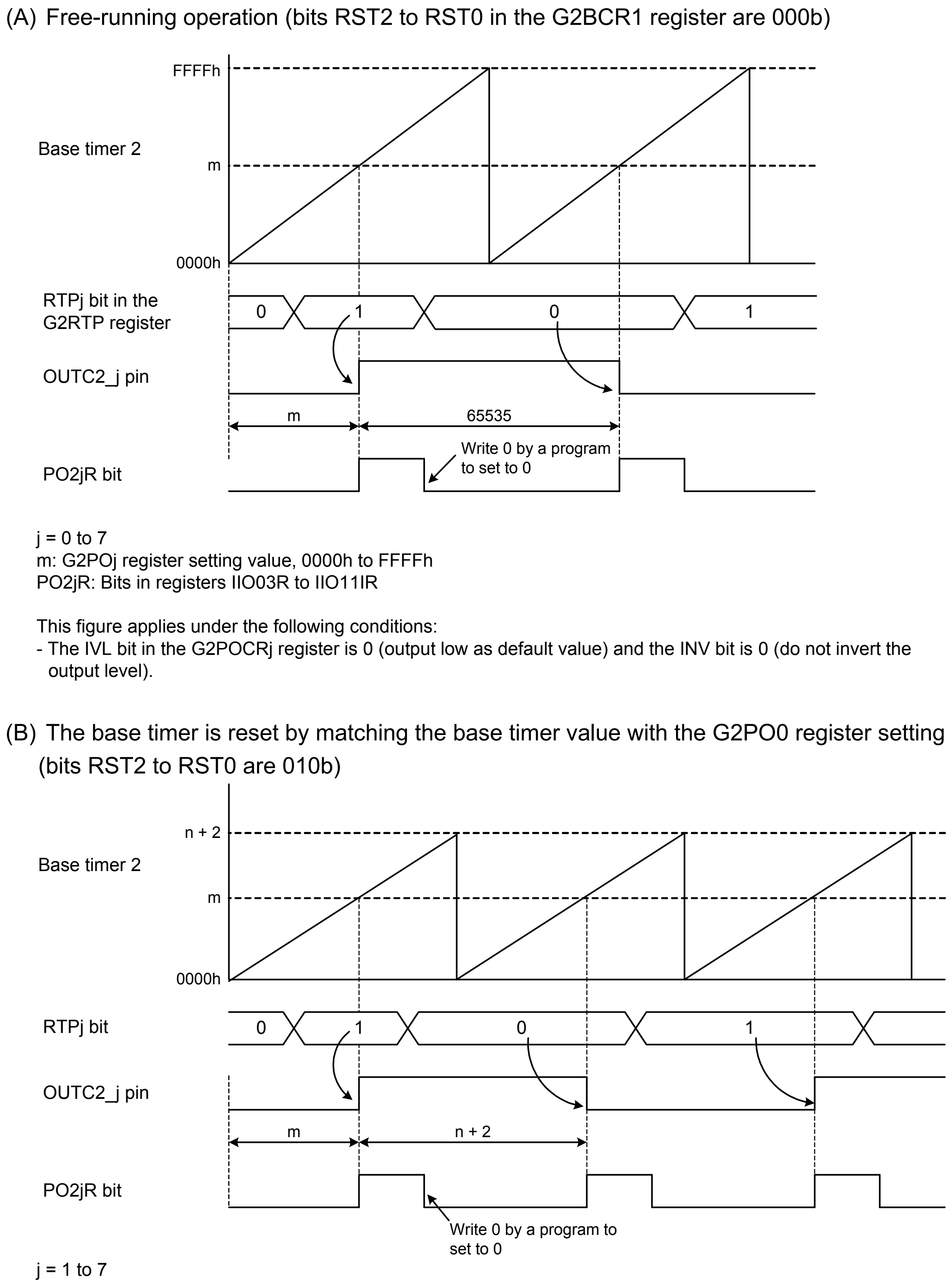


Figure 23.29 RTP Output Block Diagram



(A) Free-running operation (bits RST2 to RST0 in the G2BCR1 register are 000b)

j = 0 to 7
m: G2POj register setting value, 0000h to FFFFh
PO2jR: Bits in registers IIO03R to IIO11IR

This figure applies under the following conditions:
- The IVL bit in the G2POCRj register is 0 (output low as default value) and the INV bit is 0 (do not invert the output level).

(B) The base timer is reset by matching the base timer value with the G2PO0 register setting (bits RST2 to RST0 are 010b)

j = 1 to 7
m: G2POj register setting value, 0000h to FFFFh
n: G2POj register setting value, 0001h to FFFDh
PO2jR: Bits in registers IIO03R to IIO11IR

This figure applies under the following conditions:
- The IVL bit in the G2POCRj register is 0 (output low as default value) and the INV bit is 0 (do not invert the output level).
- m < n + 2

Figure 23.30 RTP Output Mode Operation

23.3.6 Parallel Real-time Port Output Mode (RTP Output Mode) (for Group 2)

The OUTC2_j pin outputs all the G2RTP register setting values in 1-byte units when the base timer value matches the G2POj register setting (j = 0 to 7). Table 23.14 lists specifications of parallel RTP output mode. Figure 23.7 shows the G2BCR1 register. Figure 23.31 shows a block diagram of parallel RTP output and Figure 23.32 shows an example of parallel RTP output mode operation.

Table 23.14 Parallel RTP Output Mode Specifications (j = 0 to 7)

Item	Specification
Waveform output start condition	The IFEj bit in the G2FE register is 1 (enable the channel j function)
Waveform output stop Condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	The PO2jR bit in the interrupt request register becomes 1 (interrupt requested) when the base timer value matches the G2POj register setting (0000h to FFFFh [1]) (refer to Figure 11.12)
OUTC2_j pin function	RTP output pin
Other functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and outputs the inverted signal from the OUTC2_j pin

Note:

1. The G2PO0 register should be set to between 0001h and FFFDh to set the base timer value to 0000h (bits RST2 to RST0 are set to 010b) when the base timer value matches the G2PO0 register setting.

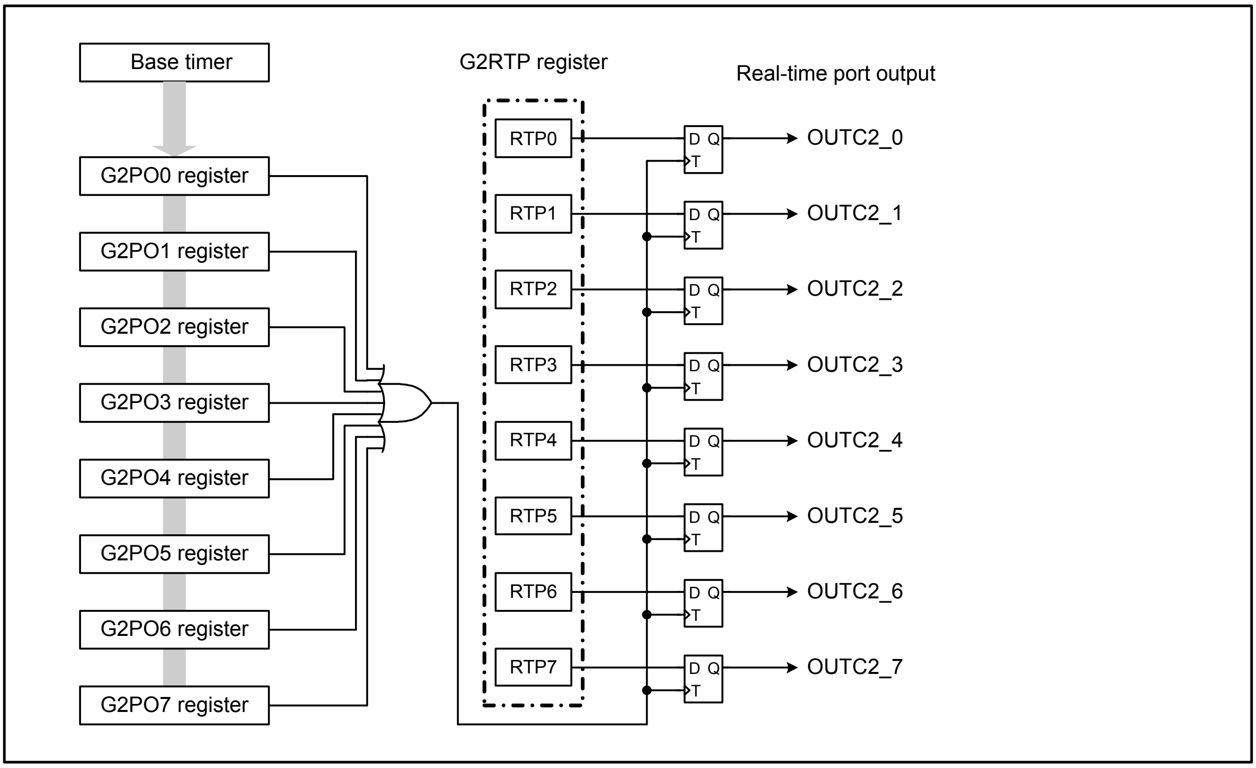
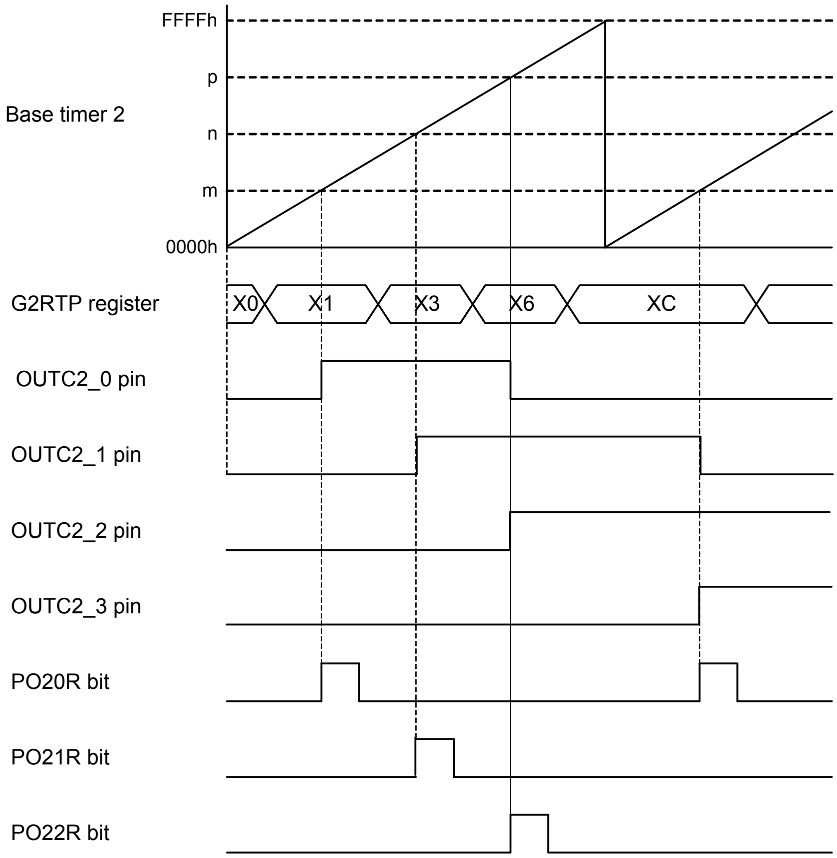


Figure 23.31 Parallel RTP Output Mode Block Diagram

(A) Free-running operation



m: G2PO0 register setting value, 0000h to FFFFh
n: G2PO1 register setting value, 0000h to FFFFh
p: G2PO2 register setting value, 0000h to FFFFh
PO20R, PO21R, and PO22R: Bits in registers IIO3IR to IIO11IR

This figure applies under the following conditions:

- The IVL bit in the G2POCRj register is 0 (output low as default value) and the IVL bit is 0 (do not invert the output level)
- Bits RST2 to RST0 in the G2BCR1 register are 000b (base timer is not reset)
- *m* < *n* < *p*

Figure 23.32 Parallel RTP Output Mode Operation

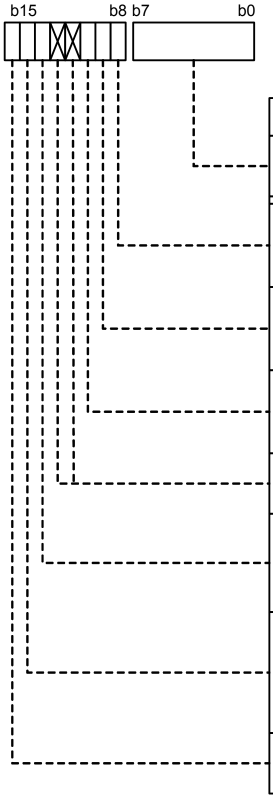
23.4 Group 2 Serial Interface

Two 8-bit shift registers and waveform generation enable the serial interface function. In group 2 of the intelligent I/O, the variable synchronous serial interface and IEBus (optional [1]) are available.
Figures 23.33 to 23.40 show associated registers.

Note:

1. Contact a Renesas Electronics sales office to use the optional features.

Group 2 SI/O Transmit Buffer Register



Symbol	Address	Reset Value
G2TB	016Dh-016Ch	Undefined

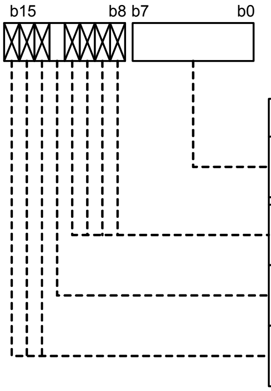
Bit Symbol	Bit Name	Function	RW
— (b7-b0)	Transmit Buffer	Data transmitted	WO
SZ0	Transmit/Receive Character Length Select Bit	b10 b9 b8 0 0 0 : 8 bits 0 0 1 : 1 bit 0 1 0 : 2 bits 0 1 1 : 3 bits 1 0 0 : 4 bits 1 0 1 : 5 bits 1 1 0 : 6 bits 1 1 1 : 7 bits	RW
SZ1			RW
SZ2			RW
— (b12-b11)	No register bits; should be written with 0 and read as undefined value		—
A	ACK Function Select Bit	0: Do not add the ACK bit 1: Add the ACK bit after last transmit bit	RW
PC	Parity Calculation Continuing Bit	0: Add the parity bit after this transmit data 1: Carry over a parity to the data to be transmitted [1]	RW
P	Parity Select Bit	0: No parity 1: Parity (even parity only)	RW

Note:

1. Set the PC bit to 1 after setting the P bit to 0.

Figure 23.33 G2TB Register

Group 2 SI/O Receive Buffer Register



Symbol	Address	Reset Value
G2RB	016Fh-016Eh	Undefined

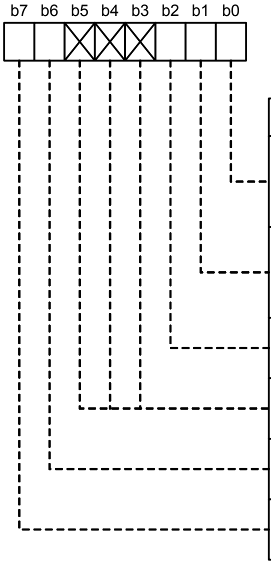
Bit Symbol	Bit Name	Function	RW
— (b7-b0)	Receive Buffer	Data received	RO
— (b11-b8)	No register bits; should be written with 0 and read as undefined value		—
OER	Overrun Error Flag [1]	0: No overrun error 1: Overrun error	RO
— (b15-b13)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. The OER bit becomes 0 when bits GMD1 and GMD0 in the G2MR register are set to 00b (communication block is reset) or the RE bit in the G2CR register is set to 0 (reception disabled).

Figure 23.34 G2RB Register

Group 2 Serial Interface Mode Register



Symbol	Address	Reset Value
G2MR	016Ah	00XX X000b

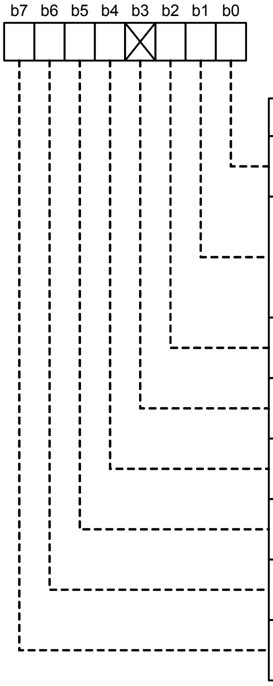
Bit Symbol	Bit Name	Function	RW
GMD0	Serial Interface Mode Select Bit	b1 b0 0 0 : Communication block is reset (OER bit is set to 0) [1] 0 1 : Variable synchronous serial interface mode [2] 1 0 : IEBus mode [2] 1 1 : Do not use this combination	RW
GMD1			RW
CKDIR	Internal/External Clock Select Bit	0: Internal clock 1: External clock	RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
UFORM	Bit Order Select Bit	0: LSB first 1: MSB first	RW
IRS	Transmit Interrupt Source Select Bit	0: Transmit buffer is empty 1: Transmission is completed	RW

Notes:

1. At least one base timer clock is required after setting bits GMD1 and GMD0 to 00b.
2. Set bits GMD1 and GMD0 to 01b or 10b while the base timer clock is stopped.

Figure 23.35 G2MR Register

Group 2 Serial Interface Control Register



Symbol
G2CR

Address
016Bh

Reset Value
0000 X110b

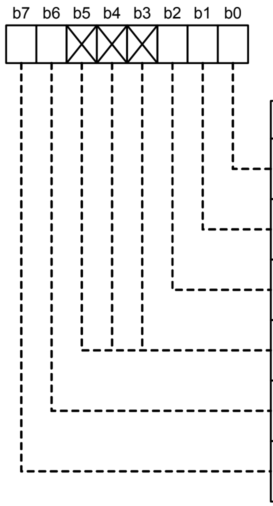
Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0: Transmission disabled 1: Transmission enabled	RW
TXEPT	Transmit Shift Register Empty Flag	0: Data in the transmit shift register (during transmission) 1: No data in the transmit shift register (transmission completed)	RO
TI	Transmit Buffer Empty Flag	0: Data held in the G2TB register 1: No data held in the G2TB register	RO
— (b3)	No register bit; should be written with 0 and read as undefined value		—
RE	Receive Enable Bit [1]	0: Reception disabled 1: Reception enabled	RW
RI	Receive Complete Flag	0: No data held in the G2RB register 1: Data held in the G2RB register	RO
OPOL	ISTXD2 Output Polarity Switching Bit	0: Not inverted 1: Inverted	RW
IPOL	ISRXD2 Input Polarity Switching Bit [1]	0: Not inverted 1: Inverted	RW

Note:

1. The group 2 base timer may be reset when these bits are rewritten. To avoid unexpected resets, set the RST2 bit in the G2BCR1 register to 0 (base timer is not reset by a reset request from the serial interface).

Figure 23.36 G2CR Register

Group 2 IEBus Control Register



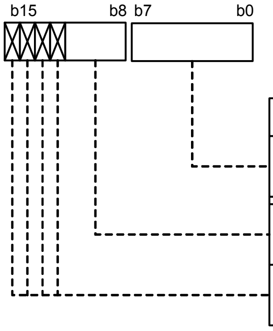
Symbol	Address	Reset Value
IECR	0172h	00XX X000b

Bit Symbol	Bit Name	Function	RW
IEB	IEBus Enable Bit [1]	0: IEBus disabled [2] 1: IEBus enabled	RW
IETS	IEBus Transmit Start Request Bit	0: Transmission completed 1: Transmission started	RW
IEBBS	IEBus Bus Busy Flag	0: Idle state 1: Busy state (start condition detected)	RO
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
DF	Digital Filter Select Bit	0: No digital filter 1: Use the digital filter	RW
IEM	IEBus Mode Select Bit	0: Mode 1 1: Mode 2	RW

Notes:
1. Rewrite the IEB bit while the base timer clock is stopped.
2. Wait at least one fBT2 cycle after setting the IEB bit to 0. To set this bit to 1, set bits BCK1 and BCK0 in the G2BCR0 register to 00b (clock is stopped).

Figure 23.37 IECR Register

Group 2 IEBus Address Register

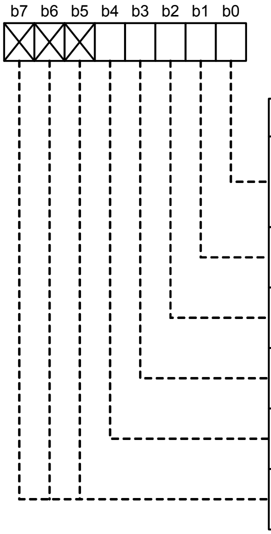


Symbol	Address	Reset Value
IEAR	0171h-0170h	Undefined

Function	RW
Address data	RW
Address data	RW
No register bits; should be written with 0 and read as undefined value	—

Figure 23.38 IEAR Register

Group 2 IEBus Transmit Interrupt Source Detect Register



Symbol	Address	Reset Value
IETIF	0173h	XXX0 0000b

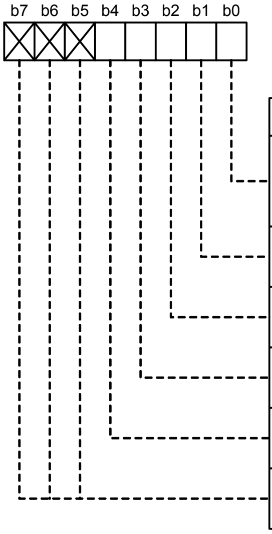
Bit Symbol	Bit Name	Function	RW
IETNF	Normal Completion Flag	0: Transmission completed in error 1: Transmission completed successfully [1]	RW
IEACK	ACK Error Flag	0: No error detected 1: Error detected [1]	RW
IETMB	Maximum Transmit Byte Error Flag	0: No error detected 1: Error detected [1]	RW
IETT	Timing Error Flag	0: No error detected 1: Error detected [1]	RW
IEABL	Arbitration Lost Flag	0: No error detected 1: Error detected [1]	RW
— (b7-b5)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. This bit can be set to 0 by a program, but cannot be set to 1. To set this bit to 0, set the IEB bit in the IECR register to 0 (IEBus disabled).

Figure 23.39 IETIF Register

Group 2 IEBus Receive Interrupt Source Detect Register



Symbol	Address	Reset Value
IERIF	0174h	XXX0 0000b

Bit Symbol	Bit Name	Function	RW
IERNF	Normal Completion Flag	0: Reception completed in error 1: Reception completed successfully [1]	RW
IEPAR	Parity Error Flag	0: No error detected 1: Error detected [1]	RW
IERMB	Maximum Receive byte Error Flag	0: No error detected 1: Error detected [1]	RW
IERT	Timing Error Flag	0: No error detected 1: Error detected [1]	RW
IERETC	Error by Other Sources Flag	0: No error detected 1: Error detected [1]	RW
— (b7-b5)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. This bit can be set to 0 by a program, but cannot be set to 1. To set this bit to 0, set the IEB bit in the IECR register to 0 (IEBus disabled).

Figure 23.40 IERIF Register

23.4.1 Variable Synchronous Serial Interface Mode (for Group 2)

This mode allows data transmission/reception synchronized with the transmit/receive clock. The character length is selectable from 1 to 8 bits. Table 23.15 lists specifications of the group 2 variable synchronous serial interface mode and Table 23.16 lists its settings. Figure 23.41 shows an operation example of data transmission/reception.

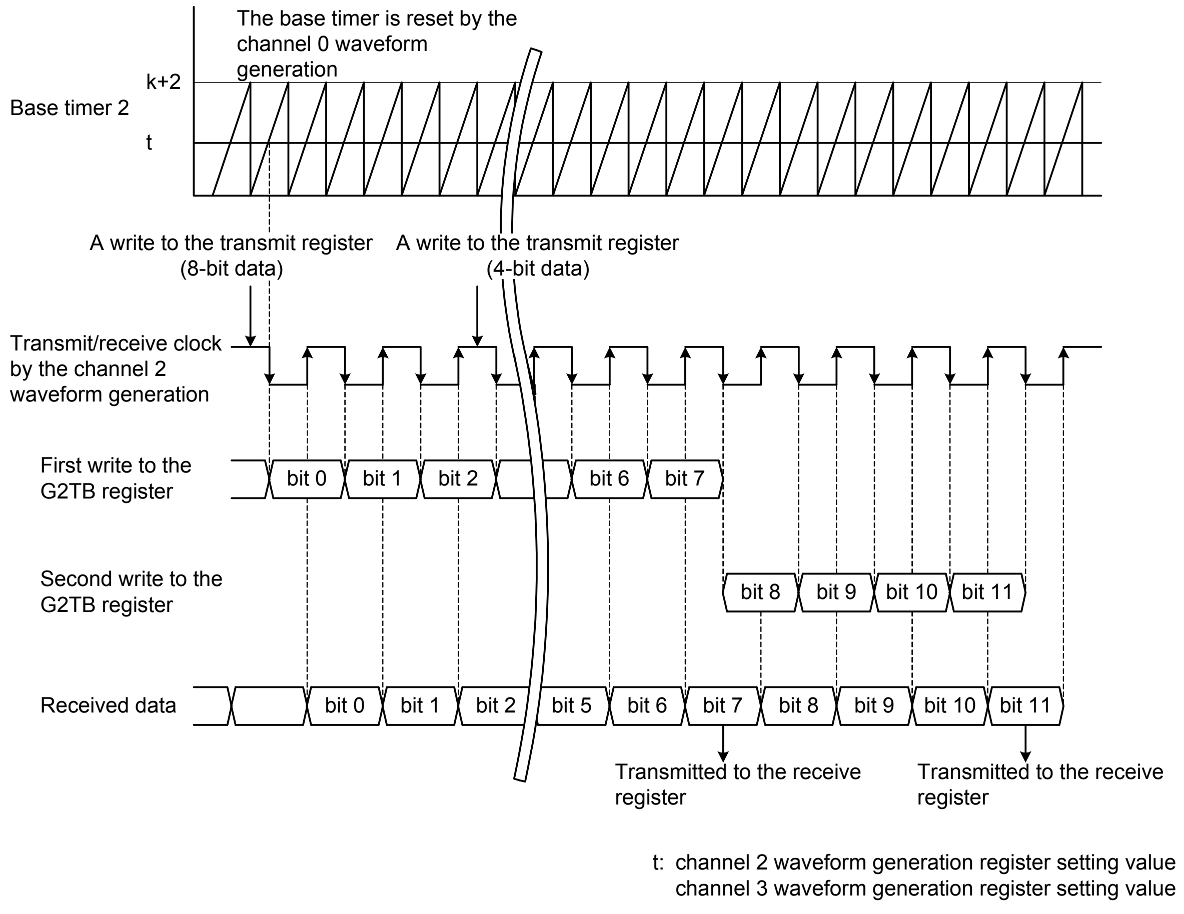
Table 23.15 Group 2 Variable Synchronous Serial Interface Mode Specifications

Item	Specification
Data format	1- to 8-bit character length
Transmit/receive clock	• The CKDIR bit in the G2MR register is 0 (internal clock selected): $\frac{fBT2}{2(n+2)}$ n: G2PO0 register setting value, 0000h to FFFFh (1) The bit rate is set using the G2PO0 register. The clock is generated in the inverted waveform output mode of the channel 2 waveform generation • The CKDIR bit is 1 (external clock selected): input into the ISCLK2 pin (2)
Transmit start conditions	The conditions for starting data transmission are as follows: • The TE bit in the G2CR register is 1 (transmission enabled) • The TI bit in the G2CR register is 0 (data held in the G2TB register)
Receive start conditions	The conditions for starting data reception are as follows: • The RE bit in the G2CR register is 1 (reception enabled) • The TE bit in the G2CR register is 1 (transmission enabled) • The TI bit in the G2CR register is 0 (data held in the G2TB register)
Interrupt request	In transmit interrupt, either of the following conditions is selected to set the SIO2TR bit in the IIO6IR register to 1 (interrupt requested) (refer to Figure 11.12): • The IRS bit in the G2MR register is 0 (transmit buffer in the G2TB register is empty): when data is transferred from the G2TB register to the transmit shift register (when the transmission has started) • The IRS bit is 1 (transmission is completed): when data transmission from the transmit shift register is completed In receive interrupt, When data is transferred from the receive shift register to the G2RB register (when the reception is completed), the SIO2PR bit in the IIO5IR register is set to 1 (interrupt requested) (refer to Figure 11.12)
Error detection	Overrun error (3) This error occurs when the last bit of the next data has been received before reading the G2RB register
Other functions	• Bit order selection LSB first or MSB first • ISTXD2 and ISRXD2 I/O polarity Output levels from the ISTXD2 pin and input levels to the ISRXD2 pin can be inverted • Character length for data transmission/reception 1- to 8-bit character length

Notes:
1. When using the serial interface, set a value greater than or equal to 1 to the G2PO0 register.
2. The highest transmit/receive clock frequency should be fBT2 divided by 20.
3. If an overrun error occurs, the G2RB register is undefined.

Table 23.16 Register Settings in Group 2 Variable Synchronous Serial Interface Mode

Register	Bits	Function
G2BCR0	BCK1 and BCK0	Set the bits to 11b
	DIV4 to DIV0	Select a divide ratio of count source
	IT	Set the bit to 0
G2BCR1	7 to 0	Set the bits to 0001 0010b
G2POCR0	7 to 0	Set the bits to 0000 0111b
G2POCR1	7 to 0	Set the bits to 0000 0111b
G2POCR2	7 to 0	Set the bits to 0000 0010b
G2PO0	15 to 0	Set a comparative value for waveform generation $\frac{fBT2}{2 \times (\text{setting value} + 2)}$ = transmit/receive clock frequency
G2PO2	15 to 0	Set to a value smaller than that in the G2PO0 register setting
G2FE	IFE2 to IFE0	Set the bits to 111b
G2MR	GMD1 and GMD0	Set the bits to 01b
	CKDIR	Select either the internal clock or the external clock
	UFORM	Select either LSB first or MSB first
	IRS	Select a source for transmit interrupt
G2CR	TE	Set the bit to 1 to enable data transmission/reception
	TXEPT	Transmit shift register empty flag
	TI	Transmit buffer empty flag
	RE	Set the bit to 1 to enable data reception
	RI	Receive complete flag
	OPOL	Select if the output level at the ISTXD2 pin is inverted (usually set the bit to 0)
	IPOL	Select if the input level at the ISRXD2 pin is inverted (usually set the bit to 0)
G2TB	15 to 0	Set the data to be transmitted/received and its character length
G2RB	15 to 0	Store received data and error flag



The base timer is reset by the channel 0 waveform generation
k+2
Base timer 2
t
A write to the transmit register (8-bit data)
A write to the transmit register (4-bit data)
Transmit/receive clock by the channel 2 waveform generation
First write to the G2TB register
bit 0
bit 1
bit 2
bit 6
bit 7
Second write to the G2TB register
bit 8
bit 9
bit 10
bit 11
Received data
bit 0
bit 1
bit 2
bit 5
bit 6
bit 7
bit 8
bit 9
bit 10
bit 11
Transmitted to the receive register
Transmitted to the receive register
t: channel 2 waveform generation register setting value
channel 3 waveform generation register setting value

Figure 23.41 Group 2 Variable Synchronous Serial Interface Mode Transmit/Receive Operation

24. Multi-master I2C-bus Interface

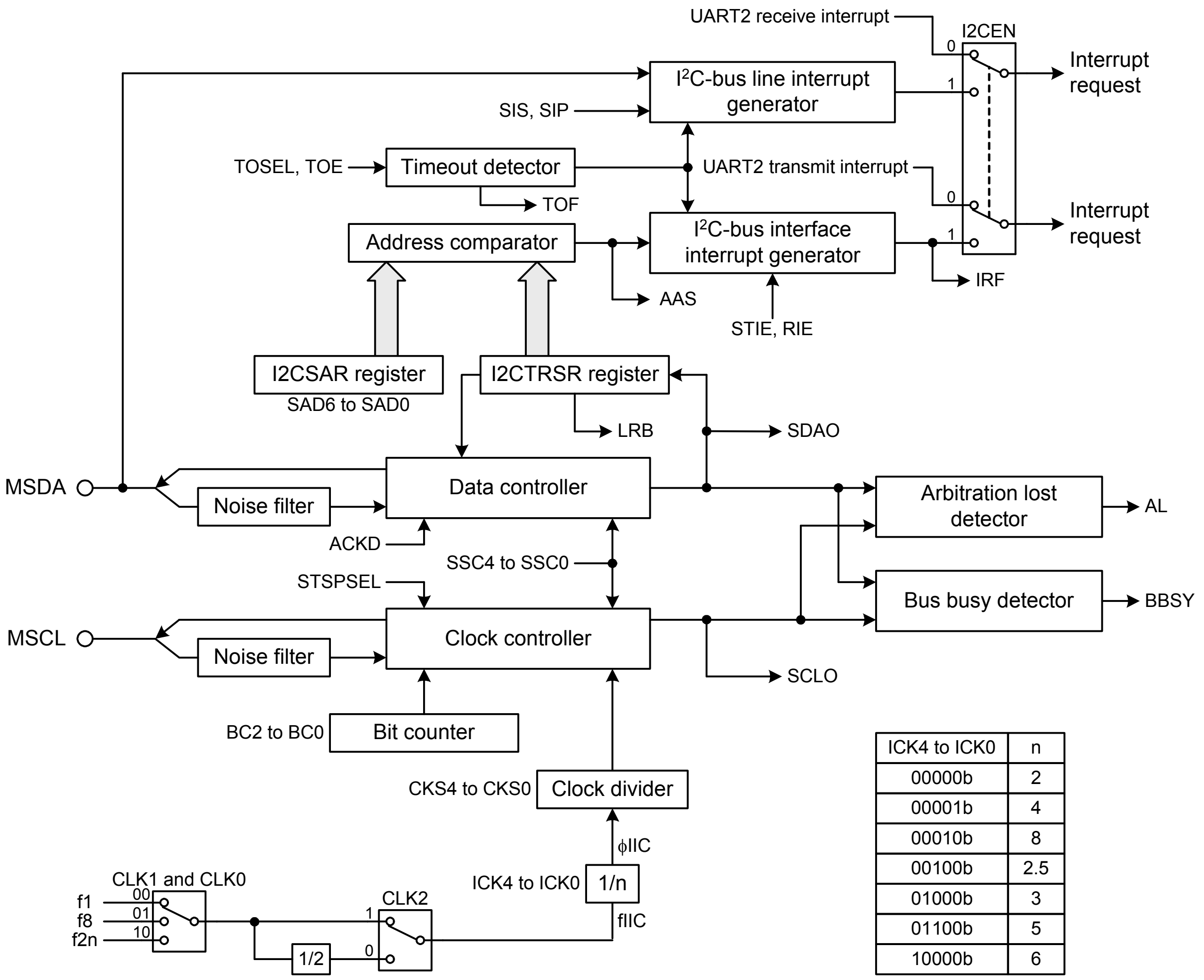
The multi-master I2C-bus interface (MMI2C) is capable of serial, bi-directional data transfer in the I2C-bus data transmit and receive format. It contains an arbitration lost detector and a clock synchronization function. Table 24.1 lists specifications of the multi-master I2C-bus interface. Table 24.2 lists detectors of the multi-master I2C-bus interface. Figure 24.1 shows a block diagram of the multi-master I2C-bus interface.

Table 24.1 Multi-master I2C-bus Interface Specifications

Item	Specification
Data format	Compliant with the I2C-bus specification • 7-bit addressing format • Fast-mode • Standard-mode
Master/Slave device	Selectable
I/O pins	Serial data line: MSDA (SDA) Serial clock line: MSCL (SCL)
Transmit/Receive clock	16.1 to 400 kbps (ϕIIC = 4 MHz) ϕIIC: I2C-bus system clock
Transmit/Receive modes	Compliant with the I2C-bus specification • Master-transmit mode • Master-receive mode • Slave-transmit mode • Slave-receive mode
Interrupt request sources	• Six I2C-bus interface interrupts: Successful transmit, successful receive, slave address match detection, general call address detection, STOP condition detection, and timeout detection • Two I2C-bus line interrupts: Rising or falling edge of pins MSDA and MSCL
Other functions	• Timeout detector This function detects that the MSCL pin level is held high for longer than the specified time while the bus is busy • Free data format selector This function selects the free data format to generate an interrupt request, regardless of the slave address value, when the first byte is received

Table 24.2 Detectors of Multi-master I2C-bus Interface

Item	Specification
Slave address match detector	In slave-receive mode, this detects whether the address sent from the master device matches the slave address. When they match, an ACK is automatically sent. When they do not, a NACK is automatically sent and communication is stopped
General call address detector	This detects a general call address when in slave-receive mode
Arbitration lost detector	This detects an arbitration lost and stops MSDA output immediately when detected
Bus busy detector	This detects that the bus is busy, and sets/resets the BBSY bit



ICK4 to ICK0	n
00000b	2
00001b	4
00010b	8
00100b	2.5
01000b	3
01100b	5
10000b	6

Only set the values listed above.

SAD6 to SAD0: Bits in the I2CSAR register
CKS4 to CKS0, and ACKD: Bits in the I2CCCR register
BC2 to BC0: Bits in the I2CCR0 register
STIE, RIE, SDAO, SCLO, and ICK0 and ICK1: Bits in the I2CCR1 register
TOE, TOF, TOSEL, and ICK4 to ICK2: Bits in the I2CCR2 register
LRB, AAS, AL, IRF, and BBSY: Bits in the I2CSR register
SSC4 to SSC0, SIP, SIS, and STSPSEL: Bits in the I2CSSCR register
I2CEN, and CLK2 to CLK0: Bits in the I2CMR register

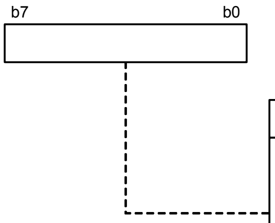
fIIC: I2C-bus interface clock
ϕIIC: I2C-bus system clock

Figure 24.1 Multi-master I2C-bus Interface Block Diagram

24.1 Multi-master I^{2}C-bus Interface-associated Registers

24.1.1 I^{2}C-bus Transmit/Receive Shift Register (I2CTRSR)

I^{2}C-bus Transmit/Receive Shift Register (1, 2, 3)



Symbol	Address	Reset Value
I2CTRSR	044400h	Undefined

Function	RW
Set transmit data in transmit mode. In receive mode, write dummy data before receiving data, and read received data after an interrupt is generated. The data being shifted can be read from this register, regardless of whether it is transmitting or receiving	RW

Notes:

1. This register is write enabled when the ICE bit in the I2CCR0 register is 1 (I^{2}C-bus interface enabled).
2. This register is used for transmitting and receiving. Read the received data before using this register for transmitting.
3. When data is written to this register, bits BC2 to BC0 in the I2CCR0 register become 000b, and bits LRB, AAS, and AL in the I2CSR register become 0.

Figure 24.2 I2CTRSR Register

The I2CTRSR register is an 8-bit shift register where received data is stored and transmit data is written. When transmit data is written to this register, the data is synchronized with the SCL clock and shifted out in descending order from bit 7. Every time a bit is shifted out, the data is shifted to the left by 1 bit. During a receive operation, the data is synchronized with the SCL clock and stored in order starting from bit 0. 1 bit of data is shifted (to the left) for every bit that is input. Figure 24.3 shows the timing when the received data is stored to the I2CTRSR register.

The I2CTRSR register is write enabled when the ICE bit in the I2CCR0 register is 1 (I^{2}C-bus interface enabled). When the ICE bit is 1 and the MST bit in the I2CSR register is 1 (master mode), writing data to the I2CTRSR register resets the bit counter and the SCL clock is output.

Write to the I2CTRSR register when a START condition is generated or the MSCL pin is low. The register can always be read.

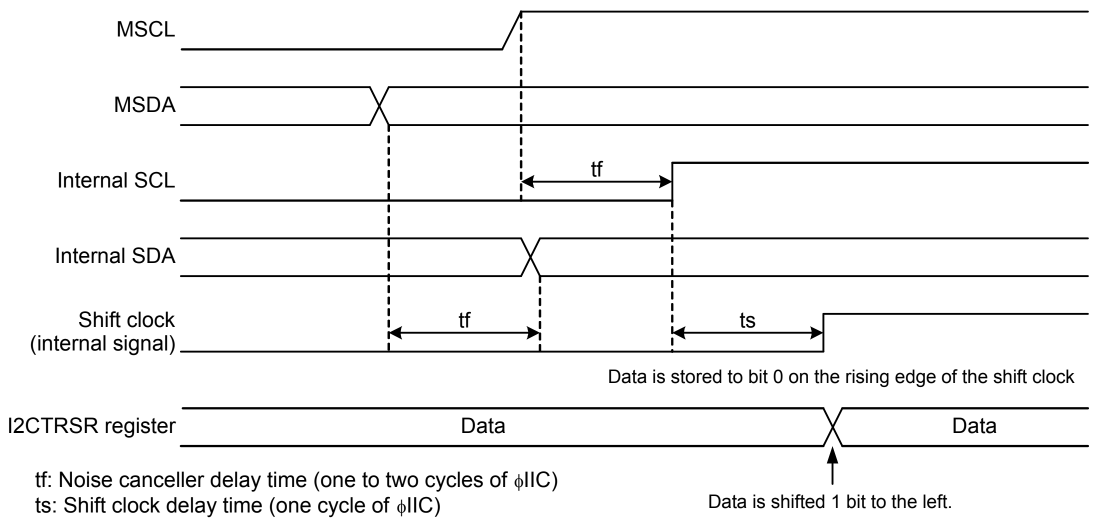
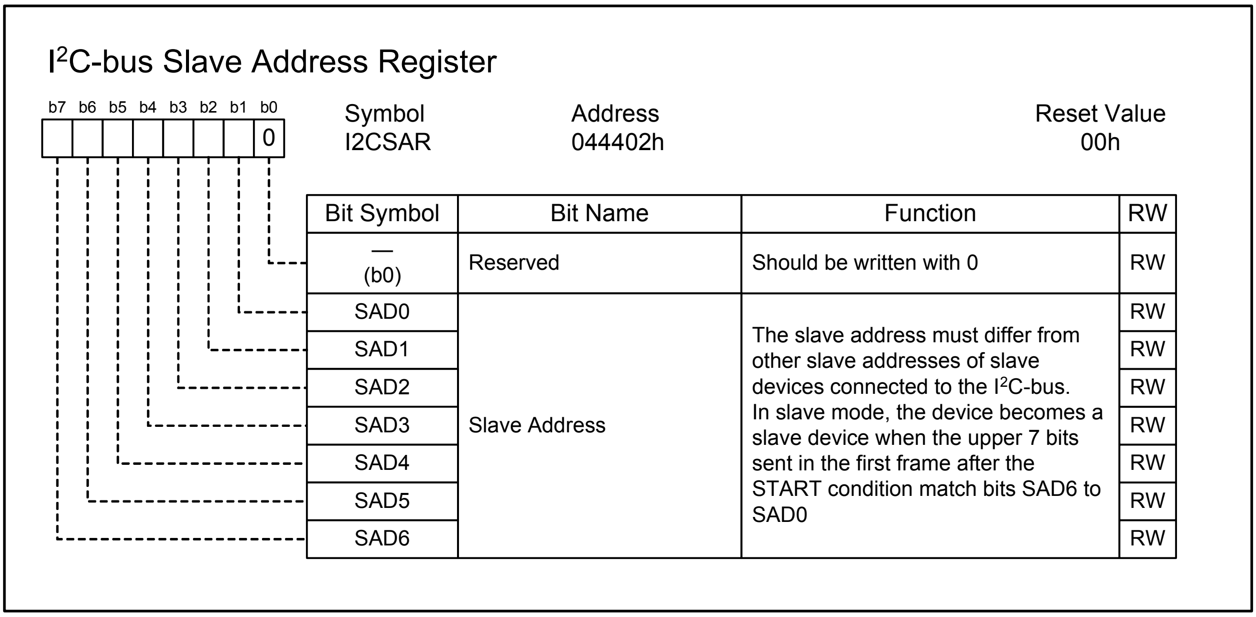

MSCL
MSDA
Internal SCL
Internal SDA
Shift clock (internal signal)
I2CTRSR register
tf
tf
ts
Data is stored to bit 0 on the rising edge of the shift clock
Data
Data
Data is shifted 1 bit to the left.
tf: Noise canceller delay time (one to two cycles of ϕIIC)
ts: Shift clock delay time (one cycle of ϕIIC)

Figure 24.3 Received Data Storing Timing to the I2CTRSR Register

24.1.2 I2C-bus Slave Address Register (I2CSAR)



I2C-bus Slave Address Register

Symbol	Address	Reset Value
I2CSAR	044402h	00h

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	RW
SAD0	Slave Address	The slave address must differ from other slave addresses of slave devices connected to the I^{2}C-bus. In slave mode, the device becomes a slave device when the upper 7 bits sent in the first frame after the START condition match bits SAD6 to SAD0	RW
SAD1			RW
SAD2			RW
SAD3			RW
SAD4			RW
SAD5			RW
SAD6			RW

Figure 24.4 I2CSAR Register

The I2CSAR register stores a slave address to automatically recognize itself as a slave device. When the received address matches the slave address, the device operates as a slave device.

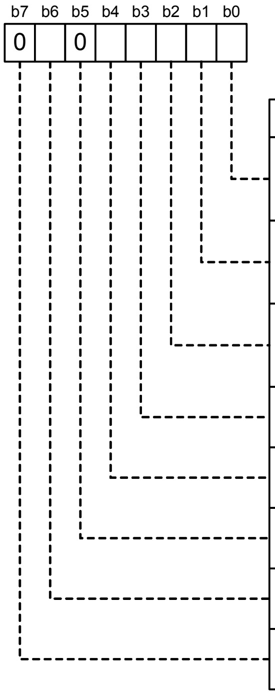
24.1.2.1 Bits SAD6 to SAD0

Bits SAD6 to SAD0 store a slave address. When the addressing format is enabled, the received 7-bit address and the slave address set in bits SAD6 to SAD0 are compared. When a match is detected, the device operates as a slave device.

24.1.3 I^{2}C-bus Control Register 0 (I2CCR0)

I^{2}C-bus Control Register 0

Symbol	Address	Reset Value
I2CCR0	044403h	0000 0000b



Bit Symbol	Bit Name	Function	RW
BC0	Transmit/Receive Bit Length Setting Bit [1]	b2 b1 b0 0 0 0 : 8 bits 0 0 1 : 7 bits 0 1 0 : 6 bits 0 1 1 : 5 bits 1 0 0 : 4 bits 1 0 1 : 3 bits 1 1 0 : 2 bits 1 1 1 : 1 bit	RW
BC1			RW
BC2			RW
ICE	I^{2}C-bus Interface Enable Bit	0: I^{2}C-bus interface disabled 1: I^{2}C-bus interface enabled	RW
DFS	Data Format Select Bit	0: Addressing format 1: Free data format	RW
— (b5)	Reserved	Should be written with 0	RW
RST	I^{2}C-bus Interface Reset Bit	Writing 1 to this bit resets the I^{2}C-bus interface circuit	RW
— (b7)	Reserved	Should be written with 0	RW

Note:

1. These bits automatically become 000b in the following cases:
   - When a START or STOP condition is detected
   - When data transmission is completed
   - When data reception is completed

Figure 24.5 I2CCR0 Register

The I2CCR0 register controls data communication format.

24.1.3.1 Bits BC2 to BC0

Bits BC2 to BC0 set the data bit length to be transmitted or received next. When data transmission or reception is completed for the data length (acknowledge clock pulse is included in the number when the ACKCLK bit in the I2CCCR register is 1) specified with bits BC2 to BC0, an I^{2}C-bus interface interrupt request is generated. Consequently, bits BC2 to BC0 become 000b. Note that these bits also become 000b when a START condition is detected. Address data is transmitted or received in 8 bits regardless of their settings.

24.1.3.2 ICE Bit

The ICE bit enables the I^2C-bus interface. Set this bit to 1 to enable the I^2C-bus interface and 0 to disable it. When this bit is 0, pins MSDA and MSCL are fixed high (these pins are high-impedance when the corresponding NOD bits in registers P7_0S and P7_1S are 1), therefore the I^2C-bus interface cannot be used.

When the ICE bit is set to 0, the following occurs:

- Bits ADZ, AAS, AL, BBSY, TRS, and MST in the I2CSR register become 0, and the IRF bit becomes 1.
- Writing to the I2CTRSR register is disabled.
- The I^2C-bus system clock (ϕIIC) is stopped, and the internal counter and flags are reset.
- The TOF bit in the I2CCR2 register becomes 0 (timeout not detected).

24.1.3.3 DFS Bit

The DFS bit enables the automatic recognition of a slave address. When the DFS bit is set to 0, the addressing format is selected and the slave address is automatically recognized. In this setting, data is received only when a general call address is received or a slave address match is detected. When the DFS bit is set to 1, the free data format is selected. In this setting, the slave address is not recognized, so all data are received.

24.1.3.4 RST Bit

The RST bit resets the I^2C-bus interface when a communication error occurs. When the ICE bit is set to 1 (I^2C-bus interface enabled), writing 1 (reset) to the RST bit has the following effects on the I^2C-bus interface:

- Bits ADZ, AAS, AL, BBSY, TRS, and MST in the I2CSR register become 0, and the IRF bit becomes 1.
- The TOF bit in the I2CCR2 register becomes 0 (timeout not detected).
- The internal counter and flags are reset.

When the RST bit is written with 1, the multi-master I^2C-bus interface is reset within a maximum of 2.5 ϕIIC cycles. Consequently, the RST bit automatically becomes 0.

Figure 24.6 shows the timing when the I^2C-bus interface is reset.

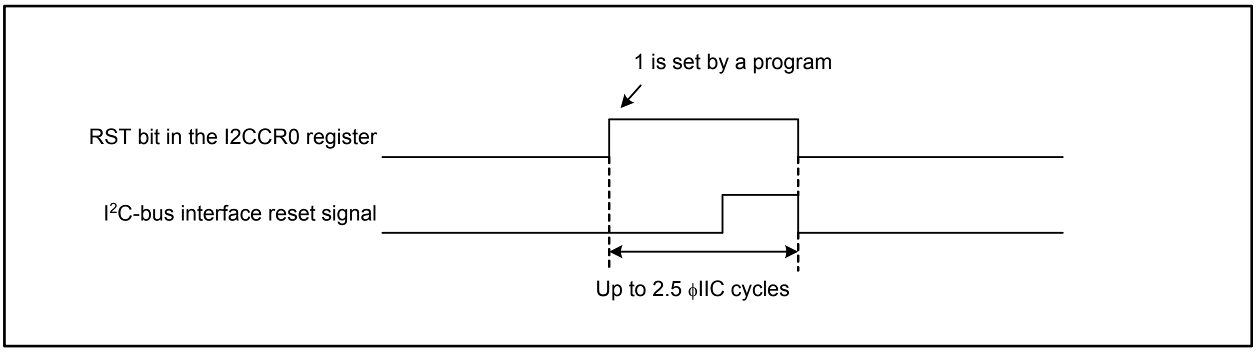
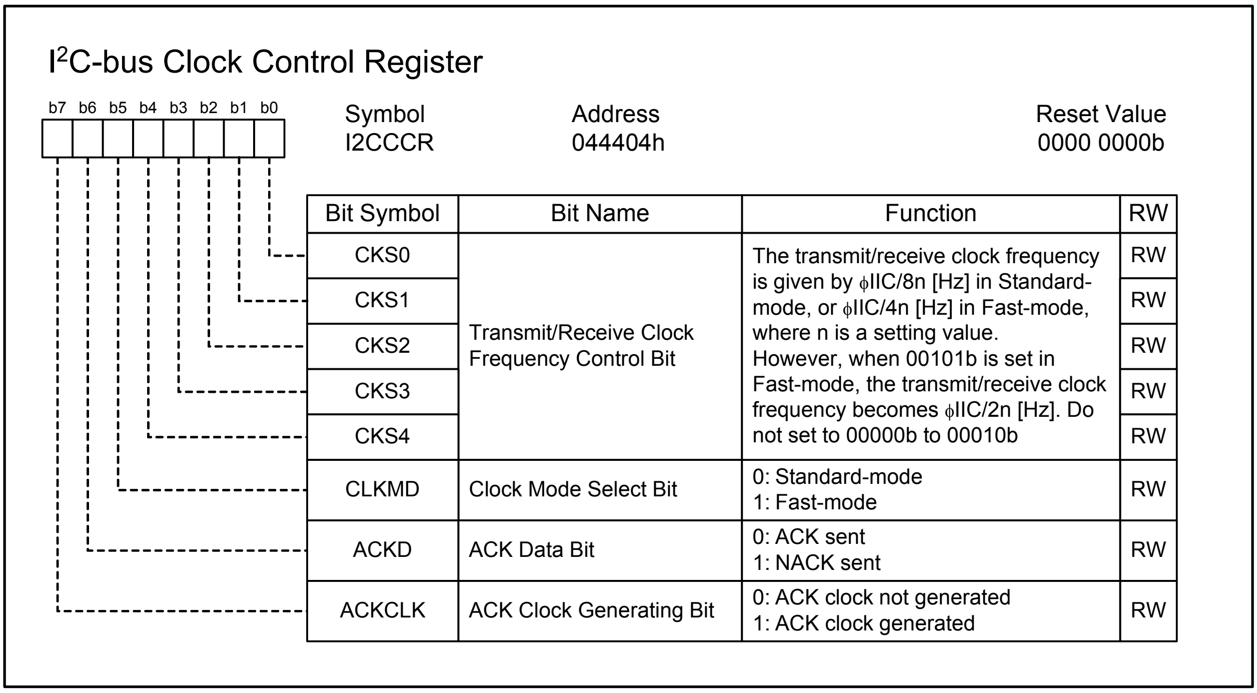

1 is set by a program
RST bit in the I2CCR0 register
I2C-bus interface reset signal
Up to 2.5 ϕIIC cycles

Figure 24.6 I^2C-bus Interface Reset Timing

24.1.4 I^{2}C-bus Clock Control Register (I2CCCR)


I^{2}C-bus Clock Control Register

Symbol I2CCCR Address 044404h Reset Value 0000 0000b

Bit Symbol	Bit Name	Function	RW
CKS0	Transmit/Receive Clock Frequency Control Bit	The transmit/receive clock frequency is given by ϕIIC/8n [Hz] in Standard-mode, or ϕIIC/4n [Hz] in Fast-mode, where n is a setting value. However, when 00101b is set in Fast-mode, the transmit/receive clock frequency becomes ϕIIC/2n [Hz]. Do not set to 00000b to 00010b	RW
CKS1			RW
CKS2			RW
CKS3			RW
CKS4			RW
CLKMD	Clock Mode Select Bit	0: Standard-mode 1: Fast-mode	RW
ACKD	ACK Data Bit	0: ACK sent 1: NACK sent	RW
ACKCLK	ACK Clock Generating Bit	0: ACK clock not generated 1: ACK clock generated	RW

Figure 24.7 I2CCCR Register

The I2CCCR register controls ACK and sets SCL mode and SCL clock frequency. While data is being transmitted or received, only rewrite the ACKD bit.

24.1.4.1 Bits CKS4 to CKS0

Bits CKS4 to CKS0 set the SCL clock frequency. The SCL clock frequency varies as shown in Table 24.3, where n is a setting value of bits CKS4 to CKS0 (n = 3 to 31). Do not rewrite these bits while data is being transmitted or received.

Table 24.3 I2CCCR Register Setting Values and SCL Frequencies

Bits CKS4 to CKS0 Setting Value (n)	SCL Frequency (When ϕIIC = 4 MHz) [1]	
	Standard-mode	Fast-mode
0 to 2	Do not set [2]	Do not set [2]
3	Do not set [3]	333 kHz (ϕIIC/4n)
4	Do not set [3]	250 kHz (ϕIIC/4n)
5	100 kHz (ϕIIC/8n)	400 kHz (ϕIIC/2n) [4]
6 to 31	83 to 16 kHz (ϕIIC/8n)	166 to 32 kHz (ϕIIC/4n)

Notes:

1. The CKS value must be set so the SCL clock frequency is 100 kHz or less in Standard-mode or 400 kHz or less in Fast-mode. The high period of the SCL clock has a margin of error of +2 to -4 ϕIIC in Standard-mode, and +2 to -2 ϕIIC in Fast-mode. Note that if the high period is shortened, the low period is lengthened, so the frequency remains unchanged.
2. Do not set the CKS value to 0 to 2 regardless of the ϕIIC frequency.
3. When ϕIIC is 4 MHz or higher, do not set the CKS value to 3 or 4. The SCL clock frequency will extend beyond the specified range.
4. The normal duty cycle of the SCL clock is 50%. When the CKS value is 5 in Fast-mode, it varies from 35% to 45%.

24.1.4.2 CLKMD Bit

Set the CLKMD bit to select the SCL mode. Set this bit to 0 to select Standard-mode and 1 for Fast-mode. To use the device under the Fast-mode I^2C-bus specification (up to 400 kbit/s), set ϕIIC to be 4 MHz or higher.

24.1.4.3 ACKD Bit

Set the ACKD bit to select the state of the MSDA pin with the ACK clock. When the ACKD bit is set to 0, the MSDA pin becomes low (acknowledged) by an ACK. When the ACKD bit is 1, the MSDA pin is held high with the ACK clock.
Table 24.4 lists the MSDA pin state with the ACK clock.

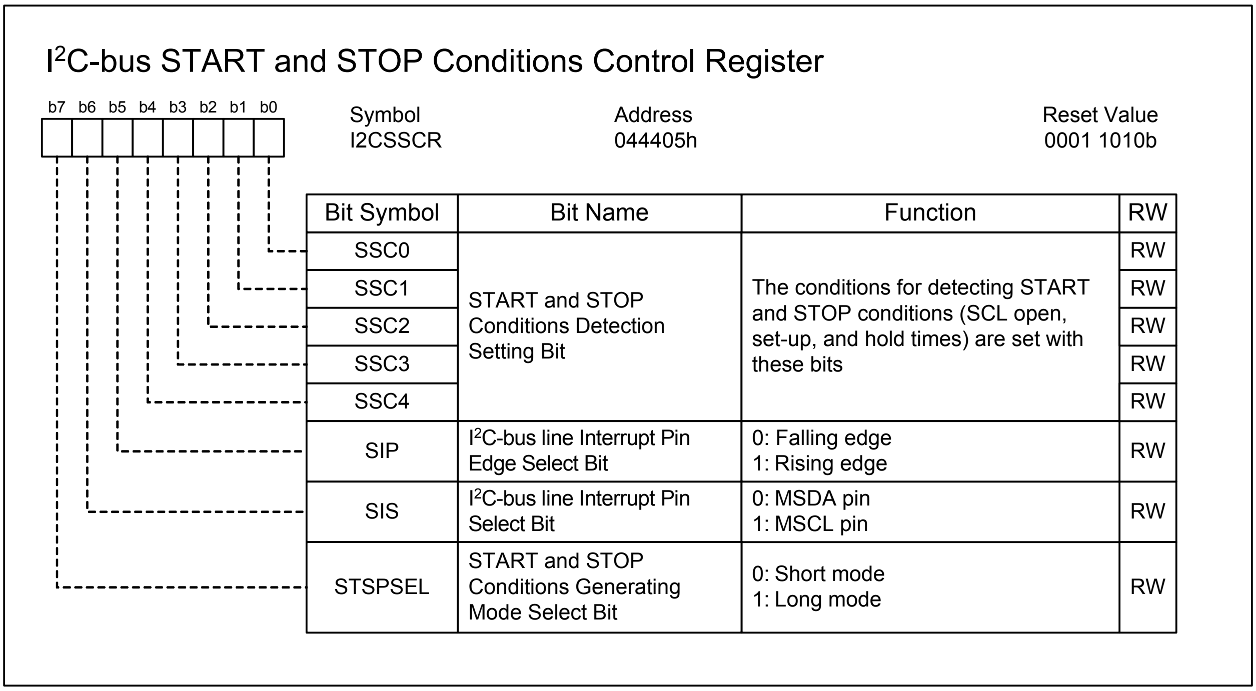
Table 24.4 MSDA Pin States with the ACK Clock

Received Content	DFS Bit	ACKD Bit	Slave Address	MSDA Pin State
Slave address	0	0	Match	Low (ACK)
			No match	High (NACK)
		1	—	High (NACK)
	1	0	—	Low (ACK)
		1	—	High (NACK)
Data	—	0	—	Low (ACK)
		1	—	High (NACK)

24.1.4.4 ACKCLK Bit

Set the ACKCLK bit to select whether or not to generate an ACK handshake. When this bit is 1 (ACK clock generated), an ACK clock pulse is generated after 1 byte of data is transmitted or received. When this bit is 0 (ACK clock not generated), the ACK clock is not generated after 1 byte of data is transmitted or received. In this case, the IR bit in the I2CIC register becomes 1 (I^2C-bus interface interrupt requested) on the last falling edge of the clock for data transmission or reception.

24.1.5 I^{2}C-bus START and STOP Conditions Control Register (I2CSSCR)



I^{2}C-bus START and STOP Conditions Control Register

Symbol	Address	Reset Value
I2CSSCR	044405h	0001 1010b

Bit Symbol	Bit Name	Function	RW
SSC0	START and STOP Conditions Detection Setting Bit	The conditions for detecting START and STOP conditions (SCL open, set-up, and hold times) are set with these bits	RW
SSC1			RW
SSC2			RW
SSC3			RW
SSC4			RW
SIP	I^{2}C-bus line Interrupt Pin Edge Select Bit	0: Falling edge 1: Rising edge	RW
SIS	I^{2}C-bus line Interrupt Pin Select Bit	0: MSDA pin 1: MSCL pin	RW
STSPSEL	START and STOP Conditions Generating Mode Select Bit	0: Short mode 1: Long mode	RW

Figure 24.8 I2CSSCR Register

The I2CSSCR register controls the detection and generation of START and STOP conditions.

24.1.5.1 Bits SSC4 to SSC0

Bits SSC4 to SSC0 select the parameters for detecting the START and STOP conditions by setting the high period of SCL pin, set-up, and hold times. This parameter is set by referencing the I^{2}C-bus system clock (ϕIIC). Therefore, it changes according to the XIN frequency and the setting of the I^{2}C-bus system clock select bits (i.e. bits ICK4 to ICK0 in registers I2CCR2 and I2CCR1). Do not set an odd number or 00000b to bits SSC4 to SSC0. Detection of START and STOP conditions starts immediately after setting the ICE bit in the I2CCR0 register to 1 (I^{2}C-bus interface enabled). Table 24.11 lists the recommended values for bits SSC4 to SSC0.

24.1.5.2 SIP Bit

Set the SIP bit to select which of the edges of MSCL or MSDA pin generates the I^{2}C-bus line interrupt. Set this bit to 0 to select the falling edge, and 1 to select the rising edge.

24.1.5.3 SIS Bit

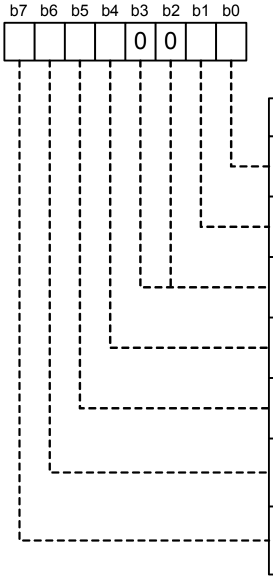
Set the SIS bit to select the input signal to be used as an I^{2}C-bus line interrupt source. To select the MSDA pin as an I^{2}C-bus line interrupt source, set this bit to 0. To select the MSCL pin, set this bit to 1.

24.1.5.4 STSPSEL Bit

Set the STSPSEL bit to select the set-up and hold times when START and STOP conditions are generated. Set this bit to 0 to select short mode and 1 to select long mode. The STSPSEL bit must be set to 1 (long mode) when the ϕIIC frequency is higher than 4 MHz. Figure 24.16 shows the START condition generation timing. Table 24.9 lists the set-up and hold times when START and STOP conditions are generated.

24.1.6 I^2C-bus Control Register 1 (I2CCR1)

I^2C-bus Control Register 1 (1)



Symbol	Address	Reset Value
I2CCR1	044406h	0011 0000b

Bit Symbol	Bit Name	Function	RW
STIE	STOP Condition Detection Interrupt Enable Bit	0: Disabled 1: Enabled	RW
RIE	Successful Receive Interrupt Enable Bit (2)	0: Disabled 1: Enabled	RW
— (b3-b2)	Reserved	Should be written with 0	RW
SDAO	Internal SDA Output Monitor Bit	0: Low 1: High	RO
SCLO	Internal SCL Output Monitor Bit	0: Low 1: High	RO
ICK0	I^2C-bus System Clock Select Bit (3)	b7 b6 0 0 : fIIC divided-by-2 0 1 : fIIC divided-by-4 1 0 : fIIC divided-by-8 1 1 : Do not use this combination	RW
ICK1			RW

Notes:
1. Do not use a bit processing instruction with this register.
2. Set this bit to 0 when the ACKCLK bit is 0 (ACK clock is not generated).
3. These bits are enabled when bits ICK4 to ICK2 in the I2CCR2 register are 000b.

Figure 24.9 I2CCR1 Register

The I2CCR1 register controls the I^2C-bus interface.

24.1.6.1 STIE Bit

Set the STIE bit to enable an interrupt when detecting a STOP condition. When this bit is set to 1, the I^2C-bus interface interrupt is generated when detecting a STOP condition. Consequently, the STOP bit in the I2CCR2 register becomes 1 (STOP condition detection interrupt requested) and the IR bit in the I2CIC register becomes 1 (I^2C-bus interface interrupt requested).

24.1.6.2 RIE Bit

Set the RIE bit to enable an interrupt when receiving the last bit of data when the ACKCLK bit in the I2CCCR register is 1 (ACK clock generated). When the RIE bit is 1, the I^2C-bus interface interrupt is generated when the last bit (the eighth falling edge of the SCL) of data is received.

The I^2C-bus interface interrupt is generated at the ACK bit transmission (the ninth falling edge of the SCL) regardless of the RIE bit setting, therefore two I^2C-bus interface interrupts are generated per data when the RIE bit is 1. The source of the interrupt can be identified by reading the RIE bit. The read value indicates the internal WAIT flag state. When the read value is 1, the last bit of data is the interrupt source. When the read value is 0, the ACK bit is the interrupt source.

Set the RIE bit to 0 when the ACKCLK bit in the I2CCCR register is 0 (ACK clock not generated). When the device is transmitting data or receiving a slave address, the I^2C-bus interface interrupt is generated only by the ACK bit (the ninth falling edge of the SCL) regardless of the RIE bit setting. In both cases, the internal WAIT flag is 0.

Table 24.5 I2C-bus Interrupt Request Generation Timings and How to Resume Communication

I2C-bus Interface Interrupt Generation Timing	Internal WAIT Flag	Resuming Transmission/Reception
Last bit of data (on eighth clock)	1	Write to the ACKD bit in the I2CCCR register
ACK bit (on ninth clock)	0	Write to the I2CTRSR register

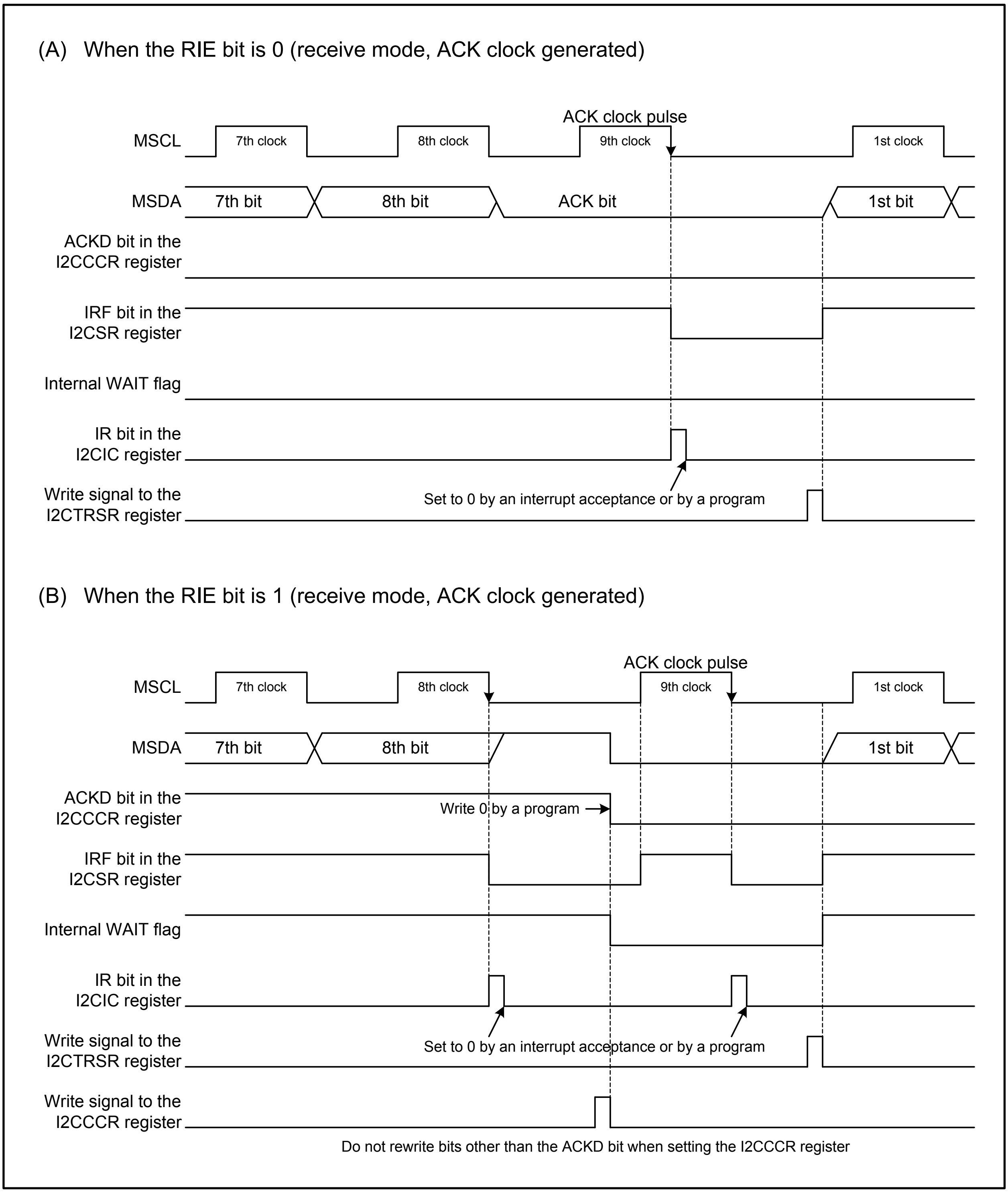

(A) When the RIE bit is 0 (receive mode, ACK clock generated)
MSCL
7th clock
8th clock
ACK clock pulse
9th clock
1st clock
MSDA
7th bit
8th bit
ACK bit
1st bit
ACKD bit in the I2CCCR register
IRF bit in the I2CSR register
Internal WAIT flag
IR bit in the I2CIC register
Write signal to the I2CTRSR register
Set to 0 by an interrupt acceptance or by a program
(B) When the RIE bit is 1 (receive mode, ACK clock generated)
MSCL
7th clock
8th clock
ACK clock pulse
9th clock
1st clock
MSDA
7th bit
8th bit
1st bit
ACKD bit in the I2CCCR register
Write 0 by a program
IRF bit in the I2CSR register
Internal WAIT flag
IR bit in the I2CIC register
Write signal to the I2CTRSR register
Set to 0 by an interrupt acceptance or by a program
Write signal to the I2CCCR register
Do not rewrite bits other than the ACKD bit when setting the I2CCCR register

Figure 24.10 Interrupt Request Generation Timing in Receive Mode

24.1.6.3 Bits SDAO and SCLO

Bits SDAO and SCLO are read-only bits and used to monitor the logical values of the internal SDA output signal and internal SCL output signal, respectively. Only set these bits to 0. Note that the internal SDA and SCL output signals indicate output levels before being affected by external devices and do not indicate MSDA and MSCL pin states.

24.1.6.4 Bits ICK1 and ICK0

Set bits ICK1 and ICK0 to select the frequency of the I^{2}C-bus system clock (ϕIIC). These bits are enabled when bits ICK4 to ICK2 in the I2CCR2 register are 000b. Rewrite these bits when the ICE bit in the I2CCR0 register is 0 (I^{2}C-bus interface disabled). The frequency of the I^{2}C-bus system clock (ϕIIC) can be selected from fIIC divided-by-2, -4, and -8 by setting these bits. fIIC divided-by-2.5, -3, -5, and -6 are also available by setting bits ICK4 to ICK2 in the I2CCR2 register. However, bits ICK1 and ICK0 are disabled in this case.

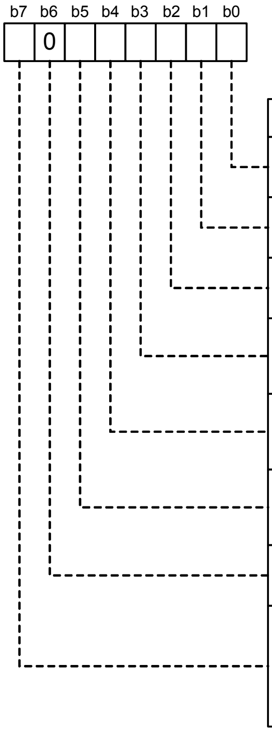
Table 24.6 I^{2}C-bus System Clock (ϕIIC) Select Bit Settings

I2CCR2 Register			I2CCR1 Register		ϕIIC
ICK4 bit	ICK3 bit	ICK2 bit	ICK1 bit	ICK0 bit	
0	0	0	0	0	fIIC divided-by-2
			0	1	fIIC divided-by-4
			1	0	fIIC divided-by-8
0	0	1	0	0	fIIC divided-by-2.5
0	1	0	0	0	fIIC divided-by-3
0	1	1	0	0	fIIC divided-by-5
1	0	0	0	0	fIIC divided-by-6

Only set the values listed above.

24.1.7 I2C-bus Control Register 2 (I2CCR2)

I2C-bus Control Register 2



Symbol I2CCR2 Address 044407h Reset Value 0X00 0000b

Bit Symbol	Bit Name	Function	RW
TOE	Timeout Detector Enable Bit	0: Timeout detector disabled 1: Timeout detector enabled	RW
TOF	Timeout Detect Flag	0: Timeout not detected 1: Timeout detected	RO
TOSEL	Timeout Detect Period Select Bit	0: Long 1: Short	RW
ICK2	I2C-bus System Clock Select Bit	b5 b4 b3 0 0 0 : ϕIIC = set by bits ICK1 and ICK0 in the I2CCR1 register 0 0 1 : ϕIIC = fIIC divided-by-2.5 0 1 0 : ϕIIC = fIIC divided-by-3 0 1 1 : ϕIIC = fIIC divided-by-5 1 0 0 : ϕIIC = fIIC divided-by-6 Only set the values listed above	RW
ICK3			RW
ICK4			RW
— (b6)	Reserved	Should be written with 0	RW
STOP	STOP Condition Detect Interrupt Request Monitor Bit	0: I2C-bus interface interrupt not requested 1: I2C-bus interface interrupt requested	RW

Figure 24.11 I2CCR2 Register

The I2CCR2 register controls communication error detection. If the SCL clock stops during transmission or reception, each device connected to the bus is halted suspending communication. To avoid this, the multi-master I2C-bus interface supports a function to generate an I2C-bus interface interrupt when the SCL clock is held high for a specified period of time during transmission or reception.

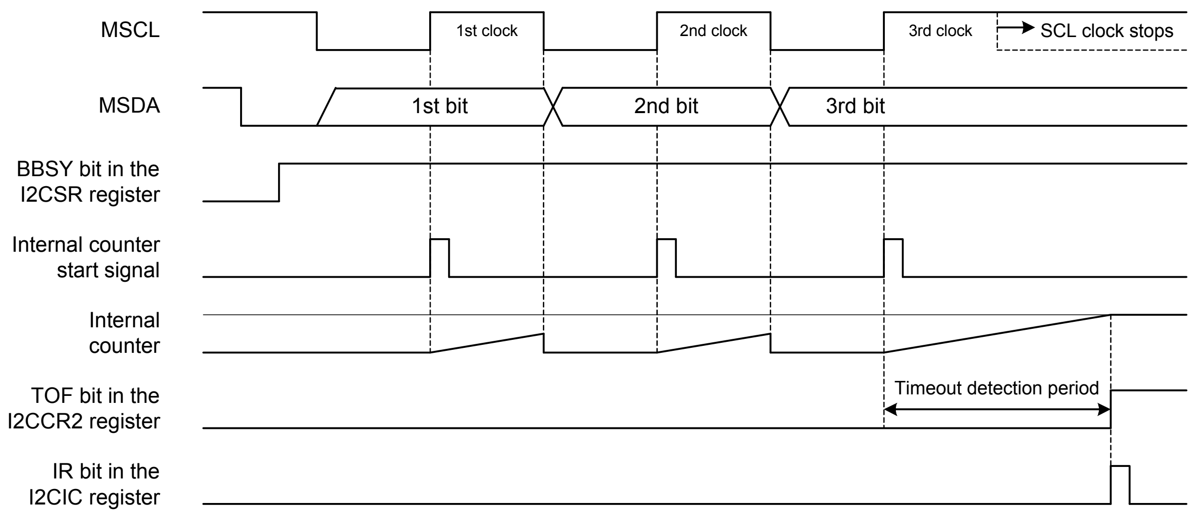

MSCL
1st clock
2nd clock
3rd clock
SCL clock stops
MSDA
1st bit
2nd bit
3rd bit
BBSY bit in the I2CSR register
Internal counter start signal
Internal counter
TOF bit in the I2CCR2 register
Timeout detection period
IR bit in the I2CIC register

Figure 24.12 Timeout Detection Timing

24.1.7.1 TOE Bit

The TOE bit enables the timeout detector. When this bit is set to 1, the timeout detector is enabled, and when the SCL clock is held high for a specified period of time while the BBSY bit in the I2CSR register is 1 (bus is busy), an I^{2}C-bus interface interrupt request is generated.

The timeout detection period is determined by 1) the internal counter that uses ϕIIC as a count source, and 2) the TOSEL bit setting (selects the timeout detection period to be either long or short). Refer to 24.1.7.3 “TOSEL bit” for details.

When a timeout is detected, set the ICE bit in the I2CCR0 register to 0 (I^{2}C-bus interface disabled) and initialize the I^{2}C-bus interface.

24.1.7.2 TOF Bit

The TOF bit is a flag that indicates the state of a timeout detection. This bit is enabled when the TOE bit is 1. When the TOF bit becomes 1 (timeout detected), the IR bit in the I2CIC register becomes 1 (I^{2}C-bus interface interrupt requested) simultaneously.

24.1.7.3 TOSEL Bit

The TOSEL bit selects a long or short length for a timeout detection period. This bit is enabled when the TOE bit is 1 (timeout detector enabled). Set this bit to 0 to select the long timeout period. In this setting, the internal counter functions as a 16-bit counter. Set this bit to 1 to select the short timeout period. In this setting, the internal counter functions as a 14-bit counter.

The internal counter increments using the I^{2}C-bus system clock (ϕIIC) as a count source.

Table 24.7 lists timeout detection periods.

Table 24.7 Example Timeout Detection Periods

ϕIIC	Long Timeout Detection Period (TOSEL = 0)	Short Timeout Detection Period (TOSEL = 1)
4 MHz	16.4 ms	4.1 ms
2 MHz	32.8 ms	8.2 ms
1 MHz	65.6 ms	16.4 ms

24.1.7.4 Bits ICK4 to ICK2

Set bits ICK4 to ICK2 to select the frequency of the I^{2}C-bus system clock (ϕIIC). Rewrite these bits when the ICE bit in the I2CCR0 register is 0 (I^{2}C-bus interface disabled).

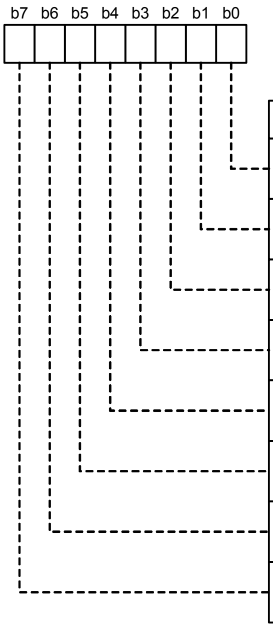
The frequency of the I^{2}C-bus system clock (ϕIIC) can be selected from fIIC divided-by-2.5, -3, -5, and -6. When bits ICK4 to ICK2 are set to 000b, fIIC divided-by-2, -4, and -8 can also be selected by setting bits ICK1 and ICK0 in the I2CCR1 register. Refer to Table 24.6.

24.1.7.5 STOP Bit

The STOP bit monitors the STOP condition detection interrupt. When the I^{2}C-bus interface interrupt is generated by the detection of a STOP condition, the STOP bit becomes 1. This bit is enabled when the STIE bit in the I2CCR1 register is 1 (STOP condition detection interrupt is enabled). This bit is set to 0 by a program. Writing 1 to this bit has no effect.

24.1.8 I^{2}C-bus Status Register (I2CSR)

I^{2}C-bus Status Register



Symbol	Address	Reset Value
I2CSR	044408h	0001 000Xb

Bit Symbol	Bit Name	Function	RW
LRB	Last Received Bit [1, 2]	0: Last received bit is 0 1: Last received bit is 1	RW
ADZ	General Call Address Detect Flag [1, 2]	0: General call address not detected 1: General call address detected	RW
AAS	Slave Address Match Flag [1, 2]	0: Address not matched 1: Address matched	RW
AL	Arbitration Lost Detect Flag [1, 2]	0: Arbitration lost not detected 1: Arbitration lost detected	RW
IRF	I^{2}C-bus Interface Interrupt Request Flag [3]	0: Requested 1: Not requested	RO
BBSY	Bus Busy Flag [2]	0: Bus is free 1: Bus is busy	RW
TRS	Transmit/Receive Switch Bit	0: Receive mode 1: Transmit mode [1]	RW
MST	Master/Slave Select Bit	0: Slave mode 1: Master mode [1]	RW

Notes:

1. Write 1111b to the lower 4 bits of this register to set the TRS or MST bit to 1 without generating a START or STOP condition.
2. These bits are read-only when using them to check the status.
3. This bit is read-only. Only set this bit to 0.

Figure 24.13 I2CSR Register

The I2CSR register monitors the state of the I^{2}C-bus interface. Write to this register only when using the functions listed in Table 24.8, and only set the values that are listed. Note that the lower 6 bits are not rewritten even when values from Table 24.8 are written to.

Table 24.8 I2CSR Register Settings and Functions

Values Written to the I2CSR Register								Function
MST	TRS	BBSY	IRF	AL	AAS	ADZ	LRB	
0	0	X	0	1	1	1	1	Select slave-receive mode
0	1							Select slave-transmit mode
1	0							Select master-receive mode
1	1							Select master-transmit mode
1	1	0	0	0	0	0	0	Select master-transmit mode and set the device to be on STOP condition standby.
		1						Select master-transmit mode and set the device to be on START condition standby.

24.1.8.1 LRB Bit

The LRB bit stores the data of the last received bit. It is used to check whether an ACK is received. When the ACKCLK bit in the I2CCCR register is 1 (ACK clock generated), the LRB bit becomes 0 when the ACK is received, and 1 when the ACK is not received. When the ACKCLK bit is 0 (ACK clock not generated), the last bit of data is stored to the LRB bit. When a value is written to the I2CTRSR register, the LRB bit becomes 0.

24.1.8.2 ADZ Bit

The ADZ bit is a flag that indicates that the general call address was received. When the DFS bit in the I2CCR0 register is 0 (addressing format) in slave-receive mode, the ADZ bit becomes 1 when the general call address is received.

The ADZ bit becomes 0 in any of the following cases:

- When a STOP or START condition is detected
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

24.1.8.3 AAS Bit

The AAS bit is a flag that indicates whether the received address matches its own slave address. The AAS bit becomes 1 when the received address matches its own slave address in bits SAD6 to SAD0 in the I2CSAR register, when the DFS bit in the I2CCR0 register is 0 (addressing format) in slave-receive mode, or when the received address is the general call address.

The AAS bit becomes 0 in any of the following cases:

- When data is written to the I2CTRSR register
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

24.1.8.4 AL Bit

The AL bit is a flag that indicates arbitration lost detection. In master transmit mode, if the MSDA pin is changed to low by another device, then the AL bit becomes 1. Consequently, the TRS bit in the I2CSR register becomes 0 (receive mode), and then the MST bit becomes 0 (slave mode) at the end of the byte in which an arbitration lost is detected.

The AL bit becomes 0 in any of the following cases:

- When data is written to the I2CTRSR register
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

24.1.8.5 IRF Bit

Set the IRF bit to generate the I^{2}C-bus interface interrupt request signal. When the I^{2}C-bus interface interrupt source is generated, first the IRF bit becomes 0, then the I^{2}C-bus interface interrupt is generated on the falling edge of the IRF bit. Refer to Figure 24.10 for the timing.

The IRF bit becomes 0 in any of the following cases:

- When 1-byte data transmission is completed (including when an arbitration lost is detected)
- When 1-byte data reception is completed
- When the slave address is matched in addressing format in slave-receive mode
- When the general call address is received in addressing format in slave-receive mode
- When address data reception is completed in free data format in slave-receive mode

The IRF bit becomes 1 in any of the following cases:

- When data is written to the I2CTRSR register
- When data is written to the I2CCCR register (the RIE bit is 1, internal WAIT flag is 1)
- When the ICE bit in the I2CCR0 register is set to 0 (I^{2}C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I^{2}C-bus interface reset)

24.1.8.6 BBSY Bit

The BBSY bit is a flag that indicates the availability of the I^{2}C-bus. The BBSY bit becomes 1 when a START condition is detected, and 0 when a STOP condition is detected. When the BBSY bit is 0, the I^{2}C-bus is not in use, and is available for the device to generate a START condition.

The detection of a START or STOP condition is dependent on the setting of bits SSC4 to SSC0 in the I2CSSCR register.

The BBSY bit becomes 0 in any of the following cases:

- When a STOP condition is detected
- When the ICE bit in the I2CCR0 register is set to 0 (I^{2}C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I^{2}C-bus interface reset)

24.1.8.7 TRS Bit

The TRS bit determines the direction of data communication. When this bit is set to 0, the device enters receive mode and waits for data to be sent from another device. When this bit is set to 1, the device enters transmit mode and transmits data and address to the SDA line synchronized with the SCL clock.

The TRS bit automatically becomes 1 (transmit mode) when the received address matches its own slave address and the received R/$\overline{W}$ bit is 1 (data requested) in addressing format in slave-receive mode.

The TRS bit becomes 0 in any of the following cases:

- When this bit is set to 0
- When an arbitration lost is detected
- When a STOP condition is detected
- When the START condition redundancy prevention function is activated
- When a START condition is detected in slave mode
- When a NACK is received in slave mode
- When the ICE bit in the I2CCR0 register is set to 0 (I^{2}C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I^{2}C-bus interface reset)

24.1.8.8 MST Bit

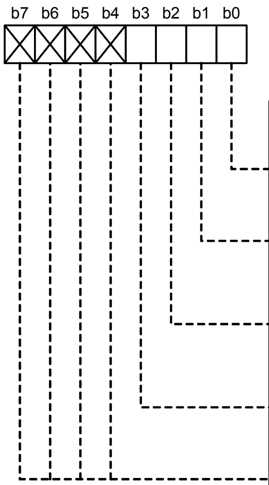
Set the MST bit to select master or slave mode. To enter slave mode, set this bit to 0. Communication is initiated in synchronization with the SCL clock generated by the master device. Set this bit to 1 to enter master mode. The device generates the SCL clock to initiate communication.

The MST bit becomes 0 in any of the following cases:

- When the MST bit is set to 0
- When an arbitration lost is detected, and transmission of the corresponding byte is completed
- When a STOP condition is detected
- When a START condition is detected
- When the START condition redundancy prevention function is enabled
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

24.1.9 I2C-bus Mode Register (I2CMR)

I2C-bus Mode Register [1]



Symbol	Address	Reset Value
I2CMR	044410h	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
I2CEN	I2C-bus Interface/UART2 Switch Bit	0: UART2 1: I2C-bus interface	RW
CLK0	I2C-bus Interface Clock Source Select Bit	b3 b2 b1 0 0 0 : f1 divided-by-2 0 0 1 : f8 divided-by-2 0 1 0 : f2n divided-by-2 0 1 1 : Do not use this combination 1 0 0 : f1 1 0 1 : f8 1 1 0 : f2n 1 1 1 : Do not use this combination	RW
CLK1			RW
CLK2			RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		RW

Note:

1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.

Figure 24.14 I2CMR Register

The I2CMR register selects signals for the I2C-bus interface and the clock source. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.

24.1.9.1 I2CEN Bit

The I2CEN bit switches between signals for UART2 and the I2C-bus interface. Set this bit to 1 to use the following signals: MSDA, MSCL, the I2C-bus interface interrupt, and the I2C-bus line interrupt. When this bit is set to 0, signals for UART2 are enabled.

24.1.9.2 Bits CLK2 to CLK0

Bits CLK2 to CLK0 select the clock source for the I2C-bus interface clock (fIIC). It is selected from f1 divided-by-2, f8 divided-by-2, f2n divided-by-2, f1, f8, or f2n.

The clock source selected for the I2C-bus interface (fIIC) is used as the clock source for the I2C-bus system clock (ϕIIC).

24.2 Generating a START Condition

To enter a START condition standby state, write E0h to the I2CSR register while the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled) and the BBSY bit in the I2CSR register is 0 (bus is free). When in standby, write a slave address to the I2CTRSR register to generate a START condition. Consequently, the bit counter becomes 000b, 1 byte of the SCL clock is output, and the slave address is transmitted. Figure 24.15 shows how to generate a START condition.

Note that after a STOP condition is generated, writing to the I2CSR register is disabled for 1.5 cycles of ϕIIC after the BBSY bit becomes 0. To generate a START condition immediately after generating a STOP condition, first write E0h to the I2CSR register, then confirm that bits TRS and MST in the I2CSR register are 1. After that, write a slave address to the I2CTRSR register.

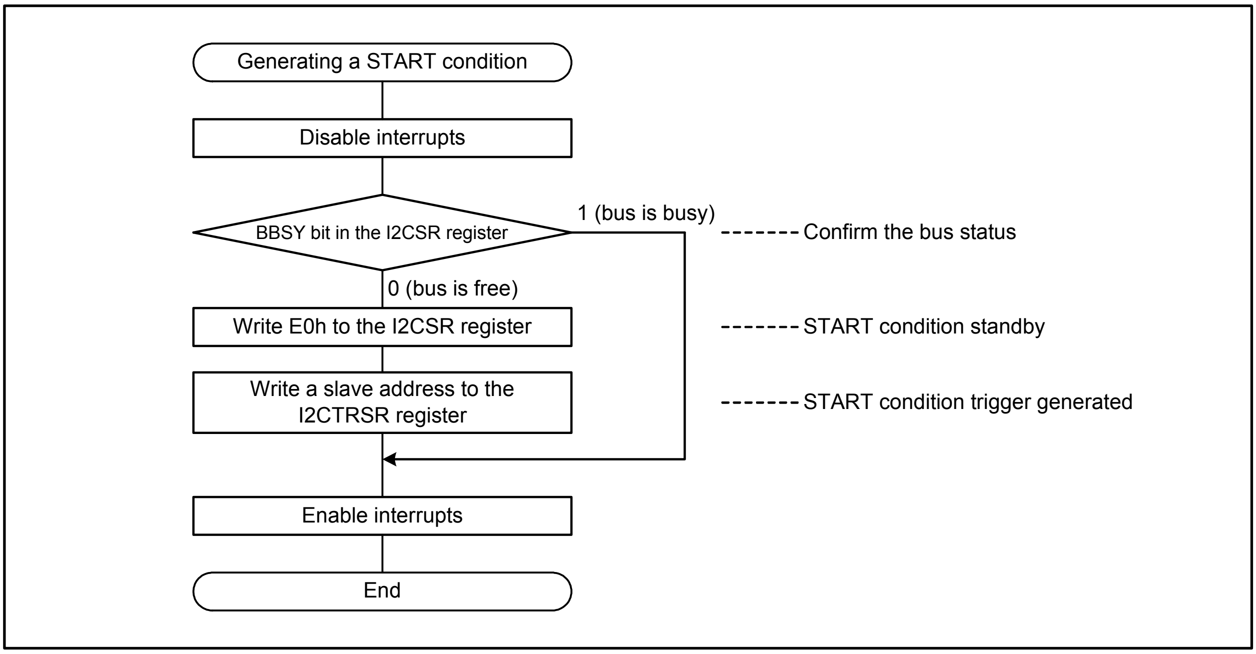

Generating a START condition
Disable interrupts
BBSY bit in the I2CSR register
1 (bus is busy)
Confirm the bus status
0 (bus is free)
Write E0h to the I2CSR register
START condition standby
Write a slave address to the I2CTRSR register
START condition trigger generated
Enable interrupts
End

Figure 24.15 Generating a START Condition

The timing to generate a START condition differs between Standard-mode and Fast-mode. Figure 24.16 shows START condition generation timing. Table 24.9 lists the set-up and hold times when a START or STOP condition is generated.

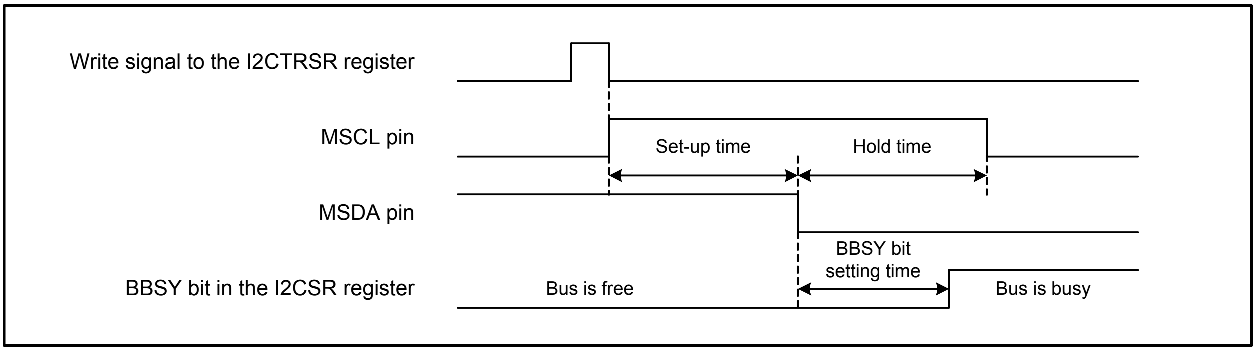


Figure 24.16 START Condition Generation Timing

Table 24.9 Set-up and Hold Times When Generating a START or STOP Condition

Parameter	SCL Mode	Short Mode (STSPSEL = 0)	Long Mode (STSPSEL = 1)
Set-up time	Standard-mode (CLKMD = 0)	5.0 μs (20)	13.0 μs (52)
	Fast-mode (CLKMD = 1)	2.5 μs (10)	6.5 μs (26)
Hold time	Standard-mode (CLKMD = 0)	5.0 μs (20)	13.0 μs (52)
	Fast-mode (CLKMD = 1)	2.5 μs (10)	6.5 μs (26)

CLKMD: Bit in the I2CCCR register
STSPSEL: Bit in the I2CSSCR register
Number of ϕIIC cycles in parentheses.

24.3 Generating a STOP Condition

To enter a STOP condition standby state, write C0h to the I2CSR register while the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled). Consequently, the MSDA pin becomes low. When in a standby state, write dummy data to the I2CTRSR register to generate a STOP condition. Figure 24.17 shows how to generate a STOP condition.

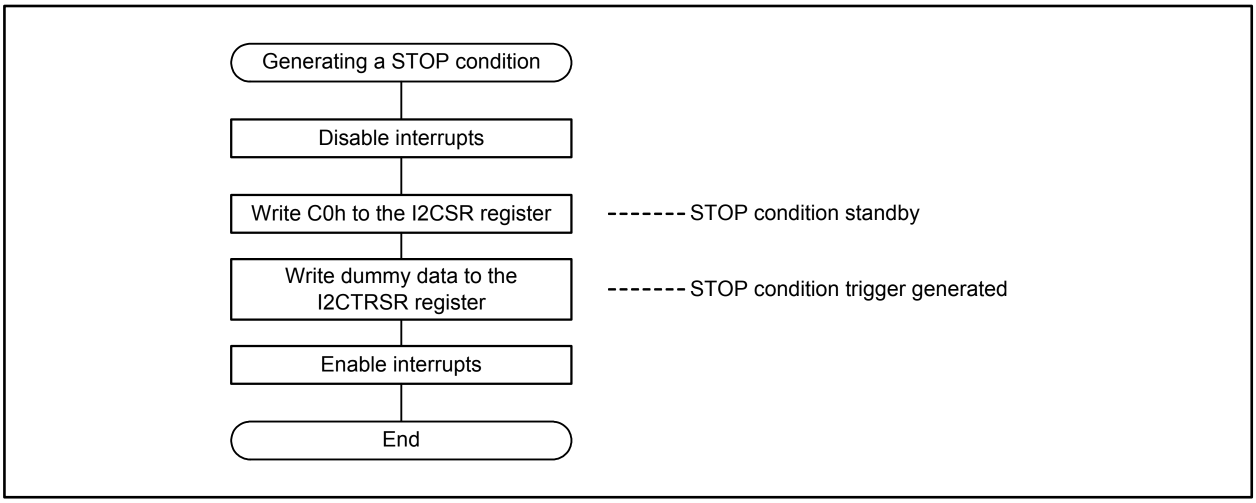

Generating a STOP condition
Disable interrupts
Write C0h to the I2CSR register
Write dummy data to the I2CTRSR register
Enable interrupts
End
STOP condition standby
STOP condition trigger generated

Figure 24.17 Generating a STOP Condition

The timing for generating a STOP condition differs between Standard-mode and Fast-mode. Figure 24.18 shows STOP condition generating timing. Table 24.9 lists the set-up and hold times when a START or STOP condition is generated.

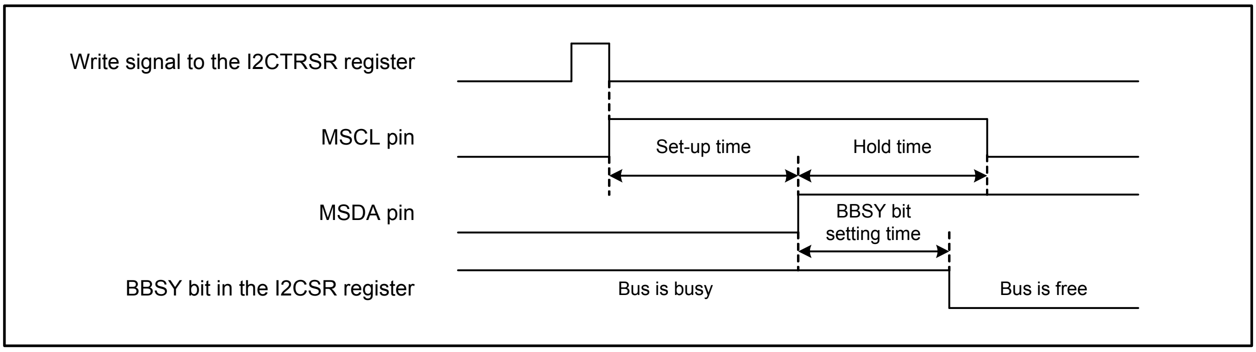


Figure 24.18 STOP Condition Generating Timing

Do not write the I2CSR or I2CTRSR register during the period after the standby setting until the BBSY bit in the I2CSR register becomes 0. Doing so may cause a failure of a successful STOP condition generation.

Furthermore, after the standby setting, the internal SCL output becomes low in the following case: after the MSCL pin becomes high and when it becomes low before the BBSY bit becomes 0. In this case, low output from the MSCL pin is stopped (clock line released) by generating a STOP condition, by setting the ICE bit in the I2CCR0 register to 0 (I2C-bus interface disabled), or by setting the RST bit to 1 (I2C-bus interface reset)

24.4 START Condition Redundancy Prevention Function

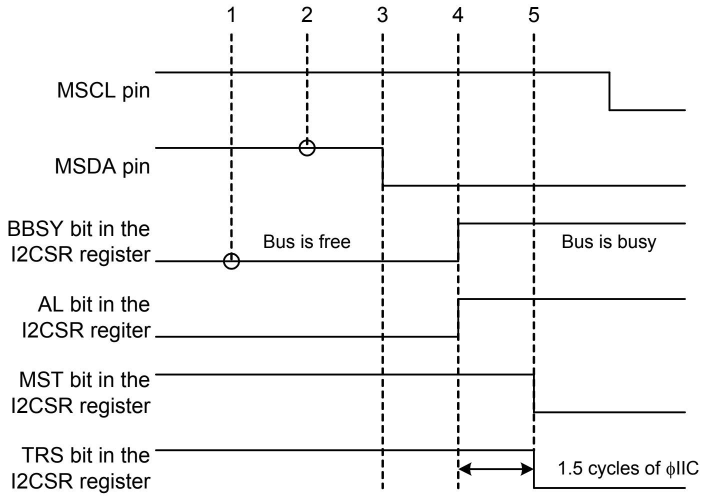
A START condition is generated when the bus is free (confirmed with the BBSY bit in the I2CSR register). However, before a START condition is generated, if a different master device generates another START condition, the BBSY bit may become 1. In this case, the START condition redundancy prevention function terminates the generation of its own START condition.

The START condition redundancy prevention functions as follows:

- The START condition standby setting is disabled (exits standby state)
- Writing to the I2CTRSR register is disabled (generation of the START condition trigger is disabled)
- Bits MST and TRS in the I2CSR register become 0 (enters slave-receive mode)
- The AL bit in the I2CSR register becomes 1 (arbitration lost is detected)

Figure 24.19 shows the operation of the START condition redundancy prevention function.

Example behavior of when a START condition from another device is generated while in a START condition standby state.


1
2
3
4
5
MSCL pin
MSDA pin
BBSY bit in the I2CSR register
Bus is free
Bus is busy
AL bit in the I2CSR regiter
MST bit in the I2CSR register
TRS bit in the I2CSR register
1.5 cycles of ϕIIC

1. Confirm the bus is free
2. START condition standby setting
3. Another START condition generated by external device
4. START condition detected
   The bus becomes busy at the same time the START condition redundancy prevention function is enabled (arbitration lost is generated).
5. Enter slave-receive mode

Figure 24.19 Example Operation of the START Condition Redundancy Prevention Function

The START condition redundancy prevention function is enabled from the falling edge of an SDA line in a START condition until the slave address is completely received. This means, when registers I2CSR and I2CTRSR are written during this period, then the START condition redundancy prevention function is enabled. Figure 24.20 shows the duration.

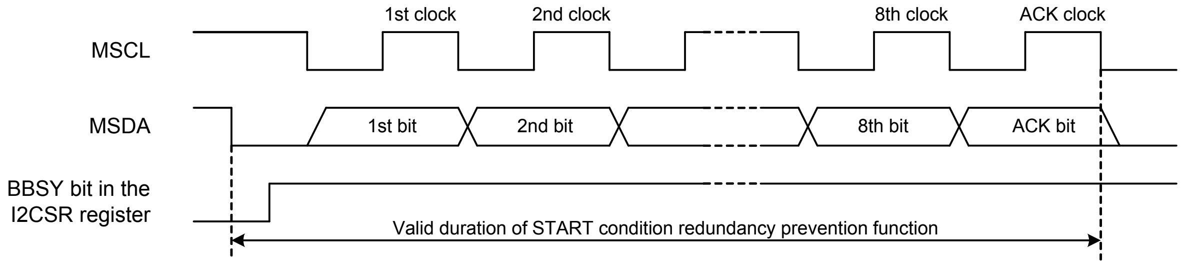

1st clock
2nd clock
8th clock
ACK clock
MSCL
MSDA
1st bit
2nd bit
8th bit
ACK bit
BBSY bit in the I2CSR register
Valid duration of START condition redundancy prevention function

Figure 24.20 Enabled Duration of the START Condition Redundancy Prevention Function

24.5 Detecting START and STOP Conditions

Figure 24.21 shows START condition detection, Figure 24.22 shows STOP condition detection, and Table 24.10 lists the parameters for detecting START and STOP conditions. The parameters to detect START and STOP conditions are set with bits SSC4 to SSC0 in the I2CSSCR register. These parameters are detectable only when the input signals of pins MSCL and MSDA meet all the conditions of the high period of MSCL pin, set-up, and hold times in Table 24.10.

The BBSY bit in the I2CSR register becomes 1 when a START condition is detected, and 0 when a STOP condition is detected. The timing for setting the BBSY bit differs between Standard-mode and Fast-mode. Refer to Table 24.11 for BBSY bit setting time. Table 24.11 lists the recommended settings for bits SSC4 to SSC0 in Standard-mode.

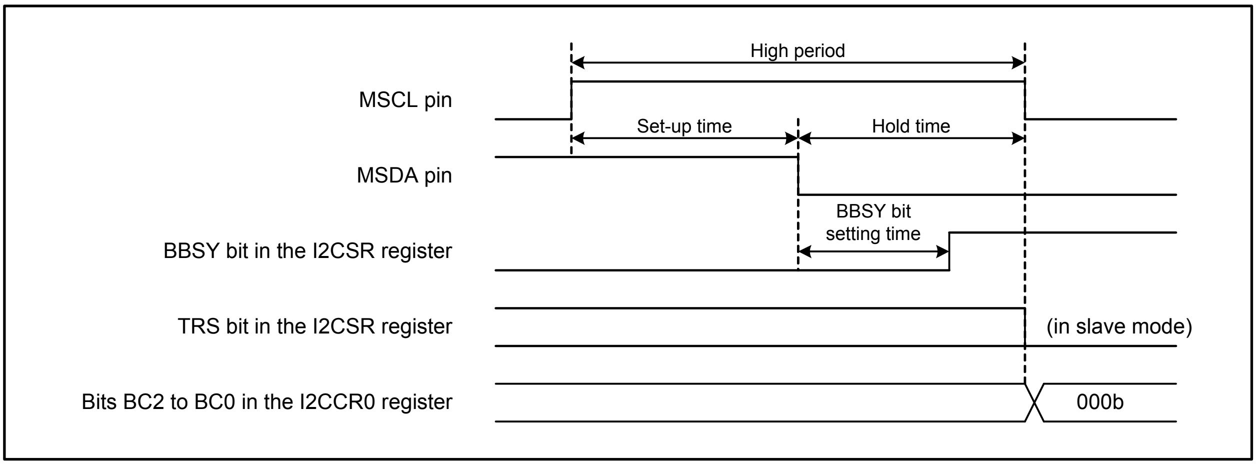

High period
MSCL pin
Set-up time
Hold time
MSDA pin
BBSY bit setting time
BBSY bit in the I2CSR register
TRS bit in the I2CSR register
(in slave mode)
Bits BC2 to BC0 in the I2CCR0 register
000b

Figure 24.21 Detecting a START Condition

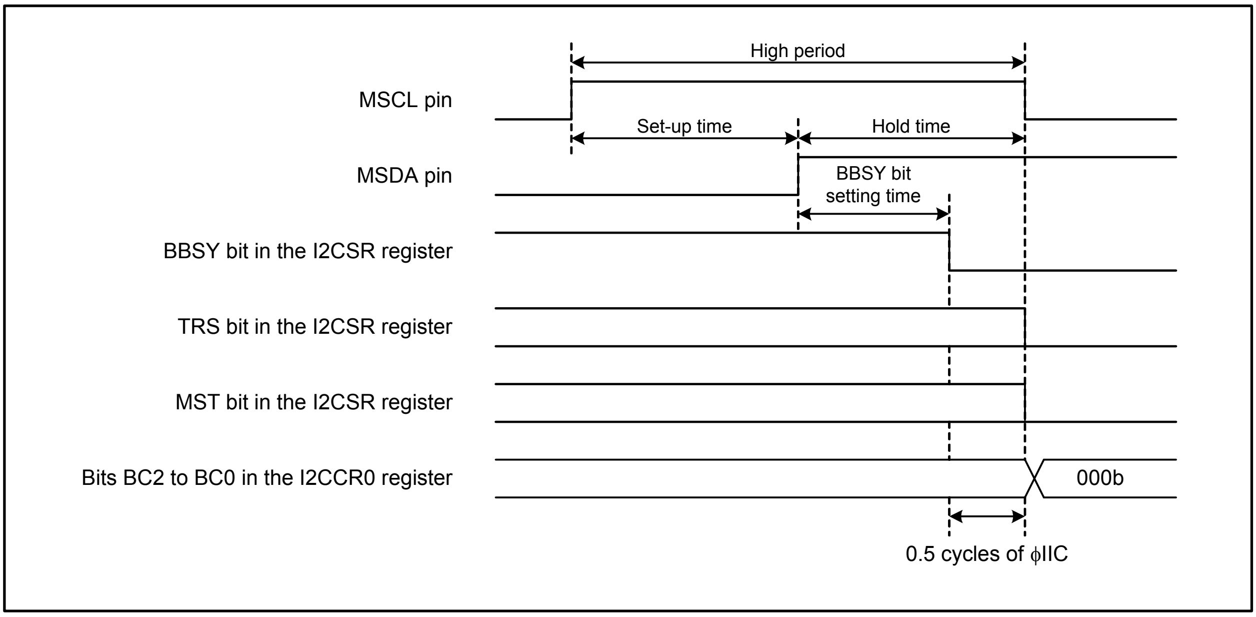

High period
MSCL pin
Set-up time
Hold time
MSDA pin
BBSY bit setting time
BBSY bit in the I2CSR register
TRS bit in the I2CSR register
MST bit in the I2CSR register
Bits BC2 to BC0 in the I2CCR0 register
000b
0.5 cycles of ϕIIC

Figure 24.22 Detecting a STOP Condition

Table 24.10 Parameters for Detecting START and STOP Conditions

Parameter	Standard-mode	Fast-mode
High period of MSCL pin	SSC value + 1 cycle (6.25 μs)	4 cycles (1.0 μs)
Set-up time	$\frac{\text{SSC value}}{2}$ + 1 cycle < 4.0 μs (3.25 μs)	2 cycles (0.5 μs)
Hold time	$\frac{\text{SSC value}}{2}$ cycles < 4.0 μs (3.0 μs)	2 cycles (0.5 μs)
BBSY bit set/reset time	$\frac{\text{SSC value - 1}}{2}$ + 2 cycles (3.375 μs)	3.5 cycles (0.875 μs)

Unit: ϕIIC cycles

SSC value: Setting value of bits SSC4 to SSC0 in the I2CSSCR register. Do not set these bits to 0 or an odd number.

Example times of when ϕIIC = 4 MHz and the I2CSSCR register = 18h are in parentheses.

Table 24.11 Recommended Values for Bits SSC4 to SSC0 in Standard-mode

ϕIIC	SSC Recommended Value	Parameters for Detecting START and STOP Conditions			BBSY Bit Set/Reset Time
		High period of MSCL pin	Set-up time	Hold time	
5 MHz	30	6.2 μs (31)	3.2 μs (16)	3.0 μs (15)	4.125 μs (16.5)
4 MHz	26	6.75 μs (27)	3.5 μs (14)	3.25 μs (13)	3.625 μs (14.5)
	24	6.25 μs (25)	3.25 μs (13)	3.0 μs (12)	3.375 μs (13.5)
2 MHz	12	6.5 μs (13)	3.5 μs (7)	3.0 μs (6)	3.75 μs (7.5)
	10	5.5 μs (11)	3.0 μs (6)	2.5 μs (5)	3.25 μs (6.5)
1 MHz	4	5.0 μs (5)	3.0 μs (3)	2.0 μs (2)	3.5 μs (3.5)

Number of ϕIIC cycles in parentheses.

SSC recommended values: Decimal value of bits SSC4 to SSC0 in the I2CSSCR register.

24.6 Data Transmission and Reception

Examples of the data transmission and reception format for master-transmission or slave-reception in a 7-bit address format are shown in section 24.6.1 “Master Transmission” and 24.6.2 “Slave Reception”. These examples assume communication starts after initialization using the parameters set in Table 24.12.

Table 24.12 Example of Initial Settings

Register	Setting Value	Parameter	Initial Setting
I2CSAR	02h	Slave address	1
I2CCCR	85h	SCL frequency	100 kHz (ϕIIC = 4 MHz)
		Clock mode	Standard-mode
		ACK clock generation	ACK clock generated
I2CCR2	00h	Timeout Detector	Disabled
I2CCR1	13h	STOP condition detection interrupt	Enabled
		Successful data receive interrupt	Enabled
		ϕIIC	fIIC divided-by-2
I2CSR	0Fh	Communication mode	Slave-receive mode
I2CSSCR	98h	SSC value (see Table 24.11)	24
		START and STOP conditions generation mode	Long mode
I2CCR0	08h	Number of bits to be transmitted or received	8 bits
		I^2C-bus interface	Enabled (communication enabled)
		Data format	Addressing format
I2CMR	09h	I^2C-bus interface/UART2	I^2C-bus interface selected
		I^2C-bus interface clock source	fIIC = f2n

24.6.1 Master Transmission

The operation and procedures of master transmission are described in this section. Figure 24.23 shows an example of master transmission operation. For (A) to (C) in the figure, see A to C in the descriptions and procedures below. (1) to (3) show the program's instructions. Arrows indicate that the procedure is performed by the MCU automatically.

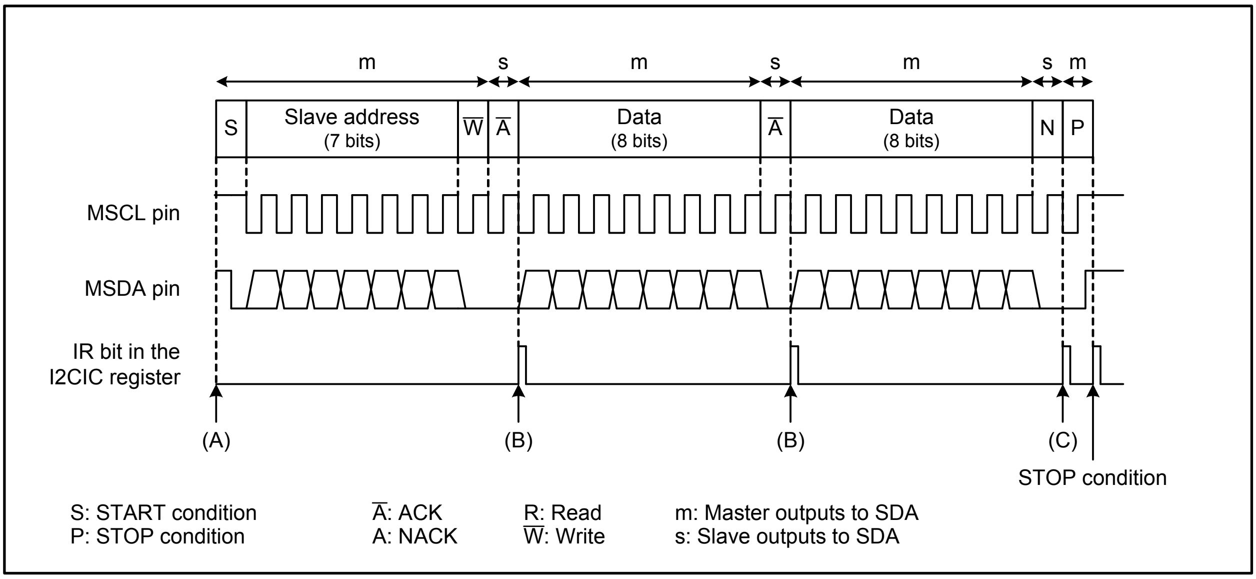


Figure 24.23 Example Operation of Master Transmission

A. Transmitting a slave address

(1) Confirm the BBSY bit in the I2CSR register is 0 (bus is free)
(2) Write E0h to the I2CSR register
→ The device enters the START condition standby state
(3) Write an address of a receiver (slave address) to the upper 7 bits of the I2CTRSR register
→ A START condition is generated
→ The slave address is sent

B. Transmitting data (processed in the I2C-bus interrupt routine)

(1) Write transmit data to the I2CTRSR register
→ Data is sent
To send multiple bytes of data, write them to the I2CTRSR register in succession

C. Completing master transmission (processed in the I2C-bus interrupt routine)

(1) Write C0h to the I2CSR register
→ The device enters the STOP condition standby state
(2) Write dummy data to the I2CTRSR register
→ A STOP condition is generated

In addition to the case where transmission is completed, procedure (C) is required when no ACK from the slave device is received (when a NACK is received as shown in Figure 24.23).

24.6.2 Slave Reception

The operation and procedures of slave reception are described in this section. Figure 24.24 shows an example of slave reception operation. For (A) to (D) in the figure, see A to D in the descriptions and procedures below. (1) to (3) show the program's instructions. Arrows indicate that the procedure is performed by the MCU automatically.

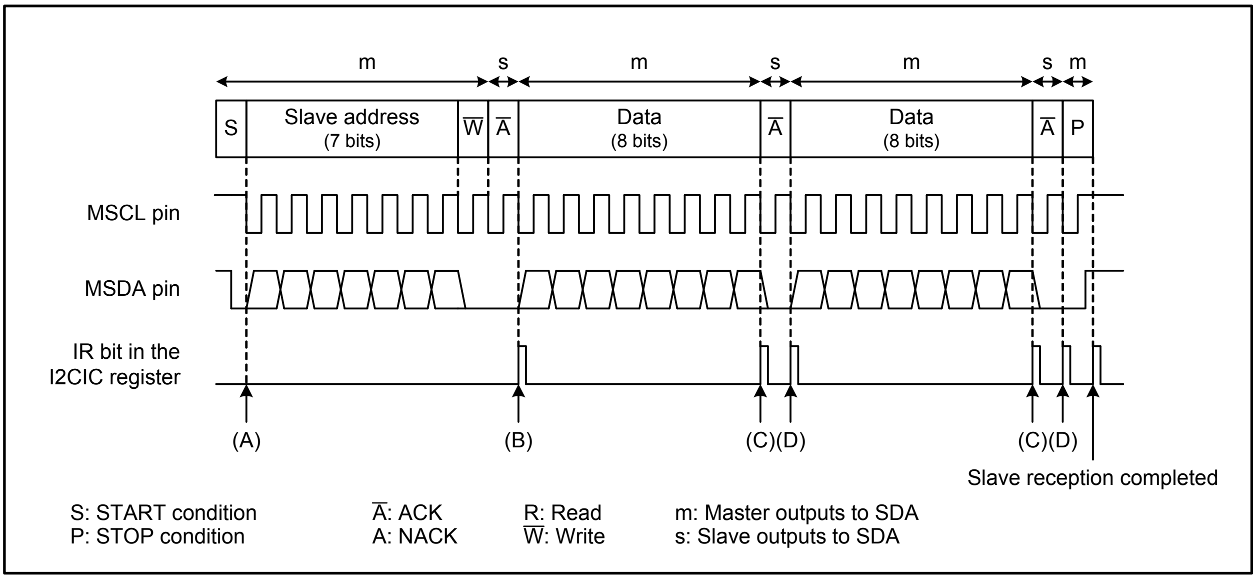


Figure 24.24 Example Operation of Slave Reception

A. Receiving a slave address (performed by the MCU automatically)

→ A START condition is detected

→ A slave address is received

→ An ACK is sent and the I^{2}C-bus interface interrupt is generated in either of the following cases

-When the general call address is received (the ADZ bit in the I2CSR register is 1)

-When an address match is detected (the AAS bit in the I2CSR register is 1)

B. Starting slave reception (processed in the I^{2}C-bus interrupt routine)

(1) Check the I2CSR register value. When the TRS bit is 0, start the slave reception.

(2) Write dummy data to the I2CTRSR register

→ Data reception starts

C. Completing slave reception (processed in the I^{2}C-bus interrupt routine)

(1) Read the received data from the I2CTRSR register

(2) Set the ACKD bit in the register to 1 (NACK) when the data is the last received data

(3) Set the ACKD bit in the register to 0 (ACK) when the data is not the last received data

→ An ACK or NACK is sent and an I^{2}C-bus interface interrupt is generated

D. Completing ACK transmission (processed in the I^{2}C-bus interrupt routine)

(1) Write dummy data to the I2CTRSR register

→ If the data is the last received data, a STOP condition is detected

→ If not, data reception restarts

24.7 Notes on Using Multi-master I2C-bus Interface

24.7.1 Accessing Multi-master I2C-bus Interface-associated Registers

Notes on writing to and reading I2C-bus interface-associated registers.

- I2CTRSR register

  Do not write to this register during data transmission or reception. Doing so resets the transmit/ receive counter and the register is unable to perform normal data transmission or reception.

- I2CCR0 register

  This register becomes 000b when a START condition is detected or 1 byte of data transmission or reception is completed. Do not write to or read this register at these two timings. Doing so may change the register value to an unexpected value. Figures 24.26 and 24.27 show the bit counter reset timings.

- I2CCCR register

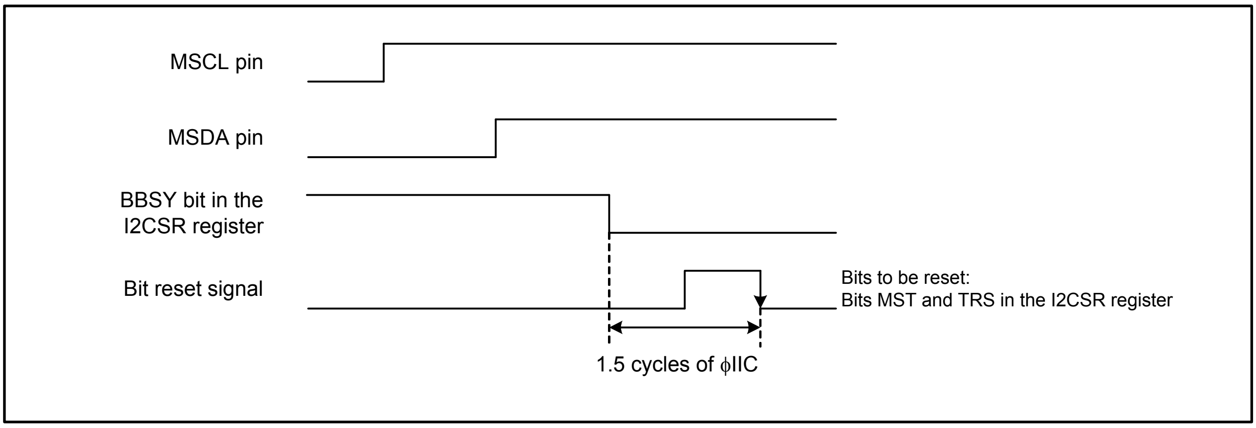
  Do not rewrite bits other than the ACKD bit during transmission or reception. Otherwise the I2C-bus clock circuit is reset and a normal transmission or reception will not be performed as a result.

- I2CCR1 register

  Rewrite bits ICK4 to ICK0 only when the ICE bit in the I2CCR0 register is 0 (I2C-bus interface disabled). When the I2CCR1 register is read, the internal WAIT flag status is read from this register. Therefore, do not use a bit processing instruction (read-modify-write instruction) with this register.

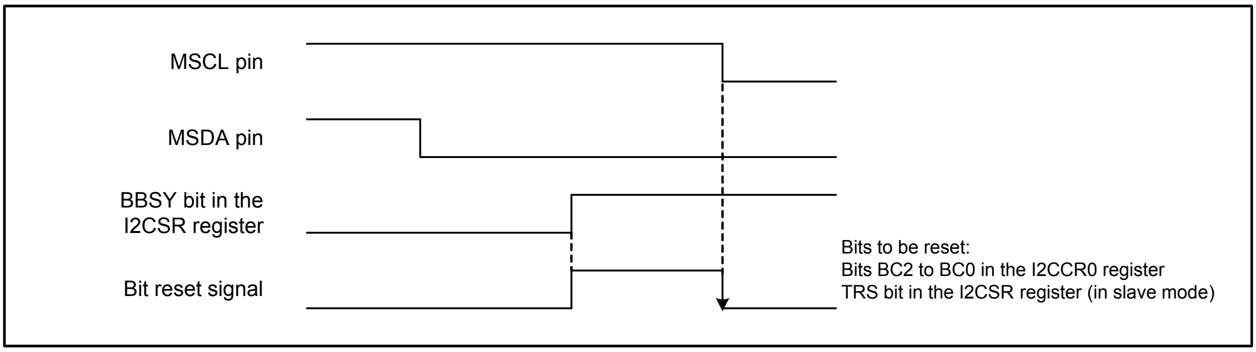
- I2CSR register

  Do not use a bit processing instruction (read-modify-write instruction) since the value of each bit in the I2CSR register changes depending on the communication state. Also, do not access this register when MST bit or TRS bit, which select the communication mode, changes. Doing so may change the register value to an unexpected value. Figures 24.25 to 24.27 show the timing of bits MST and TRS to change.



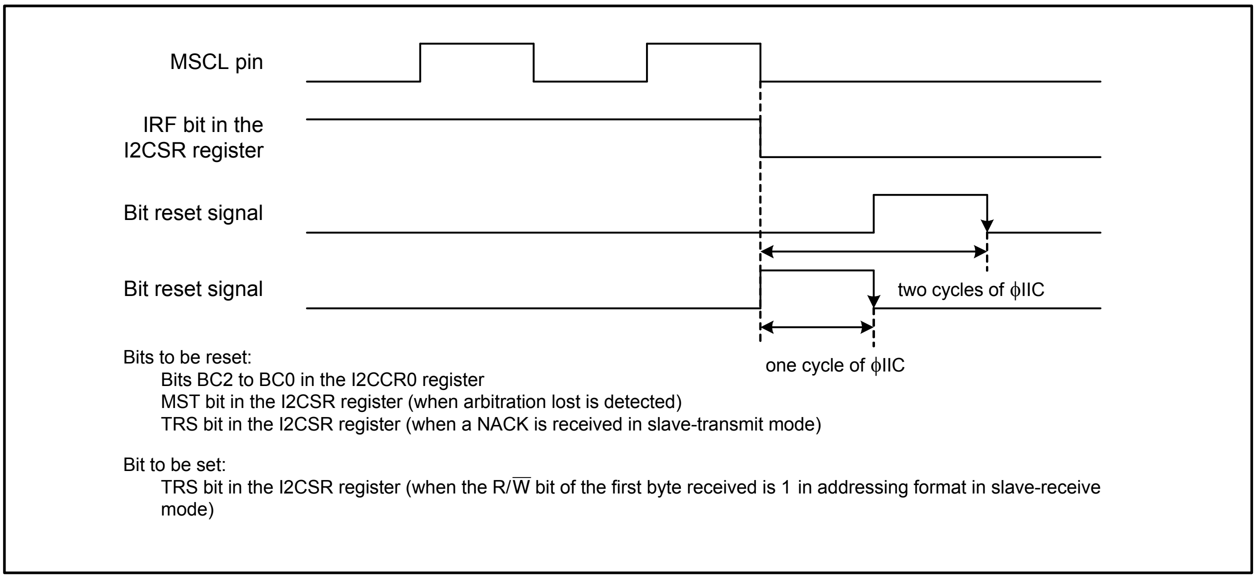
Bits to be reset:
Bits MST and TRS in the I2CSR register

Figure 24.25 Bit Resetting Timing (when a STOP condition is detected)



Bits to be reset:
Bits BC2 to BC0 in the I2CCR0 register
TRS bit in the I2CSR register (in slave mode)

Figure 24.26 Bit Resetting Timing (when a START condition is detected)



Bits to be reset:
Bits BC2 to BC0 in the I2CCR0 register
MST bit in the I2CSR register (when arbitration lost is detected)
TRS bit in the I2CSR register (when a NACK is received in slave-transmit mode)

Bit to be set:
TRS bit in the I2CSR register (when the R/$\overline{W}$ bit of the first byte received is 1 in addressing format in slave-receive mode)

Figure 24.27 Bit Setting/Resetting Timing (when data transmission/reception is completed)

24.7.2 Generating a Repeated START condition

Use the following steps to generate a repeated START condition after transmitting 1-byte of data:

(1) Write E0h (the START condition standby state, and the MSDA pin is high) to the I2CSR register
(2) Wait until the MSDA pin becomes high
(3) Write a slave address to the I2CTRSR register to generate a START condition trigger

Figure 24.28 shows the repeated START condition generating timing.

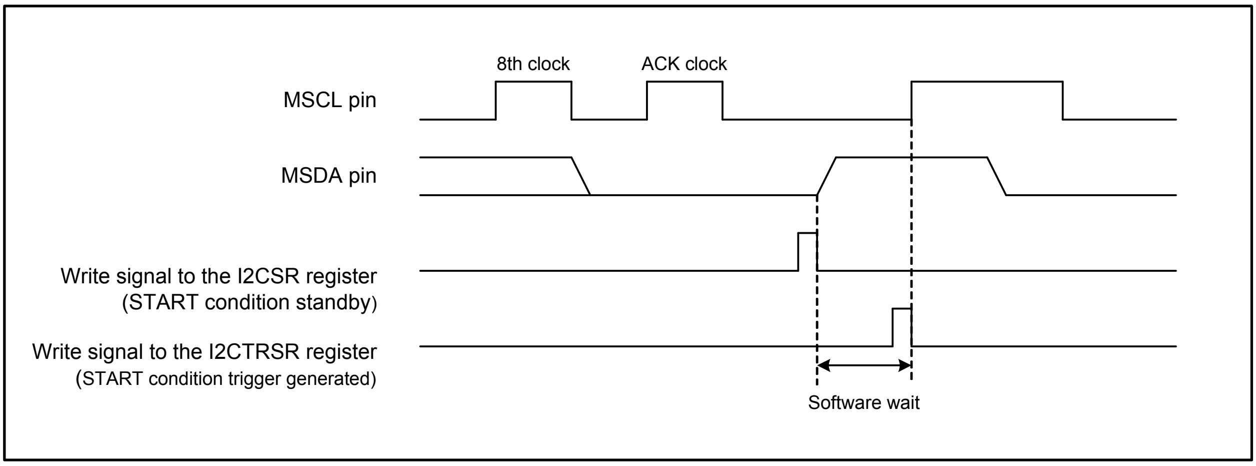

8th clock
ACK clock
MSCL pin
MSDA pin
Write signal to the I2CSR register
(START condition standby)
Write signal to the I2CTRSR register
(START condition trigger generated)
Software wait

Figure 24.28 Repeated START Condition Generating Timing

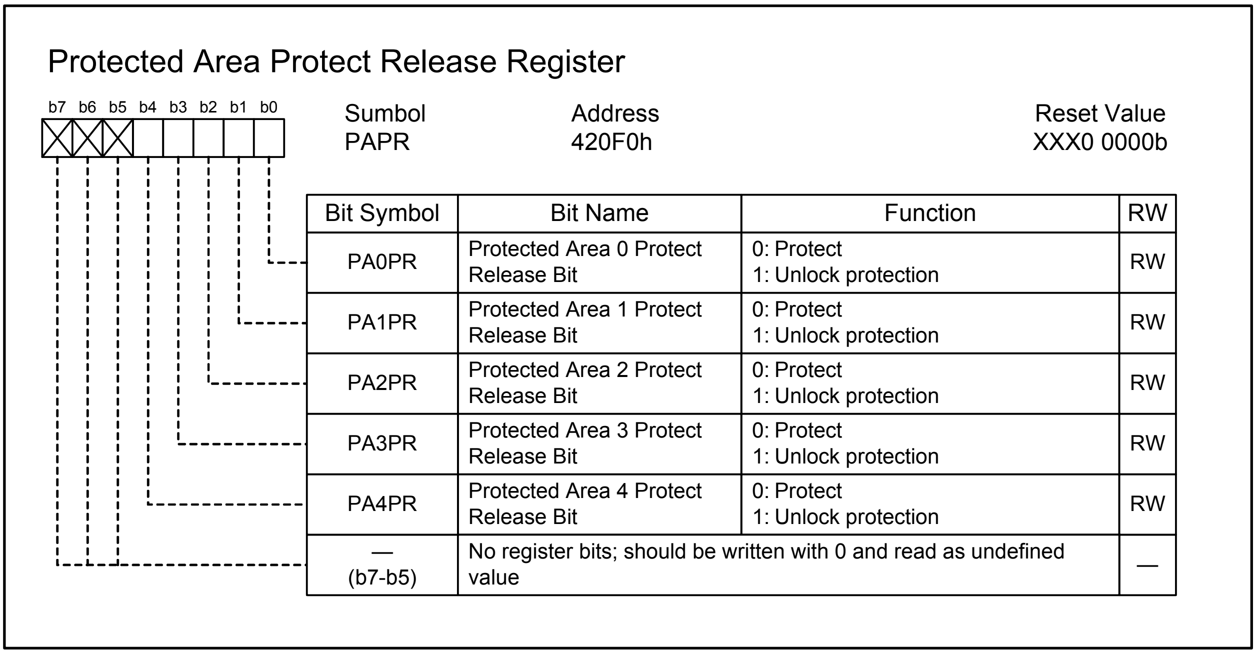
25. Protected Areas

The R32C/116A Group has five 32-bit areas protected from unexpected write accesses. The protected areas are randomly accessible as well as RAM. However, they are not rewritable while the protection is being locked.

Table 25.1 lists the location of protected areas and Figures 25.1 and 25.2 show the associated registers.

Table 25.1 Address Range of Protected Areas

Protected Area	Address Range
Protected Area 0	42000h to 4201Fh
Protected Area 1	42020h to 4203Fh
Protected Area 2	42040h to 4205Fh
Protected Area 3	42060h to 4207Fh
Protected Area 4	42080h to 4209Fh



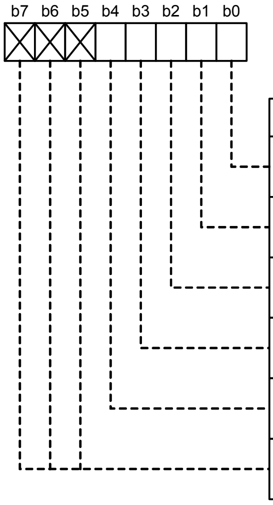
Protected Area Protect Release Register

Symbol PAPR
Address 420F0h
Reset Value XXX0 0000b

Bit Symbol	Bit Name	Function	RW
PA0PR	Protected Area 0 Protect Release Bit	0: Protect 1: Unlock protection	RW
PA1PR	Protected Area 1 Protect Release Bit	0: Protect 1: Unlock protection	RW
PA2PR	Protected Area 2 Protect Release Bit	0: Protect 1: Unlock protection	RW
PA3PR	Protected Area 3 Protect Release Bit	0: Protect 1: Unlock protection	RW
PA4PR	Protected Area 4 Protect Release Bit	0: Protect 1: Unlock protection	RW
— (b7-b5)	No register bits; should be written with 0 and read as undefined value		—

Figure 25.1 PAPR Register

Protected Area Write Access Flag Register



Symbol PAWF

Address 420F2h

Reset Value XXX0 0000b

Bit Symbol	Bit Name	Function	RW
PA0WF	Protected Area 0 Write Access Flag	0: No write access 1: Protected area is write accessed (1)	RW
PA1WF	Protected Area 1 Write Access Flag	0: No write access 1: Protected area is write accessed (1)	RW
PA2WF	Protected Area 2 Write Access Flag	0: No write access 1: Protected area is write accessed (1)	RW
PA3WF	Protected Area 3 Write Access Flag	0: No write access 1: Protected area is write accessed (1)	RW
PA4WF	Protected Area 4 Write Access Flag	0: No write access 1: Protected area is write accessed (1)	RW
— (b7-b5)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. To set the flag to 0, write this bit to 0. Writing 1 has no effect.

Figure 25.2 PAWF Register

26. I/O Pins

Each pin of the MCU functions as a programmable I/O port, an I/O pin for integrated peripherals, or a bus control pin. These functions can be switched by the function select registers or the processor mode registers. This chapter particularly addresses the function select registers. For the use as a bus control pin, refer to 7. “Processor Mode” and 9. “Bus”.
The pull-up resistors are enabled for every group of four pins. However, a pull-up resistor is separated from other peripherals even if it is enabled, when a pin functions as an output pin.
Figure 26.1 shows a block diagram of typical I/O pin.

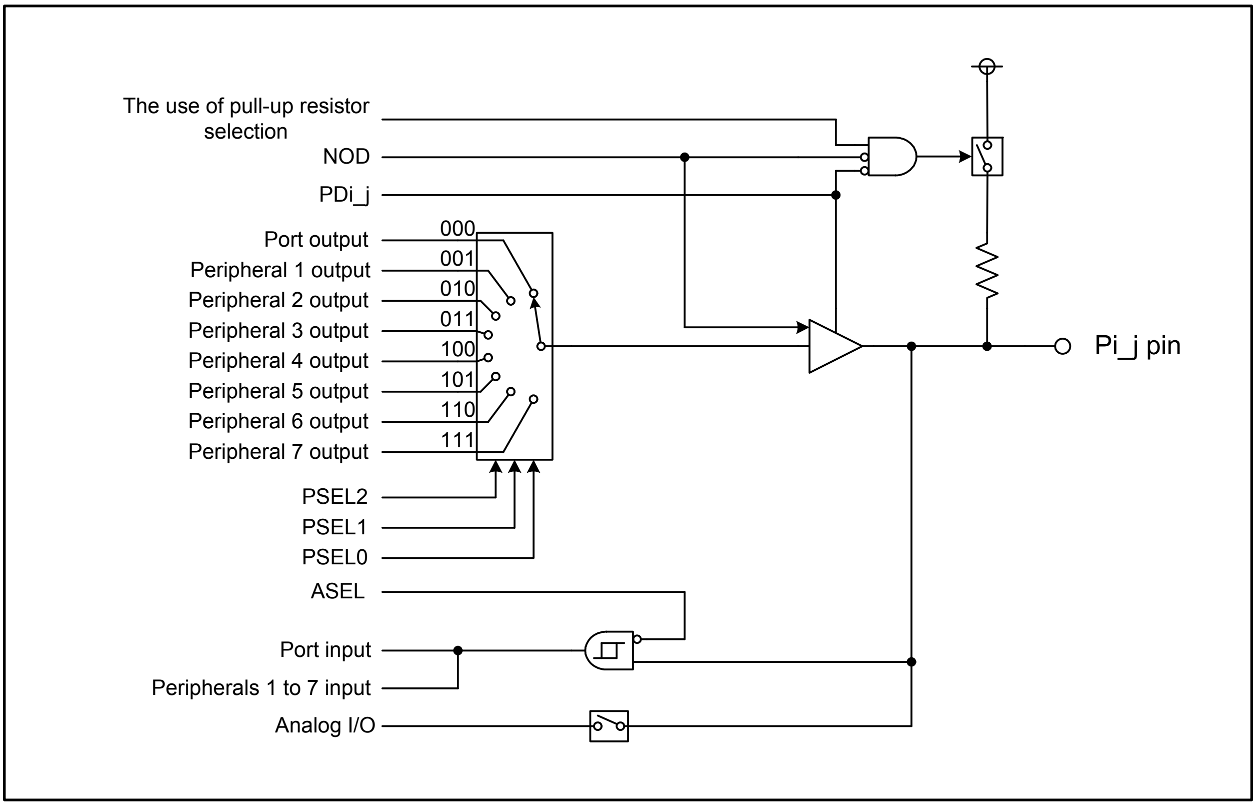


Figure 26.1 Typical I/O Pin Block Diagram (i = 0 to 19; j = 0 to 7)

The registers to control I/O pins are as follows: port Pi direction register (PDi register), output function select registers, and pull-up control registers. The PDi register selects the input or output state of pins. The output function select registers which select output function consist of bits PSEL2 to PSEL0, NOD, and ASEL. Bits PSEL2 to PSEL0 select a function as a programmable I/O or peripheral output (except analog output). The NOD bit selects the N-channel open drain output for a pin. The ASEL bit prevents the increase in power consumption of input buffer caused by an intermediate potential when a pin functions as an analog I/O pin. The pull-up control registers enable/disable the pull-up resistors.
To use a pin as an analog I/O pin, set the PDi_j bit to 0 (input), bits PSEL2 to PSEL0 to 000b, and the ASEL bit to 1.
The input-only port P8_5 shares a pin with $\overline{\text{NMI}}$ and has no function select register or bit 5 in the PD8 register. Port P14_1 also functions as an input-only port. The function select register and bit 1 in the PD14 register are reserved. Port P9 is protected from unexpected write accesses by the PRC2 bit in the PRCR register (refer to 10. “Protection”).

26.1 Port Pi Direction Register (PDi Register, i = 0 to 19)

The PDi register selects the input or output state of pins. Bits in this register correspond to respective pins.

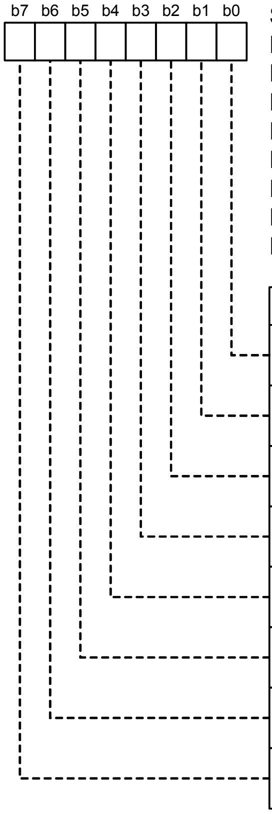
In memory expansion mode or microprocessor mode, this register cannot control pins being assigned bus control signals (A0 to A23, D0 to D31, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, WR/$\overline{\text{WR0}}$, $\overline{\text{BC0}}$, $\overline{\text{BC1}}$/$\overline{\text{WR1}}$, $\overline{\text{BC2}}$/$\overline{\text{WR2}}$, $\overline{\text{BC3}}$/$\overline{\text{WR3}}$, $\overline{\text{RD}}$, CLKOUT/BCLK, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$, ALE, and $\overline{\text{RDY}}$).

Figure 26.2 shows the PDi register.

No register bit is provided for port P8_5. For port P14_1, a reserved bit is provided.

The PD9 register is protected from unexpected write accesses by setting the PRC2 bit in the PRCR register (refer to 10. “Protection”).

Port Pi Direction Register (i = 0 to 19) (1)



Symbol	Address	Reset Value
PD0 to PD3	03C2h, 03C3h, 03C6h, 03C7h	0000 0000b
PD4 to PD7	03CAh, 03CBh, 03CEh, 03CFh	0000 0000b
PD8 (2)	03D2h	00X0 0000b
PD9 (3), PD10	03D3h, 03D6h	0000 0000b
PD11 (4) to PD13	03D7h, 03DAh, 03DBh	0000 0000b
PD14 (4, 5)	03DEh	0000 0000b
PD15	03DFh	0000 0000b
PD16 to PD19 (6)	03E2h, 03E3h, 03E6h, 03E7h	0000 0000b

Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi_0 Direction Bit (5)	0: Input port 1: Output port	RW
PDi_1	Port Pi_1 Direction Bit (5)	0: Input port 1: Output port	RW
PDi_2	Port Pi_2 Direction Bit (5)	0: Input port 1: Output port	RW
PDi_3	Port Pi_3 Direction Bit	0: Input port 1: Output port	RW
PDi_4	Port Pi_4 Direction Bit	0: Input port 1: Output port	RW
PDi_5	Port Pi_5 Direction Bit (2, 4)	0: Input port 1: Output port	RW
PDi_6	Port Pi_6 Direction Bit (4)	0: Input port 1: Output port	RW
PDi_7	Port Pi_7 Direction Bit (4)	0: Input port 1: Output port	RW

Notes:

1. In memory expansion mode or microprocessor mode, this register cannot control pins being assigned bus control signals (A0 to A23, D0 to D31, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, WR/$\overline{\text{WR0}}$, $\overline{\text{BC0}}$, $\overline{\text{BC1}}$/$\overline{\text{WR1}}$, $\overline{\text{BC2}}$/$\overline{\text{WR2}}$, $\overline{\text{BC3}}$/$\overline{\text{WR3}}$, $\overline{\text{RD}}$, CLKOUT/BCLK, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$, ALE, and $\overline{\text{RDY}}$).
2. The PD8_5 bit in the PD8 register is unavailable on this MCU. If necessary, set this bit to 0. The read value is undefined.
3. Set the PRC2 bit in the PRCR register to 1 (write enabled) just before rewriting the PD9 register. No interrupt handling or DMA transfers should be inserted between these two instructions.
4. Bits PD11_5 to PD11_7 in the PD11 register and the PD14_7 bit in the PD14 register should be rewritten after the PDWE bit in the PCR register is set to 1 (write enabled).
5. Bits PD14_0 to PD14_2 in the PD14 register are reserved. These bits should be written with 0.
6. In the 144-pin package, enabled bits in registers PD16 to PD19 should be written with 1 (output port).

Figure 26.2 Registers PD0 to PD19

26.2 Output Function Select Registers

When a programmable I/O port and peripheral output share a pin, these registers select the output function of the pin. Regardless of the register settings, signals are input to all the connected peripherals.

An output function select register consists of bits PSEL2 to PSEL0, NOD, and ASEL. Bits PSEL2 to PSEL0 select a function as programmable I/O or peripheral output (except analog output). The NOD bit selects the N-channel open drain output. The ASEL bit prevents the increase in power consumption caused by an intermediate potential generated when a pin functions as an analog I/O pin.

Table 26.1 shows the peripherals assigned to each PSEL2 to PSEL0 bit combination, and Figures 26.3 to 26.22 show the function select registers.

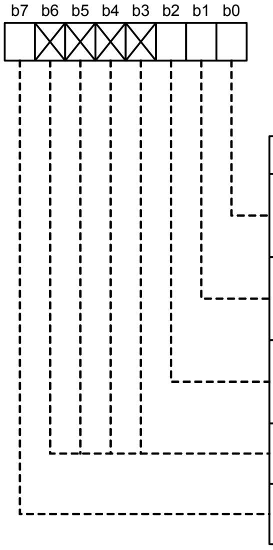
Note that ports P8_5 and P14_1 (input only) have no output function select registers.

The P9_iS register is protected from unexpected write accesses by setting the PRC2 bit in the PRCR register (refer to 10. “Protection”).

Table 26.1 Peripheral Assignment

Bits PSEL2 to PSEL0	Peripherals
001b	Timer
010b	Three-phase motor control timers
011b	UART
100b	UART special function
101b	Intelligent I/O groups 0 and 2
110b	Intelligent I/O group 1
111b	UART8

Port P0_i Function Select Register (i = 0 to 7)



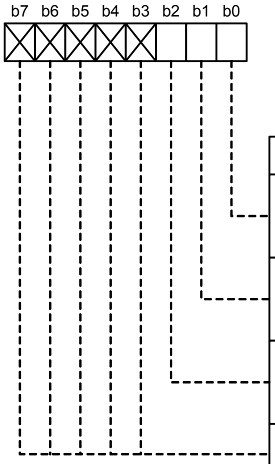
Symbol	Address	Reset Value
P0_0S to P0_2S	400A0h, 400A2h, 400A4h	0XXX X000b
P0_3S to P0_5S	400A6h, 400A8h, 400AAh	0XXX X000b
P0_6S, P0_7S	400ACh, 400AEh	0XXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P0_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P0_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b6-b3)	No register bits; should be written with 0 and read as undefined value		—
ASEL	Port P0_i Analog Function Select Bit	0: Function other than AN0_i 1: AN0_i	RW

Figure 26.3 Registers P0_0S to P0_7S

Port P0_i shares a pin with the AN0_i input for the A/D converter (i = 0 to 7).
To use it as a programmable I/O port, set the P0_iS register to 00h. To use it as an A/D converter input pin, set this register to 80h and the PD0_i bit to 0 (port P0_i functions as an input port).

Port P1_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P1_0S to P1_2S	400A1h, 400A3h, 400A5h	XXXX X000b
P1_3S to P1_5S	400A7h, 400A9h, 400ABh	XXXX X000b
P1_6S, P1_7S	400ADh, 400AFh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P1_i Output Function Select Bit [1]	b2 b1 b0 0 0 0 : I/O port P1_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : IIO0_i output 1 1 0 : IIO1_i output 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P1_0	P1_0	— [2]	— [2]	— [2]	— [2]	IIO0_0 output	IIO1_0 output	— [2]
P1_1	P1_1	— [2]	— [2]	— [2]	— [2]	IIO0_1 output	IIO1_1 output	— [2]
P1_2	P1_2	— [2]	— [2]	— [2]	— [2]	IIO0_2 output	IIO1_2 output	— [2]
P1_3	P1_3	— [2]	— [2]	— [2]	— [2]	IIO0_3 output	IIO1_3 output	— [2]
P1_4	P1_4	— [2]	— [2]	— [2]	— [2]	IIO0_4 output	IIO1_4 output	— [2]
P1_5	P1_5	— [2]	— [2]	— [2]	— [2]	IIO0_5 output	IIO1_5 output	— [2]
P1_6	P1_6	— [2]	— [2]	— [2]	— [2]	IIO0_6 output	IIO1_6 output	— [2]
P1_7	P1_7	— [2]	— [2]	— [2]	— [2]	IIO0_7 output	IIO1_7 output	— [2]

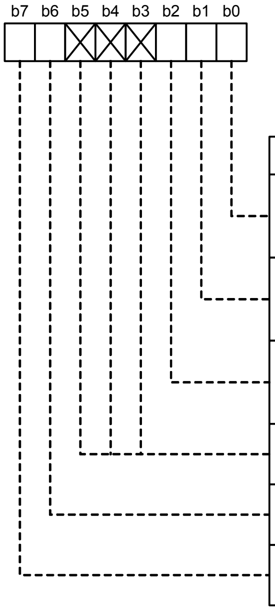
2. Do not use this combination.

Figure 26.4 Registers P1_0S to P1_7S

Port P1_i shares a pin with intelligent I/O groups 0 and 1 (IIO0 and IIO1) and the external interrupt inputs (i = 0 to 7).

To use it as an output pin, set the PD1_i bit to 1 (port P1_i functions as an output port) and select a function according to Figure 26.4. To use it as an input pin, set the PD1_i bit to 0 (port P1_i functions as an input port).

Port P2_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P2_0S to P2_2S	400B0h, 400B2h, 400B4h	00XX X000b
P2_3S to P2_5S	400B6h, 400B8h, 400BAh	00XX X000b
P2_6S, P2_7S	400BCh, 400BEh	00XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P2_i Output Function Select Bit [(1)]	b2 b1 b0 0 0 0 : I/O port P2_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART9/UART10 output 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
ASEL	Port P2_i Analog Function Select Bit	0: Function other than AN2_i 1: AN2_i	RW

Notes:

1. Refer to the table below for each pin setting.

Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P2_0	P2_0	— [(2)]	— [(2)]	$\overline{\text{RTS9}}$	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P2_1	P2_1	— [(2)]	— [(2)]	CLK9 output	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P2_2	P2_2	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P2_3	P2_3	— [(2)]	— [(2)]	TXD9	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P2_4	P2_4	— [(2)]	— [(2)]	$\overline{\text{RTS10}}$	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P2_5	P2_5	— [(2)]	— [(2)]	CLK10 output	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P2_6	P2_6	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P2_7	P2_7	— [(2)]	— [(2)]	TXD10	— [(2)]	— [(2)]	— [(2)]	— [(2)]

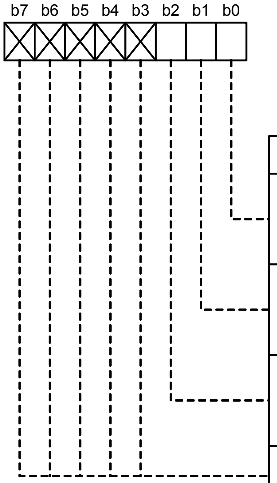
2. Do not use this combination.

Figure 26.5 Registers P2_0S to P2_7S

Port P2_i shares a pin with the serial interface (UART9 and UART10) and AN2_i for the A/D converter (i = 0 to 7).

To use it as an output pin, set the PD2_i bit to 1 (port P2_i functions as an output port) and select a function according to Figure 26.5. To use it as an input pin of functions other than the A/D converter, set the PD2_i bit to 0 (port P2_i functions as an input port). To use it as an A/D converter input pin, set this register to 80h and the PD2_i bit to 0 (port P2_i functions as an input port).

Port P3_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P3_0S to P3_2S	400B1h, 400B3h, 400B5h	XXXX X000b
P3_3S to P3_5S	400B7h, 400B9h, 400BBh	XXXX X000b
P3_6S, P3_7S	400BDh, 400BFh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P3_i Output Function Select Bit [(1)]	b2 b1 b0 0 0 0 : I/O port P3_i 0 0 1 : Timer output 0 1 0 : Three-phase motor control output 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

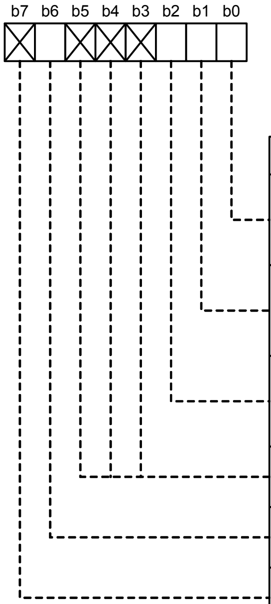
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P3_0	P3_0	TA0OUT output	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P3_1	P3_1	TA3OUT output	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P3_2	P3_2	TA1OUT output	V	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P3_3	P3_3	— [(2)]	$\overline{V}$	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P3_4	P3_4	TA2OUT output	W	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P3_5	P3_5	— [(2)]	$\overline{W}$	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P3_6	P3_6	TA4OUT output	U	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]
P3_7	P3_7	— [(2)]	$\overline{U}$	— [(2)]	— [(2)]	— [(2)]	— [(2)]	— [(2)]

2. Do not use this combination.

Figure 26.6 Registers P3_0S to P3_7S

Port P3_i shares a pin with the timer output and three-phase motor control output (i = 0 to 7).
To use it as an output pin, set the PD3_i bit to 1 (port P3_i functions as an output port) and select a function according to Figure 26.6. To use it as an input pin, set the PD3_i bit to 0 (port P3_i functions as an input port).

Port P4_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P4_0S to P4_2S	400C0h, 400C2h, 400C4h	X0XX X000b
P4_3S to P4_5S	400C6h, 400C8h, 400CAh	X0XX X000b
P4_6S, P4_7S	400CCh, 400CEh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P4_i Output Function Select Bit [1]	b2 b1 b0 0 0 0 : I/O port P4_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART3/UART6 output 1 0 0 : UART3/UART6 special function output 1 0 1 : IIO2 output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P4_0	P4_0	— [2]	— [2]	$\overline{\text{RTS3}}$	— [2]	— [2]	— [2]	— [2]
P4_1	P4_1	— [2]	— [2]	CLK3 output	— [2]	— [2]	— [2]	— [2]
P4_2	P4_2	— [2]	— [2]	SCL3 output	STXD3	— [2]	— [2]	— [2]
P4_3	P4_3	— [2]	— [2]	TXD3 SDA3 output	— [2]	OUTC2_0 ISTXD2 IEOUT	— [2]	— [2]
P4_4	P4_4	— [2]	— [2]	$\overline{\text{RTS6}}$	— [2]	— [2]	— [2]	— [2]
P4_5	P4_5	— [2]	— [2]	CLK6 output	— [2]	— [2]	— [2]	— [2]
P4_6	P4_6	— [2]	— [2]	SCL6 output	STXD6	— [2]	— [2]	— [2]
P4_7	P4_7	— [2]	— [2]	TXD6 SDA6 output	— [2]	— [2]	— [2]	— [2]

2. Do not use this combination.

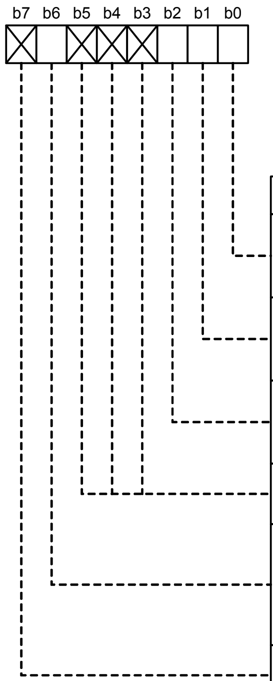
Figure 26.7 Registers P4_0S to P4_7S

Port P4_i shares a pin with the serial interface (UART3 and UART6) and intelligent I/O group 2 (IIO2) (i = 0 to 7).

To use it as an output pin, set the PD4_i bit to 1 (port P4_i functions as an output port) and select a function according to Figure 26.7. To use it as an input pin, set the PD4_i bit to 0 (port P4_i functions as an input port).

Ports P4_0 to P4_7 are 5 V tolerant inputs. To use them as I/O pins with 5 V tolerant input enabled, set the NOD bit to 1.

Port P5_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P5_0S, P5_1S	400C1h, 400C3h	XXXX X000b
P5_2S, P5_3S	400C5h, 400C7h	XXXX X000b
P5_4S, P5_5S	400C9h, 400CBh	X0XX X000b
P5_6S, P5_7S	400CDh, 400CFh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P5_i Output Function Select Bit [1]	b2 b1 b0 0 0 0 : I/O port P5_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART7 output 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
— (b6) (i = 0 to 3)	No register bit; should be written with 0 and read as undefined value		—
NOD (i = 4 to 7)	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

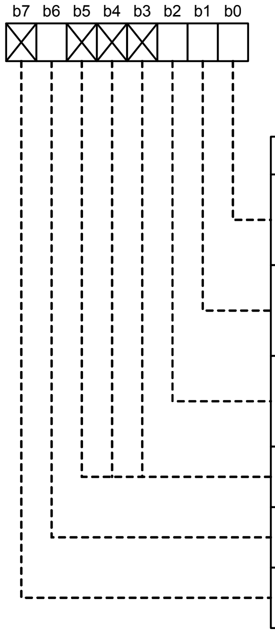
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P5_0	P5_0	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P5_1	P5_1	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P5_2	P5_2	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P5_3	P5_3	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P5_4	P5_4	— [2]	— [2]	TXD7	— [2]	— [2]	— [2]	— [2]
P5_5	P5_5	— [2]	— [2]	CLK7 output	— [2]	— [2]	— [2]	— [2]
P5_6	P5_6	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P5_7	P5_7	— [2]	— [2]	$\overline{\text{RTS7}}$	— [2]	— [2]	— [2]	— [2]

2. Do not use this combination.

Figure 26.8 Registers P5_0S to P5_7S

Port P5_i shares a pin with the serial interface (UART7) (i = 0 to 7).
To use it as an output pin, set the PD5_i bit to 1 (port P5_i functions as an output port) and select a function according to Figure 26.8. To use it as an input pin, set the PD5_i bit to 0 (port P5_i functions as an input port).
Ports P5_4 to P5_7 are 5 V tolerant inputs. To use them as I/O pins with 5 V tolerant input enabled, set the NOD bit to 1.

Port P6_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P6_0S to P6_2S	400D0h, 400D2h, 400D4h	X0XX X000b
P6_3S to P6_5S	400D6h, 400D8h, 400DAh	X0XX X000b
P6_6S, P6_7S	400DCh, 400DEh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P6_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P6_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART0/UART1 output 1 0 0 : UART0/UART1 special function output 1 0 1 : IIO2 output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

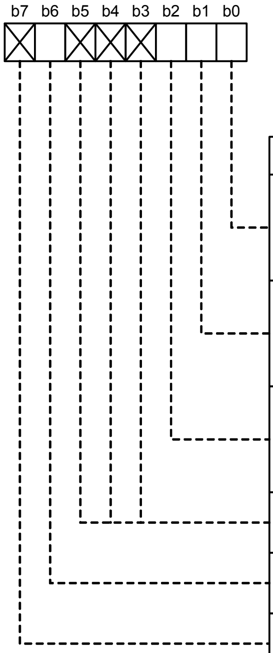
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P6_0	P6_0	— (2)	— (2)	$\overline{\text{RTS0}}$	— (2)	— (2)	— (2)	— (2)
P6_1	P6_1	— (2)	— (2)	CLK0 output	— (2)	— (2)	— (2)	— (2)
P6_2	P6_2	— (2)	— (2)	SCL0 output	STXD0	— (2)	— (2)	— (2)
P6_3	P6_3	— (2)	— (2)	TXD0 SDA0 output	— (2)	— (2)	— (2)	— (2)
P6_4	P6_4	— (2)	— (2)	$\overline{\text{RTS1}}$	— (2)	OUTC_1 ISCLK2 output	— (2)	— (2)
P6_5	P6_5	— (2)	— (2)	CLK1 output	— (2)	— (2)	— (2)	— (2)
P6_6	P6_6	— (2)	— (2)	SCL1 output	STXD1	— (2)	— (2)	— (2)
P6_7	P6_7	— (2)	— (2)	TXD1 SDA1 output	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 26.9 Registers P6_0S to P6_7S

Port P6_i shares a pin with the serial interface (UART0 and UART1) and intelligent I/O group 2 (IIO2) (i = 0 to 7).
To use it as an output pin, set the PD6_i bit to 1 (port P6_i functions as an output port) and select a function according to Figure 26.9. To use it as an input pin, set the PD6_i bit to 0 (port P6_i functions as an input port).
Ports P6_0 to P6_7 are 5 V tolerant inputs. To use them as I/O pins with 5 V tolerant input enabled, set the NOD bit to 1.

Port P7_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P7_0S to P7_2S	400D1h, 400D3h, 400D5h	X0XX X000b
P7_3S to P7_5S	400D7h, 400D9h, 400DBh	X0XX X000b
P7_6S, P7_7S	400DDh, 400DFh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P7_i Output Function Select Bit[1]	b2 b1 b0 0 0 0 : I/O port P7_i 0 0 1 : Timer output 0 1 0 : Three-phase motor control output 0 1 1 : UART2/UART5/MMI2C output 1 0 0 : UART2 special function output 1 0 1 : IIO2 output 1 1 0 : IIO1 output 1 1 1 : UART8 output	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P7_0	P7_0	TA0OUT output	—[2]	TXD2 SDA2 output MSDA output	—[2]	OUTC2_0 ISTXD2 IEOUT	IIO1_6 output	—[2]
P7_1	P7_1	—[2]	—[2]	SCL2 output MSCL output	STXD2	OUTC2_2	IIO1_7 output	—[2]
P7_2	P7_2	TA1OUT output	V	CLK2 output	—[2]	—[2]	—[2]	—[2]
P7_3	P7_3	—[2]	$\overline{V}$	$\overline{RTS2}$	—[2]	—[2]	IIO1_0 output	TXD8
P7_4	P7_4	TA2OUT output	W	—[2]	—[2]	—[2]	IIO1_1 output	CLK8 output
P7_5	P7_5	—[2]	$\overline{W}$	—[2]	—[2]	—[2]	IIO1_2 output	—[2]
P7_6	P7_6	TA3OUT output	—[2]	TXD5 SDA5 output	—[2]	—[2]	IIO1_3 output	$\overline{RTS8}$
P7_7	P7_7	—[2]	—[2]	CLK5 output	—[2]	—[2]	IIO1_4 output	—[2]

2. Do not use this combination.

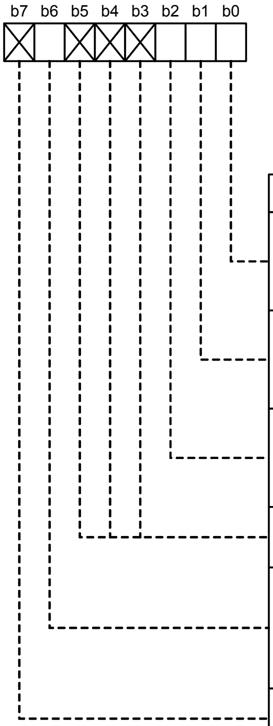
Figure 26.10 Registers P7_0S to P7_7S

Port P7_i shares a pin with the timer, three-phase motor control, serial interface (UART2, UART5, and UART8), multi-master I^2C-bus interface (MMI2C), and intelligent I/O groups 1 and 2 (IIO1 and IIO2) (i = 0 to 7).

To use it as an output pin, set the PD7_i bit to 1 (port P7_i functions as an output port) and select a function according to Figure 26.10. To use it as an input pin, set the PD7_i bit to 0 (port P7_i functions as an input port).

Ports P7_0 to P7_7 are 5 V tolerant inputs. To use them as I/O pins with 5 V tolerant input enabled, set the NOD bit to 1.

Port P8_i Function Select Register (i = 0 to 4, 6, 7)



Symbol	Address	Reset Value
P8_0S, P8_1S	400E0h, 400E2h	X0XX X000b
P8_2S, P8_3S	400E4h, 400E6h	X0XX X000b
P8_4S	400E8h	XXXX X000b
P8_6S, P8_7S	400ECh, 400EEh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P8_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P8_i 0 0 1 : Timer output 0 1 0 : Three-phase motor control output 0 1 1 : UART5 output 1 0 0 : UART5 special function output 1 0 1 : Do not use this combination 1 1 0 : IIO1 output 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD (i = 0 to 3)	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b6) (i = 4, 6, 7)	No register bit; should be written with 0 and read as undefined value		—
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

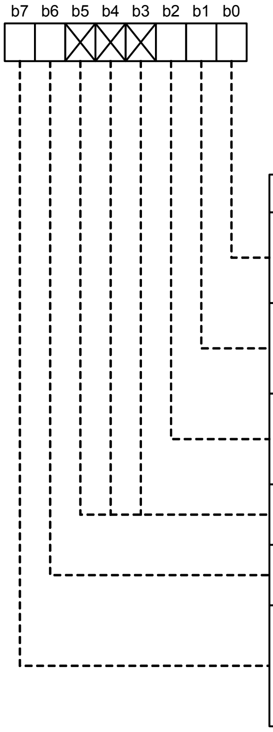
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P8_0	P8_0	TA4OUT output	U	SCL5 output	STXD5	— (2)	— (2)	— (2)
P8_1	P8_1	— (2)	$\overline{\mathrm{U}}$	$\overline{\mathrm{RTS5}}$	— (2)	— (2)	IIO1_5 output	— (2)
P8_2	P8_2	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P8_3	P8_3	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P8_4	P8_4	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P8_6	P8_6	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P8_7	P8_7	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 26.11 Registers P8_0S to P8_4S, P8_6S, and P8_7S

Port P8_i shares a pin with the timer, three-phase motor control, serial interface (UART5), intelligent I/O group 1 (IIO1), and external interrupt inputs (i = 0 to 4, 6, 7).
To use it as an output pin, set the PD8_i bit to 1 (port P8_i functions as an output port) and select a function according to Figure 26.11. To use it as an input pin, set the PD8_i bit to 0 (port P8_i functions as an input port).
Ports P8_0 to P8_3 are 5 V tolerant inputs. To use them as I/O pins with 5 V tolerant input enabled, set the NOD bit to 1.

Port P9_i Function Select Register (i = 0 to 7) (1)



Symbol	Address	Reset Value
P9_0S to P9_2S	400E1h, 400E3h, 400E5h	X0XX X000b
P9_3S to P9_5S	400E7h, 400E9h, 400EBh	00XX X000b
P9_6S	400EDh	00XX X000b
P9_7S	400EFh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P9_i Output Function Select Bit (2)	b2 b1 b0 0 0 0 : I/O port P9_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART3/UART4 output 1 0 0 : UART3/UART4 special function output 1 0 1 : IIO2 output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7) (i = 0 to 2, 7)	No register bit; should be written with 0 and read as undefined value		—
ASEL (i = 3 to 6)	Port P9_i (i = 3 to 6) Analog Functions Select Bit	0: Function other than Analog pin 1: Analog pin	RW

Notes:

1. Set the PRC2 bit in the PRCR register to 1 (write enabled) just before rewriting this register. No interrupt handling or DMA transfers should be inserted between these two instructions.
2. Refer to the following table for each pin setting.

Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P9_0	P9_0	— (3)	— (3)	CLK3 output	— (3)	— (3)	— (3)	— (3)
P9_1	P9_1	— (3)	— (3)	SCL3 output	STXD3	— (3)	— (3)	— (3)
P9_2	P9_2	— (3)	— (3)	TXD3 SDA3 output	— (3)	OUTC2_0 ISTXD2 IEOUT	— (3)	— (3)
P9_3	P9_3	— (3)	— (3)	$\overline{\text{RTS3}}$	— (3)	— (3)	— (3)	— (3)
P9_4	P9_4	— (3)	— (3)	$\overline{\text{RTS4}}$	— (3)	— (3)	— (3)	— (3)
P9_5	P9_5	— (3)	— (3)	CLK4 output	— (3)	— (3)	— (3)	— (3)
P9_6	P9_6	— (3)	— (3)	TXD4 SDA4 output	— (3)	— (3)	— (3)	— (3)
P9_7	P9_7	— (3)	— (3)	SCL4 output	STXD4	— (3)	— (3)	— (3)

3. Do not use this combination.

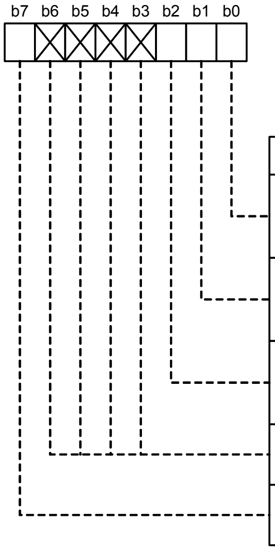
Figure 26.12 Registers P9_0S to P9_7S

Port P9_i shares a pin with the serial interface (UART3 and UART4) and intelligent I/O group 2 (IIO2) (i = 0 to 7). Ports P9_3 to P9_6 also share a pin with the A/D converter I/O (ANEX0 and ANEX1) and D/A converter output.

To use it as the A/D converter pin or the D/A converter pin, set the P9_iS register to 80h and the PD9_i bit to 0 (port P9_i functions as an input port) irrespective of the I/O state.

To use it as an output pin for functions other than the A/D converter or the D/A converter, set the PD9_i bit to 1 (port P9_i functions as an output port) and select a function according to Figure 26.12. To use it as an input pin of functions other than the A/D converter or the D/A converter, set the PD9_i bit to 0 (port P9_i functions as an input port).
When the NOD bit is set to 1, the corresponding pin functions as an N-channel open drain output.

Port P10_i Function Select Register (i = 0 to 7)



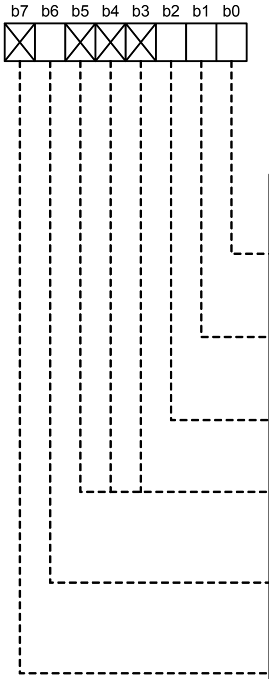
Symbol	Address	Reset Value
P10_0S to P10_2S	400F0h, 400F2h, 400F4h	0XXX X000b
P10_3S to P10_5S	400F6h, 400F8h, 400FAh	0XXX X000b
P10_6S, P10_7S	400FCh, 400FEh	0XXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P10_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P10_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b6-b3)	No register bits; should be written with 0 and read as undefined value		—
ASEL	Port P10_i Analog Functions Select Bit	0: Function other than AN_i 1: AN_i	RW

Figure 26.13 Registers P10_0S to P10_7S

Port P10_i shares a pin with the AN_i input for the A/D converter and key input interrupts (i = 0 to 7).
To use as it a programmable I/O port, set the P10_iS register to 00h. To use it as an input pin (except for the A/D converter), set the PD10_i bit to 0 (port P10_i functions as an input port). To use it as an input pin for the A/D converter, set the P10_iS register to 80h and the PD10_i bit to 0 (port P10_i functions as an input port).

Port P11_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P11_0S, P11_1S	400F1h, 400F3h,	X0XX X000b
P11_2S, P11_3S	400F5h, 400F7h	X0XX X000b
P11_4S, P11_5S	400F9h, 400FBh	XXXX X000b
P11_6S, P11_7S	400FDh, 400FFh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P11_i Output Function Select Bit [1]	b2 b1 b0 0 0 0 : I/O port P11_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART8 output 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : IIO1_i output 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD (i = 0 to 3)	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b6) (i = 4 to 7)	No register bit; should be written with 0 and read as undefined value		—
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table below for each pin setting.

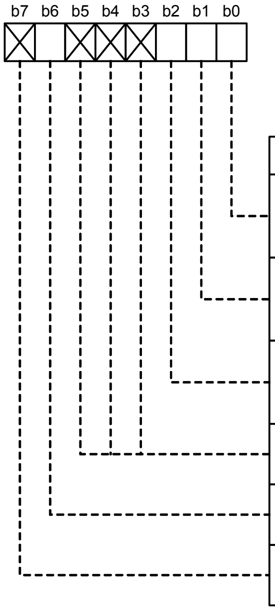
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P11_0	P11_0	— [2]	— [2]	TXD8	— [2]	— [2]	IIO1_0 output	— [2]
P11_1	P11_1	— [2]	— [2]	CLK8 output	— [2]	— [2]	IIO1_1 output	— [2]
P11_2	P11_2	— [2]	— [2]	— [2]	— [2]	— [2]	IIO1_2 output	— [2]
P11_3	P11_3	— [2]	— [2]	$\overline{\text{RTS8}}$	— [2]	— [2]	IIO1_3 output	— [2]
P11_4	P11_4	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P11_5	P11_5	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P11_6	P11_6	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P11_7	P11_7	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]

2. Do not use this combination.

Figure 26.14 Registers P11_0S to P11_7S

Port P11_i shares a pin with the serial interface (UART8) and intelligent I/O group 1 (IIO1) (i = 0 to 7).
To use it as an output pin, set the PD11_i bit to 1 (port P11_i functions as an output port) and select a function according to Figure 26.14. To use it as an input pin, set the PD11_i bit to 0 (port P11_i functions as an input port).
To use as an N-channel open drain output, set the NOD bit to 1.

Port P12_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P12_0S to P12_2S	40100h, 40102h, 40104h	X0XX X000b
P12_3S to P12_5S	40106h, 40108h, 4010Ah	X0XX X000b
P12_6S, P12_7S	4010Ch, 4010Eh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P12_i Output Function Select Bit [1]	b2 b1 b0 0 0 0 : I/O port P12_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART6 output 1 0 0 : UART6 special function 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

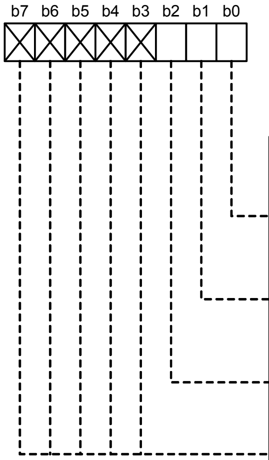
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P12_0	P12_0	— [2]	— [2]	TXD6 SDA6 output	— [2]	— [2]	— [2]	— [2]
P12_1	P12_1	— [2]	— [2]	CLK6 output	— [2]	— [2]	— [2]	— [2]
P12_2	P12_2	— [2]	— [2]	SCL6 output	STXD6	— [2]	— [2]	— [2]
P12_3	P12_3	— [2]	— [2]	$\overline{\text{RTS6}}$	— [2]	— [2]	— [2]	— [2]
P12_4	P12_4	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P12_5	P12_5	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P12_6	P12_6	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P12_7	P12_7	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]

2. Do not use this combination.

Figure 26.15 Registers P12_0S to P12_7S

Port P12_i shares a pin with the serial interface (UART6) (i = 0 to 7).
To use it as an output pin, set the PD12_i bit to 1 (port P12_i functions as an output port) and select a function according to Figure 26.15. To use it as an input pin, set the PD12_i bit to 0 (port P12_i functions as an input port).
Ports P12_0 to P12_7 are 5 V tolerant inputs. To use them as I/O pins with 5 V tolerant input enabled, set the NOD bit to 1.

Port P13_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P13_0S to P13_2S	40101h, 40103h, 40105h	XXXX X000b
P13_3S to P13_5S	40107h, 40109h, 4010Bh	XXXX X000b
P13_6S, P13_7S	4010Dh, 4010Fh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P13_i Output Function Select Bit [1]	b2 b1 b0 0 0 0 : I/O port P13_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : IIO2 output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

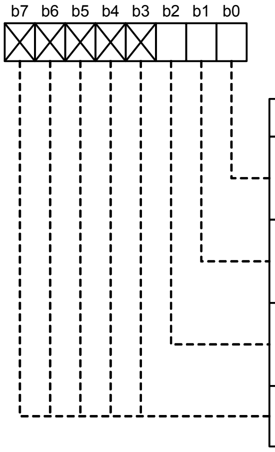
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P13_0	P13_0	— [2]	— [2]	— [2]	— [2]	OUTC2_4	— [2]	— [2]
P13_1	P13_1	— [2]	— [2]	— [2]	— [2]	OUTC2_5	— [2]	— [2]
P13_2	P13_2	— [2]	— [2]	— [2]	— [2]	OUTC2_6	— [2]	— [2]
P13_3	P13_3	— [2]	— [2]	— [2]	— [2]	OUTC2_3	— [2]	— [2]
P13_4	P13_4	— [2]	— [2]	— [2]	— [2]	OUTC2_0 ISTXD2 IEOUT	— [2]	— [2]
P13_5	P13_5	— [2]	— [2]	— [2]	— [2]	OUTC2_2	— [2]	— [2]
P13_6	P13_6	— [2]	— [2]	— [2]	— [2]	OUTC2_1 ISCLK2 output	— [2]	— [2]
P13_7	P13_7	— [2]	— [2]	— [2]	— [2]	OUTC2_7	— [2]	— [2]

2. Do not use this combination.

Figure 26.16 Registers P13_0S to P13_7S

Port P13_i shares a pin with intelligent I/O group 2 (IIO2) (i = 0 to 7).
To use it as an output pin, set the PD13_i bit to 1 (port P13_i functions as an output port) and select a function according to Figure 26.16. To use it as an input pin, set the PD13_i bit to 0 (port P13_i functions as an input port).

Port P14_i Function Select Register (i = 3 to 7)



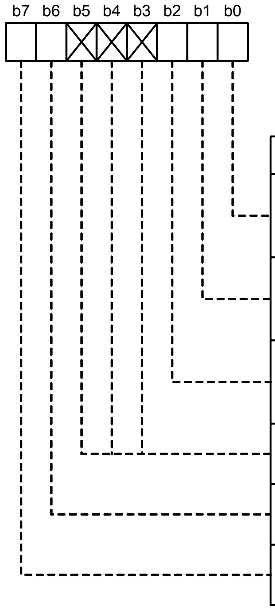
Symbol	Address	Reset Value
P14_3S to P14_5S	40116h, 40118h, 4011Ah	XXXX X000b
P14_6S, P14_7S	4011Ch, 4011Eh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P14_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P14_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Figure 26.17 Registers P14_3S to P14_7S

Port P14_i shares a pin with external interrupt inputs. Set the P14_iS register to 00h (I/O port) (i = 3 to 7).

Port P15_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P15_0S to P15_2S	40111h, 40113h, 40115h	00XX X000b
P15_3S to P15_5S	40117h, 40119h, 4011Bh	00XX X000b
P15_6S, P15_7S	4011Dh, 4011Fh	00XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P15_i Output Function Select Bit[1]	b2 b1 b0 0 0 0 : I/O port P15_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART6/UART7 output 1 0 0 : UART6 special function 1 0 1 : IIO0_i output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
ASEL	Port P15_i Analog Function Select Bit	0: Function other than AN15_i 1: AN15_i	RW

Notes:

1. Refer to the following table for each pin setting.

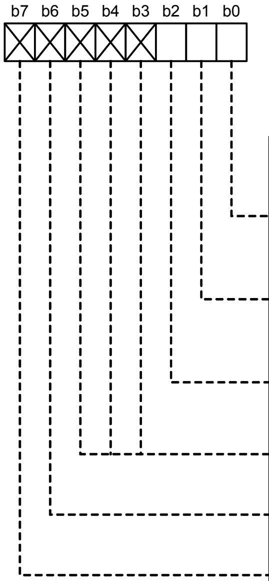
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P15_0	P15_0	—[2]	—[2]	TXD7	—[2]	IIO0_0 output	—[2]	—[2]
P15_1	P15_1	—[2]	—[2]	CLK7 output	—[2]	IIO0_1 output	—[2]	—[2]
P15_2	P15_2	—[2]	—[2]	—[2]	—[2]	IIO0_2 output	—[2]	—[2]
P15_3	P15_3	—[2]	—[2]	$\overline{\text{RTS7}}$	—[2]	IIO0_3 output	—[2]	—[2]
P15_4	P15_4	—[2]	—[2]	TXD6 SDA6 output	—[2]	IIO0_4 output	—[2]	—[2]
P15_5	P15_5	—[2]	—[2]	SCL6 output	STXD6	IIO0_5 output	—[2]	—[2]
P15_6	P15_6	—[2]	—[2]	CLK6 output	—[2]	IIO0_6 output	—[2]	—[2]
P15_7	P15_7	—[2]	—[2]	$\overline{\text{RTS6}}$	—[2]	IIO0_7 output	—[2]	—[2]

2. Do not use this combination.

Figure 26.18 Registers P15_0S to P15_7S

Port P15_i shares a pin with the serial interface (UART6 and UART7), intelligent I/O group 0 (IIO0), and AN15_i input for the A/D converter (i = 0 to 7).
To use it as an output pin, set the PD15_i bit to 1 (port P15_i functions as an output port) and select a function according to Figure 26.18. To use it as an input pin (except for the A/D converter), set the PD15_i bit to 0 (port P15_i functions as an input port). To use it as an input pin for the A/D converter, set the P15_iS register to 80h and the PD15_i bit to 0.
To use as an N-channel open drain output, set the NOD bit to 1.

Port P16_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P16_0S to P16_2S	40120h, 40122h, 40124h	XXXX X000b
P16_3S to P16_5S	40126h, 40128h, 4012Ah	XXXX X000b
P16_6S, P16_7S	4012Ch, 4012Eh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P16_i Output Function Select Bit [1]	b2 b1 b0 0 0 0 : I/O port P16_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART9/UART10 output 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting.

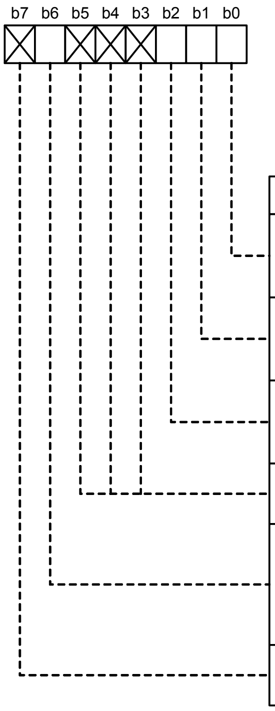
Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P16_0	P16_0	— [2]	— [2]	$\overline{\text{RTS9}}$	— [2]	— [2]	— [2]	— [2]
P16_1	P16_1	— [2]	— [2]	CLK9 output	— [2]	— [2]	— [2]	— [2]
P16_2	P16_2	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P16_3	P16_3	— [2]	— [2]	TXD9	— [2]	— [2]	— [2]	— [2]
P16_4	P16_4	— [2]	— [2]	$\overline{\text{RTS10}}$	— [2]	— [2]	— [2]	— [2]
P16_5	P16_5	— [2]	— [2]	CLK10 output	— [2]	— [2]	— [2]	— [2]
P16_6	P16_6	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]	— [2]
P16_7	P16_7	— [2]	— [2]	TXD10	— [2]	— [2]	— [2]	— [2]

2. Do not use this combination.

Figure 26.19 Registers P16_0S to P16_7S

Port P16_i shares a pin with the serial interface (UART9 and UART10) (i = 0 to 7).
To use it as an output pin, set the PD16_i bit to 1 (port P16_i functions as an output port) and select a function according to Figure 26.19. To use it as an input pin, set the PD16_i bit to 0 (port P16_i functions as an input port).
Ports P16_0 to P16_7 are 5 V tolerant inputs. To use them as I/O pins with 5 V tolerant input enabled, set the NOD bit to 1.

Port P17_i Function Select Register (i = 0 to 7)



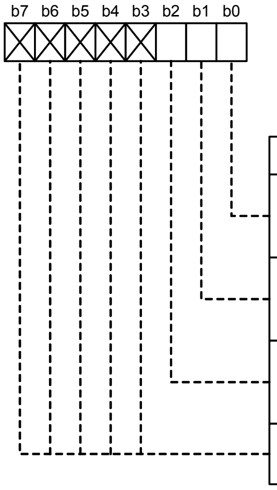
Symbol	Address	Reset Value
P17_0S, P17_1S	40121h, 40123h	X0XX X000b
P17_2S, P17_3S	40125h, 40127h	X0XX X000b
P17_4S, P17_5S	40129h, 4012Bh	XXXX X000b
P17_6S, P17_7S	4012Dh, 4012Fh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P17_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P17_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD (i = 0 to 3) — (b6) (i = 4 to 7)	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
	No register bits; should be written with 0 and read as undefined value		—
— (b7)	No register bits; should be written with 0 and read as undefined value		—

Figure 26.20 Registers P17_0S to P17_7S

Ports P17_0 to P17_3 are 5 V tolerant inputs. To use them as I/O pins with 5 V tolerant input enabled, set the NOD bit to 1.

Port P18_i Function Select Register (i = 0 to 7)

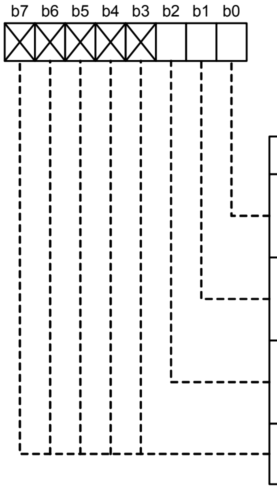


Symbol	Address	Reset Value
P18_0S to P18_2S	40130h, 40132h, 40134h	XXXX X000b
P18_3S to P18_5S	40136h, 40138h, 4013Ah	XXXX X000b
P18_6S, P18_7S	4013Ch, 4013Eh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P18_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P18_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Figure 26.21 Registers P18_0S to P18_7S

Port P19_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P19_0S to P19_2S	40131h, 40133h, 40135h	XXXX X000b
P19_3S to P19_5S	40137h, 40139h, 4013Bh	XXXX X000b
P19_6S, P19_7S	4013Dh, 4013Fh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P19_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P19_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Figure 26.22 Registers P19_0S to P19_7S

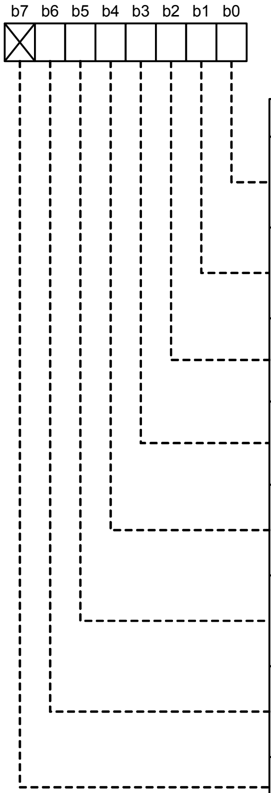
Any function other than the programmable I/O port is not assigned for ports P18 and P19. Do not set these registers to any values other than 00h.

26.3 Input Function Select Registers

When a peripheral input is assigned to multiple pins, these registers select which input pin should be connected to the peripheral.

Figures 26.23 to 26.25 show the input function select registers.

Input Function Select Register 0



Symbol IFS0 Address 40098h Reset Value X000 0000b

Bit Symbol	Bit Name	Function	RW
IFS00	Timer A Input Pin Switch Bit [1]	Assign timer A input to 0: Port P3 1: Port P7/port P8	RW
IFS01	Timer B Input Pin Switch Bit [2]	Assign timer B input to 0: Port P6 1: Port P9	RW
IFS02	UART6 Input Pin Switch Bit [3]	Assign UART6 input to b3 b2 0 0 : Port P4 0 1 : Do not use this combination 1 0 : Port P15 1 1 : Port P12	RW
IFS03			RW
IFS04	UART8 Input Pin Switch Bit [4]	Assign UART8 input to 0: Port P7 1: Port P11	RW
IFS05	UART7 Input Pin Switch Bit [5]	Assign UART7 input to 0: Port P5 1: Port P15	RW
IFS06	UART3 Input Pin Switch Bit [6]	Assign UART3 input to 0: Port P4 1: Port P9	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting of timer A.

IFS00	TA0OUT input	TA1OUT input	TA1IN	TA2OUT input	TA2IN	TA3OUT input	TA4OUT input	TA4IN
0	P3_0	P3_2	P3_3	P3_4	P3_5	P3_1	P3_6	P3_7
1	P7_0	P7_2	P7_3	P7_4	P7_5	P7_6	P8_0	P8_1

2. Refer to the following table for each pin setting of timer B.

IFS01	TB0IN	TB1IN	TB2IN
0	P6_0	P6_1	P6_2
1	P9_0	P9_1	P9_2

3. Refer to the following table for each pin setting of UART6.

IFS03	IFS02	SDA6 input/SRXD6	RXD6/SCL6 input	CLK6 input	CTS6/SS6
0	0	P4_7	P4_6	P4_5	P4_4
1	0	P15_4	P15_5	P15_6	P15_7
1	1	P12_0	P12_2	P12_1	P12_3

4. Refer to the following table for each pin setting of UART8.

IFS04	CLK8 input	RXD8	CTS8
0	P7_4	P7_5	P7_6
1	P11_1	P11_2	P11_3

5. Refer to the following table for each pin setting of UART7.

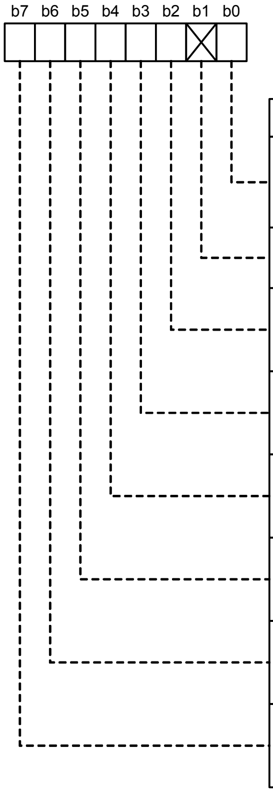
IFS05	CLK7 input	RXD7	CTS7
0	P5_5	P5_6	P5_7
1	P15_1	P15_2	P15_3

6. Refer to the following table for each pin setting of UART3.

IFS06	SDA3 input/SRXD3	RXD3/SCL3 input	CLK3 input	CTS3/SS3
0	P4_3	P4_2	P4_1	P4_0
1	P9_2	P9_1	P9_0	P9_3

Figure 26.23 IFS0 Register

Input Function Select Register 2



Symbol IFS2

Address 4009Ah

Reset Value 0000 00X0b

Bit Symbol	Bit Name	Function	RW
IFS20	Intelligent I/O Group 0 Input Pin Switch Bit [1]	Assign IIO0 input to 0: Port P1 1: Port P15	RW
— (b1)	No register bit; should be written with 0 and read as undefined value		—
IFS22	Intelligent I/O Group 0 Two-phase Pulse Input Pin Switch Bit [2]	Assign this input to b3 b2 0 0 : Port P8 and $\overline{\text{INT1}}$ 0 1 : Port P7 and $\overline{\text{INT0}}$ 1 0 : Port P3 and $\overline{\text{INT1}}$ 1 1 : Port P3 and $\overline{\text{INT0}}$	RW
IFS23			RW
IFS24	Intelligent I/O Group 1 Input Pin Switch Bit [3]	Assign IIO1 input to b5 b4 0 0 : Port P7/port P8 0 1 : Port P11 1 0 : Port P1 1 1 : Do not use this combination	RW
IFS25			RW
IFS26	Intelligent I/O Group 1 Two-phase Pulse Input Pin Switch Bit [4]	Assign this input to b7 b6 0 0 : Port P8 and $\overline{\text{INT1}}$ 0 1 : Port P7 and $\overline{\text{INT0}}$ 1 0 : Port P3 and $\overline{\text{INT1}}$ 1 1 : Port P3 and $\overline{\text{INT0}}$	RW
IFS27			RW

Notes:

1. Refer to the following table for each pin setting of intelligent I/O group 0.

IFS20	IIO0_0 input	IIO0_1 input	IIO0_2 input	IIO0_3 input	IIO0_4 input	IIO0_5 input	IIO0_6 input	IIO0_7 input
0	P1_0	P1_1	P1_2	P1_3	P1_4	P1_5	P1_6	P1_7
1	P15_0	P15_1	P15_2	P15_3	P15_4	P15_5	P15_6	P15_7

2. Refer to the following table for each pin setting of intelligent I/O group 0 in two-phase pulse signal processing mode.

IFS23	IFS22	UD0A	UD0B	UD0Z
0	0	P8_0	P8_1	P8_3 ($\overline{\text{INT1}}$)
0	1	P7_6	P7_7	P8_2 ($\overline{\text{INT0}}$)
1	0	P3_0	P3_1	P8_3 ($\overline{\text{INT1}}$)
1	1	P3_0	P3_1	P8_2 ($\overline{\text{INT0}}$)

3. Refer to the following table for each pin setting of intelligent I/O group 1.

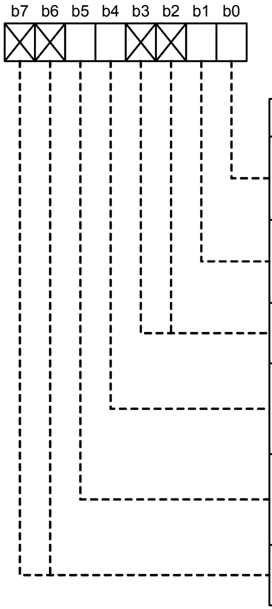
IFS25	IFS24	IIO1_0 input	IIO1_1 input	IIO1_2 input	IIO1_3 input	IIO1_4 input	IIO1_5 input	IIO1_6 input	IIO1_7 input
0	0	P7_3	P7_4	P7_5	P7_6	P7_7	P8_1	P7_0	P7_1
0	1	P11_0	P11_1	P11_2	P11_3	—	—	—	—
1	0	P1_0	P1_1	P1_2	P1_3	P1_4	P1_5	P1_6	P1_7

4. Refer to the following table for each pin setting of intelligent I/O group 1 in two-phase pulse signal processing mode.

IFS27	IFS26	UD1A	UD1B	UD1Z
0	0	P8_0	P8_1	P8_3 ($\overline{\text{INT1}}$)
0	1	P7_6	P7_7	P8_2 ($\overline{\text{INT0}}$)
1	0	P3_0	P3_1	P8_3 ($\overline{\text{INT1}}$)
1	1	P3_0	P3_1	P8_2 ($\overline{\text{INT0}}$)

Figure 26.24 IFS2 Register

Input Function Select Register 3



Symbol	Address	Reset Value
IFS3	4009Bh	XX00 XX00b

Bit Symbol	Bit Name	Function	RW
IFS30	Intelligent I/O Group 2 Input Pin Switch Bit (1)	Assign IIO2 input to b1 b0 0 0 : Port P6/port P7 0 1 : Port P6/port P9 1 0 : Port P13 1 1 : Port P6/port P4	RW
IFS31			RW
— (b3-b2)	No register bits; should be written with 0 and read as undefined value		—
IFS34	UART9 Input Pin Switch Bit (2)	Assign UART9 input to 0: Port P2 1: Port P16	RW
IFS35	UART10 Input Pin Switch Bit (3)	Assign UART10 input to 0: Port P2 1: Port P16	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the following table for each pin setting of intelligent I/O group 2.

IFS31	IFS30	ISCLK2 input	ISRXD2/IEIN
0	0	P6_4	P7_1
0	1	P6_4	P9_1
1	0	P13_6	P13_5
1	1	P6_4	P4_2

2. Refer to the following table for each pin setting of UART9. Set this bit to 0 in the 144-pin package.

IFS34	$\overline{\text{CTS9}}$	CLK9 input	RXD9
0	P2_0	P2_1	P2_2
1	P16_0	P16_1	P16_2

3. Refer to the following table for each pin setting of UART10. Set this bit to 0 in the 144-pin package.

IFS35	$\overline{\text{CTS10}}$	CLK10 input	RXD10
0	P2_4	P2_5	P2_6
1	P16_4	P16_5	P16_6

Figure 26.25 IFS3 Register

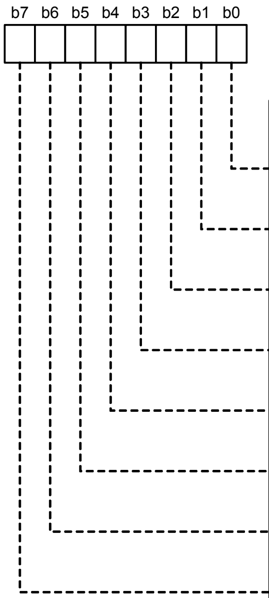
26.4 Pull-up Control Registers 0 to 5 (Registers PUR0 to PUR5)

Figures 26.26 to 26.31 show registers PUR0 to PUR5.

These registers enable/disable the pull-up resistors for every group of four pins. To enable the pull-up resistors, set the corresponding bits in registers PUR0 to PUR5 to 1 (pull-up resistor enabled) and the respective bits in the direction register to 0 (input).

In memory expansion mode or microprocessor mode, set 0 (pull-up resistor disabled) to the pull-up control bits for ports P0 to P5, P11, and P13, operating as bus control pins. The pull-up resistors are enabled for ports P0, P1, P11, and P13 when these pins function as input ports in these modes.

Pull-up Control Register 0 (1)



Symbol	Address	Reset Value
PUR0	03F0h	0000 0000b

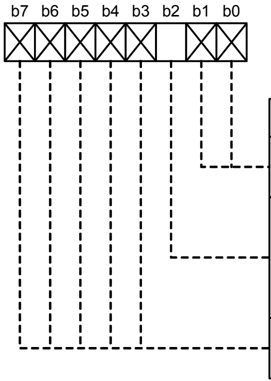
Bit Symbol	Bit Name	Function	RW
PU00	P0_0 to P0_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU01	P0_4 to P0_7 Pull-up Control Bit		RW
PU02	P1_0 to P1_3 Pull-up Control Bit		RW
PU03	P1_4 to P1_7 Pull-up Control Bit		RW
PU04	P2_0 to P2_3 Pull-up Control Bit		RW
PU05	P2_4 to P2_7 Pull-up Control Bit		RW
PU06	P3_0 to P3_3 Pull-up Control Bit		RW
PU07	P3_4 to P3_7 Pull-up Control Bit		RW

Note:

1. In memory expansion mode or microprocessor mode, each bit in the PUR0 register should be set to 0 since ports P0 to P3 are used as bus control pins. However, the pull-up resistors are enabled for ports P0 and P1 when these pins function as I/O ports with 8-bit bus or multiplexed bus format.

Figure 26.26 PUR0 Register

Pull-up Control Register 1 (1)



Symbol	Address	Reset Value
PUR1	03F1h	XXXX X0XXb

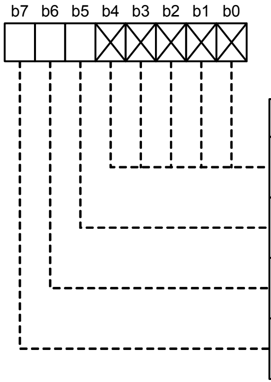
Bit Symbol	Bit Name	Function	RW
— (b1-b0)	No register bits; should be written with 0 and read as undefined value		—
PU12	P5_0 to P5_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. In memory expansion mode or microprocessor mode, each bit in the PUR1 register should be set to 0 since the port P5 functions as a bus control pin.

Figure 26.27 PUR1 Register

Pull-up Control Register 2



Symbol	Address	Reset Value
PUR2	03F2h	000X XXXXb

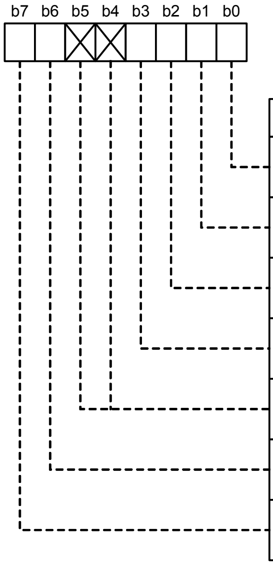
Bit Symbol	Bit Name	Function	RW
— (b4-b0)	No register bits; should be written with 0 and read as undefined value		—
PU25	P8_4 to P8_7 Pull-up Control Bit (1)	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU26	P9_0 to P9_3 Pull-up Control Bit		RW
PU27	P9_4 to P9_7 Pull-up Control Bit		RW

Note:

1. Port P8_5 has no pull-up resistor.

Figure 26.28 PUR2 Register

Pull-up Control Register 3



Symbol PUR3 | Address 03F3h | Reset Value 00XX 0000b

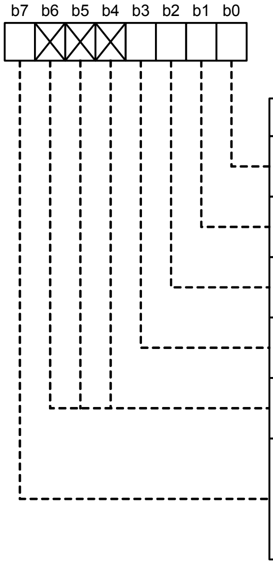
Bit Symbol	Bit Name	Function	RW
PU30	P10_0 to P10_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU31	P10_4 to P10_7 Pull-up Control Bit		RW
PU32	P11_0 to P11_3 Pull-up Control Bit [1]		RW
PU33	P11_4 to P11_7 Pull-up Control Bit [1]		RW
— (b5-b4)	No register bits; should be written with 0 and read as undefined value		—
PU36	P13_0 to P13_3 Pull-up Control Bit [1]	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU37	P13_4 to P13_7 Pull-up Control Bit [1]		RW

Note:

1. In memory expansion mode or microprocessor mode, bits PU32 to PU37 should be set to 0 since ports P11 and P13 function as bus control pins. However, the pull-up resistors are enabled for ports P11 and P13 when these pins function as I/O ports with 8-/16-bit bus or multiplexed bus format.

Figure 26.29 PUR3 Register

Pull-up Control Register 4



Symbol PUR4 | Address 03F4h | Reset Value 0XXX 0000b

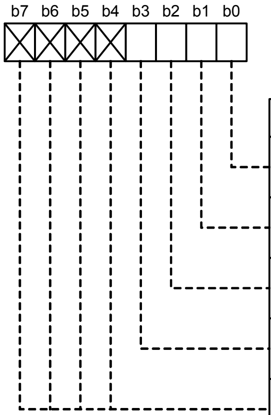
Bit Symbol	Bit Name	Function	RW
PU40	P14_1 and P14_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU41	P14_4 to P14_7 Pull-up Control Bit		RW
PU42	P15_0 to P15_3 Pull-up Control Bit		RW
PU43	P15_4 to P15_7 Pull-up Control Bit		RW
— (b6-b4)	No register bits; should be written with 0 and read as undefined value		—
PU47	P17_4 to P17_7 Pull-up Control Bit [1]	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW RW

Note:

1. Port P17 is not provided in the 144-package. The PU47 bit should be set to 0.

Figure 26.30 PUR4 Register

Pull-up Control Register 5 [1]



Symbol	Address	Reset Value
PUR5	03F5h	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
PU50	P18_0 to P18_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU51	P18_4 to P18_7 Pull-up Control Bit		RW
PU52	P19_0 to P19_3 Pull-up Control Bit		RW
PU53	P19_4 to P19_7 Pull-up Control Bit		RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. This register should be set to 00h in the 144-pin package.

Figure 26.31 PUR5 Register

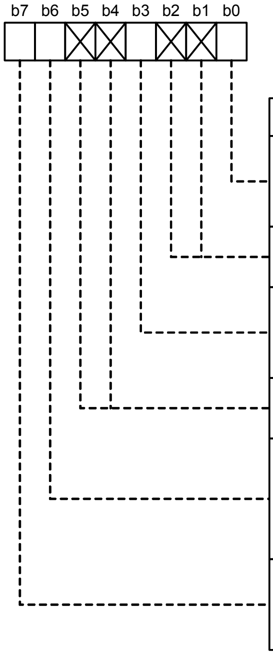
26.5 Port Control Register (PCR Register)

Figure 26.32 shows the PCR register.

This register selects an output mode for port P1 between push-pull output and pseudo-N-channel open drain output. When the PCR0 bit is set to 1, the P-channel transistor in the output buffer is turned off. Note that port P1 cannot be a perfect open drain output due to remaining parasitic diode. The absolute maximum rating of the input voltage is, therefore, -0.3 V to VCC + 0.3 V (refer to Figure 26.33).

In memory expansion mode or microprocessor mode, when port P1 is used for the data bus, the PCR0 bit should be set to 0. However, when port P1 is used as a programmable I/O port or an I/O pin for the peripheral functions, the output mode can be selected by setting the PCR0 bit even in these operating modes.

Port Control Register



Symbol	Address	Reset Value
PCR	03FFh	00XX 0XX0b (144-pin package) 10XX 0XX0b (176-pin package)

Bit Symbol	Bit Name	Function	RW
PCR0	Port P1 Output Format Control Bit [1]	0: Push-pull output 1: Pseudo-N-channel open drain output [2]	RW
— (b2-b1)	No register bits; should be written with 0 and read as undefined value		—
PRDS	Read Data Select Bit	In the output port, 0: Value in the port latch is read 1: Pin state is read	RW
— (b5-b4)	No register bits; should be written with 0 and read as undefined value		—
PDWE	PD11_5 to PD11_7, PD14_7 Write Enable Bit	0: Write disabled to bits PD11_5 to PD11_7, and PD14_7 1: Write enabled to bits PD11_5 to PD11_7, and PD14_7	RW
— (b7)	Reserved	0 should be held in the 144-pin package and 1 should be held in the 176-pin pakcage	RW

Notes:

1. In memory expansion mode or microprocessor mode, this bit should be set to 0 since port P1 is used for the data bus. However, when it is used as an I/O port or an I/O pin for the peripheral functions, the PCR0 bit can select an output format between push-pull output and pseudo-N-channel open drain output.
2. This function is designated not to make port P1 a full open drain, but to turn off the P-channel transistor in the CMOS output buffer. Therefore, the absolute maximum rating of the input voltage is -0.3 V to VCC + 0.3 V.

Figure 26.32 PCR Register

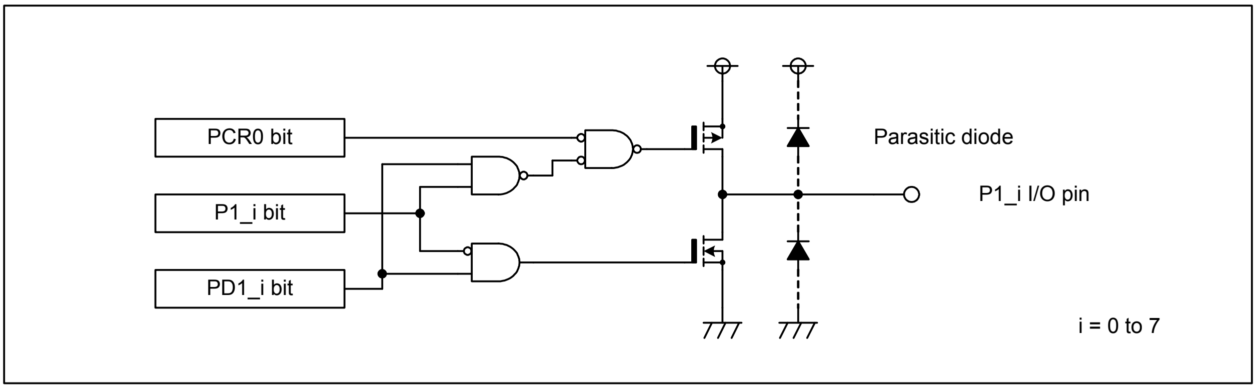


Figure 26.33 Port P1 Output Buffer Configuration

26.6 Configuring Unused Pins

Table 26.2, Table 26.3, and Figure 26.35 show examples of configuring unused pins on the board.

Table 26.2 Unused Pin Configuration in Single-chip Mode (1)

Pin Name	Setting
Ports P0 to P19 (excluding ports P8_5 and P14_1) (2, 3, 4)	Configure as input ports so that each pin is connected to VSS via its own resistor; (5) or configure as output ports to leave the pins open
P14_1	Connect the pin to VSS via a resistor (5)
XOUT (6)	Leave pin open
$\overline{\text{NMI}}$ (P8_5)	Connect the pin to VCC via a resistor (5)
AVCC	Connect the pin to VCC
AVSS, VREF	Connect the pin to VSS
NSD	Connect the pin to VCC via a resistor of 1 to 4.7 kΩ

Notes:

1. Unused pins should be wired within 2 cm of the MCU.
2. When configuring the pins as output ports to leave them open, note that ports as inputs remain unchanged from when the reset is released until the mode transition is completed. During this transition, the power supply current may increase due to an undefined voltage level of the pins. In addition, the direction register value may change due to noise or program runaway caused by the noise. To avoid these situations, reconfigure the direction register regularly by software, which may achieve higher program reliability.
3. Ports P16 to P19 are available in the 176-pin package only.
4. In the 144-pin package, set FFh to the following addresses: 03E2h, 03E3h, 03E6h, and 03E7h.
5. Select a resistance value that is appropriate for the system. A range from 10 to 100 kΩ is recommended.
6. This setting is applicable when an external clock is applied to the XIN pin.

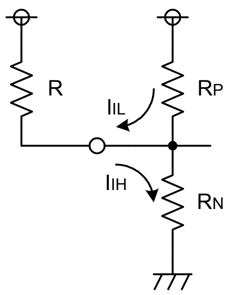
Table 26.3 Unused Pin Configuration in Memory Expansion Mode or Microprocessor Mode (1)

Pin Name	Setting
Ports P1, P6 to P19 (excluding ports P8_5 and P14_1) (2, 3, 4)	Configure as input ports so that each pin is connected to VSS via its own resistor; (5) or configure as output ports to leave the pins open
P14_1	Connect the pin to VSS via a resistor (5)
$\overline{\text{BC0}}$ to $\overline{\text{BC3}}$, $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$, ALE, $\overline{\text{HLDA}}$, XOUT (6), BCLK	Leave the pins open
$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$	Connect the pins to VCC via a resistor (5)
$\overline{\text{NMI}}$ (P8_5)	Connect the pin to VCC via a resistor (5)
AVCC	Connect the pin to VCC
AVSS, VREF	Connect the pins to VSS
NSD	Connect the pin to VCC via a resistor of 1 to 4.7 kΩ

Notes:

1. Unused pins should be wired within 2 cm of the MCU.
2. When configuring the pins as output ports to leave them open, note that ports as inputs remain unchanged from when the reset is released until the mode transition is completed. During this transition, the power supply current may increase due to an undefined voltage level of the pins. In addition, the direction register value may change due to noise or program runaway caused by the noise. To avoid these situations, reconfigure the direction register regularly by software, which may achieve higher program reliability.
3. Ports P16 to P19 are available in the 176-pin package only.
4. In the 144-pin package, set FFh to the following addresses: 03E2h, 03E3h, 03E6h, and 03E7h.
5. Select a resistance value that is appropriate for the system. A range from 10 to 100 kΩ is recommended.
6. This setting is applicable when an external clock is applied to the XIN pin.

Pull-up/pull-down resistors



The figure shows the equivalent circuit of an input pin.
The equivalent input resistors (RP and RN) are calculated using input power current (IIL and IIH).

Example: When VCC = 5.0 V, IIH = IIL = 5 μA,

$$R_P = R_N = \frac{5.0}{5 \times 10^{-6}} = 1\ \mathrm{M\Omega}$$

Since the voltage (VIH) defined as high is more than 0.8 VCC,
the resistance value R should satisfy the following expression:

$$R // R_P : R_N = 0.2 : 0.8$$

That is,

$$R = \frac{2R_PR_N}{8R_P - 2R_N}$$

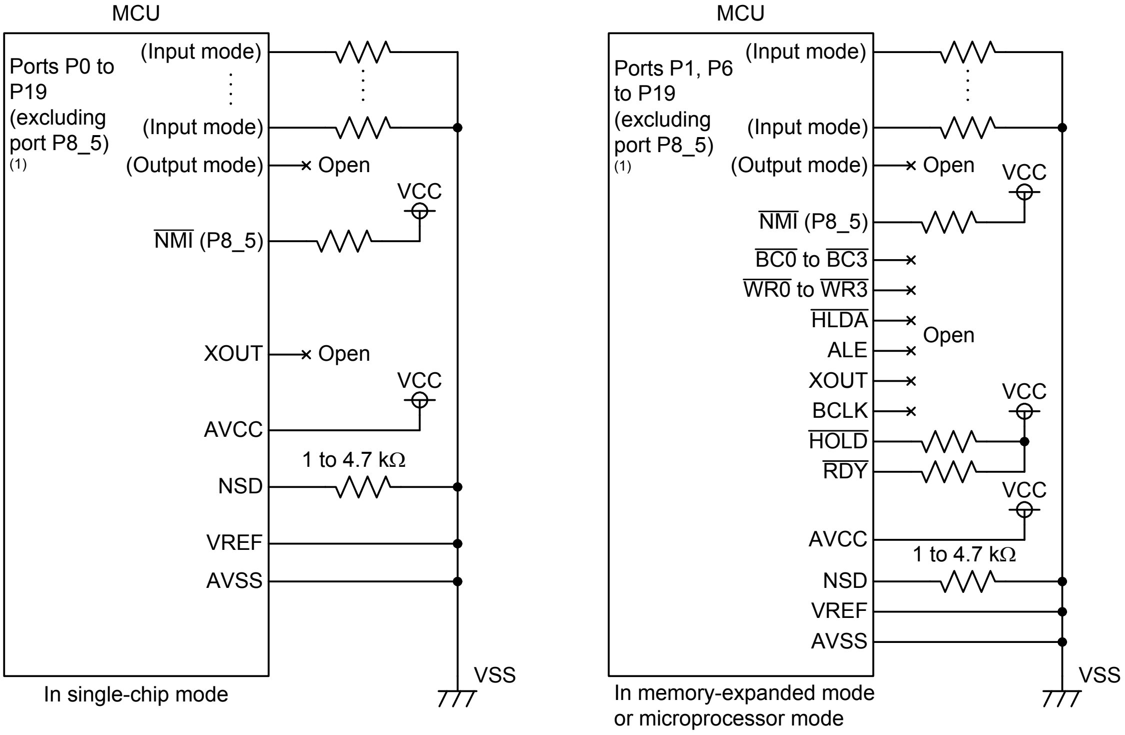
Specifically,

Example: When VCC = 5.0 V, IIH = IIL = 5 μA,

$$R = \frac{2 \times 10^6 \times 10^6}{8 \times 10^6 - 2 \times 10^6} = 333333$$

The maximum pull-up resistor R is approximately 330 kΩ.
The actual resistance value is the calculated value with some margins.

Figure 26.34 Pull-up/Pull-down Resistors



Note:

1. Ports P16 to P19 are provided in the 176-pin package only.

Figure 26.35 Unused Pin Configuration

27. Flash Memory

27.1 Overview

The flash memory can be programmed in the following three modes: CPU rewrite mode, standard serial I/O mode, and parallel I/O mode.
Table 27.1 lists specifications of the flash memory and Table 27.2 shows the overview of each rewrite mode.

Table 27.1 Flash Memory Specifications

Item	Specification
Rewrite modes	CPU rewrite mode, standard serial I/O mode, parallel I/O mode
Structure	Block architecture. Refer to Figure 27.1
Program operation	8-byte basis
Erase operation	1-block basis
Program and erase control method	Software commands
Protection types	Lock bit protect, ROM code protect, ID code protect
Software commands	9

Table 27.2 Flash Memory Rewrite Mode Overview

Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	CPU executes a software command to rewrite the flash memory EW0 mode: Rewritable in areas other than the on-chip flash memory EW1 mode: Rewritable in areas other than specified blocks to be rewritten	A dedicated serial programmer rewrites the flash memory Standard serial I/O mode 1: Synchronous serial I/O selected Standard serial I/O mode 2: UART selected	A dedicated parallel programmer rewrites the flash memory
CPU operating mode	Single-chip mode, Memory expansion mode (EW0 mode)	Standard serial I/O mode	Parallel I/O mode
Programmer	—	Serial programmer	Parallel programmer
On-board programming	Supported	Supported	Not supported

Figure 27.1 shows the on-chip flash memory structure.
The on-chip flash memory contains program area to store user programs, and data area/data flash to store the result of user programs. The program area consists of blocks 0 to 17, and data area/data flash consists of blocks A and B.
Each block can be individually protected (locked) from programming or erasing by setting the lock bit.

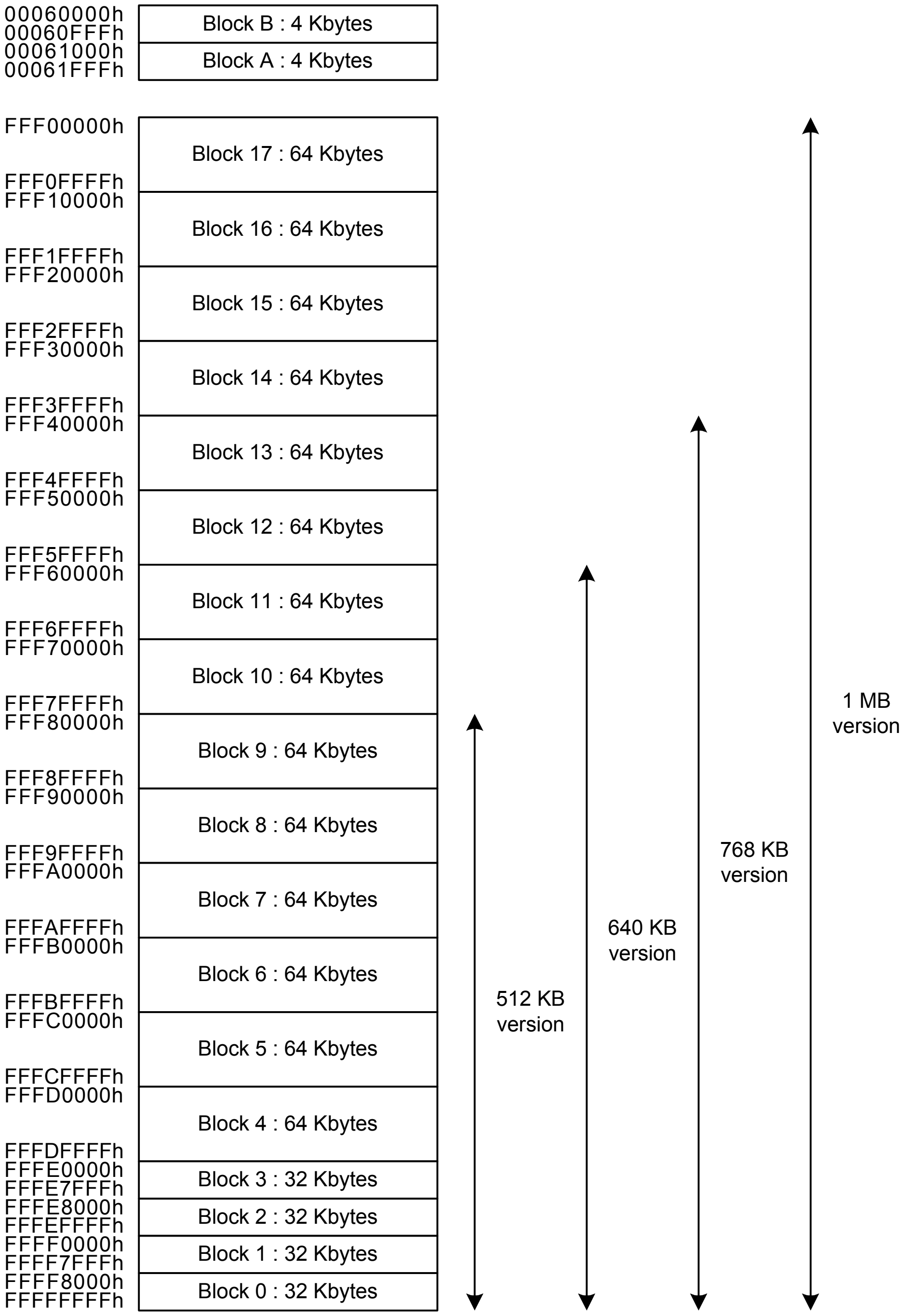


Figure 27.1 On-chip Flash Memory Block Diagram

27.2 Flash Memory Protection

There are three types of protection as shown in Table 27.3. Lock bit protection is intended to prevent accidental write or erase by program runaway. ROM code protection and ID code protection are intended to prevent read or write by a third party.

Table 27.3 Protection Types and Characteristics

Protection Type	Lock Bit Protection	ROM Code Protection	ID Code Protection
Protected operations	Erase, write	Read, write	Read, erase, write
Protection available in	CPU rewrite mode Standard serial I/O mode Parallel I/O mode	Parallel I/O mode	Standard serial I/O mode
Protection available for	Individual blocks	Entire flash memory	Entire flash memory
Protection settings	Setting 0 to the lock bit of block to be protected	Setting the protect bit of any block to 0	Writing the program which has set an ID code to specified address
Protection disabled by	Setting the LBD bit in the FMR register to 1 (lock bit protection disabled), or by erasing the blocks whose lock bits are set to 0 to permanently disable the protection	Erasing all blocks whose protect bits are set to 0	Sending a proper ID code from the serial programmer

27.2.1 Lock Bit Protection

This protection can be used in all three rewrite modes. When the lock bit protection is enabled, all blocks whose lock bits are set to 0 (locked) are protected against programming and erasing.

To set the lock bit to 0, the lock bit program command must be issued.

To temporarily disable the protection of all protected blocks, disable the lock bit protection itself by setting the LBD bit in the FMR1 register to 1 (lock bit protection disabled). The protection of a protected block is disabled permanently and its lock bit becomes 1 (unlocked) if the block is erased.

27.2.2 ROM Code Protection

This protection can only be used in parallel I/O mode. When the ROM code protection is enabled, the entire flash memory is protected against reading and writing.

To disable the protection, erase all the blocks whose protect bits are set to 0 (protected).

Each block has two protect bits. Setting any protect bit to 0 by a software command enables the protection for the entire flash memory. Table 27.4 lists protect bit addresses.

Table 27.4 **Protect Bit Addresses**

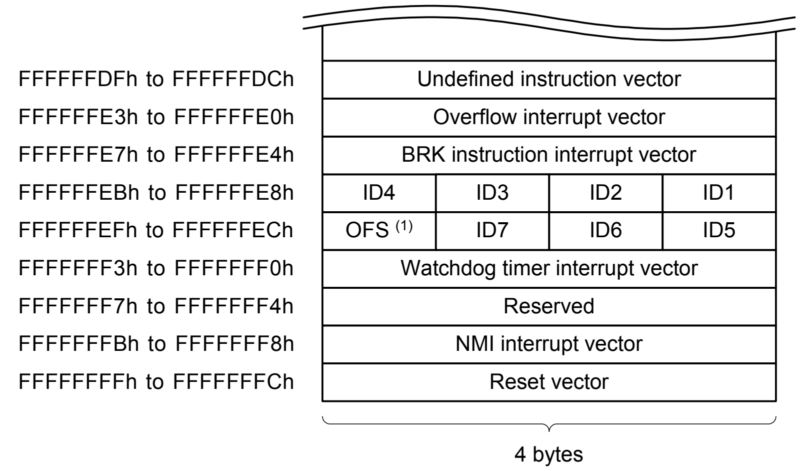
Block	Protect Bit 0	Protect Bit 1
Block B	00060100h	00060300h
Block A	00061100h	00061300h
Block 17	FFF00100h	FFF00300h
Block 16	FFF10100h	FFF10300h
Block 15	FFF20100h	FFF20300h
Block 14	FFF30100h	FFF30300h
Block 13	FFF40100h	FFF40300h
Block 12	FFF50100h	FFF50300h
Block 11	FFF60100h	FFF60300h
Block 10	FFF70100h	FFF70300h
Block 9	FFF80100h	FFF80300h
Block 8	FFF90100h	FFF90300h
Block 7	FFFA0100h	FFFA0300h
Block 6	FFFB0100h	FFFB0300h
Block 5	FFFC0100h	FFFC0300h
Block 4	FFFD0100h	FFFD0300h
Block 3	FFFE0100h	FFFE0300h
Block 2	FFFE8100h	FFFE8300h
Block 1	FFFF0100h	FFFF0300h
Block 0	FFFF8100h	FFFF8300h

27.2.3 ID Code Protection

This protection can only be used in standard serial I/O mode. A command from the serial programmer is to be accepted when the 7-byte ID code sent from the serial programmer matches the ID code programmed in the flash memory. However, when the reset vector is FFFFFFFFh, the ID code check is skipped because the flash memory is considered to be blank. When the reset vector is FFFFFFFFh and the ROM code protection is enabled, only the block erase command is accepted.

The ID codes sent from the serial programmer are consecutively numbered as ID1, ID2, ..., and ID7. ID codes programmed in the flash memory, also numbered as ID1, ID2, ..., and ID7, are assigned to addresses FFFFFFE8h, FFFFFFE9h, ..., and FFFFFFEEh as shown in Figure 27.2. The ID code protection is enabled when a program which has an ID code set in the corresponding address is written to the flash memory.

The following two ASCII code combinations are specified as reserved ID codes: “ALeRASE” and “Protect”. Refer to Table 27.5, 27.2.4 “Forcible Erase Function”, and 27.2.5 “Standard Serial I/O Mode Disable Function” for details.



Address				
FFFFFFDFh to FFFFFFDCh	Undefined instruction vector			
FFFFFFE3h to FFFFFFE0h	Overflow interrupt vector			
FFFFFFE7h to FFFFFFE4h	BRK instruction interrupt vector			
FFFFFFEBh to FFFFFFE8h	ID4	ID3	ID2	ID1
FFFFFFEFh to FFFFFFECh	OFS [1]	ID7	ID6	ID5
FFFFFFF3h to FFFFFFF0h	Watchdog timer interrupt vector			
FFFFFFF7h to FFFFFFF4h	Reserved			
FFFFFFFBh to FFFFFFF8h	NMI interrupt vector			
FFFFFFFFh to FFFFFFFCh	Reset vector			

Note:

1. Refer to the Watchdog Timer chapter for the OFS area.

Figure 27.2 Addresses for ID Code Stored

Table 27.5 Reserved ID Codes

ID Code		ID1	ID2	ID3	ID4	ID5	ID6	ID7
ALeRASE	Glyph	A	L	e	R	A	S	E
	ASCII code	41h	4Ch	65h	52h	41h	53h	45h
Protect	Glyph	P	r	o	t	e	c	t
	ASCII code	50h	72h	6Fh	74h	65h	63h	74h

27.2.4 Forcible Erase Function

The forcible erase function is available in standard serial I/O mode. With this function, all blocks of the flash memory are forcibly erased when ID codes sent from the serial programmer matches the ASCII code corresponding to the following sequential ASCII-glyphs: “A”, “L”, “e”, “R”, “A”, “S”, and “E”. However, the function is ignored when the ROM code protection is activated and ID codes other than “ALeRASE” are programmed in the flash memory.

Table 27.6 Operational Conditions for Forcible Erase Function

ID Codes Sent From the Serial Programmer	ID Codes Programmed in the Flash Memory	ROM Code Protection	Function
“ALeRASE”	“ALeRASE”	—	Erase all blocks of the flash memory
	Any codes other than “ALeRASE” or ”Protect”	Inactivated	
		Activated	Check ID codes (resulted in unmatched codes)
Any codes other than “ALeRASE”	“ALeRASE”	—	Check ID codes (resulted in unmatched codes)
	Any codes other than “ALeRASE” or ”Protect”	—	Check ID codes

27.2.5 Standard Serial I/O Mode Disable Function

With the standard serial I/O mode disable function, the flash memory in standard serial I/O mode is inaccessible from the CPU when ID code programmed in the flash memory are ASCII codes corresponding to the following sequential ASCII-glyphs: “P”, “r”, “o”, “t”, “e”, “c”, and “t”.

When the ROM code protection is activated and ID codes corresponding to “Protect” are programmed, the serial programmer cannot deactivate the ROM code protection. In this case, the flash memory is not accessible from the outside of MCU, except that the parallel programmer can delete the flash memory.

27.3 CPU Rewrite Mode

In CPU rewrite mode, the CPU executes software commands to rewrite the flash memory. The CPU accesses the flash memory not via the CPU buses, but via the dedicated flash memory rewrite buses (refer to Figure 27.3).

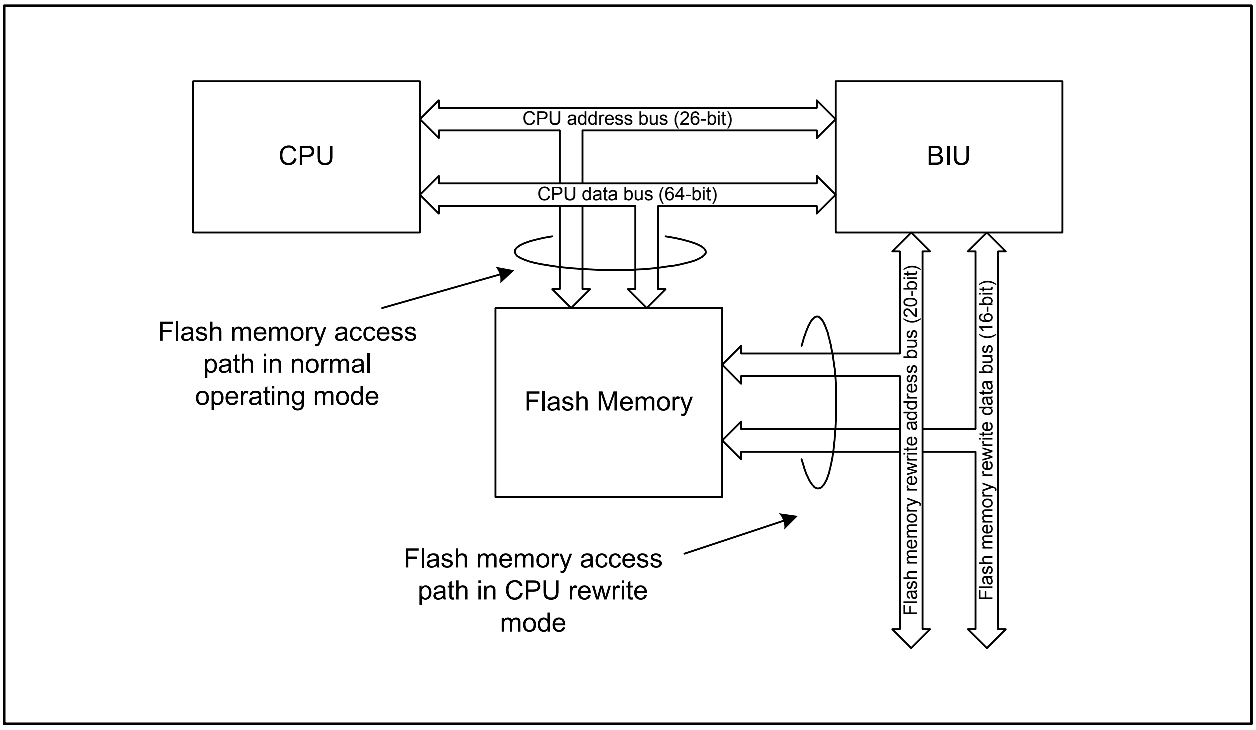

CPU
CPU address bus (26-bit)
BIU
CPU data bus (64-bit)
Flash memory access path in normal operating mode
Flash Memory
Flash memory rewrite address bus (20-bit)
Flash memory rewrite data bus (16-bit)
Flash memory access path in CPU rewrite mode

Figure 27.3 Flash Memory Access Path in CPU Rewrite Mode

Bus setting for flash memory rewrite should be performed by the FEBC register. Refer to 27.3.1 “Flash Memory Rewrite Bus Timing” and 28. “Electrical Characteristics” for the appropriate bus setting.

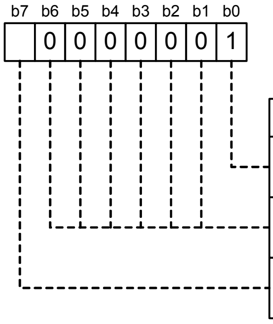
The CPU rewrite mode contains modes EW0 and EW1 as shown in Table 27.7.

Table 27.7 EW0 and EW1 Modes

Item	EW0 Mode	EW1 Mode
CPU operating modes	Single-chip mode Memory expansion mode	Single-chip mode
Rewrite program executable spaces	Spaces other than the on-chip flash memory	Internal spaces other than specified blocks to be rewritten, internal RAM
Restrictions on software commands	None	• Do not execute either the program command or the block erase command for blocks where the rewrite control programs are written to • Do not execute the enter read status register mode command • Execute the enter read lock bit status mode command in RAM • Execute the enter read protect bit status mode command in RAM
Mode after program/ erase operation	Read status register mode	Read array mode
CPU state during program/erase operation	Operating	Operating unless the CPU accesses the SFRs, flash memory or external bus. If it accesses the areas above mentioned, it stops operating
Flash memory state detection by	• Reading the FMSR0 register by a program • Executing the enter read status register mode command to read data	• Reading the FMSR0 register by a program
Suspend request timing	• When the SUSREQ bit in the FMR1 register is set to 1	• When an interrupt request with higher priority level than that is set by the SUSILVL bit in the FMR1 register is generated
Other restrictions	None	• Disable interrupts (except NMI) during program/erase operation, if suspend/ resume function is not used • Disable DMA transfer during program/ erase operation

To select CPU rewrite mode, the FEW bit in the FMCR register should be set to 1. Then, EW0 mode/EW1 mode can be selected by setting the EWM bit in the FMR0 register.
Registers FMCR and FMR0 are protected by registers PRR and FPR0, respectively.
Figures 27.4 to 27.12 show associated registers.

Flash Memory Control Register [1]



Symbol	Address	Reset Value
FMCR	0006h	0000 0001b

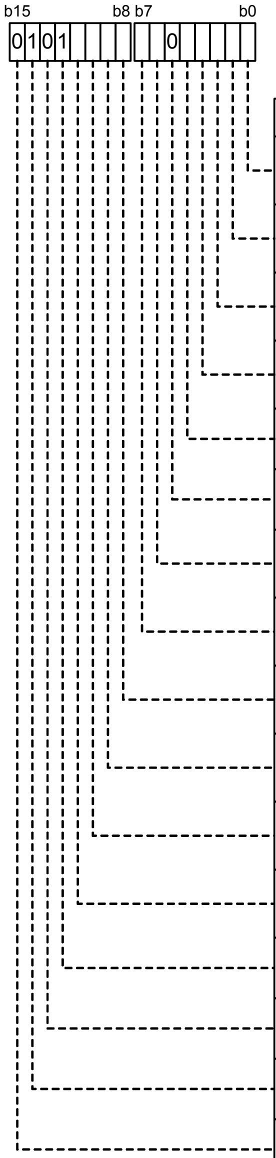
Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 1	RW
— (b6-b1)	Reserved	Should be written with 0	RW
FEW	CPU Rewrite Mode Setting Bit [2]	0: Normal operating mode 1: CPU rewrite mode	RW

Notes:

1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. Do not set this bit to 1 when the MRS bit in the VRCR register is 1 (main regulator stopped).

Figure 27.4 FMCR Register

Flash Memory Rewrite Bus Control Register [1]



Symbol FEBC

Address 0008h-0009h

Reset Value 0000h

Bit Symbol	Bit Name	Function	RW
FWR0	$\overline{\text{RD}}$ Pulse Width Setting Bit	b3 b2 b1 b0 0 0 0 0 : *wr* = 1 0 0 0 1 : *wr* = 2 0 1 0 1 : *wr* = 3 0 1 1 0 : *wr* = 4 1 0 1 0 : *wr* = 5 1 0 1 1 : *wr* = 6 1 1 1 1 : *wr* = 7 Only use the combinations listed above	RW
FWR1			RW
FWR2			RW
FWR3			RW
FWR4	$\overline{\text{RD}}$ Pulse Width Extension Select Bit	0: No pulse width extension 1: Pulse width extension selected	RW
— (b5)	Reserved	Should be written with 0	RW
MPY0	Multiplied Cycle Setting Bit	b7 b6 0 0 : Do not use this combination 0 1 : Do not use this combination 1 0 : *mpy* = 3 1 1 : *mpy* = 4	RW
MPY1			RW
FSUW0	Address Setup Before $\overline{\text{WR}}$ Setting Bit	b9 b8 0 0 : *suw* = 0 0 1 : *suw* = 1 1 0 : *suw* = 2 1 1 : *suw* = 3	RW
FSUW1			
FWW0	$\overline{\text{WR}}$ Pulse Width Setting Bit	b11 b10 0 0 : *ww* = 1 0 1 : *ww* = 2 1 0 : *ww* = 3 1 1 : *ww* = 4	RW
FWW1			
— (b12)	Reserved	Should be written with 1	RW
— (b13)	Reserved	Should be written with 0	RW
— (b14)	Reserved	Should be written with 1	RW
— (b15)	Reserved	Should be written with 0	RW

Note:

1. Set the PRR register to AAh (write enabled) before rewriting this register.

Figure 27.5 FEBC Register

Flash Register Protection Unlock Register 0

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	

Symbol FPR0 | Address 40008h | Reset Value 0000 0000b

Bit Symbol	Bit Name	Function	RW
PR0	Protection Unlock Bit	For registers FMR0 and FMR1, 0: Write disabled 1: Write enabled	RW
— (b7-b1)	Reserved	Should be written with 0	RW

Figure 27.6 FPR0 Register

Flash Memory Control Register 0 (1, 2)

b7	b6	b5	b4	b3	b2	b1	b0
0		0					

Symbol FMR0 | Address 40000h | Reset Value 0001 XX00b

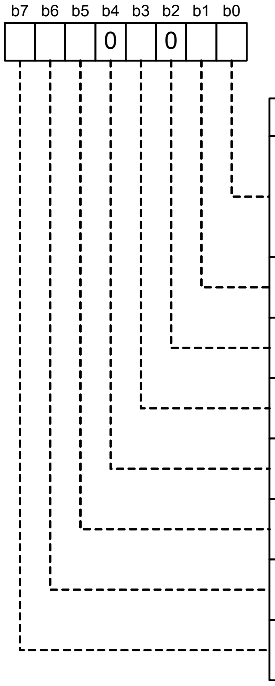
Bit Symbol	Bit Name	Function	RW
EWM	Rewrite Mode Select Bit	0: EW0 mode 1: EW1 mode	RW
LBM	Lock Bit Read Mode Setting Bit	0: Read via data bus (3) 1: Read by the LBS bit (4)	RW
LBS	Lock Bit Status Flag (4)	0: Locked 1: Unlocked	RO
RRDY	Read Ready Flag	0: Busy 1: Ready	RO
FCA	Final Command Accept Busy Flag	0: Final command accept ready 1: Final command accept busy	RO
— (b5)	Reserved	Should be written with 0	RW
SUSACK	Suspend Request Acknowledge Flag	0: No suspend request is acknowledged 1: Suspend request is acknowledged	RO
— (b7)	Reserved	Should be written with 0	RW

Notes:

1. Set the PR0 bit in the FPR0 register to 1 (write enabled) before rewriting this register.
2. This register is reset after exiting wait mode or stop mode.
3. After entering read lock bit status mode, the lock bit status is reflected to bit 6 of read data when reading any even address in the corresponding block.
4. The LBS bit reflects the lock bit status when issuing the read lock bit status command.

Figure 27.7 FMR0 Register

Flash Memory Control Register 1 [1]



Symbol FMR1

Address 40009h

Reset Value 0000 0010b

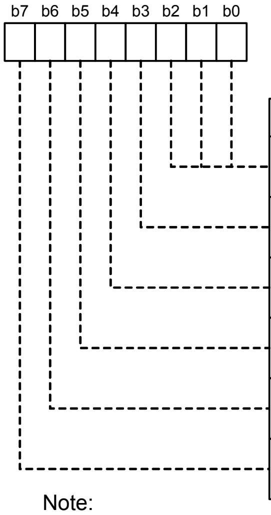
Bit Symbol	Bit Name	Function	RW
SUSILVL	Suspend Request Interrupt Level Select Bit	0: Normal interrupt [2] with request level 7, or DMAII transfer request 1: Normal interrupt [2] with request level 6 or 7 [3]	RW
RR	Reset Release Bit	0: Reset 1: Reset released	RW
— (b2)	Reserved	Should be written with 0	RW
LBD	Lock Bit Protect Disable Bit	0: Lock bit protection enabled 1: Lock bit protection disabled	RW
— (b4)	Reserved	Should be written with 0	RW
SUSREQ	Suspend Request Bit	0: No request/request to resume 1: Request to suspend	RW
SUSRDY	Suspended and Ready Flag	0: Busy or not suspended 1: Suspended and ready	RO
SUSEN	Suspend Enable Bit	0: Disable suspend function 1: Enable suspend function	RW

Notes:

1. Set the PR0 bit in the FPR0 register to 1 (write enabled) before rewriting this register.
2. Fast interrupt does not trigger a suspend request.
3. Set the DMAII bit in registers RIPL1 and RIPL2 to 0 (use interrupt request level 7 for interrupt).

Figure 27.8 FMR1 Register

Flash Memory Status Register 0



Symbol: FMSR0 Address: 40001h Reset Value: 1000 0000b

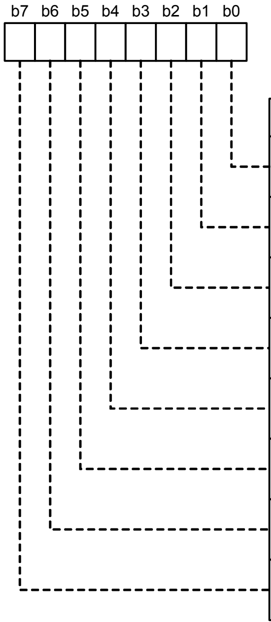
Bit Symbol	Bit Name	Function	RW
— (b2-b0)	Reserved	These bits are read as undefined value	RO
WSUS	Program Suspend Status Flag	0: Program in progress/completed 1: Program suspended	RO
WERR	Program Error Flag	0: No program error 1: Program error occurred	RO
EERR	Erase Error Flag	0: No erase error 1: Erase error occurred (1)	RO
ESUS	Erase Suspend Status Flag	0: Erase in progress/completed 1: Erase suspended	RO
RDY	Ready Flag	0: Busy 1: Ready	RO

Note:

1. If an erase error has occurred, issue the clear status register command first, then reissue the block erase command repeatedly until no more erase errors occur. After that, execute three extra block erase operations.

Figure 27.9 FMSR0 Register

Block Protect Bit Monitor Register 0 (1)



Symbol: FBPM0 Address: 4000Ah Reset Value: ??X? ????b (1)

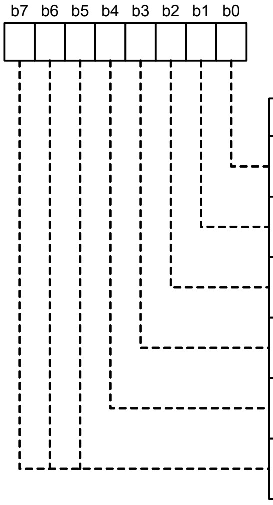
Bit Symbol	Bit Name	Function	RW
BP0	Block 0 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP1	Block 1 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP2	Block 2 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP3	Block 3 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP4	Block 4 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
— (b5)	Reserved	This bit is read as undefined value	RO
BP5	Block 5 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP6	Block 6 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO

Note:

1. This register is updated only once after reset is released. The protect bit status at that time is applied as reset value.

Figure 27.10 FBPM0 Register

Block Protect Bit Monitor Register 1 [(1)]



Symbol	Address	Reset Value
FBPM1	4000Bh	XXX? ????b [(1)]

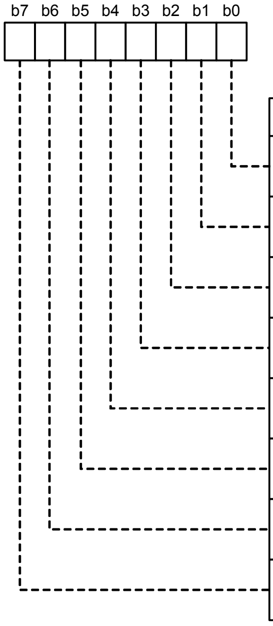
Bit Symbol	Bit Name	Function	RW
BP7	Block 7 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP8	Block 8 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP9	Block 9 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BPB	Block B Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BPA	Block A Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
— (b7-b5)	Reserved	These bits are read as undefined value	RO

Note:

1. This register is updated only once after reset is released. The protect bit status at that time is applied as the reset value.

Figure 27.11 FBPM1 Register

Block Protect Bit Monitor Register 2 [(1)]



Symbol	Address	Reset Value
FBPM2	40011h	???? ????b [(1)]

Bit Symbol	Bit Name	Function	RW
BP10	Block 10 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP11	Block 11 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP12	Block 12 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP13	Block 13 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP14	Block 14 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP15	Block 15 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP16	Block 16 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP17	Block 17 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO

Note:

1. This register is updated only once after reset is released. The protect bit status at that time is applied as the reset value.

Figure 27.12 FBPM2 Register

27.3.1 Flash Memory Rewrite Bus Timing

The bus setting for the flash memory rewrite is performed by setting the FEBC register. This section specifically describes the setting of the FEBC register.

The reference clock is the base clock set with bits BCD1 and BCD0 in the CCR register. Time duration including tsu, tw, tc, and th are specified by the number of base clock cycles.

Tables 27.8 to 27.10 show the correlation of the read cycle and setting of bits MPY1, MPY0, and FWR4 to FWR0, according to peripheral bus clock divide ratios. Tables 27.11 to 27.13 show the correlation of the write cycle and setting of bits MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0. Associated read/write timings are illustrated in Figures 27.13 and 27.14, respectively.

Read/write cycle timing is selected from the tables below to meet the timing requirements in the CPU rewrite mode described in the electrical characteristics.

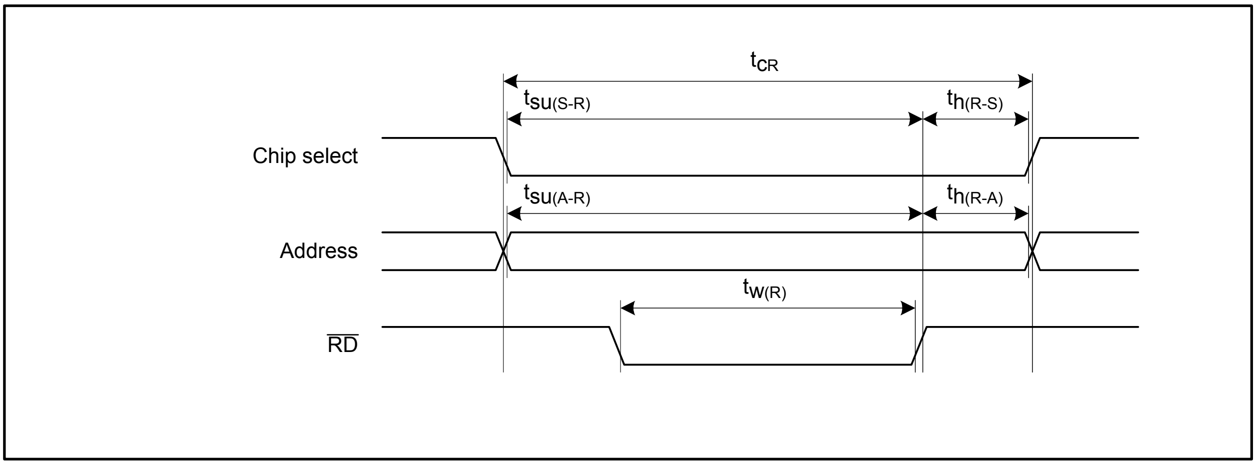


Figure 27.13 Read Timing

Table 27.8 Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 2 (unit: cycles)

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 Bit Settings							
			10b				11b			
			$mpy = 3$				$mpy = 4$			
			$t_{su(S-R)}$, $t_{su(A-R)}$	$t_{w(R)}$	t_{cR}	$t_{h(R-S)}$, $t_{h(R-A)}$	$t_{su(S-R)}$, $t_{su(A-R)}$	$t_{w(R)}$	t_{cR}	$t_{h(R-S)}$, $t_{h(R-A)}$
0000b	$wr = 1$	0	4	3	4	0	6	5	6	0
		1	6	5	6	0	6	5	6	0
0001b	$wr = 2$	0	8	7	8	0	10	9	10	0
		1	8	7	8	0	10	9	10	0
0101b	$wr = 3$	0	10	9	10	0	14	13	14	0
		1	12	11	12	0	14	13	14	0
0110b	$wr = 4$	0	14	13	14	0	18	17	18	0
		1	14	13	14	0	18	17	18	0
1010b	$wr = 5$	0	16	15	16	0	22	21	22	0
		1	18	17	18	0	22	21	22	0
1011b	$wr = 6$	0	20	19	20	0	26	25	26	0
		1	20	19	20	0	26	25	26	0
1111b	$wr = 7$	0	22	21	22	0	30	29	30	0
		1	24	23	24	0	30	29	30	0

Table 27.9 Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 3 (unit: cycles)

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 Bit Settings							
			10b				11b			
			$mpy = 3$				$mpy = 4$			
			$t_{su(S-R)}$, $t_{su(A-R)}$	$t_{w(R)}$	t_{cR}	$t_{h(R-S)}$, $t_{h(R-A)}$	$t_{su(S-R)}$, $t_{su(A-R)}$	$t_{w(R)}$	t_{cR}	$t_{h(R-S)}$, $t_{h(R-A)}$
0000b	$wr = 1$	0	6	4.5	6	0	6	4.5	6	0
		1	6	4.5	6	0	6	4.5	6	0
0001b	$wr = 2$	0	9	7.5	9	0	9	7.5	9	0
		1	9	7.5	9	0	12	10.5	12	0
0101b	$wr = 3$	0	12	10.5	12	0	15	13.5	15	0
		1	12	10.5	12	0	15	13.5	15	0
0110b	$wr = 4$	0	15	13.5	15	0	18	16.5	18	0
		1	15	13.5	15	0	18	16.5	18	0
1010b	$wr = 5$	0	18	16.5	18	0	21	19.5	21	0
		1	18	16.5	18	0	24	22.5	24	0
1011b	$wr = 6$	0	21	19.5	21	0	27	25.5	27	0
		1	21	19.5	21	0	27	25.5	27	0
1111b	$wr = 7$	0	24	22.5	24	0	30	28.5	30	0
		1	24	22.5	24	0	30	28.5	30	0

Table 27.10 Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 4 (unit: cycles)

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 Bit Settings							
			10b				11b			
			$mpy = 3$				$mpy = 4$			
			$t_{su(S-R)}$, $t_{su(A-R)}$	$t_{w(R)}$	t_{cR}	$t_{h(R-S)}$, $t_{h(R-A)}$	$t_{su(S-R)}$, $t_{su(A-R)}$	$t_{w(R)}$	t_{cR}	$t_{h(R-S)}$, $t_{h(R-A)}$
0000b	$wr = 1$	0	4	2	4	0	8	6	8	0
		1	8	6	8	0	8	6	8	0
0001b	$wr = 2$	0	8	6	8	0	12	10	12	0
		1	8	6	8	0	12	10	12	0
0101b	$wr = 3$	0	12	10	12	0	16	14	16	0
		1	12	10	12	0	16	14	16	0
0110b	$wr = 4$	0	16	14	16	0	20	18	20	0
		1	16	14	16	0	20	18	20	0
1010b	$wr = 5$	0	16	14	16	0	24	22	24	0
		1	20	18	20	0	24	22	24	0
1011b	$wr = 6$	0	20	18	20	0	28	26	28	0
		1	20	18	20	0	28	26	28	0
1111b	$wr = 7$	0	24	22	24	0	32	30	32	0
		1	24	22	24	0	32	30	32	0

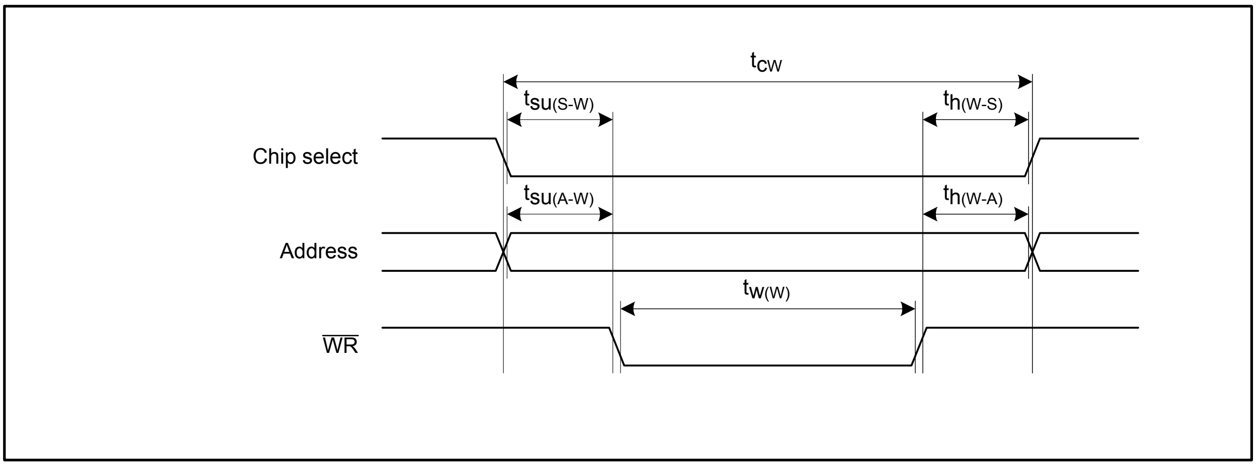


Figure 27.14 Write Timing

Table 27.11 Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 2 (unit: cycles)

FSUW1 and FSUW0 Bit Settings		FWW1 and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				$mpy = 3$				$mpy = 4$			
				$t_{su(S-W)}$, $t_{su(A-W)}$	$t_{w(W)}$	t_{CW}	$t_{h(W-S)}$, $t_{h(W-A)}$	$t_{su(S-W)}$, $t_{su(A-W)}$	$t_{w(W)}$	t_{CW}	$t_{h(W-S)}$, $t_{h(W-A)}$
00b	$suw = 0$	00b	$ww = 1$	1	3	6	2	1	4	6	1
		01b	$ww = 2$	1	6	8	1	1	8	10	1
		10b	$ww = 3$	1	9	12	2	1	12	14	1
		11b	$ww = 4$	1	12	14	1	1	16	18	1
01b	$suw = 1$	00b	$ww = 1$	4	3	8	1	5	4	10	1
		01b	$ww = 2$	4	6	12	2	5	8	14	1
		10b	$ww = 3$	4	9	14	1	5	12	18	1
		11b	$ww = 4$	4	12	18	2	5	16	22	1
10b	$suw = 2$	00b	$ww = 1$	7	3	12	2	9	4	14	1
		01b	$ww = 2$	7	6	14	1	9	8	18	1
		10b	$ww = 3$	7	9	18	2	9	12	22	1
		11b	$ww = 4$	7	12	20	1	9	16	26	1
11b	$suw = 3$	00b	$ww = 1$	10	3	14	1	13	4	18	1
		01b	$ww = 2$	10	6	18	2	13	8	22	1
		10b	$ww = 3$	10	9	20	1	13	12	26	1
		11b	$ww = 4$	10	12	24	2	13	16	30	1

Table 27.12 Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 3 (unit: cycles)

FSUW1 and FSUW0 Bit Settings		FWW1 and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				mpy = 3				*mpy* = 4			
				tsu(S-W), tsu(A-W)	tw(W)	tcW	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcW	th(W-S), th(W-A)
00b	*suw* = 0	00b	*ww* = 1	1	3	6	2	1	4	6	1
		01b	*ww* = 2	1	6	9	2	1	8	12	3
		10b	*ww* = 3	1	9	12	2	1	12	15	2
		11b	*ww* = 4	1	12	15	2	1	16	18	1
01b	*suw* = 1	00b	*ww* = 1	4	3	9	2	6	3	12	3
		01b	*ww* = 2	4	6	12	2	6	7	15	2
		10b	*ww* = 3	4	9	15	2	6	11	18	1
		11b	*ww* = 4	4	12	18	2	6	15	24	3
10b	*suw* = 2	00b	*ww* = 1	7	3	12	2	9	4	15	2
		01b	*ww* = 2	7	6	15	2	9	8	18	1
		10b	*ww* = 3	7	9	18	2	9	12	24	3
		11b	*ww* = 4	7	12	21	2	9	16	27	2
11b	*suw* = 3	00b	*ww* = 1	10	3	15	2	13	4	18	1
		01b	*ww* = 2	10	6	18	2	13	8	24	3
		10b	*ww* = 3	10	9	21	2	13	12	27	2
		11b	*ww* = 4	10	12	24	2	13	16	30	1

Table 27.13 Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 4 (unit: cycles)

FSUW1 and FSUW0 Bit Settings		FWW1 and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				mpy = 3				*mpy* = 4			
				tsu(S-W), tsu(A-W)	tw(W)	tcW	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcW	th(W-S), th(W-A)
00b	*suw* = 0	00b	*ww* = 1	1	3	8	4	1	4	8	3
		01b	*ww* = 2	1	6	8	1	1	8	12	3
		10b	*ww* = 3	1	9	12	2	1	12	16	3
		11b	*ww* = 4	1	12	16	3	1	16	20	3
01b	*suw* = 1	00b	*ww* = 1	4	3	8	1	5	4	12	3
		01b	*ww* = 2	4	6	12	2	5	8	16	3
		10b	*ww* = 3	4	9	16	3	5	12	20	3
		11b	*ww* = 4	4	12	20	4	5	16	24	3
10b	*suw* = 2	00b	*ww* = 1	8	2	12	2	9	4	16	3
		01b	*ww* = 2	8	5	16	3	9	8	20	3
		10b	*ww* = 3	8	8	20	4	9	12	24	3
		11b	*ww* = 4	8	11	20	1	9	16	28	3
11b	*suw* = 3	00b	*ww* = 1	10	3	16	3	13	4	20	3
		01b	*ww* = 2	10	6	20	4	13	8	24	3
		10b	*ww* = 3	10	9	20	1	13	12	28	3
		11b	*ww* = 4	10	12	24	2	13	16	32	3

27.3.2 Software Commands

In CPU rewrite mode, software commands enable program and erase operations for the flash memory. Writing commands and reading/writing data should be performed in 16-bit units.

Table 27.14 lists the software commands.

Table 27.14 Software Commands

Command	First Command Cycle		Second Command Cycle	
	Address	Data	Address	Data
Enter read array mode	FFFFF800h	00FFh	—	—
Enter read status register mode [1]	FFFFF800h	0070h	—	—
Clear status register	FFFFF800h	0050h	—	—
Program [2]	FFFFF800h	0043h	WA	WD
Block erase	FFFFF800h	0020h	BA	00D0h
Lock bit program	FFFFF800h	0077h	BA	00D0h
Read lock bit status	FFFFF800h	0071h	BA	00D0h
Enter read lock bit status mode [3]	FFFFF800h	0071h	—	—
Protect bit program	FFFFF800h	0067h	PBA	00D0h
Enter read protect bit status mode [3]	FFFFF800h	0061h	—	—

WA: Even address to be written
WD: 16-bit data to be written
BA: Even address within a specific block
PBA: Protect bit address (refer to Table 27.4)

Notes:

1. This command cannot be executed in EW1 mode.
2. The program is performed in 64-bit (4-word) units. A sequence of commands consists of commands from the second to fifth. The upper 29 bits of the address WA should be fixed and the lower 3 bits of respective commands from the second to fifth should be set to 000b, 010b, 100b, and 110b for the addresses 0h, 2h, 4h, and 6h, or 8h, Ah, Ch, and Eh.
3. This command should be executed in RAM.

27.3.3 Mode Transition

CPU rewrite mode supports four flash memory operating modes:

- Read array mode
- Read status register mode
- Read lock bit status mode
- Read protect bit status mode

When reading the flash memory in these modes, the memory data, the status register value, the state of the lock bit in the read block, and the state of the protect bit are individually read. Details are listed in Tables 27.15 to 27.17.

Table 27.15 Status Register

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b8	—	Disabled bit	—	—
b7	SR7	Sequencer status	BUSY	READY
b6	SR6	Erase suspend status	Erase in progress/ completed	Erase suspended
b5	SR5	Erase status	Successfully completed	Error
b4	SR4	Program status	Successfully completed	Error
b3	SR3	Program suspend status	Program in progress/ completed	Program suspended
b2	—	Reserved bit	—	—
b1	—	Reserved bit	—	—
b0	—	Reserved bit	—	—

Table 27.16 Lock Bit Status

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b7	—	Disabled bit	—	—
b6	LBS	Lock bit status	Locked	Unlocked
b5-b0	—	Disabled bit	—	—

Table 27.17 Protect Bit Status

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b7	—	Disabled bit	—	—
b6	PBS	Protect bit status	Protected	Unprotected
b5-b0	—	Disabled bit	—	—

In these operating modes, program or erase operation can be performed by software commands. After an operation is completed, the flash memory module automatically enters read array mode (in EW1 mode) or read status register mode (in EW0 mode).

27.3.4 Issuing Software Commands

This section describes how to issue software commands.
These commands should be issued while the RDY bit in the FMSR0 register is 1 (ready).

27.3.4.1 Enter Read Array Mode Command

Execute this command to enter read array mode.
When 00FFh is written to address FFFFF800h, the flash memory enters read array mode. In this mode, the value stored to a given address in memory can be read.
In EW1 mode, the flash memory is always in read array mode.

27.3.4.2 Enter Read Status Register Mode

Execute this command to enter read status register mode.
When 0070h is written to address FFFFF800h, the status register value is read in any address of the flash memory.
Do not issue this command in EW1 mode.

27.3.4.3 Clear Status Register

Execute this command to reset the status register in the flash memory.
When 0050h is written to address FFFFF800h, bits SR5 and SR4 in the status register become 0 (successfully completed) (refer to Table 27.15). Consequently, bits EERR and WERR in the FMSR0 register become 0 (no errors).

27.3.4.4 Program Command

Execute this command to program the flash memory in 8-byte (4-word) units.

To start automatic programming (program and program-verify operations), write 0043h to address FFFFF800h, then write data to addresses 8n + 0 to 8n + 6. Verify that the FCA bit in the FMR0 register is 0 just before executing the final command.

To monitor the automatic program operation, read the RDY bit in the FMSR0 register. This bit becomes 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

The operation result can be verified by the WERR bit in the FMSR0 register (refer to 27.3.5 “Status Check”).

Do not write additional data to an address that is already programmed.

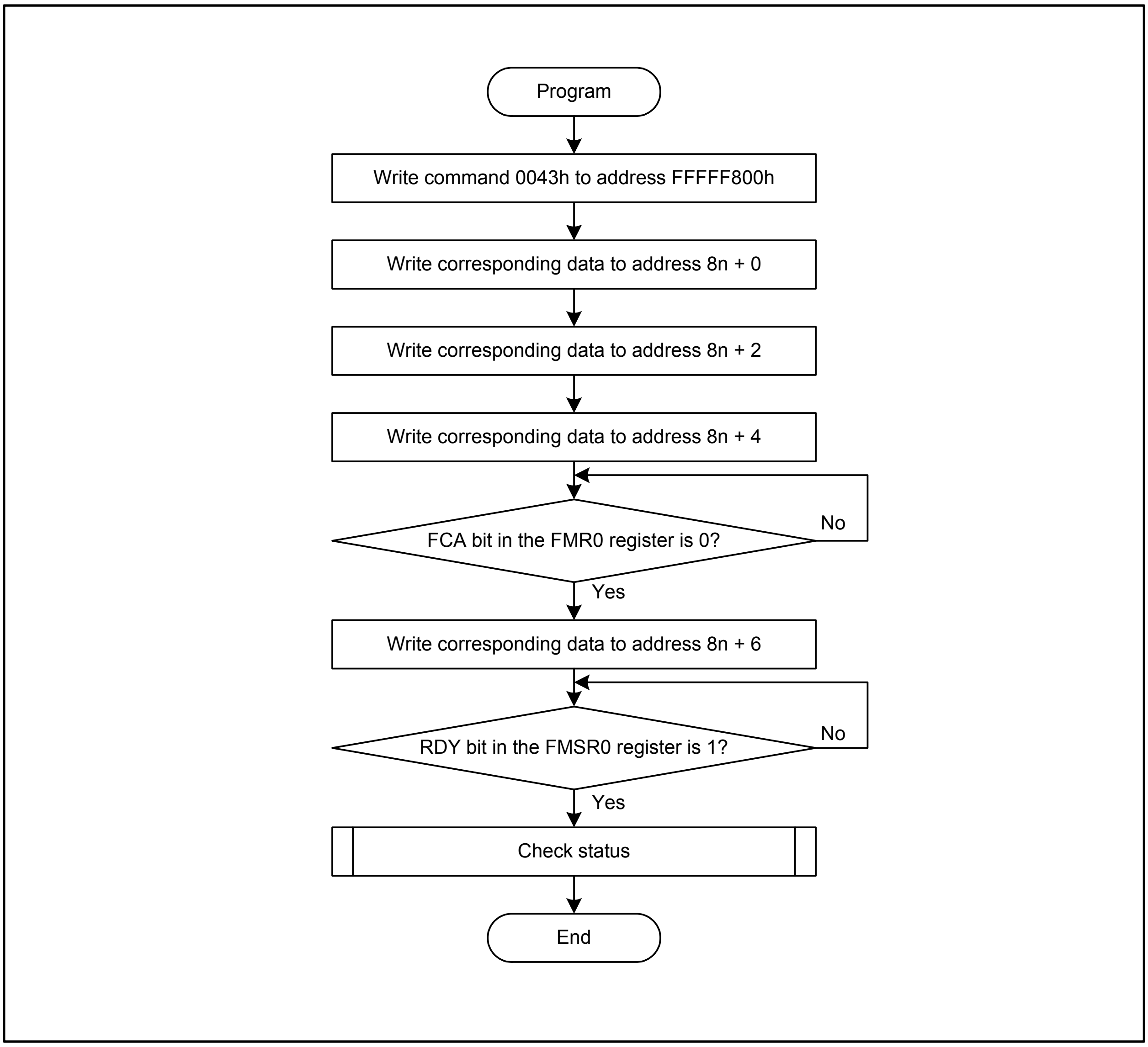

Program
Write command 0043h to address FFFFF800h
Write corresponding data to address 8n + 0
Write corresponding data to address 8n + 2
Write corresponding data to address 8n + 4
FCA bit in the FMR0 register is 0?
No
Yes
Write corresponding data to address 8n + 6
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End

Figure 27.15 Program Command Execution Flowchart

27.3.4.5 Block Erase Command

Execute this command to erase a specified block in the flash memory.

To start automatic erasing of a specified block (erase and erase-verify operations), write 0020h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address in the corresponding block.

To monitor the automatic erase operation, read the RDY bit in the FMSR0 register. This bit becomes 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

The operation result can be verified by the EERR bit in the FMSR0 register (refer to 27.3.5 “Status Check”).

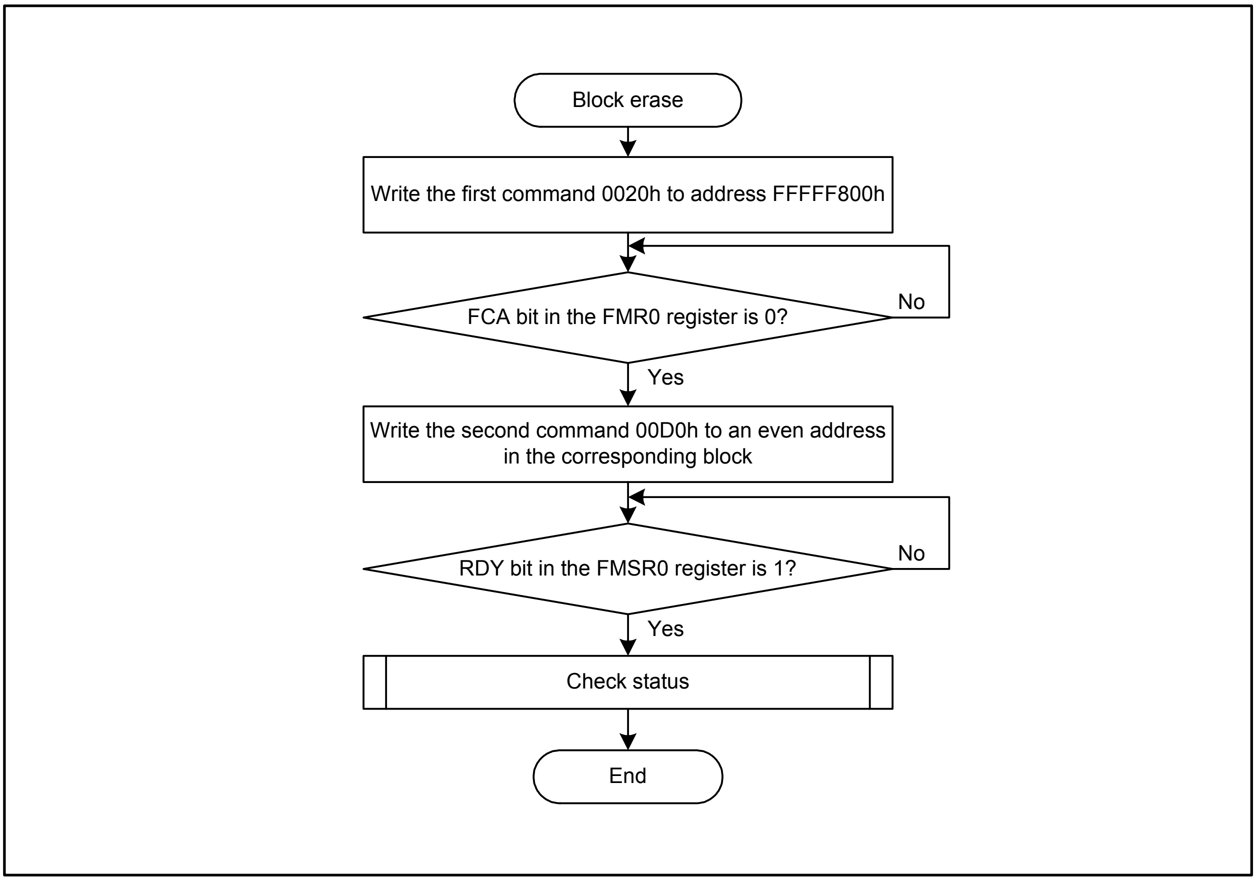

Block erase
Write the first command 0020h to address FFFFF800h
FCA bit in the FMR0 register is 0?
No
Yes
Write the second command 00D0h to an even address in the corresponding block
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End

Figure 27.16 Block Erase Command Execution Flowchart

27.3.4.6 Lock Bit Program Command

Execute this command to lock a specified block in the flash memory.

To lock the block, write 0077h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address in the corresponding block. Then the lock bit of the block becomes 0 (locked).

To monitor the lock bit program, read the RDY bit in the FMSR0 register. This bit becomes 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

The state of the lock bit can be verified by the read lock bit status command if the LBM bit in the FMR0 register is 1 (read by the LBS bit) (refer to 27.3.4.7 “Read Lock Bit Status Command”). If the LBM bit is 0 (read via data bus), enter read lock bit status mode (refer to 27.3.4.8 “Enter Read Lock Bit Status Mode Command”).

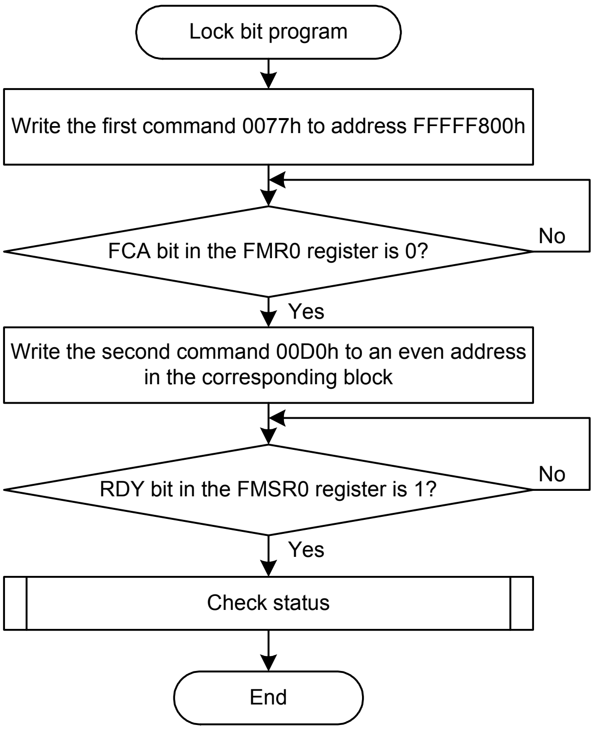

Lock bit program
Write the first command 0077h to address FFFFF800h
FCA bit in the FMR0 register is 0?
No
Yes
Write the second command 00D0h to an even address in the corresponding block
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End

Figure 27.17 Lock Bit Program Command Execution Flowchart

27.3.4.7 Read Lock Bit Status Command

Execute this command to verify if a specified block in the flash memory is locked. This command can be used when the LBM bit in the FMR0 register is 1 (read by the LBS bit).

The LBS bit in the FMSR0 register reflects the lock bit status of the specified block when the following is performed: first write 0071h to address FFFFF800h and verify that the FCA bit in the FMR0 register becomes 0. Then write 00D0h to an even address of the corresponding block.

Read the LBS bit after the RDY bit in the FMSR0 register becomes 1 (ready).

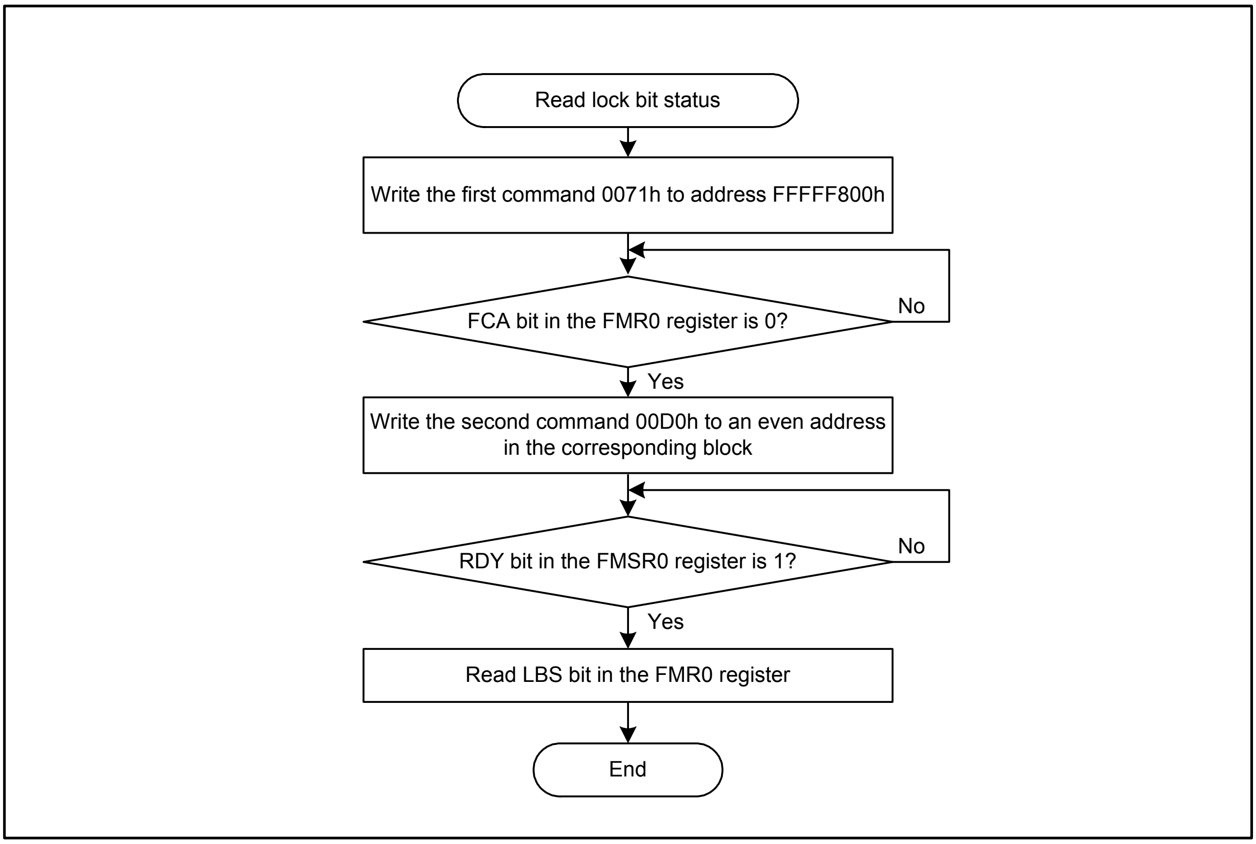

Read lock bit status
Write the first command 0071h to address FFFFF800h
FCA bit in the FMR0 register is 0?
No
Yes
Write the second command 00D0h to an even address in the corresponding block
RDY bit in the FMSR0 register is 1?
No
Yes
Read LBS bit in the FMR0 register
End

Figure 27.18 Read Lock Bit Status Command Execution Flowchart

27.3.4.8 Enter Read Lock Bit Status Mode Command

Execute this command to enter read lock bit status mode. This command is enabled when the LBM bit in the FMR0 register is 0 (read via data bus).

To read the lock bit status of the read block, write 0071h to address FFFFF800h (refer to Table 27.16).

The status is read in any address of the flash memory.

Execute this command in RAM.

27.3.4.9 Protect Bit Program Command

Execute this command to protect a specific block in the flash memory. ROM code protection is enabled by setting one of the protect bits of the block to 0.

To set the protect bit of the designated block to 0 (protected), write 0067h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, and then write 00D0h to the protect bit of the corresponding block (refer to Table 27.4).

To monitor the protect bit program, read the RDY bit in the FMSR0 register. This bit becomes 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

To verify the state of protect bit, enter read protect bit status mode (refer to 27.3.4.10 “Enter Read Protect Bit Status Mode Command”), then read the flash memory.

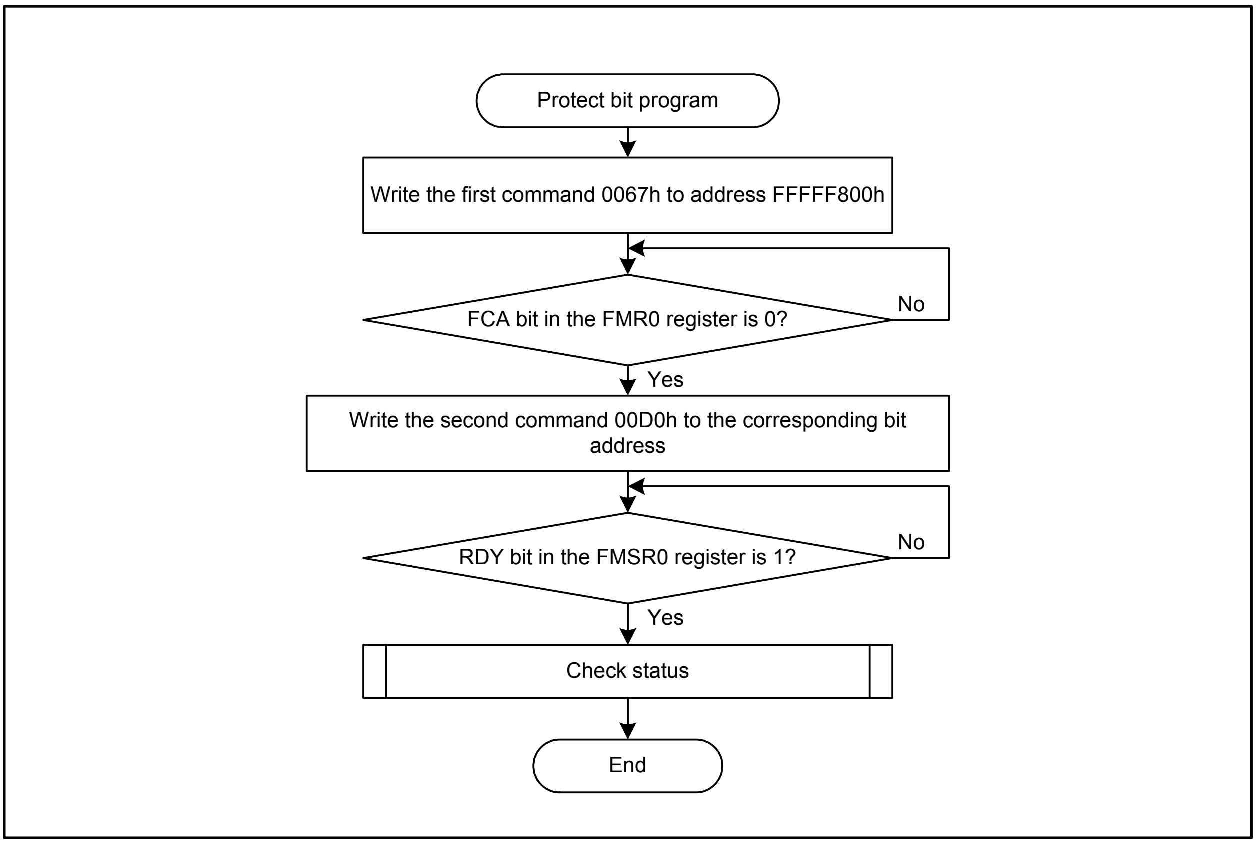

Protect bit program
Write the first command 0067h to address FFFFF800h
FCA bit in the FMR0 register is 0?
No
Yes
Write the second command 00D0h to the corresponding bit address
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End

Figure 27.19 Protect Bit Program Command Execution Flowchart

27.3.4.10 Enter Read Protect Bit Status Mode Command

Execute this command to enter read protect bit status mode.

To read the protect bit status of the read block, write 0061h to address FFFFF800h (refer to Table 27.17). The status is read from any address in the flash memory.

Execute this command in RAM.

27.3.5 Status Check

To verify if a software command is successfully executed, read the EERR or WERR bit in the FMSR0 register, or the SR5 bit or SR4 bit in the status register.

Table 27.18 lists status and errors indicated by these bits and Figure 27.20 shows the flowchart of the status check.

Table 27.18 Status and Errors

FMSR0 Register (Status Register)		Error	Source of Error
EERR bit (SR5 bit)	WERR bit (SR4 bit)		
1	1	Command sequence error	• Data other than 00D0h or 00FFh (command to cancel) was written as the last command of two commands • An unavailable address was specified by an address specifying command
1	0	Erase error	• Attempted to erase a locked block • Corresponding block was not erased properly
0	1	Program error	• Attempted to program a locked block • Data was not programmed properly • Lock bit was not programmed properly • Protect bit was not programmed properly
0	0	No error	

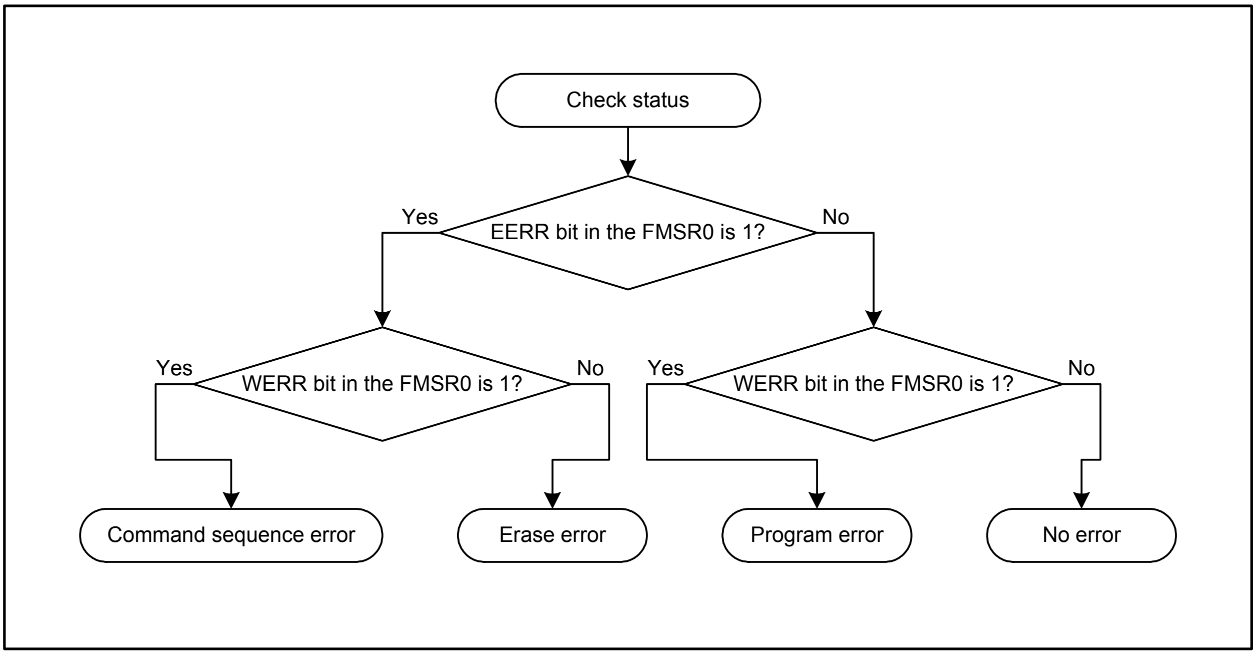

Check status
Yes
EERR bit in the FMSR0 is 1?
No
Yes
WERR bit in the FMSR0 is 1?
No
Yes
WERR bit in the FMSR0 is 1?
No
Command sequence error
Erase error
Program error
No error

Figure 27.20 Status Check Flowchart

When an error occurs, execute the clear status register command and then handle the error.

If erase errors or program errors occur frequently even though the program is correct, the corresponding block may be disabled.

27.3.6 Suspend/Resume Operation

The R32C/116A Group supports program-suspend and erase-suspend operations to execute other operations with higher priority. Unlike non-maskable interrupts which abort an operation in progress, a suspended operation is able to resume on demand. The following two software commands are suspendable: block erase and program.

27.3.6.1 Suspend Request

To generate a suspend request, set the SUSEN bit in the FMR1 register to 1 (suspend function enabled) before issuing a software command.
This section describes how to manage a suspend request in modes EW0 and EW1, respectively.

(1) Suspend request in EW0 mode

A suspend is requested by a program. Issue a software command, then set the SUSREQ bit in the FMR1 register to 1 (suspend requested) while a program or erase operation is being executed, that is, the RDY bit in the FMSR0 register is 0 (busy).

(2) Suspend Request in EW1 Mode

A suspend is requested by an interrupt. After issuing a software command, a suspend is requested when an interrupt with higher request level than the level set in the SUSILVL bit in the FMR1 register is generated while a program or erase operation is being executed, that is, the RDY bit in the FMSR0 register is 0 (busy). Note that a fast interrupt does not trigger a suspend request.

Note that set all interrupt-associated registers before entering CPU rewrite mode. When setting the SUSILVL bit in the FMR1 register to 0 (normal interrupt with request level 7 or DMA II transfer request), set the IPL to 6. When setting this bit to 1 (normal interrupt with request level 6 or 7), set the IPL to 5. Once the IPL is set, do not rewrite the interrupt-associated registers.

27.3.6.2 Operation in Suspend State

The operation with higher priority than the in-progress program or erase operation should be performed after verifying the suspend operation has completed successfully.

When a suspend request is acknowledged and the SUSRDY bit in the FMR1 register becomes 1 (suspended and ready), the CPU enables access to the flash memory. At the same moment, the SUSACK bit in the FMR0 register and the RDY bit in the FMSR0 register become 1. In addition, bits WSUS and ESUS in the FMSR0 register also become 1 if the above mentioned suspend request is generated while program and erase operations are being executed, respectively.

Figure 27.21 shows the flow to verify a successful suspend operation.

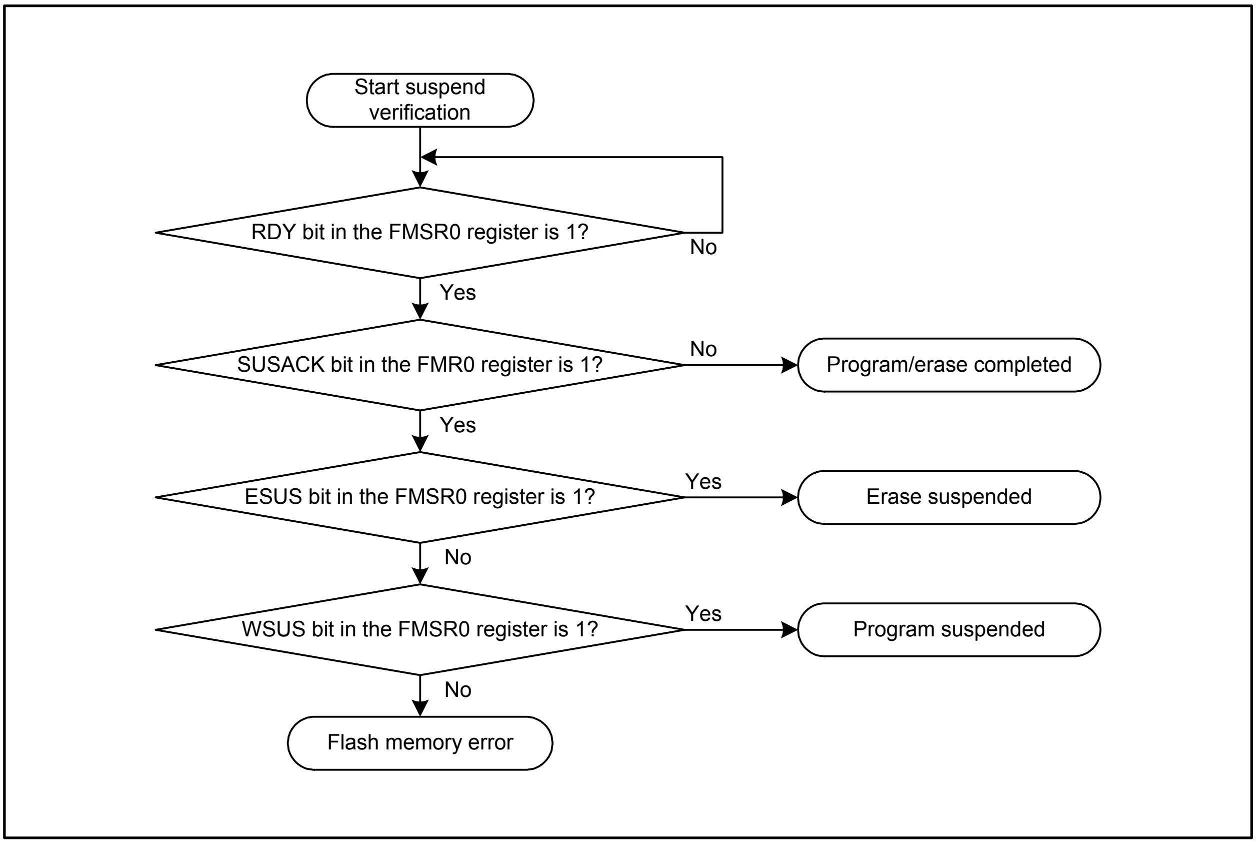

Start suspend verification
RDY bit in the FMSR0 register is 1?
No
Yes
SUSACK bit in the FMR0 register is 1?
No
Program/erase completed
Yes
ESUS bit in the FMSR0 register is 1?
Yes
Erase suspended
No
WSUS bit in the FMSR0 register is 1?
Yes
Program suspended
No
Flash memory error

Figure 27.21 Verification of Suspend Operation

When the RDY bit in the FMSR0 register is 1 (ready) and the SUSACK bit in the FMR0 register is 0 (no suspend request is acknowledged), the program or erase operation has been completed. Resume requests, if any, are ignored.

When the RDY bit is 1 (ready) and SUSACK bit is 1 (suspend request is acknowledged), a suspend request is accepted. However, if both bits ESUS and WSUS in the FMSR0 register are 0, the flash memory may have an error.

The software commands listed in Table 27.19 are valid while a program or erase operation is suspended. Do not issue any unlisted command while being in program-suspend or erase-suspend state.

Table 27.19 **Valid Software Commands in Program-Suspend/Erase-Suspend State**

Valid Commands in Program-Suspend State	Valid Commands in Erase-Suspend State
• Enter read array mode • Enter read status register mode • Clear status register	• Enter read array mode • Enter read status register mode • Clear status register • Program [1]

Note:

1. Issuing the command to a block being erased causes a command sequence error. The program commands executed in this program cannot be suspended.

27.3.6.3 Resume Request

To resume the original operation being processed before a suspend operation, set the SUSREQ bit in the FMR1 register to 0 (no suspend requested or resume requested) while the SUSACK bit in the FMR0 register is 1 (suspend request is acknowledged). When the SUSACK bit is 0 (no suspend request is acknowledged), setting of the SUSREQ bit to 0 is ignored.

If the SUSACK bit in the FMR0 register in suspend state becomes 0 (no suspend request is acknowledged), the flash memory may have an error. In this case, erase the corresponding block following the steps below:

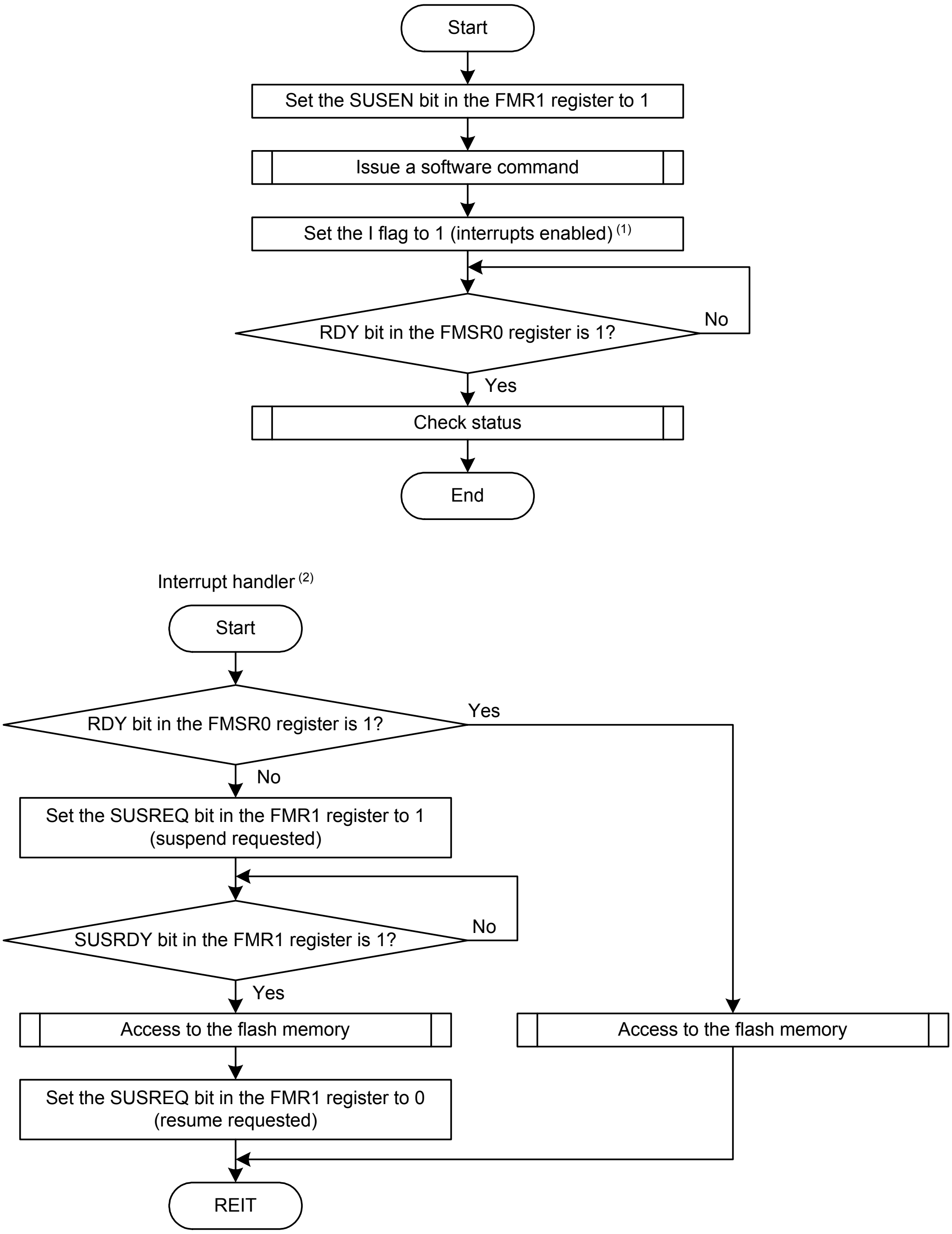
(1) Set the SUSREQ bit in the FMR1 register to 0 (resume requested).
(2) Wait at least 150 μs, or until the RDY bit in the FMSR0 register becomes 1 (ready).
(3) Set the RR bit in the FMR1 register to 0 (reset).
(4) Wait at least 20 μs.
(5) Set the RR bit to 1 (reset released).
(6) Set the LBD bit in the FMR1 register to 1 (lock bit protection disabled).
(7) Erase the corresponding block.

27.3.6.4 Suspend/Resume Usage Procedure

This section describes the usage procedure of suspend/resume function. For issuing software commands, refer to 27.3.4 “Issuing Software Commands”.

Figure 27.22 shows a flow of suspend/resume operation in EW0 mode. In this example, since the resume operation is executed in the interrupt handler, all other interrupts must be disabled. If any of these interrupts must be enabled, do not execute suspend/resume operation in that interrupt handler.

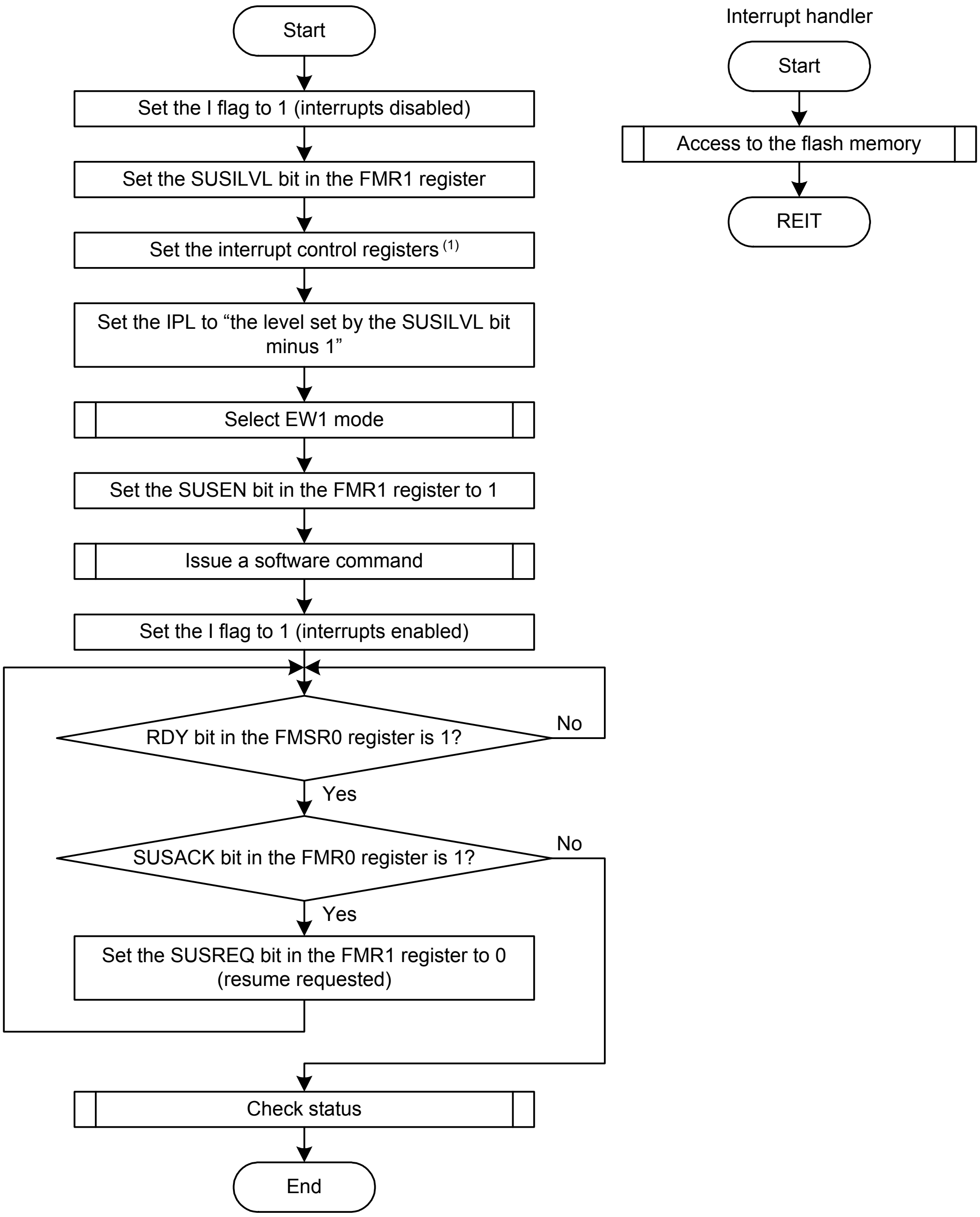
Figure 27.23 shows a flow of suspend/resume operation in EW1 mode. After issuing a software command, the CPU continues to operate unless it accesses the SFR, flash memory, or external bus. To stop the operation, set the I flag to 1 (interrupts enabled) and read the RDY bit in the FMSR0 register. When an interrupt request is generated and the program or erase operation is suspended, the CPU resumes the operation and just after that, the interrupt sequence is executed to branch an interrupt handler. The resume operation is executed in the main routine.


Start
Set the SUSEN bit in the FMR1 register to 1
Issue a software command
Set the I flag to 1 (interrupts enabled) (1)
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End
Interrupt handler (2)
Start
RDY bit in the FMSR0 register is 1?
Yes
No
Set the SUSREQ bit in the FMR1 register to 1 (suspend requested)
SUSRDY bit in the FMR1 register is 1?
No
Yes
Access to the flash memory
Access to the flash memory
Set the SUSREQ bit in the FMR1 register to 0 (resume requested)
REIT

Notes:

1. When not using interrupts, this step is unnecessary.
2. Allocate the respective interrupt vector table and interrupt handler on the RAM.

Figure 27.22 Flow of Suspend/Resume Operation in EW0 Mode


Start
Set the I flag to 1 (interrupts disabled)
Set the SUSILVL bit in the FMR1 register
Set the interrupt control registers (1)
Set the IPL to "the level set by the SUSILVL bit minus 1"
Select EW1 mode
Set the SUSEN bit in the FMR1 register to 1
Issue a software command
Set the I flag to 1 (interrupts enabled)
RDY bit in the FMSR0 register is 1?
No
Yes
SUSACK bit in the FMR0 register is 1?
No
Yes
Set the SUSREQ bit in the FMR1 register to 0 (resume requested)
Check status
End
Interrupt handler
Start
Access to the flash memory
REIT

Note:

1. Set the interrupt to request a suspend to level 6 or 7. When the DMAC II is to be used, set also the DMAC II index. The interrupt control registers can be rewritten in this step only.

Figure 27.23 Flow of Suspend/Resume Operation in EW1 Mode

27.4 Standard Serial I/O Mode

In standard serial I/O mode, an R32C/116A Group compatible serial programmer can be used to rewrite the flash memory while the MCU is mounted on a board.

For further information on the serial programmer, contact your serial programmer manufacturer and refer to the user’s manual included with the serial programmer for instructions.

As shown in Table 27.20, this mode provides two types of transmit/receive mode: Standard serial I/O mode 1 which uses a synchronous serial interface, and standard serial I/O mode 2 which uses UART.

Table 27.20 Standard Serial I/O Mode Specifications

Item		Standard Serial I/O Mode 1	Standard Serial I/O Mode 2
Transmit/receive mode		Synchronous serial I/O	UART
Transmit/receive bit rate		High	Low
Serial interface to be used		UART1	UART1
Pin settings	CNVSS	High	High
	$\overline{\text{CE}}$ (P5_0)	High	High
	$\overline{\text{EPM}}$ (P5_5)	Low	Low
Pin functions	SCLK (P6_5)	In reset: Low In transmission/reception: Transmit/receive clock	In reset: Low In transmission/reception: Unused
	BUSY (P6_4)	BUSY signal	Monitor to check program operation
	RXD (P6_6)	Serial data input	Serial data input
	TXD (P6_7)	Serial data output	Serial data output

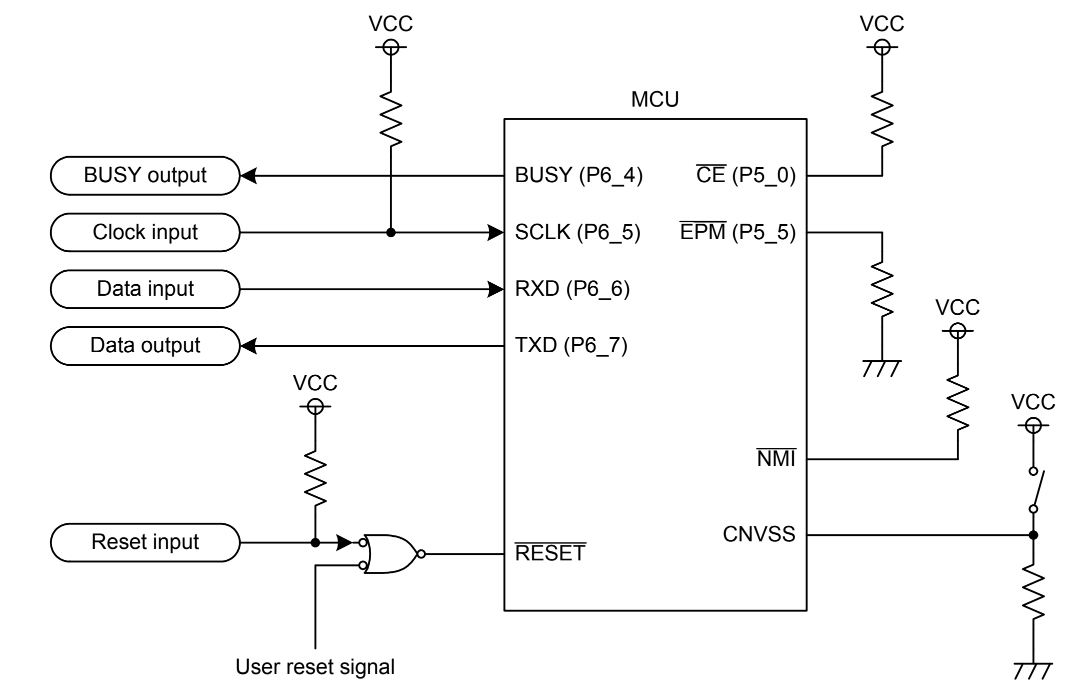
Table 27.21 lists the pin definitions and functions in standard serial I/O mode. Figures 27.24 and 27.25 show examples of a circuit application in standard serial I/O modes 1 and 2, respectively. Refer to the serial programmer user manual to handle pins controlled by the serial programmer.

Table 27.21 Pin Definitions and Functions in Standard Serial I/O Mode

Pin Name	Function	I/O	Description
VCC, VSS	Power supply input	I	Applicable as follows: VCC = guaranteed voltage for program/ erase operations, VSS = 0 V
VDC1, VDC0	Connecting pins for decoupling capacitor	—	A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1
CNVSS	CNVSS	I	This pin should be connected to VCC via a resistor
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While the $\overline{\text{RESET}}$ pin is driven low, at least 20 clock cycles should be input at the XIN pin
XIN	Main clock input	I	A ceramic resonator or a crystal oscillator should be connected between pins XIN and XOUT. An external clock should be input at XIN while leaving XOUT open
XOUT	Main clock output	O	
NSD	Debug port	I/O	This pin should be connected to VCC via a resistor of 1 to 4.7 kΩ
AVCC, AVSS	Analog power supply	I	AVCC and AVSS should be connected to VCC and VSS, respectively
VREF	Reference voltage input	I	Reference voltage input for the A/D converter and D/A converter
P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7	Input port	I	High or low should be input, or the ports should be left open
P5_0	$\overline{\text{CE}}$ input	I	High should be input
P5_1 to P5_4	Input port	I	High or low should be input, or the ports should be left open
P5_5	$\overline{\text{EPM}}$ input	I	Low should be input
P5_6, P5_7, P6_0 to P6_3	Input port	I	High or low should be input, or the ports should be left open
P6_4	BUSY output	O	Standard serial I/O mode 1: BUSY output pin Standard serial I/O mode 2: Program operation monitor
P6_5	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Low should be input
P6_6	Data input RXD	I	Serial data input pin
P6_7	Data output TXD	O	Serial data output pin
P7_0 to P7_7, P8_0 to P8_4	Input port	I	High or low should be input, or the ports should be left open
P8_5	$\overline{\text{NMI}}$ input	I	This pin should be connected to VCC via a resistor
P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [1]	Input port	I	High or low should be input, or the ports should be left open

Note:

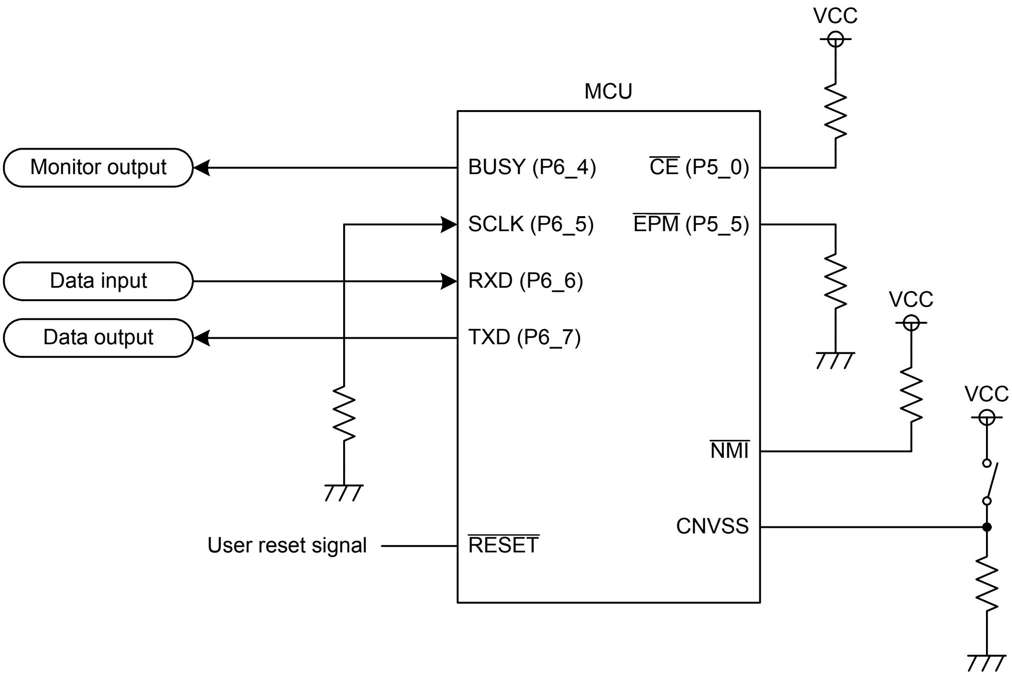
1. Ports P16 to P19 are available in the 176-pin package only.



Notes:

1. Control pins and external circuitry vary with the serial programmer. Refer to the user’s manual included with the serial programmer.
2. In this example, a selector controls the voltage applied to the CNVSS pin to switch between single-chip mode and standard serial I/O mode 1.
3. If the user reset signal becomes low while the MCU is communicating with the serial programmer, cut off the connection between the user reset signal and the RESET pin by, for example, a jumper selector.

Figure 27.24 Circuit Application in Standard Serial I/O Mode 1



Notes:

1. Control pins and external circuitry vary with the serial programmer. Refer to the user’s manual included with the serial programmer.
2. In this example, a selector controls the voltage applied to the CNVSS pin to switch between single-chip mode and standard serial I/O mode 2.
3. If the user reset signal becomes low while the MCU is communicating with the serial programmer, cut off the connection between the user reset signal and the RESET pin by, for example, a jumper selector.

Figure 27.25 Circuit Application in Standard Serial I/O Mode 2

27.5 Parallel I/O mode

In parallel I/O mode, an R32C/116A Group compatible parallel programmer can be used to rewrite the flash memory.

For further information on the parallel programmer, contact your parallel programmer manufacturer and refer to the user’s manual included with your parallel programmer for instructions.

27.6 Notes on Flash Memory Rewriting

27.6.1 Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on the flash memory is in progress. If the supply voltage goes beyond the guaranteed value, the device cannot be guaranteed.

27.6.2 Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on the flash memory is in progress.

27.6.3 Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation on the flash memory in standard serial I/O mode is disabled.

27.6.4 Notes on Programming

- Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode. In addition, do not change the current mode to wait mode or stop mode during CPU rewrite mode.
- The program, block erase, lock bit program, and protect bit program are interrupted by an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt. If any of the software commands above are interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, the lock bit and protect bit values become undefined. Therefore, disable the lock bit, and then execute the block erase command again.
- Do not use CPU rewrite mode in the interrupt handler of non-maskable interrupts.

27.6.5 Notes on Interrupts

- EW0 mode
  - To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
  - When an NMI, watchdog timer interrupt, oscillator stop detection interrupt, or low voltage detection interrupt occurs, the flash memory module automatically enters read array mode. Therefore, these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by the interrupts and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation. Note that the FMR1 register is not reset. Set the SUSEN bit to 0 (suspend function disabled) in an interrupt handler.
  - Instructions BRK, INTO, and UND, which refer to data on the flash memory, cannot be used in this mode.

- EW1 mode
  - Interrupts assigned to the relocatable vector table should not be accepted during program or block erase operation in the area in which the relocatable vector table is.
  - The watchdog timer interrupt should not be generated.
  - When an NMI, watchdog timer interrupt, oscillator stop detection interrupt, or low voltage detection interrupt occurs, the flash memory module automatically enters read array mode. Therefore, these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by the interrupts and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the EWM bit in the FMR0 register to 1 (EW1 mode) and the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation. Note that the FMR1 register is not reset. Set the SUSEN bit to 0 (suspend function disabled) in an interrupt handler.

27.6.6 Notes on Rewrite Control Program

- EW0 mode
  - If the supply voltage drops during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, and the rewrite operation itself may not be performed. In this case, perform the rewrite operation by serial programmer or parallel programmer.

- EW1 mode
  - Do not rewrite blocks having the rewrite control program.

27.6.7 Notes on Number of Program/Erase Cycles and Software Command Execution Time

- The time to execute software commands (program, block erase, lock bit program, and protect bit program) increases as the number of program/erase cycles increases. If the number of program/erase cycles exceeds the endurance value specified in the electrical characteristics, it may take an unpredictable amount of time to execute the software commands. The wait time for executing software commands should be set much longer than the execution time specified in the electrical characteristics.

27.6.8 Other Notes

- The minimum values of program/erase cycles specified in the electrical characteristics are the maximum values that can guarantee the initial performance of the flash memory. The program/erase operation may still be performed even if the number of program/erase cycles exceeds the guaranteed values.
- Chips repeatedly programmed and erased for debugging should not be used for commercial products.

28. Electrical Characteristics

Table 28.1 Absolute Maximum Ratings (1)

Symbol	Characteristic		Condition	Value	Unit
V_{CC}	Supply voltage		$V_{CC} = AV_{CC}$	-0.3 to 6.0	V
AV_{CC}	Analog supply voltage		$V_{CC} = AV_{CC}$	-0.3 to 6.0	V
V_I	Input voltage	XIN, $\overline{\text{RESET}}$, CNVSS, NSD, V_{REF}, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P13_0 to P13_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P17_4 to P17_7, P18_0 to P18_7, P19_0 to P19_7 (2)		-0.3 to V_{CC} + 0.3	V
		P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_3, P12_0 to P12_7, P16_0 to P16_7, P17_0 to P17_3 (2)		-0.3 to 6.0	V
V_O	Output voltage	XOUT, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 (2)		-0.3 to V_{CC} + 0.3	V
P_d	Power consumption		T_a = 25°C	500	mW
—	Operating temperature range			-40 to 85	°C
T_{stg}	Storage temperature range			-65 to 150	°C

Notes:

1. Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. Ports P16 to P19 are available in the 176-pin package only.

Table 28.2 Operating Conditions (1/5) [1]

Symbol	Characteristic			Value			Unit
				Min.	Typ.	Max.	
V_{CC}	Digital supply voltage			3.0	5.0	5.5	V
AV_{CC}	Analog supply voltage				V_{CC}		V
V_{REF}	Reference voltage			3.0		V_{CC}	V
V_{SS}	Digital ground voltage				0		V
AV_{SS}	Analog ground voltage				0		V
dV_{CC}/dt	V_{CC} ramp up rate (V_{CC} < 2.0 V)			0.05			V/ms
V_{IH}	High level input voltage	XIN, $\overline{\text{RESET}}$, CNVSS, NSD, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4 to P8_7 [2], P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P17_4 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [3]		0.8 × V_{CC}		V_{CC}	V
		P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_3, P16_0 to P16_7, P17_0 to P17_3 [3]		0.8 × V_{CC}		6.0	V
		P0_0 to P0_7, P1_0 to P1_7, P13_0 to P13_7	in single-chip mode	0.8 × V_{CC}		V_{CC}	V
			in memory expansion mode or microprocessor mode	0.5 × V_{CC}		V_{CC}	V
		P12_0 to P12_7	in single-chip mode	0.8 × V_{CC}		6.0	V
			in memory expansion mode or microprocessor mode	0.5 × V_{CC}		6.0	V
V_{IL}	Low level input voltage	XIN, $\overline{\text{RESET}}$, CNVSS, NSD, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7 [2], P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [3]		0		0.2 × V_{CC}	V
		P0_0 to P0_7, P1_0 to P1_7, P12_0 to P12_7, P13_0 to P13_7	in single-chip mode	0		0.2 × V_{CC}	V
			in memory expansion mode or microprocessor mode	0		0.16 × V_{CC}	V
T_{opr}	Operating temperature range	N version		-20		85	°C
		D version		-40		85	°C
		P version		-40		85	°C

Notes:

1. The device is operationally guaranteed under these operating conditions.
2. V_{IH} and V_{IL} for P8_7 are specified for P8_7 as a programmable port. These values are not applicable for P8_7 as XCIN.
3. Ports P16 to P19 are available in the 176-pin package only.

Table 28.3 Operating Conditions (2/5)
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted) (1)

Symbol	Characteristic		Value (2)			Unit
			Min.	Typ.	Max.	
C_{VDC}	Decoupling capacitance for voltage regulator	Inter-pin voltage: 1.5 V	2.4		10.0	µF

Notes:
1. The device is operationally guaranteed under these operating conditions.
2. This value should be met with due consideration to the following conditions: operating temperature, DC bias, aging, etc.

Table 28.4 Operating Conditions (3/5)
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted) [1]

Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
$I_{OH(peak)}$	High level peak output current [2]	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [3]			-10.0	mA
$I_{OH(avg)}$	High level average output current [4]	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [3]			-5.0	mA
$I_{OL(peak)}$	Low level peak output current [2]	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [3]			10.0	mA
$I_{OL(avg)}$	Low level average output current [4]	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [3]			5.0	mA

Notes:

1. The device is operationally guaranteed under these operating conditions.
2. The following conditions should be satisfied:
   - The sum of $I_{OL(peak)}$ of ports P0, P1, P2, P8_6, P8_7, P9, P10, P11_0 to P11_4, P14_3 to P14_6, P15, P18_2 to P18_7, P19_0, P19_1, P19_6, and P19_7 is 80 mA or less.
   - The sum of $I_{OL(peak)}$ of ports P3, P4, P5, P6, P7, P8_0 to P8_4, P11_5 to P11_7, P12, P13, P14_7, P16, P17, P18_0, P18_1, and P19_2 to P19_5 is 80 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P0, P1, P2, P11_0 to P11_4, P18_2 to P18_7, P19_0, and P19_1 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P8_6, P8_7, P9, P10, P14_3 to P14_6, P15, P19_6, and P19_7 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P3, P4, P5, P11_5, P11_6, P12, P13, P16, P17_0 to P17_3, and P19_2 to P19_5 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P6, P7, P8_0 to P8_4, P11_7, P14_7, P17_4 to P17_7, P18_0, and P18_1 is -40 mA or less.
3. Ports P16 to P19 are available in the 176-pin package only.
4. Average value within 100 ms.

Table 28.5 Operating Conditions (4/5)
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted) (1)

Symbol	Characteristic	Value			Unit
		Min.	Typ.	Max.	
$f_{(XIN)}$	Main clock oscillator frequency	4		16	MHz
$f_{(XRef)}$	Reference clock frequency	2		4	MHz
$f_{(PLL)}$	PLL clock oscillator frequency	96		128	MHz
$f_{(Base)}$	Base clock frequency			64	MHz
$t_{c(Base)}$	Base clock cycle time	15.625			ns
$f_{(CPU)}$	CPU operating frequency			64	MHz
$t_{c(CPU)}$	CPU clock cycle time	15.625			ns
$f_{(BCLK)}$	Peripheral bus clock operating frequency			32	MHz
$t_{c(BCLK)}$	Peripheral bus clock cycle time	31.25			ns
$f_{(PER)}$	Peripheral clock source frequency			32	MHz
$f_{(XCIN)}$	Sub clock oscillator frequency		32.768	62.5	kHz

Note:

1. The device is operationally guaranteed under these operating conditions.

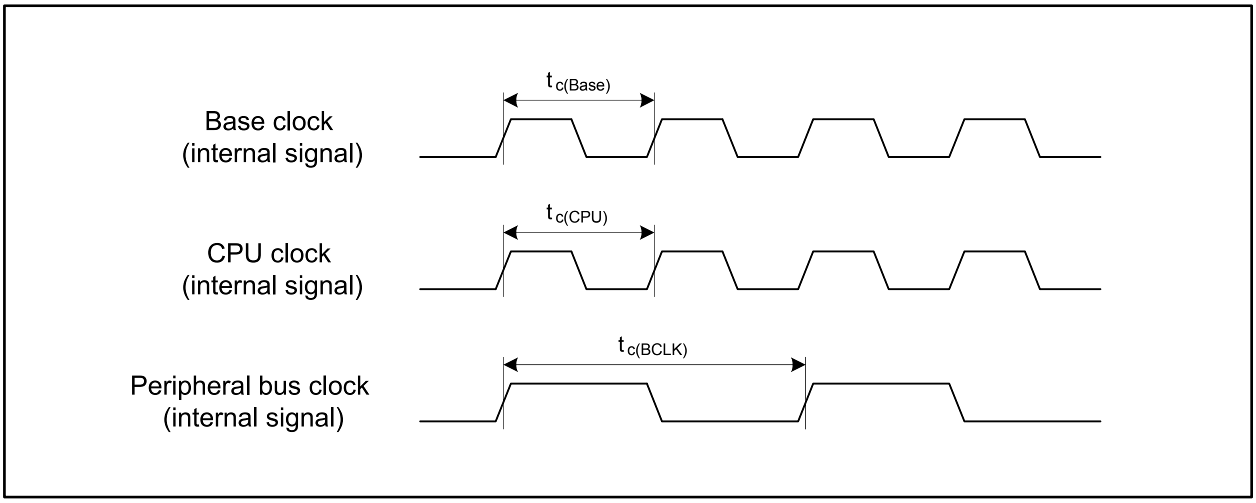


Figure 28.1 Clock Cycle Time

Table 28.6 Operating Conditions (5/5)
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted) [1]

Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
$V_{r(VCC)}$	Allowable ripple voltage	V_{CC} = 5.0 V			0.5	Vp-p
		V_{CC} = 3.0 V			0.3	Vp-p
$dV_{r(VCC)}/dt$	Ripple voltage gradient	V_{CC} = 5.0 V			±0.3	V/ms
		V_{CC} = 3.0 V			±0.3	V/ms
$f_{r(VCC)}$	Allowable ripple frequency				10	kHz

Note:

1. The device is operationally guaranteed under these operating conditions.

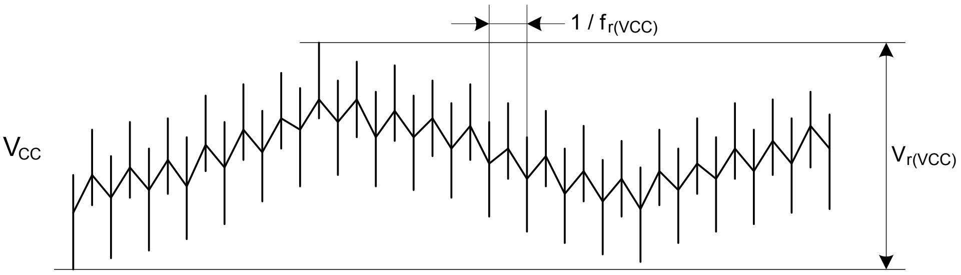


Figure 28.2 Ripple Waveform

Table 28.7 **Electrical Characteristics of RAM**
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and Ta = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
V_{RDR}	RAM data retention voltage	In stop mode	2.0			V

Table 28.8 **Electrical Characteristics of Flash Memory**
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and Ta = T_{opr}, unless otherwise noted)

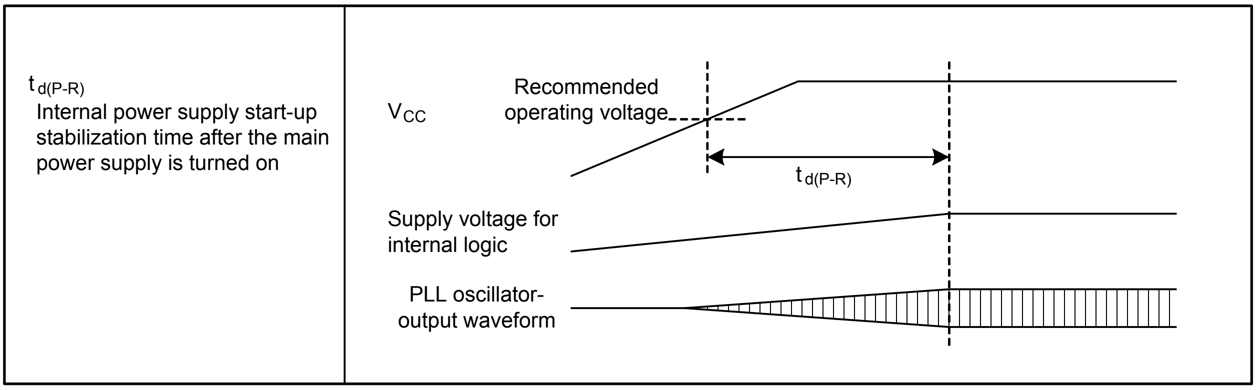
Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
—	Program/erase cycles [1]	Program area	1000			Cycles
		Data area	10000			Cycles
—	4-word program time	Program area		150	900	μs
		Data area		300	1700	μs
—	Lock bit program time	Program area		70	500	μs
		Data area		140	1000	μs
—	Block erasure time	4-Kbyte block		0.12	3.0	s
		32-Kbyte block		0.17	3.0	s
		64-Kbyte block		0.20	3.0	s
t_{SUSP}	Suspend latency				250	μs
—	Data retention [2]	T_a = 55°C [3]	10			Years

Notes:

1. Program/erase definition
   This value represents the number of erasures per block.
   When the number of program/erase cycles is n, each block can be erased n times.
   For example, if a 4-word write is performed in 512 different addresses in the 4-Kbyte block A and then the block is erased, this is counted as a single program/erase operation.
   However, the same address cannot be written to more than once per erasure (overwrite disabled).
2. Data retention includes periods when no supply voltage is applied and no clock is provided.
3. Contact a Renesas Electronics sales office for data retention times other than the above condition.

Table 28.9 Power Supply Circuit Timing Characteristics
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Internal power supply start-up stabilization time after the main power supply is turned on				2	ms



$t_{d(P-R)}$
Internal power supply start-up stabilization time after the main power supply is turned on
V_{CC}
Recommended operating voltage
$t_{d(P-R)}$
Supply voltage for internal logic
PLL oscillator-output waveform

Figure 28.3 Power Supply Circuit Timing

Table 28.10 Electrical Characteristics of Voltage Regulator for Internal Logic
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristics	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
V_{VDC1}	Output voltage			1.5		V

Table 28.11 Electrical Characteristics of Low Voltage Detector
(V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristics	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
ΔVdet	Detected voltage error				±0.3	V
Vdet(R)-Vdet(F)	Hysteresis width		0			V
—	Self-consuming current	V_{CC} = 5.0 V, low voltage detector enabled		4		μA
$t_{d(E-A)}$	Operation start time of low voltage detector				150	μs

Table 28.12 Electrical Characteristics of Oscillator
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristics	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$f_{SO(PLL)}$	PLL clock self-oscillation frequency		35	55	80	MHz
$t_{LOCK(PLL)}$	PLL lock time (1)				1	ms
$t_{jitter(p-p)}$	PLL jitter period (p-p)				2.0	ns
$f_{(OCO)}$	On-chip oscillator frequency		62.5	125	250	kHz

Note:

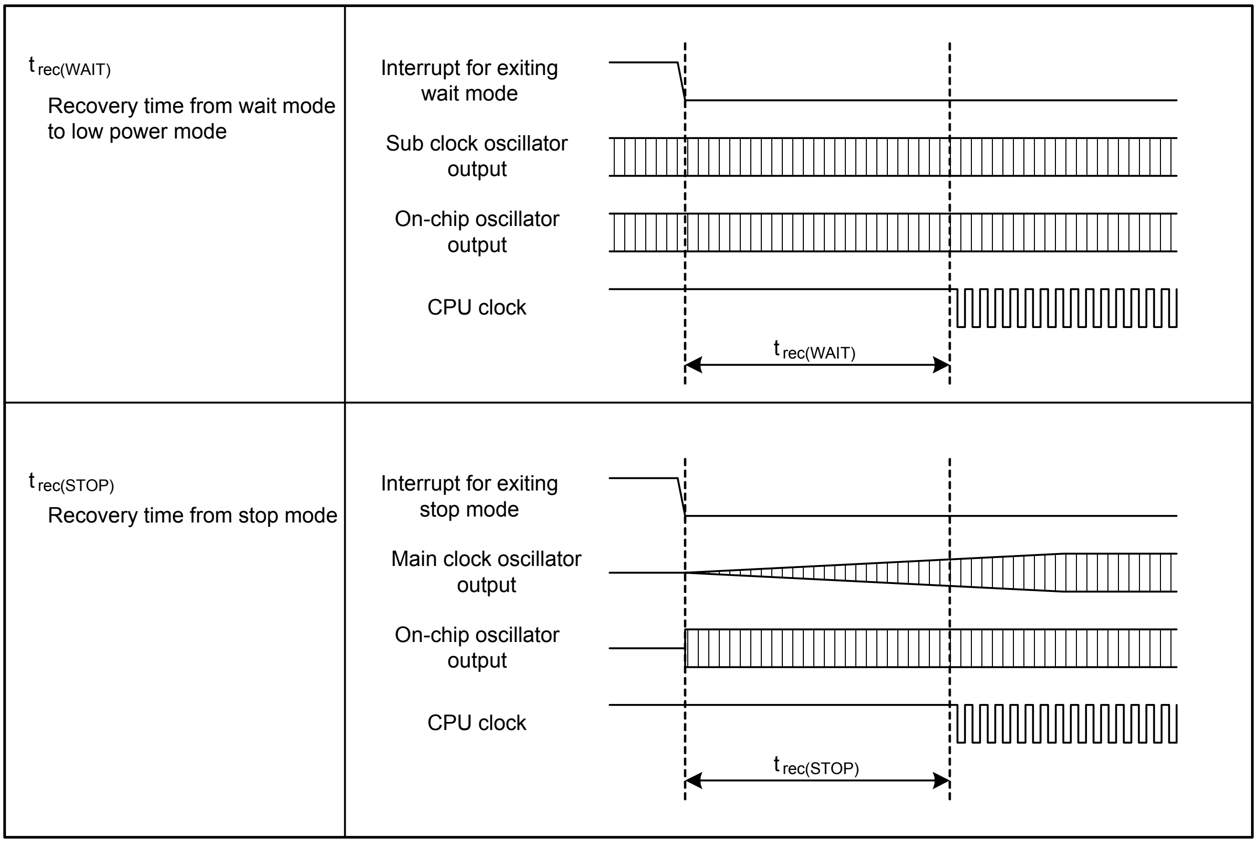
1. This value is applicable only when the main clock oscillation is stable.

Table 28.13 Electrical Characteristics of Clock Circuitry
(V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristics	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$t_{rec(WAIT)}$	Recovery time from wait mode to low power mode				225	µs
$t_{rec(STOP)}$	Recovery time from stop mode (1)				225	µs

Note:

1. The recovery time from stop mode does not include the main clock oscillation stabilization time. The CPU starts operating before the oscillator is stabilized.



$t_{rec(WAIT)}$
Recovery time from wait mode to low power mode
Interrupt for exiting wait mode
Sub clock oscillator output
On-chip oscillator output
CPU clock
$t_{rec(WAIT)}$
$t_{rec(STOP)}$
Recovery time from stop mode
Interrupt for exiting stop mode
Main clock oscillator output
On-chip oscillator output
CPU clock
$t_{rec(STOP)}$

Figure 28.4 Clock Circuit Timing

Timing Requirements (V_{CC} = 3.0 to 5.5 V, V_{SS} = 0 V, and Ta = T_{opr}, unless otherwise noted)

Table 28.14 Flash Memory CPU Rewrite Mode Timing

Symbol	Characteristics	Value		Unit
		Min.	Max.	
t_{cR}	Read cycle time	200		ns
$t_{su(S-R)}$	Chip-select setup time before read	200		ns
$t_{h(R-S)}$	Chip-select hold time after read	0		ns
$t_{su(A-R)}$	Address setup time before read	200		ns
$t_{h(R-A)}$	Address hold time after read	0		ns
$t_{w(R)}$	Read pulse width	100		ns
t_{cW}	Write cycle time	200		ns
$t_{su(S-W)}$	Chip-select setup time before write	0		ns
$t_{h(W-S)}$	Chip-select hold time after write	30		ns
$t_{su(A-W)}$	Address setup time before write	0		ns
$t_{h(W-A)}$	Address hold time after write	30		ns
$t_{w(W)}$	Write pulse width	50		ns

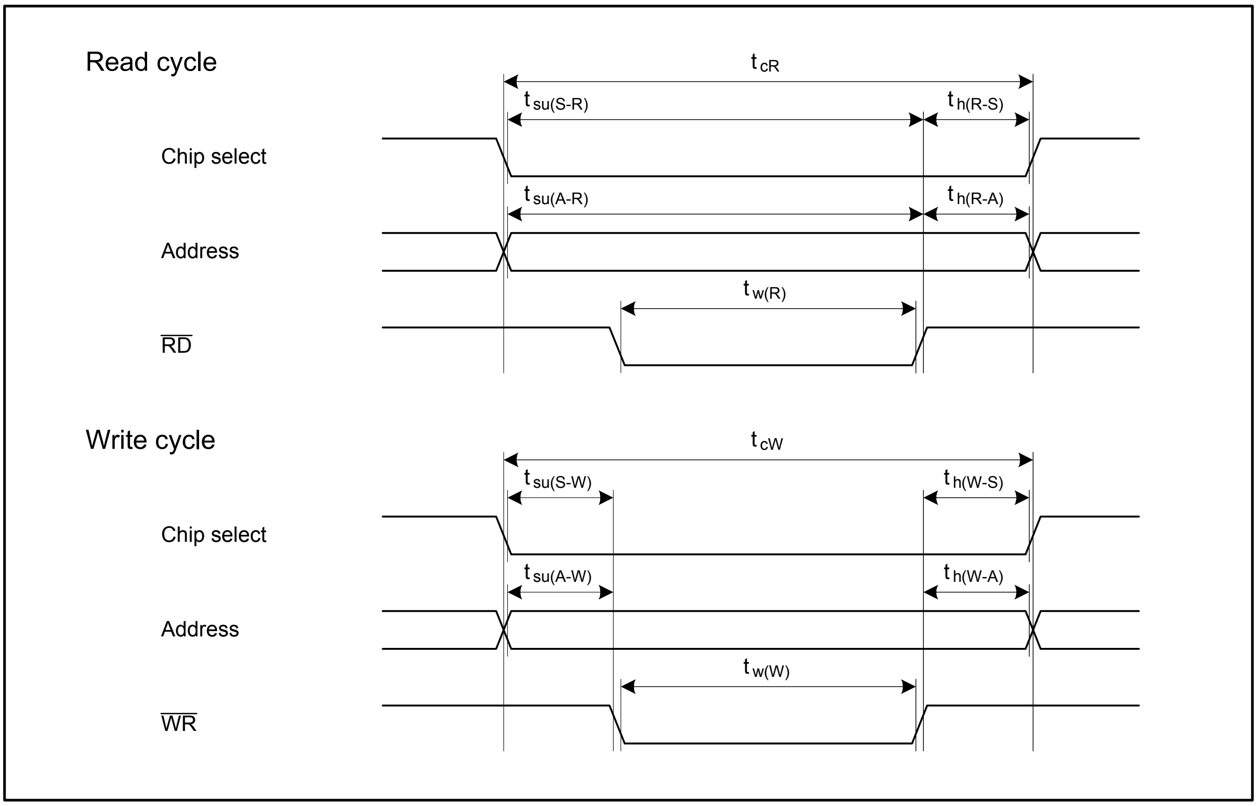


Figure 28.5 Flash Memory CPU Rewrite Mode Timing

V_{CC} = 5 V

Table 28.15 Electrical Characteristics (1/3)
(V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, T_a = T_{opr}, and $f_{(CPU)}$ = 64 MHz, unless otherwise noted)

Symbol	Characteristic		Measurement Condition	Value			Unit
				Min.	Typ.	Max.	
V_{OH}	High level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 (1)	I_{OH} = -5 mA	V_{CC} - 2.0		V_{CC}	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 (1)	I_{OH} = -200 μA	V_{CC} - 0.3		V_{CC}	V
V_{OL}	Low level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 (1)	I_{OL} = 5 mA			2.0	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 (1)	I_{OL} = 200 μA			0.45	V

Note:

1. Ports P16 to P19 are available in the 176-pin package only.

V_{CC} = 5 V

Table 28.16 Electrical Characteristics (2/3)
(V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, T_a = T_{opr}, and $f_{(CPU)}$ = 64 MHz, unless otherwise noted)

Symbol	Characteristic		Measurement Condition	Value			Unit
				Min.	Typ.	Max.	
V_{T+} - V_{T-}	Hysteresis	$\overline{\mathrm{HOLD}}$, $\overline{\mathrm{RDY}}$, $\overline{\mathrm{NMI}}$, $\overline{\mathrm{INT0}}$ to $\overline{\mathrm{INT8}}$, $\overline{\mathrm{KI0}}$ to $\overline{\mathrm{KI3}}$, TA0IN to TA4IN, TA0OUT to TA4OUT, TB0IN to TB5IN, $\overline{\mathrm{CTS0}}$ to $\overline{\mathrm{CTS10}}$, CLK0 to CLK10, RXD0 to RXD10, SCL0 to SCL6, SDA0 to SDA6, $\overline{\mathrm{SS0}}$ to $\overline{\mathrm{SS6}}$, SRXD0 to SRXD6, $\overline{\mathrm{ADTRG}}$, IIO0_0 to IIO0_7, IIO1_0 to IIO1_7, UD0A, UD0B, UD1A, UD1B, ISCLK2, ISRXD2, IEIN, MSCL, MSDA		0.2		1.0	V
		$\overline{\mathrm{RESET}}$		0.2		1.8	V
I_{IH}	High level input current	XIN, $\overline{\mathrm{RESET}}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [1]	V_I = 5 V			5.0	µA
I_{IL}	Low level input current	XIN, $\overline{\mathrm{RESET}}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [1]	V_I = 0 V			-5.0	µA
R_{PULLUP}	Pull-up resistor	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P13_0 to P13_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P17_4 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [1]	V_I = 0 V	30	50	170	kΩ
R_{fXIN}	Feedback resistor	XIN			1.5		MΩ
R_{fXCIN}	Feedback resistor	XCIN			15		MΩ

Note:

[1] Ports P16 to P19 are available in the 176-pin package only.

V_{CC} = 5 V

Table 28.17 Electrical Characteristics (3/3)
(V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement Condition		Value			Unit
				Min.	Typ.	Max.	
I_{CC}	Power supply current	In single-chip mode, output pins are left open and others are connected to V_{SS} XIN-XOUT Drive strength: low XCIN-XCOUT Drive strength: low	$f_{(CPU)}$ = 64 MHz, $f_{(BCLK)}$ = 32 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, PLL, Stopped: XCIN, OCO		45	60	mA
			$f_{(CPU)}$ = $f_{SO(PLL)}$/24 MHz, Active: PLL (self-oscillation), Stopped: XIN, XCIN, OCO		12		mA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, Stopped: PLL, XCIN, OCO		1.2		mA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz, Active: XCIN, Stopped: XIN, PLL, OCO, Main regulator: shutdown		220		µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz, Active: OCO, Stopped: XIN, PLL, XCIN, Main regulator: shutdown		230		µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, Stopped: PLL, XCIN, OCO, T_a = 25°C, Wait mode		960	1600	µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz, Active: XCIN, Stopped: XIN, PLL, OCO, Main regulator: shutdown, T_a = 25°C, Wait mode		8	140	µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz, Active: OCO, Stopped: XIN, PLL, XCIN, Main regulator: shutdown, T_a = 25°C, Wait mode		10	150	µA
			Stopped: all clocks, Main regulator: shutdown, T_a = 25°C		5	70	µA

V_{CC} = 5 V

Table 28.18 A/D Conversion Characteristics (V_{CC} = AV_{CC} = V_{REF} = 4.2 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = T_{opr}, and $f_{(BCLK)}$ = 32 MHz, unless otherwise noted)

Symbol	Characteristic	Measurement Condition		Value			Unit
				Min.	Typ.	Max.	
—	Resolution	V_{REF} = V_{CC}				10	Bits
—	Absolute error	V_{REF} = V_{CC} = 5 V	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1			±3	LSB
			External op-amp connection mode			±7	LSB
INL	Integral non-linearity error	V_{REF} = V_{CC} = 5 V	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1			±3	LSB
			External op-amp connection mode			±7	LSB
DNL	Differential non-linearity error					±1	LSB
—	Offset error					±3	LSB
—	Gain error					±3	LSB
R_{LADDER}	Resistor ladder	V_{REF} = V_{CC}		4		20	kΩ
t_{CONV}	Conversion time (10 bits)	ϕ_{AD} = 16 MHz, with sample and hold function		2.06			μs
		ϕ_{AD} = 16 MHz, without sample and hold function		3.69			μs
t_{CONV}	Conversion time (8 bits)	ϕ_{AD} = 16 MHz, with sample and hold function		1.75			μs
		ϕ_{AD} = 16 MHz, without sample and hold function		3.06			μs
t_{SAMP}	Sampling time	ϕ_{AD} = 16 MHz		0.188			μs
V_{IA}	Analog input voltage			0		V_{REF}	V
ϕ_{AD}	Operating clock frequency	Without sample and hold function		0.25		16	MHz
		With sample and hold function		1		16	MHz
$R_{PU(AST)}$	Pull-up resistor for open-circuit detection			5	10	15	kΩ
$R_{PD(AST)}$	Pull-down resistor for open-circuit detection			5	10	15	kΩ

V_{CC} = 5 V

Table 28.19 D/A Conversion Characteristics (V_{CC} = AV_{CC} = V_{REF} = 4.2 to 5.5 V, V_{SS} = AV_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute precision				1.0	%
t_S	Settling time				3	μs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference input current	See Note 1			1.5	mA

Note:

1. One D/A converter is used. The DAi register (i = 0, 1) of the other unused converter is set to 00h. The resistor ladder for the A/D converter is not considered.
   Even when the VCUT bit in the AD0CON1 register is set to 0 (V_{REF} disconnected), I_{VREF} is supplied.

V_{CC} = 5 V

Timing Requirements (V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.20 External Clock Input

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(X)}$	External clock input period	62.5	250	ns
$t_{w(XH)}$	External clock input high level pulse width	25		ns
$t_{w(XL)}$	External clock input low level pulse width	25		ns
$t_{r(X)}$	External clock input rise time		5	ns
$t_{f(X)}$	External clock input fall time		5	ns
t_w / t_c	External clock input duty	40	60	%

Table 28.21 External Bus Timing

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{su(D-R)}$	Data setup time before read	40		ns
$t_{h(R-D)}$	Data hold time after read	0		ns
$t_{dis(R-D)}$	Data disable time after read		$0.5 \times t_{c(Base)} + 10$	ns

V_{CC} = 5 V

Timing Requirements (V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.22 Timer A Input (counting input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.23 Timer A Input (gating input in timer mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	400		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	180		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	180		ns

Table 28.24 Timer A Input (external trigger input in one-shot timer mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.25 Timer A Input (external trigger input in pulse-width modulation mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.26 Timer A Input (increment/decrement switching input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input clock cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input high level pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input low level pulse width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	400		ns

V_{CC} = 5 V

Timing Requirements (V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.27 Timer B Input (counting input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time (one edge counting)	200		ns
$t_{w(TBH)}$	TBiIN input high level pulse width (one edge counting)	80		ns
$t_{w(TBL)}$	TBiIN input low level pulse width (one edge counting)	80		ns
$t_{c(TB)}$	TBiIN input clock cycle time (both edges counting)	200		ns
$t_{w(TBH)}$	TBiIN input high level pulse width (both edges counting)	80		ns
$t_{w(TBL)}$	TBiIN input low level pulse width (both edges counting)	80		ns

Table 28.28 Timer B Input (pulse period measure mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{w(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{w(TBL)}$	TBiIN input low level pulse width	180		ns

Table 28.29 Timer B Input (pulse-width measure mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{w(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{w(TBL)}$	TBiIN input low level pulse width	180		ns

$V_{CC} = 5\ V$

Timing Requirements (V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and $T_a = T_{opr}$, unless otherwise noted)

Table 28.30 Serial Interface

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input clock cycle time	200		ns
$t_{w(CKH)}$	CLKi input high level pulse width	80		ns
$t_{w(CKL)}$	CLKi input low level pulse width	80		ns
$t_{su(D-C)}$	RXDi input setup time	80		ns
$t_{h(C-D)}$	RXDi input hold time	90		ns

Table 28.31 A/D Trigger Input

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{w(ADH)}$	$\overline{\text{ADTRG}}$ input high level pulse width Hardware trigger input high level pulse width	$\frac{3}{\phi_{AD}}$		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ input low level pulse width Hardware trigger input high level pulse width	125		ns

Table 28.32 External Interrupt $\overline{\text{INTi}}$ Input

Symbol	Characteristic		Value		Unit
			Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input high level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)}$ + 200		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input low level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)}$ + 200		ns

Table 28.33 Intelligent I/O

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(ISCLK2)}$	ISCLK2 input clock cycle time	600		ns
$t_{w(ISCLK2H)}$	ISCLK2 input high level pulse width	270		ns
$t_{w(ISCLK2L)}$	ISCLK2 input low level pulse width	270		ns
$t_{su(RXD-ISCLK2)}$	ISRXD2 input setup time	150		ns
$t_{h(ISCLK2-RXD)}$	ISRXD2 input hold time	100		ns

V_{CC} = 5 V

Timing Requirements (V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.34 Multi-master I²C-bus Interface

Symbol	Characteristic	Value				Unit
		Standard-mode		Fast-mode		
		Min.	Max.	Min.	Max.	
$t_{w(SCLH)}$	MSCL input high level pulse width	600		600		ns
$t_{w(SCLL)}$	MSCL input low level pulse width	600		600		ns
$t_{r(SCL)}$	MSCL input rise time		1000		300	ns
$t_{f(SCL)}$	MSCL input fall time		300		300	ns
$t_{r(SDA)}$	MSDA input rise time		1000		300	ns
$t_{f(SDA)}$	MSDA input fall time		300		300	ns
$t_{h(SDA-SCL)S}$	MSCL high level hold time after start condition/restart condition	(1)		2 × $t_{c(\phi IIC)}$ + 40		ns
$t_{su(SCL-SDA)P}$	MSCL high level setup time for restart condition/stop condition	(1)		2 × $t_{c(\phi IIC)}$ + 40		ns
$t_{w(SDAH)P}$	MSDA high level pulse width after stop condition	(1)		4 × $t_{c(\phi IIC)}$ + 40		ns
$t_{su(SDA-SCL)}$	MSDA input setup time	100		100		ns
$t_{h(SCL-SDA)}$	MSDA input hold time	0		0		ns

Note:

1. The value is calculated by the following formulas based on a value SSC by setting bits SSC4 to SSC0 in the I2CSSCR register:

   $t_{h(SDA-SCL)S}$ = SSC ÷ 2 × $t_{c(\phi IIC)}$ + 40 [ns]

   $t_{su(SCL-SDA)P}$ = (SSC ÷ 2 + 1) × $t_{c(\phi IIC)}$ + 40 [ns]

   $t_{w(SDAH)P}$ = (SSC + 1) × $t_{c(\phi IIC)}$ + 40 [ns]

V_{CC} = 5 V

Switching Characteristics (V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.35 External Bus Timing (separate bus)

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{su(S-R)}$	Chip-select setup time before read	Refer to Figure 28.6	(1)		ns
$t_{h(R-S)}$	Chip-select hold time after read		$t_{c(Base)}$ -15		ns
$t_{su(A-R)}$	Address setup time before read		(1)		ns
$t_{h(R-A)}$	Address hold time after read		$t_{c(Base)}$ -15		ns
$t_{w(R)}$	Read pulse width		(1)		ns
$t_{su(S-W)}$	Chip-select setup time before write		(1)		ns
$t_{h(W-S)}$	Chip-select hold time after write		1.5 × $t_{c(Base)}$ -15		ns
$t_{su(A-W)}$	Address setup time before write		(1)		ns
$t_{h(W-A)}$	Address hold time after write		1.5 × $t_{c(Base)}$ -15		ns
$t_{w(W)}$	Write pulse width		(1)		ns
$t_{su(D-W)}$	Data setup time before write		(1)		ns
$t_{h(W-D)}$	Data hold time after write		0		ns

Note:

1. The value is calculated using the formulas below based on the base clock cycles ($t_{c(Base)}$) and respective cycles of Tsu(A-R), Tw(R), Tsu(A-W), and Tw(W) set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For details on how to set values, refer to 9.3.5 “External Bus Timing”.

   $t_{su(S-R)} = t_{su(A-R)} = Tsu(A-R) \times t_{c(Base)} - 15$ [ns]

   $t_{w(R)} = Tw(R) \times t_{c(Base)} - 10$ [ns]

   $t_{su(S-W)} = t_{su(A-W)} = Tsu(A-W) \times t_{c(Base)} - 15$ [ns]

   $t_{w(W)} = t_{su(D-W)} = Tw(W) \times t_{c(Base)} - 10$ [ns]

V_{CC} = 5 V

Switching Characteristics (V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and $T_a = T_{opr}$, unless otherwise noted)

Table 28.36 External Bus Timing (multiplexed bus)

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{su(S-ALE)}$	Chip-select setup time before ALE	Refer to Figure 28.6	(1)		ns
$t_{h(R-S)}$	Chip-select hold time after read		1.5 × $t_{c(Base)}$ -15		ns
$t_{su(A-ALE)}$	Address setup time before ALE		(1)		ns
$t_{h(ALE-A)}$	Address hold time after ALE		$t_{c(Base)}$ - 5 (2)		ns
$t_{h(R-A)}$	Address hold time after read		1.5 × $t_{c(Base)}$ -15		ns
$t_{d(ALE-R)}$	ALE-read delay time		$t_{c(Base)}$ - 5 (2)	$t_{c(Base)}$ + 10 (2)	ns
$t_{w(ALE)}$	ALE pulse width		(1)		ns
$t_{dis(R-A)}$	Address disable time after read			8	ns
$t_{w(R)}$	Read pulse width		(1)		ns
$t_{h(W-S)}$	Chip-select hold time after write		1.5 × $t_{c(Base)}$ -15		ns
$t_{h(W-A)}$	Address hold time after write		1.5 × $t_{c(Base)}$ -15		ns
$t_{d(ALE-W)}$	ALE-write delay time		$t_{c(Base)}$ - 5 (2)	$t_{c(Base)}$ + 10 (2)	ns
$t_{w(W)}$	Write pulse width		(1)		ns
$t_{su(D-W)}$	Data setup time before write		(1)		ns
$t_{h(W-D)}$	Data hold time after write		0.5 × $t_{c(Base)}$		ns

Notes:

1. The value is calculated using the formulas below based on the base clock cycles ($t_{c(Base)}$) and respective cycles of Tsu(A-R), Tw(R), Tsu(A-W), and Tw(W) set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For details on how to set values, refer to 9.3.5 “External Bus Timing”.
   $t_{su(S-ALE)} = t_{su(A-ALE)} = t_{w(ALE)}$ = (Tsu(A-R) - 1) × $t_{c(Base)}$ -15 [ns]
   (when Tsu(A-R) is greater than 1)
   $t_{su(S-ALE)} = t_{su(A-ALE)} = t_{w(ALE)}$ = 0.5 × $t_{c(Base)}$ -15 [ns] (when Tsu(A-R) is 1)
   $t_{w(R)}$ = Tw(R) × $t_{c(Base)}$ -10 [ns]
   $t_{w(W)} = t_{su(D-W)}$ = Tw(W) × $t_{c(Base)}$ -10 [ns]
2. When Tsu(A-R) is greater than 1 or Tsu(A-W) is greater than 1. Change “$t_{c(Base)}$” to “0.5 × $t_{c(Base)}$” when Tsu(A-R) is 1 or Tsu(A-W) is 1.

$V_{CC} = 5\text{ V}$

Switching Characteristics (V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V, and $T_a = T_{opr}$, unless otherwise noted)

Table 28.37 Serial Interface

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{d(C-Q)}$	TXDi output delay time	Refer to Figure 28.6		80	ns
$t_{h(C-Q)}$	TXDi output hold time		0		ns

Table 28.38 Intelligent I/O

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{d(ISCLK2-TXD)}$	ISTXD2 output delay time	Refer to Figure 28.6		180	ns
$t_{h(ISCLK2-RXD)}$	ISTXD2 output hold time		0		ns

Table 28.39 Multi-master I²C-bus Interface (standard-mode)

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{f(SCL)}$	MSCL output fall time	Refer to Figure 28.6	2		ns
$t_{f(SDA)}$	MSDA output fall time		2		ns
$t_{d(SDA-SCL)S}$	MSCL output delay time after start condition/restart condition		$20 \times t_{c(\phi IIC)} - 120$	$52 \times t_{c(\phi IIC)} - 40$	ns
$t_{d(SCL-SDA)P}$	Restart condition/stop condition output delay time after MSCL becomes high		$20 \times t_{c(\phi IIC)} + 40$	$52 \times t_{c(\phi IIC)} + 120$	ns
$t_{d(SCL-SDA)}$	MSDA output delay time		$2 \times t_{c(\phi IIC)} + 40$	$3 \times t_{c(\phi IIC)} + 120$	ns

Table 28.40 Multi-master I²C-bus Interface (fast-mode)

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{f(SCL)}$	MSCL output fall time	Refer to Figure 28.6	2 (1)		ns
$t_{f(SDA)}$	MSDA output fall time		2 (1)		ns
$t_{d(SDA-SCL)S}$	MSCL output delay time after start condition/restart condition		$10 \times t_{c(\phi IIC)} - 120$	$26 \times t_{c(\phi IIC)} - 40$	ns
$t_{d(SCL-SDA)P}$	Restart condition/stop condition output delay time after MSCL becomes high		$10 \times t_{c(\phi IIC)} + 40$	$26 \times t_{c(\phi IIC)} + 120$	ns
$t_{d(SCL-SDA)}$	MSDA output delay time		$2 \times t_{c(\phi IIC)} + 40$	$3 \times t_{c(\phi IIC)} + 120$	ns

Note:

1. External circuits are required to satisfy the I²C-bus specification.

V_{CC} = 3.3 V

Table 28.41 Electrical Characteristics (1/3) (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, T_a = T_{opr}, and $f_{(CPU)}$ = 64 MHz, unless otherwise noted)

Symbol	Characteristic		Measurement Condition	Value			Unit
				Min.	Typ.	Max.	
V_{OH}	High level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 (1)	I_{OH} = -1 mA	V_{CC} - 0.6		V_{CC}	V
V_{OL}	Low level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 (1)	I_{OL} = 1 mA			0.5	V

Note:

1. Ports P16 to P19 are available in the 176-pin package only.

V_{CC} = 3.3 V

Table 28.42 Electrical Characteristics (2/3) (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, T_a = T_{opr}, and $f_{(CPU)}$ = 64 MHz, unless otherwise noted)

Symbol	Characteristic		Measurement Condition	Value			Unit
				Min.	Typ.	Max.	
V_{T+} - V_{T-}	Hysteresis	$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, $\overline{\text{NMI}}$, $\overline{\text{INT0}}$ to $\overline{\text{INT8}}$, $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$, TA0IN to TA4IN, TA0OUT to TA4OUT, TB0IN to TB5IN, $\overline{\text{CTS0}}$ to $\overline{\text{CTS10}}$, CLK0 to CLK10, RXD0 to RXD10, SCL0 to SCL6, SDA0 to SDA6, $\overline{\text{SS0}}$ to $\overline{\text{SS6}}$, SRXD0 to SRXD6, $\overline{\text{ADTRG}}$, IIO0_0 to IIO0_7, IIO1_0 to IIO1_7, UD0A, UD0B, UD1A, UD1B, ISCLK2, ISRXD2, IEIN, MSCL, MSDA		0.2		1.0	V
		$\overline{\text{RESET}}$		0.2		1.8	V
I_{IH}	High level input current	XIN, $\overline{\text{RESET}}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [1]	V_I = 3.3 V			4.0	µA
I_{IL}	Low level input current	XIN, $\overline{\text{RESET}}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P16_0 to P16_7, P17_0 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [1]	V_I = 0 V			-4.0	µA
R_{PULLUP}	Pull-up resistor	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P13_0 to P13_7, P14_1, P14_3 to P14_7, P15_0 to P15_7, P17_4 to P17_7, P18_0 to P18_7, P19_0 to P19_7 [1]	V_I = 0 V	50	100	500	kΩ
R_{fXIN}	Feedback resistor	XIN			3		MΩ
R_{fXCIN}	Feedback resistor	XCIN			25		MΩ

Note:

1. Ports P16 to P19 are available in the 176-pin package only.

V_{CC} = 3.3 V

Table 28.43 Electrical Characteristics (3/3)
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement Condition		Value			Unit
				Min.	Typ.	Max.	
I_{CC}	Power supply current	In single-chip mode, output pins are left open and others are connected to V_{SS} XIN-XOUT Drive strength: low XCIN-XCOUT Drive strength: low	$f_{(CPU)}$ = 64 MHz, $f_{(BCLK)}$ = 32 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, PLL, Stopped: XCIN, OCO		40	55	mA
			$f_{(CPU)}$ = $f_{SO(PLL)}$/24 MHz, Active: PLL (self-oscillation), Stopped: XIN, XCIN, OCO		9		mA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, Stopped: PLL, XCIN, OCO		670		µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz, Active: XCIN, Stopped: XIN, PLL, OCO, Main regulator: shutdown		180		µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz, Active: OCO, Stopped: XIN, PLL, XCIN, Main regulator: shutdown		190		µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, Stopped: PLL, XCIN, OCO, T_a = 25°C, Wait mode		500	900	µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz, Active: XCIN, Stopped: XIN, PLL, OCO, Main regulator: shutdown, T_a = 25°C, Wait mode		8	140	µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz, Active: OCO, Stopped: XIN, PLL, XCIN, Main regulator: shutdown, T_a = 25°C, Wait mode		10	150	µA
			Stopped: all clocks, Main regulator: shutdown, T_a = 25°C		5	70	µA

V_{CC} = 3.3 V

Table 28.44 A/D Conversion Characteristics (V_{CC} = AV_{CC} = V_{REF} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = 0 V, T_a = T_{opr}, and $f_{(BCLK)}$ = 32 MHz, unless otherwise noted)

Symbol	Characteristic	Measurement Condition		Value			Unit
				Min.	Typ.	Max.	
—	Resolution	V_{REF} = V_{CC}				10	Bits
—	Absolute error	V_{REF} = V_{CC} = 3.3 V	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1			±5	LSB
			External op-amp connection mode			±7	LSB
INL	Integral non-linearity error	V_{REF} = V_{CC} = 3.3 V	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1			±5	LSB
			External op-amp connection mode			±7	LSB
DNL	Differential non-linearity error	V_{REF} = V_{CC} = 3.3 V				±1	LSB
—	Offset error					±3	LSB
—	Gain error					±3	LSB
R_{LADDER}	Resistor ladder	V_{REF} = V_{CC}		4		20	kΩ
t_{CONV}	Conversion time (10 bits)	ϕ_{AD} = 10 MHz, with sample and hold function		3.3			μs
t_{CONV}	Conversion time (8 bits)	ϕ_{AD} = 10 MHz, with sample and hold function		2.8			μs
t_{SAMP}	Sampling time	ϕ_{AD} = 10 MHz		0.3			μs
V_{IA}	Analog input voltage			0		V_{REF}	V
ϕ_{AD}	Operating clock frequency	Without sample and hold function		0.25		10	MHz
		With sample and hold function		1		10	MHz
$R_{PU(AST)}$	Pull-up resistor for open-circuit detection			5	10	15	kΩ
$R_{PD(AST)}$	Pull-down resistor for open-circuit detection			5	10	15	kΩ

V_{CC} = 3.3 V

Table 28.45 D/A Conversion Characteristics (V_{CC} = AV_{CC} = V_{REF} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute precision				1.0	%
t_S	Settling time				3	μs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference input current	See Note 1			1.0	mA

Note:

1. One D/A converter is used. The DAi register (i = 0, 1) of the other unused converter is set to 00h. The resistor ladder for the A/D converter is not considered.
   Even when the VCUT bit in the AD0CON1 register is set to 0 (V_{REF} disconnected), I_{VREF} is supplied.

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.46 External Clock Input

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(X)}$	External clock input period	62.5	250	ns
$t_{w(XH)}$	External clock input high level pulse width	25		ns
$t_{w(XL)}$	External clock input low level pulse width	25		ns
$t_{r(X)}$	External clock input rise time		5	ns
$t_{f(X)}$	External clock input fall time		5	ns
t_w / t_c	External clock input duty	40	60	%

Table 28.47 External Bus Timing

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{su(D-R)}$	Data setup time before read	40		ns
$t_{h(R-D)}$	Data hold time after read	0		ns
$t_{dis(R-D)}$	Data disable time after read		0.5 × $t_{c(Base)}$ + 10	ns

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.48 Timer A Input (counting input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.49 Timer A Input (gating input in timer mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	400		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	180		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	180		ns

Table 28.50 Timer A Input (external trigger input in one-shot timer mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.51 Timer A Input (external trigger input in pulse-width modulation mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.52 Timer A Input (increment/decrement switching input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input clock cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input high level pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input low level pulse width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	400		ns

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.53 Timer B Input (counting input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time (one edge counting)	200		ns
$t_{w(TBH)}$	TBiIN input high level pulse width (one edge counting)	80		ns
$t_{w(TBL)}$	TBiIN input low level pulse width (one edge counting)	80		ns
$t_{c(TB)}$	TBiIN input clock cycle time (both edges counting)	200		ns
$t_{w(TBH)}$	TBiIN input high level pulse width (both edges counting)	80		ns
$t_{w(TBL)}$	TBiIN input low level pulse width (both edges counting)	80		ns

Table 28.54 Timer B Input (pulse period measure mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{w(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{w(TBL)}$	TBiIN input low level pulse width	180		ns

Table 28.55 Timer B Input (pulse-width measure mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{w(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{w(TBL)}$	TBiIN input low level pulse width	180		ns

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.56 Serial Interface

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input clock cycle time	200		ns
$t_{w(CKH)}$	CLKi input high level pulse width	80		ns
$t_{w(CKL)}$	CLKi input low level pulse width	80		ns
$t_{su(D-C)}$	RXDi input setup time	80		ns
$t_{h(C-D)}$	RXDi input hold time	90		ns

Table 28.57 A/D Trigger Input

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{w(ADH)}$	$\overline{\text{ADTRG}}$ input high level pulse width Hardware trigger input high level pulse width	$\frac{3}{\phi_{AD}}$		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ input low level pulse width Hardware trigger input high level pulse width	125		ns

Table 28.58 External Interrupt $\overline{\text{INTi}}$ Input

Symbol	Characteristic		Value		Unit
			Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input high level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)}$ + 200		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input low level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)}$ + 200		ns

Table 28.59 Intelligent I/O

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(ISCLK2)}$	ISCLK2 input clock cycle time	600		ns
$t_{w(ISCLK2H)}$	ISCLK2 input high level pulse width	270		ns
$t_{w(ISCLK2L)}$	ISCLK2 input low level pulse width	270		ns
$t_{su(RXD-ISCLK2)}$	ISRXD2 input setup time	150		ns
$t_{h(ISCLK2-RXD)}$	ISRXD2 input hold time	100		ns

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.60 Multi-master I²C-bus Interface

Symbol	Characteristic	Value				Unit
		Standard-mode		Fast-mode		
		Min.	Max.	Min.	Max.	
$t_{w(SCLH)}$	MSCL input high level pulse width	600		600		ns
$t_{w(SCLL)}$	MSCL input low level pulse width	600		600		ns
$t_{r(SCL)}$	MSCL input rise time		1000		300	ns
$t_{f(SCL)}$	MSCL input fall time		300		300	ns
$t_{r(SDA)}$	MSDA input rise time		1000		300	ns
$t_{f(SDA)}$	MSDA input fall time		300		300	ns
$t_{h(SDA-SCL)S}$	MSCL high level hold time after start condition/restart condition	(1)		2 × $t_{c(\phi IIC)}$ + 40		ns
$t_{su(SCL-SDA)P}$	MSCL high level setup time for restart condition/stop condition	(1)		2 × $t_{c(\phi IIC)}$ + 40		ns
$t_{w(SDAH)P}$	MSDA high level pulse width after stop condition	(1)		4 × $t_{c(\phi IIC)}$ + 40		ns
$t_{su(SDA-SCL)}$	MSDA input setup time	100		100		ns
$t_{h(SCL-SDA)}$	MSDA input hold time	0		0		ns

Note:

1. The value is calculated using the formulas below based on a value SSC set by bits SSC4 to SSC0 in the I2CSSCR register:

$t_{h(SDA-SCL)S}$ = SSC ÷ 2 × $t_{c(\phi IIC)}$ + 40 [ns]

$t_{su(SCL-SDA)P}$ = (SSC ÷ 2 + 1) × $t_{c(\phi IIC)}$ + 40 [ns]

$t_{w(SDAH)P}$ = (SSC + 1) × $t_{c(\phi IIC)}$ + 40 [ns]

V_{CC} = 3.3 V

Switching Characteristics (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.61 External Bus Timing (separate bus)

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{su(S-R)}$	Chip-select setup time before read	Refer to Figure 28.6	(1)		ns
$t_{h(R-S)}$	Chip-select hold time after read		$t_{c(Base)}$ - 15		ns
$t_{su(A-R)}$	Address setup time before read		(1)		ns
$t_{h(R-A)}$	Address hold time after read		$t_{c(Base)}$ - 15		ns
$t_{w(R)}$	Read pulse width		(1)		ns
$t_{su(S-W)}$	Chip-select setup time before write		(1)		ns
$t_{h(W-S)}$	Chip-select hold time after write		1.5 × $t_{c(Base)}$ - 15		ns
$t_{su(A-W)}$	Address setup time before write		(1)		ns
$t_{h(W-A)}$	Address hold time after write		1.5 × $t_{c(Base)}$ - 15		ns
$t_{w(W)}$	Write pulse width		(1)		ns
$t_{su(D-W)}$	Data setup time before write		(1)		ns
$t_{h(W-D)}$	Data hold time after write		0		ns

Note:

1. The value is calculated using the formulas below based on the base clock cycles ($t_{c(Base)}$) and respective cycles of Tsu(A-R), Tw(R), Tsu(A-W), and Tw(W) set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For details on how to set values, refer to 9.3.5 “External Bus Timing”.

   $t_{su(S-R)} = t_{su(A-R)} = Tsu(A-R) \times t_{c(Base)} - 15$ [ns]

   $t_{w(R)} = Tw(R) \times t_{c(Base)} - 10$ [ns]

   $t_{su(S-W)} = t_{su(A-W)} = Tsu(A-W) \times t_{c(Base)} - 15$ [ns]

   $t_{w(W)} = t_{su(D-W)} = Tw(W) \times t_{c(Base)} - 10$ [ns]

V_{CC} = 3.3 V

Switching Characteristics (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.62 External Bus Timing (multiplexed bus)

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{su(S-ALE)}$	Chip-select setup time before ALE	Refer to Figure 28.6	(1)		ns
$t_{h(R-S)}$	Chip-select hold time after read		1.5 × $t_{c(Base)}$ - 15		ns
$t_{su(A-ALE)}$	Address setup time before ALE		(1)		ns
$t_{h(ALE-A)}$	Address hold time after ALE		$t_{c(Base)}$ - 5 (2)		ns
$t_{h(R-A)}$	Address hold time after read		1.5 × $t_{c(Base)}$ - 15		ns
$t_{d(ALE-R)}$	ALE-read delay time		$t_{c(Base)}$ - 5 (2)	$t_{c(Base)}$ + 10 (2)	ns
$t_{w(ALE)}$	ALE pulse width		(1)		ns
$t_{dis(R-A)}$	Address disable time after read			8	ns
$t_{w(R)}$	Read pulse width		(1)		ns
$t_{h(W-S)}$	Chip-select hold time after write		1.5 × $t_{c(Base)}$ - 15		ns
$t_{h(W-A)}$	Address hold time after write		1.5 × $t_{c(Base)}$ - 15		ns
$t_{d(ALE-W)}$	ALE-write delay time		$t_{c(Base)}$ - 5 (2)	$t_{c(Base)}$ + 10 (2)	ns
$t_{w(W)}$	Write pulse width		(1)		ns
$t_{su(D-W)}$	Data setup time before write		(1)		ns
$t_{h(W-D)}$	Data hold time after write		0.5 × $t_{c(Base)}$		ns

Notes:

1. The value is calculated using the formulas below based on the base clock cycles ($t_{c(Base)}$) and respective cycles of Tsu(A-R), Tw(R), Tsu(A-W), and Tw(W) set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For details on how to set values, refer to 9.3.5 “External Bus Timing”.
   $t_{su(S-ALE)}$ = $t_{su(A-ALE)}$ = (Tsu(A-R) - 1) × $t_{c(Base)}$ -15 [ns] (when Tsu(A-R) is greater than 1)
   $t_{su(S-ALE)}$ = $t_{su(A-ALE)}$ = 0.5 × $t_{c(Base)}$ -15 [ns] (when Tsu(A-R) is 1)
   $t_{w(ALE)}$ = (Tsu(A-R) - 1) × $t_{c(Base)}$ - 20 [ns] (when Tsu(A-R) is greater than 1)
   $t_{w(ALE)}$ = 0.5 × $t_{c(Base)}$ - 20 [ns] (when Tsu(A-R) is 1)
   $t_{w(R)}$ = Tw(R) × $t_{c(Base)}$ -10 [ns]
   $t_{w(W)}$ = $t_{su(D-W)}$ = Tw(W) × $t_{c(Base)}$ -10 [ns]
2. When Tsu(A-R) is greater than 1 or Tsu(A-W) is greater than 1. Change “$t_{c(Base)}$” to “0.5 × $t_{c(Base)}$” when Tsu(A-R) is 1 or Tsu(A-W) is 1.

V_{CC} = 3.3 V

Switching Characteristics (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.63 Serial Interface

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{d(C-Q)}$	TXDi output delay time	Refer to Figure 28.6		80	ns
$t_{h(C-Q)}$	TXDi output hold time		0		ns

Table 28.64 Intelligent I/O

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{d(ISCLK2-TXD)}$	ISTXD2 output delay time	Refer to Figure 28.6		180	ns
$t_{h(ISCLK2-RXD)}$	ISTXD2 output hold time		0		ns

Table 28.65 Multi-master I^{2}C-bus Interface (Standard-mode)

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{f(SCL)}$	MSCL output fall time	Refer to Figure 28.6	2		ns
$t_{f(SDA)}$	MSDA output fall time		2		ns
$t_{d(SDA-SCL)S}$	MSCL output delay time after start condition/restart condition		20 × $t_{c(\phi IIC)}$ - 120	52 × $t_{c(\phi IIC)}$ - 40	ns
$t_{d(SCL-SDA)P}$	Restart condition/stop condition output delay time after MSCL becomes high		20 × $t_{c(\phi IIC)}$ + 40	52 × $t_{c(\phi IIC)}$ + 120	ns
$t_{d(SCL-SDA)}$	MSDA output delay time		2 ×$t_{c(\phi IIC)}$ + 40	3 × $t_{c(\phi IIC)}$ + 120	ns

Table 28.66 Multi-master I^{2}C-bus Interface (Fast-mode)

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{f(SCL)}$	MSCL output fall time	Refer to Figure 28.6	2 (1)		ns
$t_{f(SDA)}$	MSDA output fall time		2 (1)		ns
$t_{d(SDA-SCL)S}$	MSCL output delay time after start condition/restart condition		10 × $t_{c(\phi IIC)}$ - 120	26 × $t_{c(\phi IIC)}$ - 40	ns
$t_{d(SCL-SDA)P}$	Restart condition/stop condition output delay time after MSCL becomes high		10 × $t_{c(\phi IIC)}$ + 40	26 × $t_{c(\phi IIC)}$ + 120	ns
$t_{d(SCL-SDA)}$	MSDA output delay time		2 × $t_{c(\phi IIC)}$ + 40	3 × $t_{c(\phi IIC)}$ + 120	ns

Note:

1. External circuits are required to satisfy the I^{2}C-bus specification.

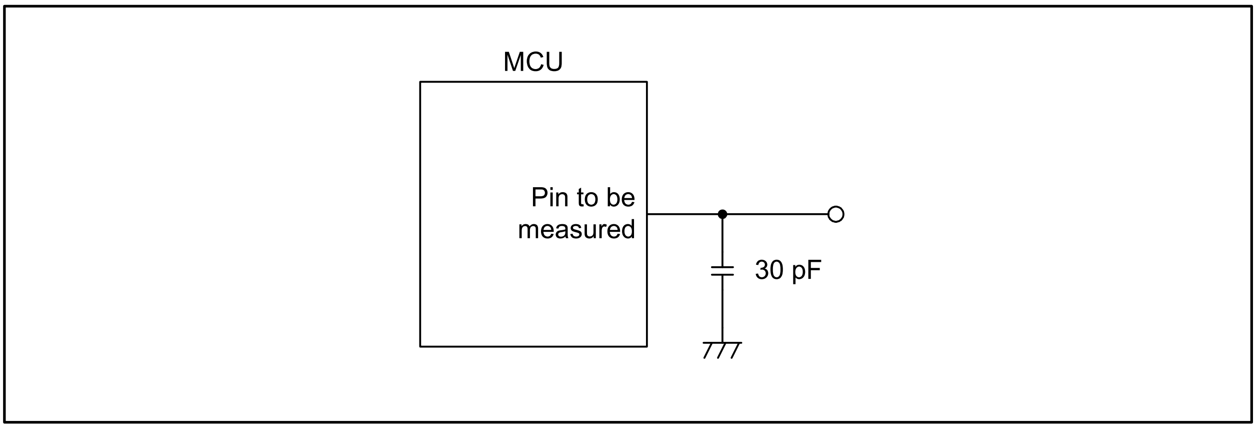


Figure 28.6 Switching Characteristic Measurement Circuit

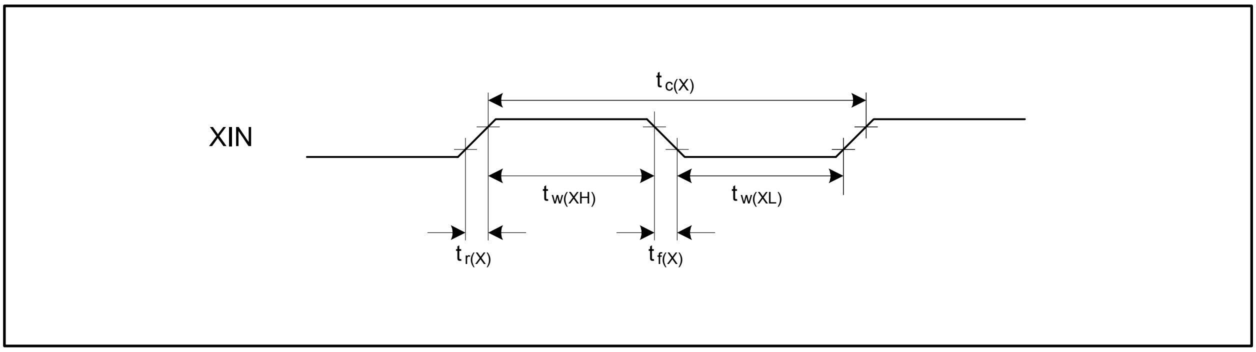
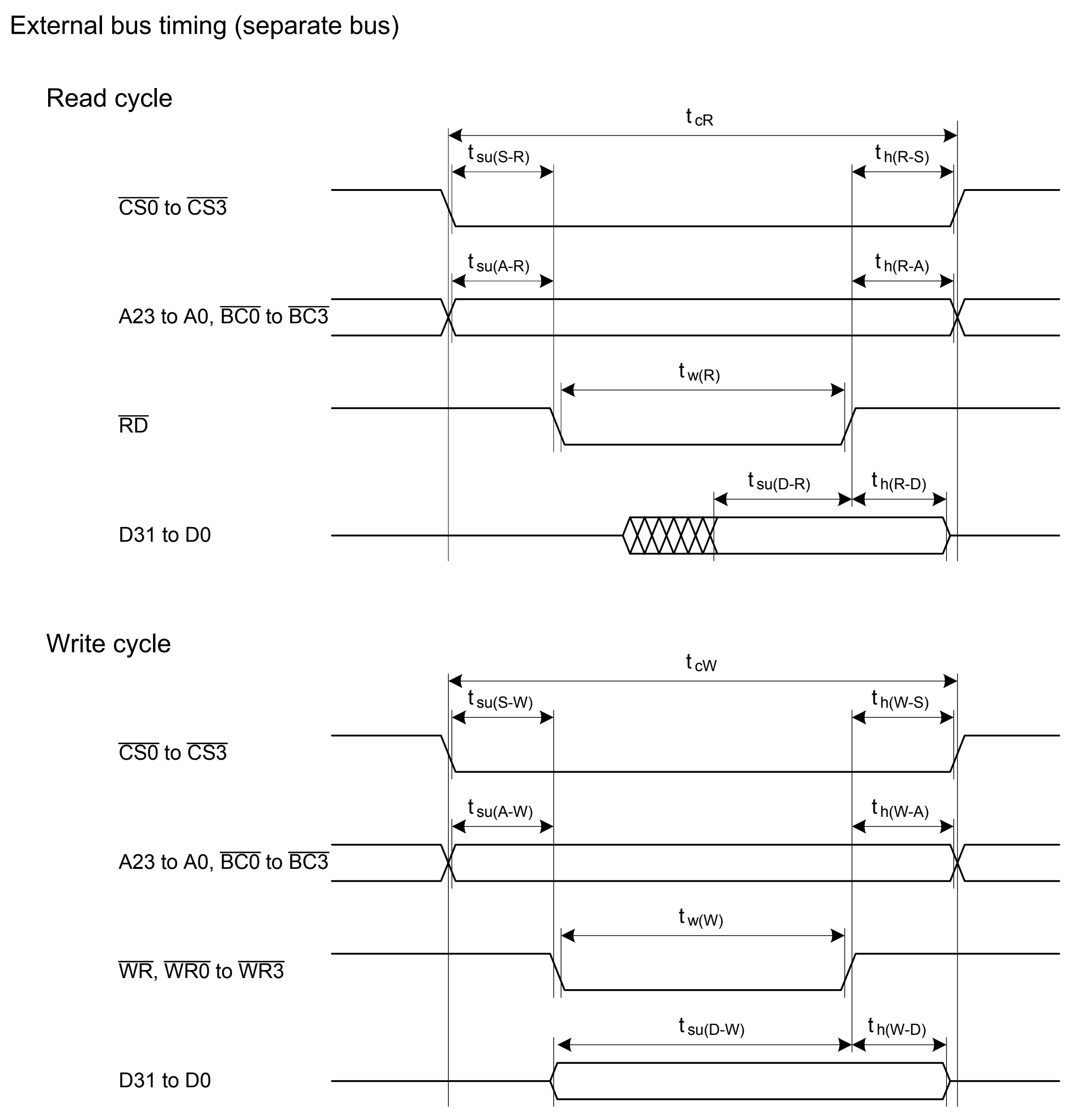


Figure 28.7 External Clock Input Timing

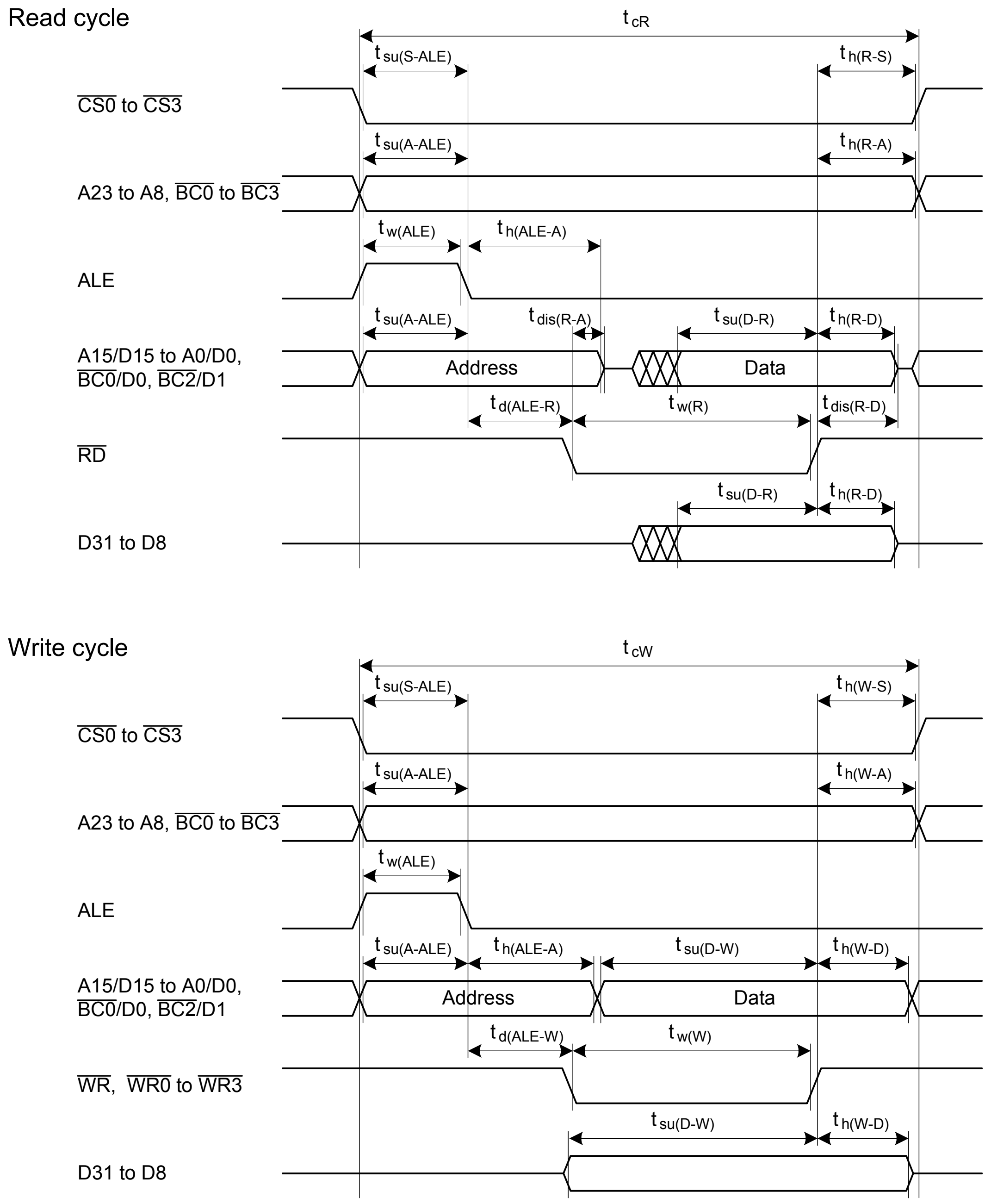


Measurement conditions

Item		V_{CC} = 4.2 to 5.5 V	V_{CC} = 3.0 to 3.6 V
Criterion for input voltage	V_{IH}	2.5 V	1.5 V
	V_{IL}	0.8 V	0.5 V
Criterion for output voltage	V_{OH}	2.0 V	2.4 V
	V_{OL}	0.8 V	0.5 V

Figure 28.8 External Bus Timing for Separate Bus

External bus timing (multiplexed bus)



Measurement conditions

Item		V_{CC} = 4.2 to 5.5 V	V_{CC} = 3.0 to 3.6 V
Criterion for input voltage	V_{IH}	2.5 V	1.5 V
	V_{IL}	0.8 V	0.5 V
Criterion for output voltage	V_{OH}	2.0 V	2.4 V
	V_{OL}	0.8 V	0.5 V

Figure 28.9 External Bus Timing for Multiplexed Bus

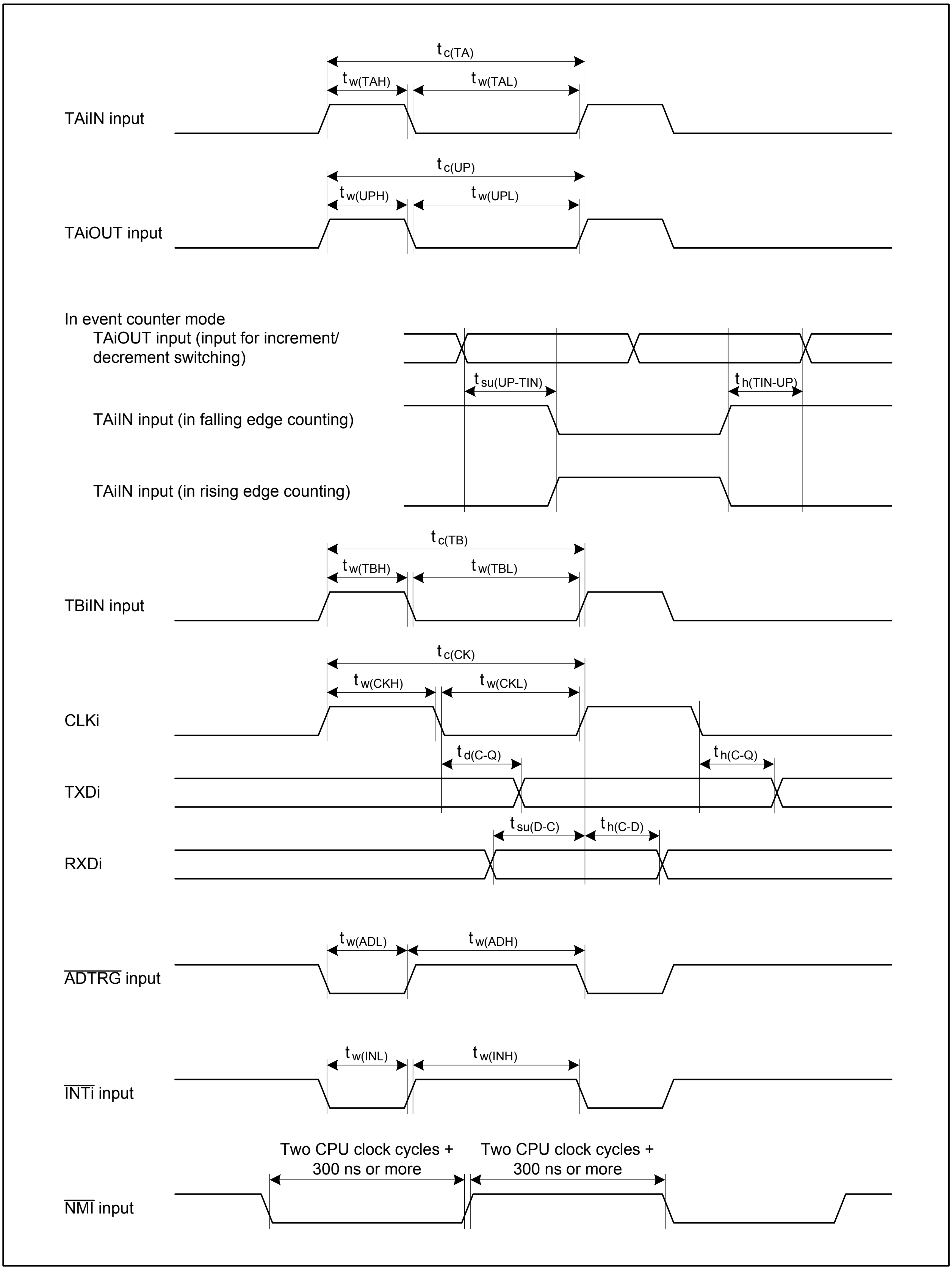


Figure 28.10 Timing of Peripherals

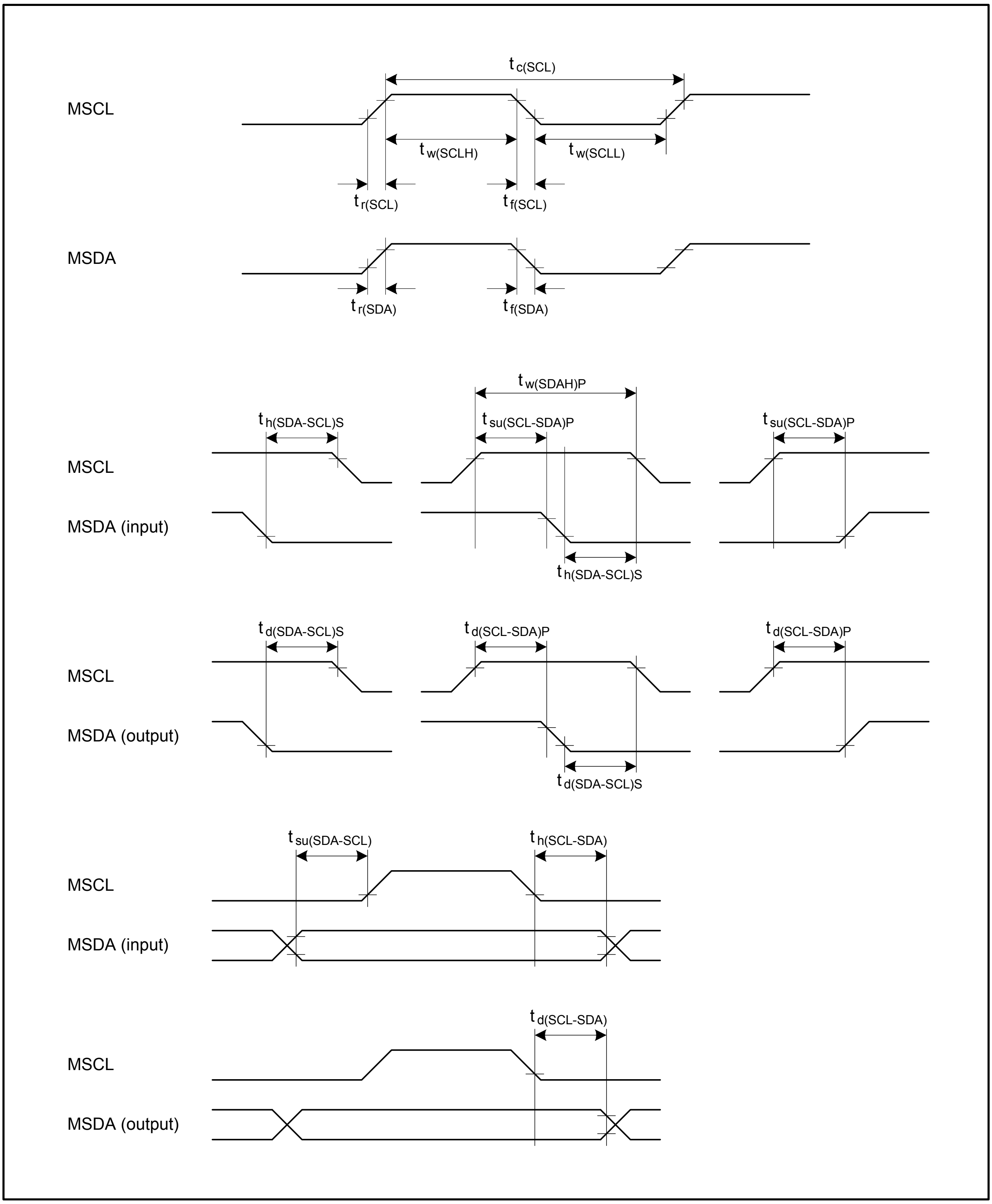


Figure 28.11 Timing of Multi-master I^{2}C-bus Interface

29. Usage Notes

29.1 Notes on Board Designing

29.1.1 Power Supply Pins

The board should be designed so there is no potential difference between pins with the same name. Note the following points:

- Connect all VSS pins to the same GND. Traces for the pins should be as wide as physically possible so the same voltage can be applied to every VSS pin.
- Connect all VCC pins to the same power supply. Traces for the pins should be as wide as physically possible so the same voltage can be applied to every VCC pin.

Insert a capacitor between each VCC pin and the VSS pin to prevent operation errors due to noise. The capacitor should be beneficially effective at high and low frequencies and should have a capacitance of approximately 0.1 μF. The traces for the capacitor and the power supply pins should be as short and wide as physically possible.

29.1.2 Supply Voltage

The device is operationally guaranteed under operating conditions specified in electrical characteristics.
Drive the $\overline{\text{RESET}}$ pin low before the supply voltage becomes lower than the recommended value.

29.2 Notes on Register Setting

29.2.1 Registers with Write-only Bits

Read-modify-write instructions cannot be used when setting a register containing write-only bits. Read-modify-write instructions read a value of an address, modify the value, and write the modified value to the same address. Table 29.1 lists read-modify-write instructions, and Table 29.2 lists registers containing write-only bits. To set a new value by modifying the previous one, write the previous value into RAM as well as to the register, change the contents of the RAM and then transfer the new value to the register by the MOV instruction.

Table 29.1 Read-modify-write Instructions

Function	Mnemonic
Transfer	MOV*Dir*
Bit processing	BCLR, BM*Cnd*, BNOT, BSET, BTSTC, and BTSTS
Shifting	ROLC, RORC, ROT, SHA, and SHL
Arithmetic operation	ABS, ADC, ADCF, ADD, ADSF, DEC, DIV, DIVU, DIVX, EXTS, EXTZ, INC, MUL, MULU, NEG, SBB, and SUB
Decimal operation	DADC, DADD, DSBB, and DSUB
Floating-point operation	ADDF, DIVF, MULF, and SUBF
Logical operation	AND, NOT, OR, and XOR

Table 29.2 Registers with Write-only Bits

Module	Register	Symbol	Address
Watchdog timer	Watchdog timer start register	WDTS	04404Eh
Timer A	Timer A0 register [1]	TA0	0347h-0346h
	Timer A1 register [1]	TA1	0349h-0348h
	Timer A2 register [1]	TA2	034Bh-034Ah
	Timer A3 register [1]	TA3	034Dh-034Ch
	Timer A4 register [1]	TA4	034Fh-034Eh
	Increment/decrement select register	UDF	0344h
Three-phase motor control timers	Timer B2 interrupt generating frequency set counter	ICTB2	030Dh
	Timer A1-1 register	TA11	0303h-0302h
	Timer A2-1 register	TA21	0305h-0304h
	Timer A4-1 register	TA41	0307h-0306h
	Dead time timer	DTT	030Ch
Serial interface	UART0 bit rate register	U0BRG	0369h
	UART1 bit rate register	U1BRG	02E9h
	UART2 bit rate register	U2BRG	0339h
	UART3 bit rate register	U3BRG	0329h
	UART4 bit rate register	U4BRG	02F9h
	UART5 bit rate register	U5BRG	01C9h
	UART6 bit rate register	U6BRG	01D9h
	UART7 bit rate register	U7BRG	01E1h
	UART8 bit rate register	U8BRG	01E9h
	UART9 bit rate register	U9BRG	040301h
	UART10 bit rate register	U10BRG	040309h
	UART0 transmit buffer register	U0TB	036Bh-036Ah
	UART1 transmit buffer register	U1TB	02EBh-02EAh
	UART2 transmit buffer register	U2TB	033Bh-033Ah
	UART3 transmit buffer register	U3TB	032Bh-032Ah
	UART4 transmit buffer register	U4TB	02FBh-02FAh
	UART5 transmit buffer register	U5TB	01CBh-01CAh
	UART6 transmit buffer register	U6TB	01DBh-01DAh
	UART7 transmit buffer register	U7TB	01E3h-01E2h
	UART8 transmit buffer register	U8TB	01EBh-01EAh
	UART9 transmit buffer register	U9TB	040303h-040302h
	UART10 transmit buffer register	U10TB	04030Bh-04030Ah
Intelligent I/O	Group 2 SIO transmit buffer register	G2TB	016Dh-016Ch

1. The register has write-only bits in one-shot timer mode and pulse-width modulation mode.

29.3 Notes on Clock Generator

29.3.1 Sub Clock

29.3.1.1 Oscillator Constant Matching

The constant matching of the sub clock oscillator should be evaluated in both cases when the drive strength is high and low.
Contact the oscillator manufacturer for details on the oscillation circuit constant matching.

29.3.2 Power Control

Do not switch the base clock source until the oscillation of the clock to be used has stabilized. However, this does not apply to the on-chip oscillator since it starts running immediately after the CM31 bit in the CM3 register is set to 1.
To switch the base clock source from the PLL clock to a low speed clock, use the MOV.L or OR.L instruction to set the BCS bit in the CCR register to 1.

- Program example in assembly language

```
        OR.L    #80h, 0004h
```

- Program example in C language

```
        asm("OR.L  #80h, 0004h");
```

29.3.2.1 Stop Mode

- To exit stop mode using a reset, apply a low signal to the $\overline{\text{RESET}}$ pin until the main clock oscillation stabilizes.

29.3.2.2 Suggestions for Power Saving

The following are suggestions to reduce power consumption when programming or designing systems.

- I/O pins:
  If inputs are floating, both transistors may be conducting. Set unassigned pins to input mode and connect each of them to VSS via a resistor, or set them to output mode and leave them open.
- A/D converter:
  When not performing the A/D conversion, set the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). To perform the A/D conversion, set the VCUT bit to 1 (VREF connected) and wait at least 1 µs before starting conversion.
- D/A converter:
  When not performing the D/A conversion, set the DAiE bit in the DACON register (i = 0, 1) to 0 (output disabled) and the DAi register to 00h.
- Peripheral clock stop:
  When entering wait mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop the peripheral clock source. However, this setting does not stop the fC32.

29.4 Notes on Bus

29.4.1 Notes on Register Settings

29.4.1.1 Chip Select Boundary Select Registers

When not using memory expansion mode, do not change values after a reset for registers CB01, CB12, and CB23.

When using memory expansion mode, set all of these registers to a value within the specified range whether or not each chip select space is used.

29.5 Notes on Interrupts

29.5.1 ISP Setting

The interrupt stack pointer (ISP) is initialized to 00000000h after a reset. Set a value to the ISP before an interrupt is accepted, otherwise the program may go out of control. A multiple of 4 should be set to the ISP, which enables faster interrupt sequence due to less memory access.
When using NMI, in particular, since this interrupt cannot be disabled, set the PM24 bit in the PM2 register to 1 (NMI enabled) after setting the ISP at the beginning of the program.

29.5.2 NMI

- NMI cannot be disabled once the PM24 bit in the PM2 register is set to 1 (NMI enabled). This bit setting should be done only when using NMI.

- When the PM24 bit in the PM2 register is 1 (NMI enabled), the P8_5 bit in the P8 register is enabled just for monitoring the $\overline{\text{NMI}}$ pin state. It is not enabled as a general port.

29.5.3 External Interrupts

- The input signal to the $\overline{\text{INTi}}$ pin requires the pulse width specified in the electrical characteristics (i = 0 to 8). If the pulse width is narrower than the specification, an external interrupt may not be accepted.

- When the effective level or edge of the $\overline{\text{INTi}}$ pin (i = 0 to 8) is changed by the following bits: bits POL, LVS in the INTiIC register, the IFSR0i bit (i = 0 to 5) in the IFSR0 register, and the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register, the corresponding IR bit may become 1 (interrupt requested). When setting the above mentioned bits, preset bits ILVL2 to ILVL0 in the INTiIC register to 000b (interrupt disabled). After setting the above mentioned bits, set the corresponding IR bit to 0 (no interrupt requested), then rewrite bits ILVL2 to ILVL0.

- The interrupt input signals to pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ are also connected to bits INT6R to INT8R in registers IIO9IR to IIO11IR. Therefore, these input signals, when assigned to the intelligent I/O, can be used as a source for exiting wait mode or stop mode. Note that these signals are enabled only on the falling edge and not affected by the following bit settings: bits POL and LVS in the INTiIC register (i = 0 to 8), IFSR0i bit (i = 0 to 5) in the IFSR0 register, and the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register.

29.6 Notes on DMAC

29.6.1 DMAC-associated Register Settings

- Set DMAC-associated registers while bits MDi1 and MDi0 in the DMDi register are 00b (DMA transfer disabled) (i = 0 to 3). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure. This procedure also applies when rewriting bits UDAi, USAi, and BWi1 and BWi0 in the DMDi register.

- When rewriting the DMAC-associated registers while DMA transfer is enabled, stop the peripherals that can be DMA triggers so that no DMA transfer request is generated, then set bits MDi1 and MDi0 in the DMDi register of the corresponding channel to 00b (DMA transfer disabled).

- Once a DMA transfer request is accepted, DMA transfer cannot be disabled even if setting bits MDi1 and MDi0 in the DMDi register to 00b (DMA transfer disabled). Do not change the settings of any DMAC-associated registers other than bits MDi1 and MDi0 until the DMA transfer is completed.

- After setting registers DMiSL and DMiSL2, wait at least six peripheral bus clocks to set bits MDi1 and MDi0 in the DMDi register to 01b (single transfer) or 11b (repeat transfer).

29.6.2 Reading DMAC-associated Registers

- Use the following read order to sequentially read registers DMiSL and DMiSL2:
  DM0SL, DM1SL, DM2SL, and DM3SL
  DM0SL2, DM1SL2, DM2SL2, and DM3SL2

29.7 Notes on Timers

29.7.1 Timer A and Timer B

All timers are stopped after a reset. To restart timers, configure parameters such as operating mode, count source, and counter value, then set the TAiS bit or TBjS bit in the TABSR or TBSR register to 1 (count starts) (i = 0 to 4; j = 0 to 5).

The following registers and bits should be set while the TAiS bit or TBjS bit is 0 (count stops):

- Registers TAiMR and TBjMR
- UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- TRGSR register

29.7.2 Timer A

29.7.2.1 Timer Mode

- While the timer counter is running, the TAi register indicates a counter value at any given time. However, FFFFh is read while reloading is in progress. A set value is read if the TAi register is set while the timer counter is stopped.

29.7.2.2 Event Counter Mode

- While the timer counter is running, the TAi register indicates a counter value at any given time. However, FFFFh is read if the timer counter underflows or 0000h if overflows while reloading is in progress. A set value is read if the TAi register is set while the timer counter is stopped.

29.7.2.3 One-shot Timer Mode

- If the TAiS bit in the TABSR register is set to 0 (count stops) while the timer counter is running, the following operations are performed:
  - The timer counter stops and the setting value of the TAi register is reloaded.
  - A low signal is output at the TAiOUT pin.
  - The IR bit in the TAiIC register becomes 1 (interrupts requested) after one CPU clock cycle.

- The one-shot timer is operated by an internal count source. When the trigger is an input to the TAiIN pin, the signal is output with a maximum one count source clock delay after a trigger input to the TAiIN pin.

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done:
  - Select one-shot timer mode after a reset.
  - Switch operating modes from timer mode to one-shot timer mode.
  - Switch operating modes from event counter mode to one-shot timer mode.

- If a retrigger occurs while counting, the timer counter decrements by one, reloads the setting value of the TAi register, and then continues counting. To generate a retrigger while counting, wait at least one count source cycle after the last trigger is generated.

- When an external trigger input is selected to start counting in timer A one-shot mode, do not provide an external retrigger for 300 ns before the timer counter reaches 0000h. Otherwise, it may stop counting.

29.7.2.4 Pulse-width Modulation Mode

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done (i = 0 to 4):
  - Select pulse-width modulation mode after a reset.
  - Switch operating modes from timer mode to pulse-width modulation mode.
  - Switch operating modes from event counter mode to pulse-width modulation mode.

- If the TAiS bit in the TABSR register is set to 0 (count stops) while PWM pulse is output, the following operations are performed:
  - The timer counter stops.
  - The output level at the TAiOUT pin changes from high to low. The IR bit becomes 1.
  - When a low signal is output at the TAiOUT pin, it does not change. The IR bit does not change, either.

29.7.3 Timer B

29.7.3.1 Timer Mode and Event Counter Mode

- While the timer counter is running, the TBj register indicates a counter value at any given time (j = 0 to 5). However, FFFFh is read while reloading is in progress. When a value is set to the TBj register while the timer counter is stopped, if the TBj register is read before the count starts, the set value is read.

29.7.3.2 Pulse Period/Pulse-width Measure Mode

- While the TBjS bit in the TABSR or TBSR register is 1 (start counter), after the MR3 bit becomes 1 (overflow) and at least one count source cycle has elapsed, a write operation to the TBjMR register sets the MR3 bit to 0 (no overflow).

- Use the IR bit in the TBjIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt handler.

- The counter value is undefined when the timer counter starts. Therefore, the timer counter may overflow before a measured pulse is applied on the initial valid edge and cause a timer Bj interrupt request to be generated.

- When the measured pulse is applied on the initial valid edge after the timer counter starts, an undefined value is transferred to the reload register. At this time, a timer Bj interrupt request is not generated.

- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBjMR register after the timer counter starts. However, if the same value is rewritten to bits MR1 and MR0, the IR bit does not change.

- Pulse width is continuously measured in pulse-width measure mode. Whether the measurement result is high-level width or not is determined by a program.

- When an overflow occurs at the same time a pulse is applied on the valid edge, this pulse is not recognized since an interrupt request is generated only once. Do not let an overflow occur in pulse period measure mode.

- In pulse-width measure mode, determine whether an interrupt source is a pulse applied on the valid edge or an overflow by reading the port level in the timer Bj interrupt handler.

29.8 Notes on Three-phase Motor Control Timers

29.8.1 Shutdown

- When a low signal is applied to the $\overline{\text{NMI}}$ pin with the following bit settings, pins TA1OUT, TA2OUT, and TA4OUT become high-impedance: the PM24 bit in the PM2 register is 1 (NMI enabled), the INV02 bit in the INVC0 register is 1 (three-phase motor control timers used), and the INV03 bit is 1 (three-phase motor control timer output enabled).

29.8.2 Register Setting

- Do not write to the TAi1 register before and after timer B2 underflows (i = 1, 2, 4). Before writing to the TAi1 register, read the TB2 register to verify that sufficient time remains until timer B2 underflows. Then, immediately write to the TAi1 register so no interrupt handling is performed during this write procedure. If the TB2 register indicates little time remains until the underflow, write to the TAi1 register after timer B2 underflows.

29.9 Notes on Serial Interface

29.9.1 Changing the UiBRG Register (i = 0 to 10)

- Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When these bits are changed, the UiBRG register must be set again.
- When a clock is input immediately after the UiBRG register is set to 00h, the counter may become FFh. In this case, it requires extra 256 clocks to reload 00h to the register. Once 00h is reloaded, the counter performs the operation without dividing the count source according to the setting.

29.9.2 Synchronous Serial Interface Mode

29.9.2.1 Selecting an External Clock

- If an external clock is selected, the following conditions must be met while the external clock is held high when the CKPOL bit in the UiC0 register is 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge), or while the external clock is held low when the CKPOL bit is 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge) (i = 0 to 10):
  - The TE bit in the UiC1 register is 1 (transmission enabled).
  - The RE bit in the UiC1 register is 1 (reception enabled). This bit setting is not required when only transmitting.
  - The TI bit in the UiC1 register is 0 (data held in the UiTB register).

29.9.2.2 Receive Operation

- In synchronous serial interface mode, the transmit/receive clock is controlled by the transmit control circuit. Set UARTi-associated registers for a transmit operation, even if the MCU is used only for receive operation (i = 0 to 10). Dummy data is output from the TXDi pin while receiving when the TXDi pin is set to output mode.

- When data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data held in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. Then, the OER bit in the UiRB register becomes 1 (overrun error occurred). In this case, the UiRB register becomes undefined. If an overrun error occurs, the IR bit in the SiRIC register does not change to 1.

29.9.3 Special Mode 1 (I^2C Mode)

- To generate a start condition, stop condition, or restart condition, set the STSPSEL bit in the UiSMR4 register to 0 (i = 0 to 6). Then, wait at least a half clock cycle of the transmit/receive clock to change the condition generate bits (STAREQ, RSTAREQ, or STPREQ bit) from 0 to 1.

29.9.4 Reset Procedure on Communication Error

Operations which result in communication errors such as rewriting function select registers during transmission/reception should not be performed. Follow the procedure below to reset the internal circuit once the communication error occurs in the following cases: when the operation above is performed by a receiver or transmitter or when a bit slip is caused by noise.

A. Synchronous Serial Interface Mode

(1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled) (i = 0 to 10).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode).
(4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

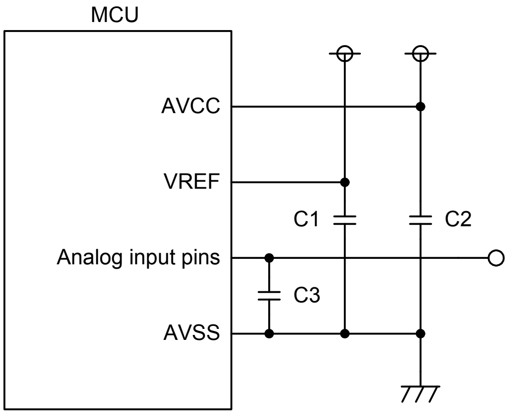
B. UART Mode

(1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 100b (UART mode, 7-bit character length), 101b (UART mode, 8-bit character length), or 110b (UART mode, 9-bit character length).
(4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

29.10 Notes on A/D Converter

29.10.1 Notes on Designing Boards

- Three capacitors should be placed between the AVSS pin and pins such as AVCC, VREF, and analog inputs (AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, and AN15_0 to AN15_7) to avoid erroneous operations caused by noise or latchup, and to reduce conversion errors. Figure 29.1 shows an example of pin configuration for A/D converter.



Notes:

1. C1 ≥ 0.47 μF, C2 ≥ 0.47 μF, and C3 ≥ 100 pF (reference values)
2. The traces for the capacitor and the MCU should be as short and wide as physically possible.

Figure 29.1 Pin Configuration for the A/D Converter

- Do not use AN_4 to AN_7 for analog input if the key input interrupt is to be used. Otherwise, a key input interrupt request occurs when the A/D input voltage becomes VIL or lower.

- When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1 should be VCC or lower.

29.10.2 Notes on Programming

- The following registers should be written while A/D conversion is stopped. That is, before a trigger occurs: AD0CON0 (except the ADST bit), AD0CON1, AD0CON2, AD0CON3, AD0CON4, and AD0CON5.

- When the VCUT bit in the AD0CON1 register is changed from 0 (VREF connected) to 1 (VREF disconnected), wait for at least 1 μs before starting A/D conversion. When not performing A/D conversion, set the VCUT bit to 0 to reduce power consumption.

- Set the port direction bit for the pin to be used as an analog input pin to 0 (input). Set the ASEL bit of the corresponding port function select register to 1 (port is used as A/D input).

- When the TRG bit in the AD0CON0 register is 1 (external trigger or hardware trigger), set the corresponding port direction bit (PD9_7 bit) for the $\overline{\text{ADTRG}}$ pin to 0 (input).

- The ϕAD frequency should be 16 MHz or lower when VCC is 4.2 to 5.5 V, and 10 MHz or lower when VCC is 3.0 to 4.2 V. It should be 1 MHz or higher when the sample and hold function is enabled. If not, it should be 250 kHz or higher.

- When A/D operating mode (bits MD1 and MD0 in the AD0CON0 register or the MD2 bit in the AD0CON1 register) has been changed, reselect analog input pins by setting bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register.

- If the AD0i register is read when the A/D converted result is stored to the register, the stored value may have an error (i = 0 to 7). Read the AD0i register after A/D conversion is completed.
  In one-shot mode or single sweep mode, read the AD0i register after the IR bit in the AD0IC register becomes 1 (interrupt requested).
  In repeat mode, repeat sweep mode 0, or repeat sweep mode 1, an interrupt request can be generated each time A/D conversion is completed when the DUS bit in the AD0CON3 register is 1 (DMAC operating mode enabled). Similar to the other modes above, read the AD00 register after the IR bit in the AD0IC register becomes 1 (interrupt requested).

- When an A/D conversion is halted by setting the ADST bit in the AD0CON0 register to 0, the converted result is undefined. In addition, the unconverted AD0i register may also become undefined. Consequently, the AD0i register should not be used just after A/D conversion is halted.

- External triggers cannot be used in DMAC operating mode. When the DMAC is configured to transfer converted results, do not read the AD00 register by a program.

- While in single sweep mode, if A/D conversion is halted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion is stopped), an interrupt request may be generated even though the sweep is not completed. To halt A/D conversion, disable interrupts before setting the ADST bit to 0.

29.11 Notes on Flash Memory Rewriting

29.11.1 Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on the flash memory is in progress. If the supply voltage goes beyond the guaranteed value, the device cannot be guaranteed.

29.11.2 Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on the flash memory is in progress.

29.11.3 Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation on the flash memory in standard serial I/O mode is disabled.

29.11.4 Notes on Programming

- Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode. In addition, do not change the current mode to wait mode or stop mode during CPU rewrite mode.
- The program, block erase, lock bit program, and protect bit program are interrupted by an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt. If any of the software commands above are interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, the lock bit and protect bit values become undefined. Therefore, disable the lock bit, and then execute the block erase command again.
- Do not use CPU rewrite mode in the interrupt handler of non-maskable interrupts.

29.11.5 Notes on Interrupts

- EW0 mode
  - To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
  - When an NMI, watchdog timer interrupt, oscillator stop detection interrupt, or low voltage detection interrupt occurs, the flash memory module automatically enters read array mode. Therefore, these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by the interrupts and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation. Note that the FMR1 register is not reset. Set the SUSEN bit to 0 (suspend function disabled) in an interrupt handler.
  - Instructions BRK, INTO, and UND, which refer to data on the flash memory, cannot be used in this mode.

- EW1 mode
  - Interrupts assigned to the relocatable vector table should not be accepted during program or block erase operation in the area in which the relocatable vector table is.
  - The watchdog timer interrupt should not be generated.
  - When an NMI, watchdog timer interrupt, oscillator stop detection interrupt, or low voltage detection interrupt occurs, the flash memory module automatically enters read array mode. Therefore, these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by the interrupts and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the EWM bit in the FMR0 register to 1 (EW1 mode) and the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation. Note that the FMR1 register is not reset. Set the SUSEN bit to 0 (suspend function disabled) in an interrupt handler.

29.11.6 Notes on Rewrite Control Program

- EW0 mode
  - If the supply voltage drops during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, and the rewrite operation itself may not be performed. In this case, perform the rewrite operation by serial programmer or parallel programmer.

- EW1 mode
  - Do not rewrite blocks having the rewrite control program.

29.11.7 Notes on Number of Program/Erase Cycles and Software Command Execution Time

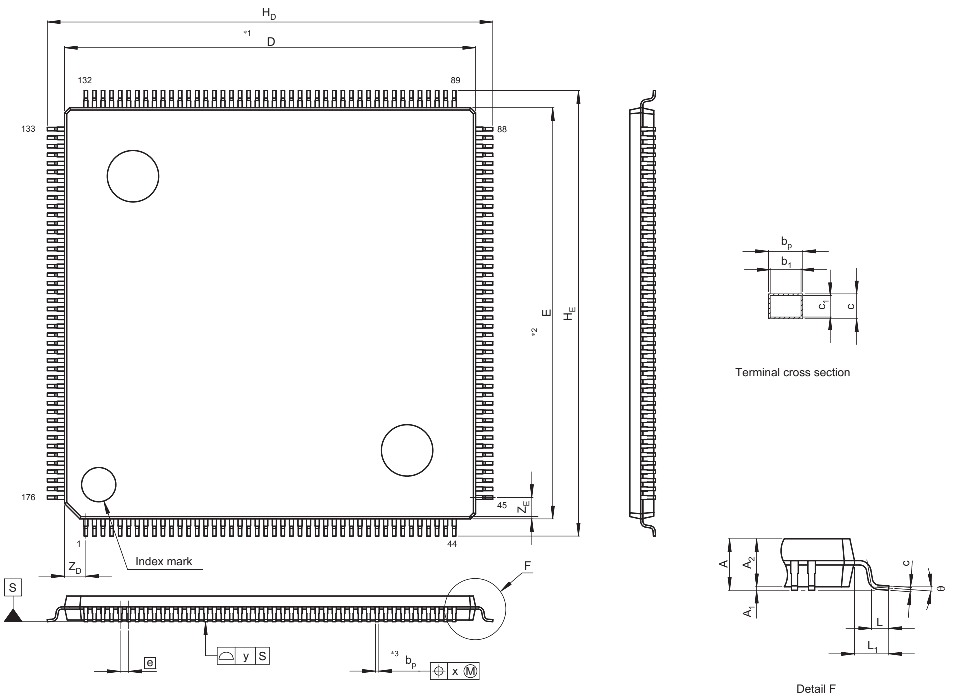
- The time to execute software commands (program, block erase, lock bit program, and protect bit program) increases as the number of program/erase cycles increases. If the number of program/erase cycles exceeds the endurance value specified in the electrical characteristics, it may take an unpredictable amount of time to execute the software commands. The wait time for executing software commands should be set much longer than the execution time specified in the electrical characteristics.

29.11.8 Other Notes

- The minimum values of program/erase cycles specified in the electrical characteristics are the maximum values that can guarantee the initial performance of the flash memory. The program/erase operation may still be performed even if the number of program/erase cycles exceeds the guaranteed values.
- Chips repeatedly programmed and erased for debugging should not be used for commercial products.

Appendix 1. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP176-24x24-0.50	PLQP0176KB-A	176P6Q-A/FP-176E/FP-176EV	1.8g



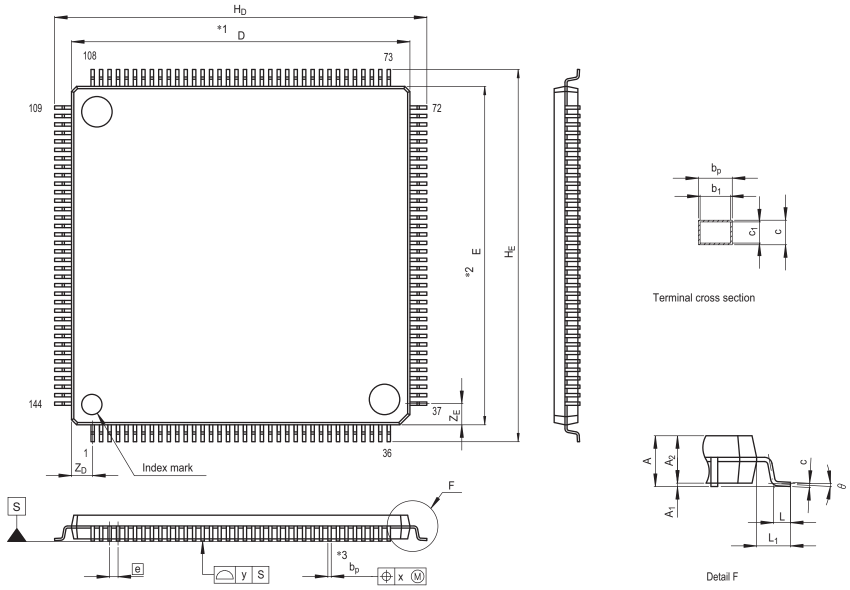
Terminal cross section

Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	23.9	24.0	24.1
E	23.9	24.0	24.1
A_2	—	1.4	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

INDEX

Revision History | R32C/116A Group User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
0.50	Mar 09, 2010	—	Initial release
1.00	Aug 05, 2010	—	Second edition released
			This manual in general
		—	• Applied new Renesas templates and formats to the manual
			• Changed company name to “Renesas Electronics Corporation” and changed related descriptions due to business merger of Renesas Technology Corporation and NEC Electronics Corporation (under Chapters 1, 7, 18, 23, and 28)
			• Modified expressions “version N”, “version D”, and “version P” to “N version”, “D version”, and “P version”, respectively (under Chapters 1 and 28)
			Chapter 1. Overview
		3, 5	• Specified current consumptions in **Tables 1.2 and 1.4**; Deleted Note 1
		6	• Corrected package codes in **Table 1.5**
		8	• Deleted Note 1 from **Figure 1.2**
		20	• Modified expression “fC” in **Table 1.15** to “low speed clocks”
		22	• Changed expression “I^2C bus” in **Table 1.17** to “I^2C-bus”
			Chapter 4. SFRs
		31, 56	• Changed expressions “I^2C Bus” and “I^2C-Bus” in **Tables 4.2 and 4.27** to “I^2C-bus”
		35, 38	• Changed register name “Group i Timer Measurement Prescaler Register” in **Tables 4.6 and 4.9** to “Group i Time Measurement Prescaler Register”
		40	• Modified expression “XY Control Register” in **Table 4.11** to “X-Y Control Register”
		42	• Changed register name “UART2 Transmission/Receive Mode Register” in **Table 4.13** to “UART2 Transmit/Receive Mode Register”; Changed hexadecimal format of reset values for registers TABSR, ONSF, and TRGSR to binary
		44	• Modified reset value “X00X X000b” for AD0CON2 register in **Table 4.15** to “XX0X X000b”
		55	• Changed register name “External Interrupt Source Select Register i” in **Table 4.26** to “External Interrupt Request Source Select Register i”
		56	• Modified reset values for the following registers in **Table 4.27**: I2CSSCR, I2CCR1, I2CCR2, I2CSR, and I2CMR; Added addresses “044420h to 0467FFh”; Modified register name “I^2C Bus START Condition/STOP Condition Control Register” to “I^2C-bus START and STOP Conditions Control Register
			Chapter 5. Resets
		57	• Changed expression “operating level” in (2) of B in **5.1** to “operating voltage”
			Chapter 6. Power Management
		61	• Modified descriptions “main clock oscillator active” and “PLL clock oscillator active” in Note 2 of **Figure 6.2** to “main clock oscillator enabled” and “PLL oscillator enabled”

Rev.	Date	Description	
		Page	Summary
			Chapter 7. Processor Mode
		69	• Modified address “44044h” for PM0 in **Figure 7.1** to “40044h”
			Chapter 8. Clock Generator
		—	• Made minor text modifications to this chapter • Modified function descriptions of bits CM05 and CM10 to the following: “main clock oscillator enabled”, “main clock oscillator disabled”, “PLL oscillator enabled”, and “PLL oscillator disabled”
		72	• Modified expression “fC” in **Figure 8.1** to “low speed clock”; Modified “low speed clock” associated items
		73	• Deleted the last sentence from Note 2 in **Figure 8.2**; Added description for bit settings to Note 6
		74	• Modified expression “fC” in **Figure 8.3** to “low speed clock”; Added Note 8
		75	• Modified bit name of CM10 bit in **Figure 8.4**, descriptively; Added Note 4
		76	• Added description to Note 1 in **Figure 8.6**
		78	• Modified Note 5 in **Figure 8.9**; Added Note 8
		79	• Added description to Note 1 in **Figure 8.10**
		83	• Changed SEO bit name and its function description in **Figure 8.15**
		86	• Added description of stopping main clock to lines 6 to 7 of **8.2.1**
		88, 95, 98	• Modified expression “fC” in line 1 of **8.6** to “Low speed clocks” and “fC” in **Tables 8.3 to 8.5 and 8.7** to “low speed clock”, respectively
		89	• Revised the entire paragraph of **8.7**
		90-91	• Revised **8.7.1** entirely
		92-94	• Add state transition for “Main clock stop (damaged)” to **Figures 8.17 to 8.19**; Added transition flows and conditions partially; Added SEO bit description; Moved figures into **8.7.1** • Corrected a typo “f(XPLL)” in the third row of **Figure 8.17** to “f(PLL)”; Deleted Note 4 • Corrected “PLC1 = 0Xh” in the first row and the fifth row of **Figures 8.17 and Figure 8.19** to “PLC1 = 1Xh” • Corrected a typo “CM0 = 1” in the fifth row of **Figure 8.18** to “CM05 = 1”; Deleted Note 3 • Corrected “CM31 = 1” in the first row and “CM10 = 0” in the second row of **Figure 8.19** to “CM31 = 0” and “CM10 = 1”, respectively; Deleted Note 3
		96	• Revised **8.7.2.4**
		97	• Added Note 1 to **Table 8.6**
		98	• Modified description in line 1 of **8.7.3**
		99	• Added interrupt numbers 0 to 63 to line 2 of **8.7.3.3** as factors of exiting stop mode • Added usage condition to “External interrupt” in **Table 8.8**
			Chapter 9. Bus
		113	• Added EXMPX bit value to column of each bus format in **Table 9.2**; Modified function of P4_0 to P4_3 for memory expansion mode to I/O ports only

Rev.	Date	Description	
		Page	Summary
		121	• Added period of address becoming undefined to "(1) 8-bit data bus" in **Figure 9.15**; Added Note 2
			• Changed timing of ALE signal becoming low below **Figure 9.15**
			Chapter 10. Protection
		—	• Changed title of each sub chapter of this chapter
			Chapter 11. Interrupts
		149	• Deleted "Bits RLVL2 to RLVL0 in the RIPL2 register" and associated signal lines from **Figure 11.8**; Changed expression "DMAC II" to "DMA II transfer complete"
		154	• Changed function description of b0 in registers IIO0IR to IIO11IR in **Figure 11.13**; Changed description of Note 3 descriptively
		156	• Revised the third bullet point of description in **11.14.3**
			Chapter 12. Watchdog Timer
		158	• Added Note 1 to **Figure 12.2**; Modified Note 2 descriptively
		159	• Modified Note 3 in **Figure 12.3** descriptively
		160	• Added description to Note 3 in **Figure 12.5**
			Chapter 13. DMAC
		162	• Modified expression "DMA transfer startup" in **Table 13.1** to "DMA transfer start-up"
		175	• Modified description "peripheral clocks" in the fourth bullet point of **13.4.1** to "peripheral bus clocks"
			Chapter 16. Timers
		190	• Separated signal for overflow or underflow from interrupt signal in **Figure 16.2**
			Chapter 17. Three-phase Motor Control Timers
		225	• Changed expressions "Timer A reload control signal is 0" and "Timer A reload control signal is 1" for INV13 bit in **Figure 17.3** to "Timer A1 reload control signal is 0" and "Timer A1 reload control signal is 1", respectively
		230	• Changed bit functions of bits MR2 and MR3 for TB2MR register in **Figure 17.8**
		231	• Changed function description of PWCON bit for TB2SC register in **Figure 17.9**
		237	• Changed description of Case 1 for "INV00 = 1, and ICTB2 = 1h" in **Figure 17.17**
		239	• Modified text representation in **17.6.1**
			• Modified "overflow" in **17.6.2** to "underflow"
			Chapter 18. Serial Interface
		—	• Made minor text modifications to this chapter
		244	• Deleted "I^{2}C mode" from "Function" of registers U7MR to U10MR in **Figure 18.4**
		252, 280	• Corrected a typo "STARREQ" in Note 3 in **Figure 18.14** and line 1 of **18.3.2** to "STAREQ"
		253	• Modified Note 1 in **Figure 18.16**
		272, 273	• Changed expression "Transmit/receive clock" in **Figures 18.30 and 18.31** to "CLKi"

Rev.	Date	Description	
		Page	Summary
		290	• Moved description in the fourth dash in **18.5.2.1** to the second dash
		291	• Added new paragraph for "Reset Procedure on Communication Error" as **18.5.4**
			Chapter 21. CRC Calculator
		317	• Corrected a typo "CRC_CCITT" in line 2 of **21. CRC Calculator** to "CRC-CCITT"
			Chapter 22. X-Y Conversion
		320, 321	• Changed figure titles "XiR Register" and "YjR Register" for **Figures 22.2 and 22.3** to "Registers X0R to X15R" and "Registers Y0R to Y15R", respectively; Changed preposition "to" in between addresses to "-"
			Chapter 23. Intelligent I/O
		—	• Made minor text modifications to this chapter
		325, 326	• Changed expression "the $\overline{\text{INT0}}$ pin" in **Figure 23.1** and "the $\overline{\text{INT1}}$ pin" **in Figure 23.2** to "the $\overline{\text{INT0}}$ pin or the $\overline{\text{INT1}}$ pin"
		327	• Corrected a typo "IE_IN" in **Figure 23.3** to "IEIN"
		330	• Changed expression "$\overline{\text{INTi}}$ pin" in **Figure 23.6** to "$\overline{\text{INT0}}$/$\overline{\text{INT1}}$ pin"; Modified Note 3 descriptively
		332	• Corrected "BT0S to BT3S" in (2) of Note 1 in **Figure 23.8** to "BT0S to BT2S"
		339	• Modified description of "Reset conditions" in **Table 23.2**
		340	• Changed expression "$\overline{\text{INTi}}$ pin" in **Figure 23.18** to "the $\overline{\text{INT0}}$/$\overline{\text{INT1}}$ pin"; Moved "i = 0 to 2" to figure title
		342, 343	• Moved "(i = 0, 1)" in **Figures 23.19 and 23.20** to respective figure titles
		363	• Deleted Note 1 from **23.4 Group 2 Serial Interface**
			Chapter 24. Multi-master I^{2}C-bus Interface
		—	• Modified the entire chapter descriptively
			• Modified expression "general call" to "general call address"
			• Modified expression "flag" to "bit" when it is used with bit symbols
			• Modified expressions "standard-mode" and "fast-mode" to "Standard-mode" and "Fast-mode", respectively
			• Modified expression "set to" for the RST bit in the I2CCR0 register to "written with"
		372	• Modified "Bus is busy detector" in **Table 24.2** to "Bus busy detector"
		375	• Changed bit name "Transmit/Receive Bit Number Set Bit" in **Figure 24.5** to "Transmit/Receive Bit Length Setting Bit"
		376	• Modified expression "slave address data" in line 2 of **24.1.3.3** to "slave address"
			• Modified expression "I^{2}C reset signal" in **Figure 24.6** to "I^{2}C-bus interface reset signal"
		377	• Modified "ACKCLK bit" in line 2 of **24.1.4** to "ACKD bit"
		378	• Corrected a typo "ϕIIO" in line 2 of **24.1.4.2** to "ϕIIC"
			• Modified **24.1.4.3** descriptively; Modified expressions "MSDA pin level" in line 4 and **Table 24.4** and "MSDA Pin Levels" for table title of Table 24.4 to "MSDA pin state" and "MSDA Pin States", respectively

Rev.	Date	Description	
		Page	Summary
		379	• Changed expression “START Condition and STOP Condition” in register name and bit name for I2CSSCR register in **Figure 24.8** to “START and STOP Conditions”; Changed expression “I^{2}C Bus-line” to “I^{2}C-bus line” • Changed expression “I^{2}C bus line” in **24.1.5.2 and 24.1.5.3** to “I^{2}C-bus line”
		380	• Moved “(2)” from “Function” to “Bit Name” in **Figure 24.9**; Switched Notes 2 and 3
		383	• Changed symbol “/” in function description of bits ICK4 to ICK2 in **Figure 24.11** to “divided-by-”
		384	• Modified setting value of TOSEL bit in **24.1.7.3**
		385	• Moved “(1)”, “(2), and “(3)” from “Function” to “Bit Name” in **Figure 24.13**; Added “(1)” to “Function” of bits TRS and MST
		386	• Deleted explanation in parentheses in line 1 of **24.1.8.2** • Deleted “data” from line 1 of **24.1.8.3**
		387	• Modified “R/W bit” in line 6 of **24.1.8.7** to “R/$\overline{W}$ bit”
		388	• Modified description “lost byte of data” in the second bullet point of **24.1.8.8** to “corresponding byte”
		392	• Modified “(I^{2}C-bus interface enabled)” in the last line below **Figure 24.18** to “(I^{2}C-bus interface disabled)”
		394	• Modified expression “high period of MSCL” in line 5 in the first paragraph of **24.5** to “high period of MSCL pin”
		395	• Changed “Standard Clock Mode” and “Fast Clock Mode” in **Table 24.10** to “Standard-mode” and “Fast-mode” • Changed “BBSY flag setting time” in **Tables 24.10 and 24.11** to “BBSY bit set/reset time”
		396	• Changed parameter “Successful receive interrupt” for I2CCR1 register in **Table 24.12** to “Successful data receive interrupt”
		398	• Modified description “For (A) to (C) in the figure, see A to C” in **24.6.2** to “For (A) to (D) in the figure, see A to D”
		400	• Modified expression “Bits to be zero” in **Figures 24.25 and 24.26** to “Bits to be reset” • Modified “TRX bit” in **Figure 24.27** to “TRS bit”; Modified expressions “Bits to be zero” and “Bit to be zero” to “Bits to be reset” and “Bit to be set”, respectively; Modified description of TRS bit
		401	• Modified “By a program” in **Figure 24.28** to “Software wait”
			Chapter 27. Flash Memory
		442	• Deleted descriptions “erase” and “by using the serial programmer” from “ROM Code Protection” in **Table 27.3**; Added “erase” to “ID Code Protection” • Deleted description “use the serial programmer” from line 3 of **27.2.2**
		444	• Assigned “OFS” to the reserved ID code area in address FFFFFFEFh in **Figure 27.2**; Added Note 1
		445	• Changed description of lines 5 to 6 in **27.2.5**

Rev.	Date	Description	
		Page	Summary
		447	• Changed the following expressions in **Table 27.7**: “the program or the block erase command” to “the program command or the block erase command” and “the read status register command” and “the ready status register command” to “the enter read status register mode command” • Modified figure number “Figure 27.11” in the last line below **Table 27.7** to “Figure 27.12”
		451	• Modified Notes 2 and 3 in **Figure 27.8**
		454, 457	• Changed expressions “$\overline{\text{CS0}}$” and “A23 to A0, $\overline{\text{BC3}}$ to $\overline{\text{BC0}}$” in **Figures 27.13 and 27.14** to “Chip select” and “Address”, respectively
			Chapter 28. Electrical Characteristics
		—	• Initially added
			Chapter 29. Usage Notes
		519, 520	• Switched table numbers for the following tables: “Read-Modify-Write Instructions” to **Table 29.2** and “Registers with Write-only Bit(s)” to **Table 29.1**
		524	• Revised the third bullet point of description in **29.5.3**
		525	• Modified description “peripheral clocks” in the fourth bullet point of **29.6.1** to “peripheral bus clocks”
		529	• Modified text representation in **29.8.1** • Modified “overflow” in **29.8.2** to “underflow”
		530	• Moved description in the fourth dash in **29.9.2.1** to the second dash
		531	• Added new paragraph for “Reset Procedure on Communication Error” as **29.9.4**
			Appendix 1. Package Dimensions
		536, 537	• Added a seating plane to the drawing of package dimension
1.10	Sep 28, 2012	—	Third edition released
			This manual in general
		—	• Changed document number “REJ09B0575-0100” to “R01UH0213EJ0110”
			Chapter 1. Overview
		—	• Modified wording and enhanced description in this chapter
		2, 4	• Modified expressions “Main clock oscillator stop/re-oscillation detection”, “calculation transfer”, “chained transfer”, and “inputs/outputs” in **Tables 1.1 and 1.3** to “Main clock oscillator stop/restart detection”, “calculation result transfer”, “chain transfer”, and “I/O ports”, respectively
		6	• Completed “on planning” and “under development” phases of versions D and P products in **Table 1.5**
		9, 15	• Changed order of signals in **Figures 1.3 and 1.4**
		11, 16	• Changed order of timer pins “TB5IN/TA0IN” in **Tables 1.7 and 1.11** to “TA0IN/TB5IN”
		24	• Added N-channel open drain for pins P2_0 to P2_7 as a selectable function in **Table 1.19**; Modified Note 1
			Chapter 2. CPU
		—	• Modified wording and enhanced description in this chapter
		26	• Corrected a typo “R3R0” in line 3 of **2.1.1** to “R3R1”

Rev.	Date	Description	
		Page	Summary
			Chapter 3. Memory
		—	• Modified wording and enhanced description in this chapter
			Chapter 4. SFRs
		35, 36, 38	• Changed hexadecimal format of reset values for registers G1BCR0, G2BCR0, and G0BCR0 in **Tables 4.6, 4.7, and 4.9** to binary
		42	• Changed register name “Increment/Decrement Counting Select Register” in **Table 4.13** to “Increment/Decrement Select Register”
			Chapter 5. Resets
		—	• Modified wording and enhanced description in this chapter
			Chapter 6. Power Management
		—	• Modified wording and enhanced description in this chapter
		64	• Modified VDEN bit name in **Figure 6.4** to “Low Voltage Detector Enable Bit”; Modified its function descriptions “low voltage detection disabled” and “low voltage detection enabled” to “low voltage detector disabled” and low voltage detector enabled”, respectively
		66	• Modified description “has re-risen above Vdet(R)” in line 7 of **6.2.1** to “rises to or above Vdet(R) again”
			Chapter 7. Processor Mode
		—	• Modified wording and enhanced description in this chapter
			Chapter 8. Clock Generator
		—	• Modified wording and enhanced description in this chapter
		72	• Added BCS bit to **Figure 8.1**
		74	• Modified CM03 bit name “XCIN-XOUT Drive Power Select Bit” in **Figure 8.3** to “XCIN-XOUT Drive Strength Select Bit”; Added description to Note 8
		75	• Modified bit name of bits CM16 and CM15 “XIN-XOUT Drive Power Select Bit” in **Figure 8.4** to “XIN-XOUT Drive Strength Select Bit”; Modified description of Note 2
		76	• Modified function descriptions of CM20 bit in **Figure 8.5** to “Disable oscillator stop detection” when it is 0 and “Enable oscillator stop detection” when it is 1; Corrected “CM02 bit” in Note 3 to “CM20 bit”
			• Modified description of Note 1 in **Figure 8.6**
		85	• Corrected description “When the CSPM bit in the OFS area is 1” in line 5 of **8.1.4** to “When the WCSS bit in the OFS area is 1 and the WPCS bit is 0”; Added description of bit settings of bits PM23 and WDK4 to lines 6 to 7; Modified the last sentence
		86	• Deleted the last sentence in parenthesis in **8.2**
		95	• Modified descriptions in lines 1 to 3 of **8.7.2**
			Chapter 9. Bus
		—	• Modified wording and enhanced description in this chapter
		104	• Changed mathematical symbol “<” in formulas in **9.3.1** to “≤”
		106-107	• Deleted Note 2 from **Figures 9.4 to 9.6**
		107, 108	• Changed minimum value for registers CB01 and CB12 in **Figures 9.7 and 9.8** to “02h”

Rev.	Date	Description	
		Page	Summary
		108	• Changed maximum value for registers CB12 and CB23 in **Figures 9.8 and 9.9** to “F8h” in memory expansion mode and “FFh” in microprocessor mode
		107, 108	• Modified description of Note 2 in **Figures 9.7 to 9.9**
		110	• Added description “(except for the $\overline{\text{CS0}}$ signal)” to Note 1 in **Figure 9.11**
		111	• Modified bit names of bits ESUR1 and ESUR0, bits ESUW1 and ESUW0, bits EWR1 and EWR0, and bits EWW1 and EWW0 in **Figure 9.12** to “Address Setup Cycles Before Read Setting Bit”, “Address Setup Cycles Before Write Setting Bit”, “Read Pulse Width Setting Bit”, and “Write Pulse Width Setting Bit”, respectively
		117	• Modified descriptions “(address setup before $\overline{\text{RD}}$)”, “(address setup before $\overline{\text{WR}}$)”, “($\overline{\text{RD}}$ pulse width)”, and “($\overline{\text{WR}}$ pulse width)” in the second paragraph of **9.3.5** to “(address setup cycles before read)”, “(address setup cycles before write)”, “(read pulse width)”, and “(write pulse width)”, respectively
			Chapter 10. Protection
		—	• Made minor text modifications to this chapter
		127	• Modified description of Note 1 in **Figure 10.1**
			Chapter 11. Interrupts
		—	• Modified wording and enhanced description in this chapter
		131	• Modified description of Note 1 in **Figure 11.1**
		132	• Modified descriptions in the second paragraph in (5) of **11.2**
		134, 135, 142	• Modified description of jump operation in **11.5, Table 11.1, and below Figure 11.4**
		145	• Moved description of Note 1 to (2) of **11.6.4**
		146	• Modified description of Note 1 of **Table 11.7**
		153	• Corrected register symbol “IIOiE” in the last line of **11.13** to “IIOiIE”
			Chapter 12. Watchdog Timer
		—	• Modified wording and enhanced description in this chapter
		157	• Modified description of lines 4 to 5 in **12. Watchdog Timer**; Corrected register symbol “WKD” in line 11 to “WDK”
			Chapter 13. DMAC
		—	• Modified wording and enhanced description in this chapter
		162	• Modified descriptions of timer- and UART-associated interrupt requests in “DMAC request sources” in **Table 13.1**; Modified description “more than 00000001h” in “DMA transfer start-up” to “other than 00000000h”
		170	• Modified descriptions in **13.1**
		171	• Corrected address of external bus “00060000h” in **Table 13.5** to “00080000h”
			Chapter 14. DMAC II
		—	• Modified wording and enhanced description in this chapter • Modified expressions “calculation transfer” and “chained transfer” to “calculation result transfer” and “chain transfer”, respectively
		176	• Corrected source address “FFFFFFFh” in Note 1 of **Table 14.1** to “FFFFFFFFh”

Rev.	Date	Description	
		Page	Summary
		176	• Corrected bit name "IIRLT" in the fifth bullet point of **14.1** to "IRLT"
		178, 179	• Changed expression "DMA II transfer complete interrupt vector address" in lines 3 to 4 and the seventh bullet point of **14.1.2 and Figure 14.2** to "jump address for the DMA II transfer complete interrupt handler"
		178, 181	• Modified expression "interrupt vector" in **Figure 14.2** and line 1 of **14.1.4** to "interrupt vector space"
		179	• Changed expression "jump address" in the seventh bullet point of **14.1.2** to "start address"
		180	• Changed bit names of OPER bit and bits CNT2 to CNT0 in **Figure 14.3** to "Calculation Result Transfer Select Bit" and "Number of Transfers Setting Bit", respectively
		184	• Modified descriptions in **Figure 14.5**
			Chapter 15. Programmable I/O Ports
		—	• Modified wording and enhanced description in this chapter
			Chapter 16. Timers
		—	• Modified wording and enhanced description in this chapter
		194	• Deleted "Counting" from UDF register name and bit names of bits TA4UD to TA0UD in **Figure 16.7**
		202	• Changed MR2 bit name "Increment/Decrement Count Switching Source Select Bit" in **Figure 16.12** to "Increment/Decrement Switching Source Select Bit"; Corrected bit symbols "TAiTGH and TAiTGL" in Note 5 to "TAjTGH and TAjTGL"
		203	• Corrected pin name "INT2" in **Figures 16.13 and 16.14** to "$\overline{\text{INT2}}$"
		204	• Corrected a typo "TA4NR" in line 3 of **16.1.3** to "TA4MR"
		221	• Modified description in **16.3.3.1**
			• Modified description "TBjS bit" in the first bullet point of **16.3.3.2** to "TBjS bit in the TABSR or TBSR register"; Modified "TBj" in the eighth bullet point to "timer Bj"
			Chapter 17. Three-phase Motor Control Timers
		—	• Modified wording and enhanced description in this chapter
		225	• Modified Note 1 in **Figure 17.3**
		232	• Deleted description in line 7 of **17.3**
		236	• Corrected bit symbol "INV06" in Note 3 of **Figure 17.15** to "INV16"
		237	• Corrected register symbol "INV1" in Note 2 of **Figure 17.17** to "INVC1"
			Chapter 18. Serial Interface
		—	• Modified wording and enhanced description in this chapter
		245, 246	• Modified CRD bit name in **Figures 18.5 and 18.6** to "CTS Function Disable Bit"; Modified their function descriptions
		247	• Modified bit description of UiIRS bit when it is 0 in **Figure 18.7** to "Transmit buffer is empty (TI = 1)"; Modified bit name of UiLCH bit to "Logic Inversion Select Bit"
		250	• Modified CSC bit name in **Figure 18.12** to "Clock Synchronization Bit"; Deleted "of the SCLi is received" from function description of SWC bit
		251	• Corrected "UiBRG count source" in function description of bits DL2 to DL0 in **Figure 18.13** to "baud rate generator count source"

Rev.	Date	Description	
		Page	Summary
		252	• Deleted “of the SCLi” from function description of SWC9 bit in **Figure 18.14**
		260	• Modified description “Pulse stops because the TE bit is set to 0” in **Figure 18.21** to “Data is transferred from the UiTB register to the UARTi transmit register”; Modified “TXEPT flag” to “TXEPT bit”; Corrected bit symbol “UiRS” in the fourth dash to “UiIRS”
			• Modified “TXEPT flag” in **Figure 18.21** to “TXEPT bit”; Corrected bit symbol “UiRS” in the fourth dash to “UiIRS”
		268, 269	• Corrected bit functions of UiIRS bit in the fourth dash in **Figures 18.26 and 18.27**
		277, 278	• Divided Table 18.11 into **Tables 18.11 and 18.12**
			Chapter 19. A/D Converter
		—	• Modified wording and enhanced description in this chapter
		300	• Modified “DMA” in Note 4 of **Figure 19.8** to “DMAC”
		302, 304, 305	• Modified “DMA” in the first bullet point in “Reading A/D converted result” of **Tables 19.3, 19.5, and 19.6** to “DMAC”
		305	• Modified description of the number of prioritized pins in line 1 of **19.1.5** and “Function” in **Table 19.6**
		309	• Deleted description “(AN0 to AN7, ANEX0, ANEX1 as analog input port)” from line 9 of **19.2.6**
		314	• Corrected “AD0i” in the ninth bullet point of **19.3.2** to “AD00”
			Chapter 20. D/A Converter
		—	• Made minor text modifications to this chapter
			Chapter 21. CRC Calculator
		—	• Made minor text modifications to this chapter
		317	• Modified **Figure 21.1**
			Chapter 22. X-Y Conversion
		—	• Made minor text modifications to this chapter
			Chapter 23. Intelligent I/O
		—	• Modified wording and enhanced description in this chapter
		327	• Corrected pin name “IE_OUT” in **Figure 23.3** to “IEOUT”
		330	• Changed bit name of bits UD1 and UD0 “Increment/Decrement Counting Control Bit” in **Figure 23.6** to “Increment/Decrement Control Bit”
		333	• Deleted Note 3 from **Figure 23.9**
		340	• Added “($\overline{\text{INT0}}$ or $\overline{\text{INT1}}$)” to the second bullet point for “Reset conditions” in **Table 23.2**; Changed description of the timer counter to start decrementing in the first bullet point of “Other functions”
		351, 353, 356	• Modified “Input to the IIOi_j pin” in **Figures 23.25 to 23.27** “IIOi_j pin”
		354, 355	• Divided Table 23.9 into **Tables 23.9 and 23.10**
		358, 360	• Corrected “Input to the OUTC2_j pin” in **Figures 23.28 and 23.30** to “OUTC2_j pin”
		360	• Deleted the second dash of (A) in **Figure 23.30**
			Chapter 24. Multi-master I^{2}C-bus Interface
		—	• Modified wording and enhanced description in this chapter

Rev.	Date	Description	
		Page	Summary
		372	• Deleted slave-transmit mode from “Specification” of “Slave address match detector” in **Table 24.2**
			• Modified **Figure 24.1**
		377	• Modified description of Note 1 of **Table 24.3**
		390	• Corrected bit symbol “STR” in line 8 of **24.2** to “TRS”
		393, 394	• Modified “VIIC” in **Figures 24.19 and 24.22** to “ϕIIC”
			Chapter 26. I/O Pins
		—	• Modified wording and enhanced description in this chapter
		404	• Deleted ASEL from a factor of pull-up resistor being separated from peripheral functions in **26. I/O Pins and Figure 26.1**
			• Modified descriptions “has neither the bit 5 of the function select register nor the PDi register” in lines 10 to 11 and “The bit 1 of the function select register and the PDi register” in lines 11 to 12 below **Figure 26.1** to “has no function select register or bit 5 in the PD8 register” and “The function select register and bit 1 in the PD14 register”, respectively
		405	• Corrected pin symbols “WR/WR0”, “BC1/WR1”, and “BC2/WR2” in line 4 of **26.1** to “$\overline{\text{WR}}$/$\overline{\text{WR0}}$”, “$\overline{\text{BC1}}$/$\overline{\text{WR1}}$”, and “$\overline{\text{BC2}}$/$\overline{\text{WR2}}$”, respectively
		408, 419, 423	• Changed expression “IIOj output” in **Figures 26.4, 26.14, and 26.18** to “IIOj_i output”
		410	• Corrected description “PD3_i register” in line 3 below **Figure 27.6** to “PD3_i bit”
			Chapter 27. Flash Memory
		—	• Modified wording and enhanced description in this chapter
		468	• Corrected register symbol “FMSR” in line 6 of **27.3.6.2** to “FMSR0”
		477	• Modified descriptions of the first bullet points in **27.6.7 and 27.6.8**
			Chapter 28. Electrical Characteristics
		—	• Modified wording and enhanced description in this chapter
		484	• Changed expression “Programming and erasure endurance of flash memory” in **Table 28.8** to “Program/erase cycles”; Changed its unit “times” to “Cycles”
		489, 502	• Added “MSCL” and “MSDA” to **Tables 28.16 and 28.42**
		490	• Modified description “Drive power:” in **Table 28.17** to “Drive strength: low”
		496, 509	• Corrected “INTi” in the title of **Tables 28.32 and 28.58** to “$\overline{\text{INTi}}$”
			Chapter 29. Usage Notes
		—	• Modified wording and enhanced description in this chapter
		520, 521	• Changed the order of **Tables 29.1 and 29.2**
		521	• Deleted “counting” from UDF register name in **Table 29.2**
		528	• Modified description in **29.7.3.1**
			• Modified description “TBjS bit” in the first bullet point of **29.7.3.2** to “TBjS bit in the TABSR or TBSR register”; Modified “TBj” in the eighth bullet point to “timer Bj”
		533	• Corrected “AD0i” in the ninth bullet point of **29.10.2** to “AD00”
		535	• Modified descriptions of the first bullet points in **29.11.7 and 29.11.8**

R32C/116A Group User’s Manual: Hardware

Publication Date: Rev.0.50 Mar 09, 2010
Rev.1.10 Sep 28, 2012

Published by: Renesas Electronics Corporation

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.3

R32C/116A Group

Renesas Electronics Corporation

R01UH0213EJ0110
(Previous Number: REJ09B0575-0100)